RxDock

0.1.0

Generated by Doxygen 1.9.7

1 Main Page 1

1.1 Introduction . . . 1

1.2 Documentation . . . 1

2 Hierarchical Index 3

2.1 Class Hierarchy . . . 3

3 Class Index 11

3.1 Class List . . . 11

4 File Index 17

4.1 File List . . . 17

5 Class Documentation 21

5.1 rxdock::AddCommand Class Reference . . . 21

5.1.1 Member Function Documentation . . . 22

5.1.1.1 Execute() . . . 22

5.1.1.2 GetName() . . . 22

5.1.1.3 GetNArgs() . . . 22

5.1.1.4 SetArg() . . . 22

5.1.1.5 SetNameArg() . . . 22

5.2 rxdock::AlignTransform Class Reference . . . 23

5.2.1 Member Function Documentation . . . 25

5.2.1.1 Execute() . . . 25

5.2.1.2 SetupLigand() . . . 25

5.2.1.3 SetupReceptor() . . . 26

5.2.1.4 SetupTransform() . . . 26

5.3 rxdock::neldermead::AndCriteria< Criteria1, Criteria2 > Class Template Reference . . . 26

5.4 rxdock::Ann_Cmp_AtomId2 Class Reference . . . 26

5.5 rxdock::Annotation Class Reference . . . 27

5.6 rxdock::AnnotationHandler Class Reference . . . 27

5.7 rxdock::AromIdxSF Class Reference . . . 28

5.7.1 Member Function Documentation . . . 32

5.7.1.1 ParameterUpdated() . . . 32

5.7.1.2 RawScore() . . . 32

5.7.1.3 SetupLigand() . . . 32

5.7.1.4 SetupReceptor() . . . 32

5.7.1.5 SetupScore() . . . 32

5.8 rxdock::Assertion Class Reference . . . 32

5.9 rxdock::Atom Class Reference . . . 33

5.10 rxdock::AtomPtrCmp_AtomicNo Class Reference . . . 35

5.11 rxdock::AtomPtrCmp_AtomId Class Reference . . . 36

5.12 rxdock::AtomPtrCmp_AtomName Class Reference . . . 36

5.13 rxdock::AtomPtrCmp_Ptr Class Reference . . . 36

5.14 rxdock::BadArgument Class Reference . . . 36
5.15 rxdock::BadReceptorFile Class Reference . . . 37
5.16 rxdock::BaseBiMolTransform Class Reference . . . 38
5.16.1 Member Function Documentation . . . 40
5.16.1.1 Update() . . . 40
5.17 rxdock::BaseFileSink Class Reference . . . 41
5.18 rxdock::BaseFileSource Class Reference . . . 41
5.19 rxdock::BaseGrid Class Reference . . . 42
5.20 rxdock::BaseIdxSF Class Reference . . . 44
5.21 rxdock::BaseInterSF Class Reference . . . 47
5.21.1 Member Function Documentation . . . 49
5.21.1.1 SetupLigand() . . . 49
5.21.1.2 SetupReceptor() . . . 49
5.21.1.3 SetupScore() . . . 50
5.21.1.4 Update() . . . 50
5.22 rxdock::BaseIntraSF Class Reference . . . 50
5.22.1 Member Function Documentation . . . 52
5.22.1.1 ScoreMap() . . . 52
5.22.1.2 Update() . . . 53
5.23 rxdock::BaseMolecularDataSource Class Reference . . . 53
5.24 rxdock::BaseMolecularFileSink Class Reference . . . 54
5.25 rxdock::BaseMolecularFileSource Class Reference . . . 55
5.25.1 Member Function Documentation . . . 57
5.25.1.1 GetAtomList() . . . 57
5.25.1.2 GetBondList() . . . 57
5.25.1.3 GetDataFieldList() . . . 57
5.25.1.4 GetDataMap() . . . 57
5.25.1.5 GetDataValue() . . . 58
5.25.1.6 GetNumAtoms() . . . 58
5.25.1.7 GetNumBonds() . . . 58
5.25.1.8 GetNumData() . . . 58
5.25.1.9 GetNumSegments() . . . 58
5.25.1.10 GetNumTitles() . . . 58
5.25.1.11 GetSegmentMap() . . . 58
5.25.1.12 GetTitleList() . . . 59
5.25.1.13 isDataFieldPresent() . . . 59
5.25.1.14 Reset() . . . 59
5.26 rxdock::BaseObject Class Reference . . . 59
5.26.1 Member Function Documentation . . . 60
5.26.1.1 Deleted() . . . 60
5.26.1.2 HandleRequest() . . . 61
5.26.1.3 ParameterUpdated() . . . 61

5.26.1.4 Print() . . . 61
5.27 rxdock::BaseSF Class Reference . . . 61
5.27.1 Member Function Documentation . . . 63
5.27.1.1 GetFullName() . . . 63
5.27.1.2 ParameterUpdated() . . . 64
5.27.1.3 RawScore() . . . 64
5.28 rxdock::BaseTransform Class Reference . . . 64
5.28.1 Member Function Documentation . . . 66
5.28.1.1 GetFullName() . . . 66
5.29 rxdock::BaseUniMolTransform Class Reference . . . 66
5.29.1 Member Function Documentation . . . 69
5.29.1.1 Update() . . . 69
5.30 rxdock::BiMolWorkSpace Class Reference . . . 69
5.30.1 Member Function Documentation . . . 71
5.30.1.1 Save() . . . 71
5.30.1.2 SaveHistory() . . . 71
5.31 rxdock::Bond Class Reference . . . 71
5.32 rxdock::BondPCmp_BondId Class Reference . . . 72
5.33 rxdock::Cavity Class Reference . . . 72
5.34 rxdock::CavityFillSF Class Reference . . . 73
5.34.1 Member Function Documentation . . . 76
5.34.1.1 ParameterUpdated() . . . 76
5.34.1.2 RawScore() . . . 76
5.34.1.3 SetupLigand() . . . 76
5.34.1.4 SetupReceptor() . . . 76
5.34.1.5 SetupScore() . . . 76
5.35 rxdock::CavityGridSF Class Reference . . . 77
5.35.1 Member Function Documentation . . . 80
5.35.1.1 ParameterUpdated() . . . 80
5.35.1.2 RawScore() . . . 80
5.35.1.3 SetupLigand() . . . 80
5.35.1.4 SetupReceptor() . . . 80
5.35.1.5 SetupScore() . . . 80
5.35.1.6 SetupSolvent() . . . 80
5.36 rxdock::CavityPtrCmp_Distance Class Reference . . . 81
5.37 rxdock::CavityPtrCmp_Volume Class Reference . . . 81
5.38 rxdock::Cell Class Reference . . . 81
5.39 rxdock::CellContext Class Reference . . . 82
5.39.1 Member Function Documentation . . . 82
5.39.1.1 Assign() [1/2] . . . 82
5.39.1.2 Assign() [2/2] . . . 83
5.39.1.3 GetVble() [1/2] . . . 83

5.39.1.4 GetVble() [2/2] . . . 83
5.39.1.5 SetVble() . . . 83
5.40 rxdock::CellTokenIter Class Reference . . . 83
5.40.1 Member Function Documentation . . . 84
5.40.1.1 Current() . . . 84
5.40.1.2 Next() . . . 84
5.41 rxdock::CharmmDataSource Class Reference . . . 84
5.42 rxdock::CharmmType Class Reference . . . 85
5.43 rxdock::CharmmTypesFileSource Class Reference . . . 85
5.44 rxdock::Chrom Class Reference . . . 86
5.44.1 Member Function Documentation . . . 88
5.44.1.1 Add() . . . 88
5.44.1.2 clone() . . . 88
5.44.1.3 CompareVector() . . . 88
5.44.1.4 GetLength() . . . 88
5.44.1.5 GetStepVector() . . . 89
5.44.1.6 GetVector() [1/2] . . . 89
5.44.1.7 GetVector() [2/2] . . . 89
5.44.1.8 GetXOverLength() . . . 89
5.44.1.9 Mutate() . . . 89
5.44.1.10 Print() . . . 89
5.44.1.11 Randomise() . . . 89
5.44.1.12 Reset() . . . 90
5.44.1.13 SetVector() [1/2] . . . 90
5.44.1.14 SetVector() [2/2] . . . 90
5.44.1.15 SyncFromModel() . . . 90
5.44.1.16 SyncToModel() . . . 90
5.45 rxdock::ChromDihedralElement Class Reference . . . 90
5.45.1 Member Function Documentation . . . 92
5.45.1.1 clone() . . . 92
5.45.1.2 CompareVector() . . . 92
5.45.1.3 GetLength() . . . 92
5.45.1.4 GetStepVector() . . . 93
5.45.1.5 GetVector() [1/2] . . . 93
5.45.1.6 GetVector() [2/2] . . . 93
5.45.1.7 GetXOverLength() . . . 93
5.45.1.8 Mutate() . . . 93
5.45.1.9 Print() . . . 93
5.45.1.10 Randomise() . . . 93
5.45.1.11 Reset() . . . 94
5.45.1.12 SetVector() [1/2] . . . 94
5.45.1.13 SetVector() [2/2] . . . 94

5.45.1.14 SyncFromModel() . . . 94
5.45.1.15 SyncToModel() . . . 94
5.46 rxdock::ChromDihedralRefData Class Reference . . . 94
5.47 rxdock::ChromElement Class Reference . . . 95
5.48 rxdock::ChromFactory Class Reference . . . 96
5.48.1 Member Function Documentation . . . 97
5.48.1.1 VisitLigandFlexData() . . . 97
5.48.1.2 VisitReceptorFlexData() . . . 97
5.48.1.3 VisitSolventFlexData() . . . 97
5.49 rxdock::ChromOccupancyElement Class Reference . . . 97
5.49.1 Member Function Documentation . . . 99
5.49.1.1 clone() . . . 99
5.49.1.2 CompareVector() . . . 99
5.49.1.3 GetLength() . . . 99
5.49.1.4 GetStepVector() . . . 99
5.49.1.5 GetVector() [1/2] . . . 99
5.49.1.6 GetVector() [2/2] . . . 100
5.49.1.7 GetXOverLength() . . . 100
5.49.1.8 Mutate() . . . 100
5.49.1.9 Print() . . . 100
5.49.1.10 Randomise() . . . 100
5.49.1.11 Reset() . . . 100
5.49.1.12 SetVector() [1/2] . . . 100
5.49.1.13 SetVector() [2/2] . . . 101
5.49.1.14 SyncFromModel() . . . 101
5.49.1.15 SyncToModel() . . . 101
5.50 rxdock::ChromOccupancyRefData Class Reference . . . 101
5.51 rxdock::ChromPositionElement Class Reference . . . 102
5.51.1 Member Function Documentation . . . 103
5.51.1.1 clone() . . . 103
5.51.1.2 CompareVector() . . . 104
5.51.1.3 GetLength() . . . 104
5.51.1.4 GetStepVector() . . . 104
5.51.1.5 GetVector() [1/2] . . . 104
5.51.1.6 GetVector() [2/2] . . . 104
5.51.1.7 GetXOverLength() . . . 104
5.51.1.8 Mutate() . . . 104
5.51.1.9 Print() . . . 105
5.51.1.10 Randomise() . . . 105
5.51.1.11 Reset() . . . 105
5.51.1.12 SetVector() [1/2] . . . 105
5.51.1.13 SetVector() [2/2] . . . 105

5.51.1.14 SyncFromModel() . . . 105
5.51.1.15 SyncToModel() . . . 105
5.52 rxdock::ChromPositionRefData Class Reference . . . 106
5.53 rxdock::unittest::ChromTest Class Reference . . . 106
5.54 rxdock::Command Class Reference . . . 107
5.55 rxdock::Commands Class Reference . . . 108
5.56 rxdock::Constraint Class Reference . . . 109
5.57 rxdock::ConstSF Class Reference . . . 110
5.57.1 Member Function Documentation . . . 113
5.57.1.1 ParameterUpdated() . . . 113
5.57.1.2 RawScore() . . . 113
5.57.1.3 ScoreMap() . . . 113
5.57.1.4 SetupLigand() . . . 113
5.57.1.5 SetupReceptor() . . . 113
5.57.1.6 SetupScore() . . . 113
5.58 rxdock::Context Class Reference . . . 114
5.59 rxdock::Coord Class Reference . . . 114
5.60 rxdock::CoordCmp Class Reference . . . 115
5.61 rxdock::CrdFileSink Class Reference . . . 115
5.61.1 Member Function Documentation . . . 117
5.61.1.1 isMultiConfSupported() . . . 117
5.61.1.2 Render() . . . 117
5.62 rxdock::CrdFileSource Class Reference . . . 117
5.62.1 Member Function Documentation . . . 119
5.62.1.1 isAtomListSupported() . . . 119
5.62.1.2 isBondListSupported() . . . 120
5.62.1.3 isCoordinatesSupported() . . . 120
5.62.1.4 isTitleListSupported() . . . 120
5.62.1.5 Parse() . . . 120
5.63 rxdock::CSVFileSink Class Reference . . . 120
5.63.1 Member Function Documentation . . . 122
5.63.1.1 Render() . . . 122
5.64 rxdock::CyclicAtom Class Reference . . . 122
5.65 rxdock::DihedralElement Class Reference . . . 122
5.66 rxdock::DihedralIntraSF Class Reference . . . 123
5.66.1 Member Function Documentation . . . 125
5.66.1.1 RawScore() . . . 125
5.66.1.2 SetupScore() . . . 125
5.67 rxdock::DihedralSF Class Reference . . . 126
5.68 rxdock::DihedralTargetSF Class Reference . . . 128
5.68.1 Member Function Documentation . . . 131
5.68.1.1 RawScore() . . . 131

5.68.1.2 SetupLigand() . . . 131
5.68.1.3 SetupReceptor() . . . 131
5.68.1.4 SetupScore() . . . 131
5.69 rxdock::DirectorySource Class Reference . . . 132
5.69.1 Detailed Description . . . 132
5.69.2 Constructor & Destructor Documentation . . . 132
5.69.2.1 DirectorySource() . . . 132
5.70 rxdock::DirIsNotAccessible Class Reference . . . 133
5.71 rxdock::DivCommand Class Reference . . . 134
5.71.1 Member Function Documentation . . . 134
5.71.1.1 Execute() . . . 134
5.71.1.2 GetName() . . . 135
5.71.1.3 GetNArgs() . . . 135
5.71.1.4 SetArg() . . . 135
5.71.1.5 SetNameArg() . . . 135
5.72 rxdock::DockingError Class Reference . . . 135
5.73 rxdock::DockingSite Class Reference . . . 136
5.74 rxdock::ElementData Class Reference . . . 137
5.75 rxdock::ElementFileSource Class Reference . . . 137
5.76 rxdock::EnumerateSymCoords Class Reference . . . 139
5.77 rxdock::EnvNotDefined Class Reference . . . 139
5.78 rxdock::Error Class Reference . . . 140
5.79 rxdock::Euler Class Reference . . . 141
5.80 rxdock::EvaluateVisitor Class Reference . . . 141
5.80.1 Member Function Documentation . . . 142
5.80.1.1 VisitAddExp() . . . 142
5.80.1.2 VisitAndExp() . . . 142
5.80.1.3 VisitDivExp() . . . 142
5.80.1.4 VisitExpExp() . . . 143
5.80.1.5 VisitIfExp() . . . 143
5.80.1.6 VisitLogExp() . . . 143
5.80.1.7 VisitMulExp() . . . 143
5.80.1.8 VisitSubExp() . . . 143
5.80.1.9 VisitVbleExp() . . . 143
5.81 rxdock::ExpCommand Class Reference . . . 144
5.81.1 Member Function Documentation . . . 144
5.81.1.1 Execute() . . . 144
5.81.1.2 GetName() . . . 145
5.81.1.3 GetNArgs() . . . 145
5.81.1.4 SetArg() . . . 145
5.81.1.5 SetNameArg() . . . 145
5.82 rxdock::PolarSF::f1prms Struct Reference . . . 145

5.83 rxdock::FFTGrid Class Reference . . . 146
5.83.1 Member Function Documentation . . . 148
5.83.1.1 Print() . . . 148
5.84 rxdock::FFTPeak Class Reference . . . 149
5.85 rxdock::FileError Class Reference . . . 149
5.86 rxdock::FileMissingParameter Class Reference . . . 150
5.87 rxdock::FileParseError Class Reference . . . 151
5.88 rxdock::FileReadError Class Reference . . . 152
5.89 rxdock::FileWriteError Class Reference . . . 153
5.90 rxdock::Filter Class Reference . . . 154
5.90.1 Member Function Documentation . . . 156
5.90.1.1 Update() . . . 156
5.91 rxdock::FilterAddExp Class Reference . . . 156
5.91.1 Member Function Documentation . . . 157
5.91.1.1 Accept() . . . 157
5.91.1.2 GetNOps() . . . 157
5.91.1.3 GetOp() . . . 157
5.91.1.4 Print() . . . 157
5.92 rxdock::FilterAndExp Class Reference . . . 158
5.92.1 Member Function Documentation . . . 158
5.92.1.1 Accept() . . . 158
5.92.1.2 GetNOps() . . . 159
5.92.1.3 GetOp() . . . 159
5.92.1.4 Print() . . . 159
5.93 rxdock::FilterDivExp Class Reference . . . 159
5.93.1 Member Function Documentation . . . 160
5.93.1.1 Accept() . . . 160
5.93.1.2 GetNOps() . . . 160
5.93.1.3 GetOp() . . . 160
5.93.1.4 Print() . . . 160
5.94 rxdock::FilterExpExp Class Reference . . . 161
5.94.1 Member Function Documentation . . . 161
5.94.1.1 Accept() . . . 161
5.94.1.2 GetNOps() . . . 162
5.94.1.3 GetOp() . . . 162
5.94.1.4 Print() . . . 162
5.95 rxdock::FilterExpression Class Reference . . . 162
5.96 rxdock::FilterExpressionVisitor Class Reference . . . 163
5.97 rxdock::FilterIfExp Class Reference . . . 164
5.97.1 Member Function Documentation . . . 164
5.97.1.1 Accept() . . . 164
5.97.1.2 GetNOps() . . . 165

5.97.1.3 GetOp() . . . 165
5.97.1.4 Print() . . . 165
5.98 rxdock::FilterLogExp Class Reference . . . 165
5.98.1 Member Function Documentation . . . 166
5.98.1.1 Accept() . . . 166
5.98.1.2 GetNOps() . . . 166
5.98.1.3 GetOp() . . . 166
5.98.1.4 Print() . . . 166
5.99 rxdock::FilterMulExp Class Reference . . . 167
5.99.1 Member Function Documentation . . . 167
5.99.1.1 Accept() . . . 167
5.99.1.2 GetNOps() . . . 168
5.99.1.3 GetOp() . . . 168
5.99.1.4 Print() . . . 168
5.100 rxdock::FilterSubExp Class Reference . . . 168
5.100.1 Member Function Documentation . . . 169
5.100.1.1 Accept() . . . 169
5.100.1.2 GetNOps() . . . 169
5.100.1.3 GetOp() . . . 169
5.100.1.4 Print() . . . 169
5.101 rxdock::FilterVbleExp Class Reference . . . 170
5.101.1 Member Function Documentation . . . 170
5.101.1.1 Accept() . . . 170
5.101.1.2 GetNOps() . . . 171
5.101.1.3 GetOp() . . . 171
5.101.1.4 Print() . . . 171
5.102 rxdock::FlexAtomFactory Class Reference . . . 171
5.102.1 Member Function Documentation . . . 172
5.102.1.1 VisitLigandFlexData() . . . 172
5.102.1.2 VisitReceptorFlexData() . . . 172
5.102.1.3 VisitSolventFlexData() . . . 172
5.103 rxdock::FlexData Class Reference . . . 172
5.104 rxdock::FlexDataVisitor Class Reference . . . 173
5.105 rxdock::GATransform Class Reference . . . 174
5.105.1 Member Function Documentation . . . 176
5.105.1.1 Execute() . . . 176
5.105.1.2 SetupLigand() . . . 177
5.105.1.3 SetupReceptor() . . . 177
5.105.1.4 SetupTransform() . . . 177
5.106 rxdock::Genome Class Reference . . . 177
5.107 rxdock::GenomeCmp_Score Class Reference . . . 178
5.108 rxdock::geneticprogram::GPChromosome Class Reference . . . 178

5.109 rxdock::geneticprogram::GPFFCHK1 Class Reference . . . 179
5.109.1 Member Function Documentation . . . 180
5.109.1.1 CalculateFitness() [1/2] . . . 180
5.109.1.2 CalculateFitness() [2/2] . . . 180
5.109.1.3 ReadTables() . . . 180
5.110 rxdock::geneticprogram::GPFFCrossDock Class Reference . . . 181
5.110.1 Member Function Documentation . . . 182
5.110.1.1 CalculateFitness() [1/2] . . . 182
5.110.1.2 CalculateFitness() [2/2] . . . 182
5.110.1.3 ReadTables() . . . 182
5.111 rxdock::geneticprogram::GPFFGold Class Reference . . . 182
5.111.1 Member Function Documentation . . . 183
5.111.1.1 CalculateFitness() [1/2] . . . 183
5.111.1.2 CalculateFitness() [2/2] . . . 184
5.111.1.3 ReadTables() . . . 184
5.112 rxdock::geneticprogram::GPFFHSP90 Class Reference . . . 184
5.112.1 Member Function Documentation . . . 185
5.112.1.1 CalculateFitness() [1/2] . . . 185
5.112.1.2 CalculateFitness() [2/2] . . . 185
5.112.1.3 ReadTables() . . . 185
5.113 rxdock::geneticprogram::GPFFSpike Class Reference . . . 186
5.113.1 Member Function Documentation . . . 187
5.113.1.1 CalculateFitness() [1/2] . . . 187
5.113.1.2 CalculateFitness() [2/2] . . . 187
5.113.1.3 ReadTables() . . . 187
5.114 rxdock::geneticprogram::GPFitnessFunction Class Reference . . . 187
5.115 rxdock::geneticprogram::GPGenome Class Reference . . . 188
5.116 rxdock::geneticprogram::GPParser Class Reference . . . 189
5.117 rxdock::geneticprogram::GPPopulation Class Reference . . . 190
5.118 rxdock::HBAConstraint Class Reference . . . 190
5.118.1 Member Function Documentation . . . 191
5.118.1.1 AddAtomList() . . . 191
5.119 rxdock::HBDConstraint Class Reference . . . 192
5.119.1 Member Function Documentation . . . 193
5.119.1.1 AddAtomList() . . . 193
5.120 rxdock::CavityGridSF::HeavyAtomFactory Class Reference . . . 193
5.120.1 Member Function Documentation . . . 193
5.120.1.1 VisitLigandFlexData() . . . 193
5.120.1.2 VisitReceptorFlexData() . . . 194
5.120.1.3 VisitSolventFlexData() . . . 194
5.121 rxdock::HeavyConstraint Class Reference . . . 194
5.121.1 Member Function Documentation . . . 195

5.121.1.1 AddAtomList() . . . 195
5.121.2 Member Data Documentation . . . 195
5.121.2.1 counter . . . 195
5.122 rxdock::HHS_Solvation Class Reference . . . 196
5.122.1 Member Data Documentation . . . 196
5.122.1.1 d_s . . . 196
5.122.1.2 Pij_12 . . . 196
5.123 rxdock::HHSType Class Reference . . . 197
5.124 rxdock::HydroAliphaticConstraint Class Reference . . . 197
5.124.1 Member Function Documentation . . . 198
5.124.1.1 AddAtomList() . . . 198
5.125 rxdock::HydroAromaticConstraint Class Reference . . . 199
5.125.1 Member Function Documentation . . . 200
5.125.1.1 AddAtomList() . . . 200
5.126 rxdock::HydroConstraint Class Reference . . . 200
5.126.1 Member Function Documentation . . . 201
5.126.1.1 AddAtomList() . . . 201
5.127 rxdock::IfCommand Class Reference . . . 201
5.127.1 Member Function Documentation . . . 202
5.127.1.1 Execute() . . . 202
5.127.1.2 GetName() . . . 202
5.127.1.3 GetNArgs() . . . 202
5.127.1.4 SetArg() . . . 202
5.127.1.5 SetNameArg() . . . 203
5.128 rxdock::TriposAtomType::info Struct Reference . . . 203
5.129 rxdock::InitHHS Class Reference . . . 203
5.130 rxdock::InteractionCenter Class Reference . . . 203
5.131 rxdock::InteractionCenterCmp Class Reference . . . 204
5.132 rxdock::InteractionGrid Class Reference . . . 204
5.132.1 Member Function Documentation . . . 206
5.132.1.1 Print() . . . 206
5.133 rxdock::InteractionGridTemplate< T > Class Template Reference . . . 207
5.133.1 Member Function Documentation . . . 209
5.133.1.1 Print() . . . 209
5.134 rxdock::InteractionGridTemplatePtr< T > Class Template Reference . . . 209
5.135 rxdock::InteractionNode< T > Class Template Reference . . . 210
5.136 rxdock::InteractionNodeList< T > Class Template Reference . . . 210
5.137 rxdock::InteractionNodeListMap< T > Class Template Reference . . . 210
5.138 rxdock::InteractionNodeListMapConstIter< T > Class Template Reference . . . 211
5.139 rxdock::InteractionNodeListMapIter< T > Class Template Reference . . . 211
5.140 rxdock::InvalidRequest Class Reference . . . 211
5.141 rxdock::InvertSelectAtom Class Reference . . . 212

5.142 rxdock::isAtom_12Connected Class Reference . . . 212
5.143 rxdock::isAtom_13Connected Class Reference . . . 213
5.144 rxdock::isAtom_eq Class Reference . . . 213
5.145 rxdock::isAtomAnionic Class Reference . . . 214
5.146 rxdock::isAtomBridgehead Class Reference . . . 214
5.147 rxdock::isAtomCationic Class Reference . . . 214
5.148 rxdock::isAtomCharged Class Reference . . . 215
5.149 rxdock::isAtomCyclic Class Reference . . . 215
5.150 rxdock::isAtomEnabled Class Reference . . . 216
5.151 rxdock::isAtomExtended Class Reference . . . 216
5.152 rxdock::isAtomGuanidiniumCarbon Class Reference . . . 216
5.153 rxdock::isAtomHBondAcceptor Class Reference . . . 217
5.154 rxdock::isAtomHBondDonor Class Reference . . . 217
5.155 rxdock::isAtomicNo_eq Class Reference . . . 218
5.156 rxdock::isAtomId_eq Class Reference . . . 218
5.157 rxdock::DockingSite::isAtomInRange Class Reference . . . 218
5.158 rxdock::isAtomInsideCuboid Class Reference . . . 219
5.159 rxdock::isAtomInsideSphere Class Reference . . . 219
5.160 rxdock::isAtomIonic Class Reference . . . 220
5.161 rxdock::isAtomLipophilic Class Reference . . . 220
5.162 rxdock::isAtomMetal Class Reference . . . 220
5.163 rxdock::isAtomName_eq Class Reference . . . 221
5.164 rxdock::isAtomNearCoordList Class Reference . . . 221
5.165 rxdock::isAtomNegCharged Class Reference . . . 222
5.166 rxdock::isAtomNitroN Class Reference . . . 222
5.167 rxdock::isAtomPlanar Class Reference . . . 222
5.168 rxdock::isAtomPosCharged Class Reference . . . 223
5.169 rxdock::isAtomPtr_eq Class Reference . . . 223
5.170 rxdock::isAtomRNA Class Reference . . . 224
5.171 rxdock::isAtomSelected Class Reference . . . 224
5.172 rxdock::isBond_eq Class Reference . . . 224
5.173 rxdock::isBondAmide Class Reference . . . 225
5.174 rxdock::isBondCyclic Class Reference . . . 225
5.175 rxdock::isBondPtr_eq Class Reference . . . 226
5.176 rxdock::isBondRotatable Class Reference . . . 226
5.177 rxdock::isBondSelected Class Reference . . . 226
5.178 rxdock::isBondToNH3 Class Reference . . . 227
5.179 rxdock::isBondToOH Class Reference . . . 227
5.180 rxdock::isCavityNearCoord Class Reference . . . 228
5.181 rxdock::isCoordinationNumber_eq Class Reference . . . 228
5.182 rxdock::isFFType_eq Class Reference . . . 229
5.183 rxdock::isGenome_eq Class Reference . . . 229

5.184 rxdock::isHHSSelected Class Reference . . . 229
5.185 rxdock::isHHSType_eq Class Reference . . . 230
5.186 rxdock::isHybridState_eq Class Reference . . . 230
5.187 rxdock::isInteractionCenterSelected Class Reference . . . 231
5.188 rxdock::isInteractionD_lt Class Reference . . . 231
5.189 rxdock::isPiAtom Class Reference . . . 231
5.190 rxdock::isSegmentName_eq Class Reference . . . 232
5.191 rxdock::isSubunitId_eq Class Reference . . . 232
5.192 rxdock::isSubunitName_eq Class Reference . . . 233
5.193 rxdock::neldermead::IterationCriterion Class Reference . . . 233
5.194 rxdock::LigandError Class Reference . . . 233
5.195 rxdock::LigandFlexData Class Reference . . . 234
5.195.1 Member Function Documentation . . . 235
5.195.1.1 Accept() . . . 235
5.196 rxdock::LigandSiteMapper Class Reference . . . 236
5.196.1 Member Function Documentation . . . 237
5.196.1.1 operator()() . . . 237
5.196.2 Member Data Documentation . . . 238
5.196.2.1 _VOL_INCR . . . 238
5.197 rxdock::LogCommand Class Reference . . . 238
5.197.1 Member Function Documentation . . . 239
5.197.1.1 Execute() . . . 239
5.197.1.2 GetName() . . . 239
5.197.1.3 GetNArgs() . . . 239
5.197.1.4 SetArg() . . . 239
5.197.1.5 SetNameArg() . . . 239
5.198 rxdock::MCStats Class Reference . . . 240
5.199 rxdock::MdlFileSink Class Reference . . . 240
5.199.1 Member Function Documentation . . . 242
5.199.1.1 Render() . . . 242
5.200 rxdock::MdlFileSource Class Reference . . . 242
5.200.1 Member Function Documentation . . . 244
5.200.1.1 isAtomListSupported() . . . 244
5.200.1.2 isBondListSupported() . . . 244
5.200.1.3 isCoordinatesSupported() . . . 245
5.200.1.4 isDataSupported() . . . 245
5.200.1.5 isTitleListSupported() . . . 245
5.200.1.6 Parse() . . . 245
5.201 rxdock::Model Class Reference . . . 245
5.202 rxdock::ModelError Class Reference . . . 247
5.203 rxdock::ModelMutator Class Reference . . . 248
5.204 rxdock::MOEGrid Class Reference . . . 249

5.205 rxdock::MOEGridPoint Class Reference . . . 249
5.206 rxdock::MOEGridShape Class Reference . . . 249
5.207 rxdock::MOL2FileSource Class Reference . . . 250
5.207.1 Member Function Documentation . . . 252
5.207.1.1 isAtomListSupported() . . . 252
5.207.1.2 isBondListSupported() . . . 252
5.207.1.3 isCoordinatesSupported() . . . 253
5.207.1.4 isDataSupported() . . . 253
5.207.1.5 isTitleListSupported() . . . 253
5.207.1.6 Parse() . . . 253
5.207.2 Member Data Documentation . . . 253
5.207.2.1 _TRIPOS_DELIM_SIZE . . . 253
5.208 rxdock::MOL2Substructure Class Reference . . . 253
5.209 rxdock::MulCommand Class Reference . . . 254
5.209.1 Member Function Documentation . . . 255
5.209.1.1 Execute() . . . 255
5.209.1.2 GetName() . . . 255
5.209.1.3 GetNArgs() . . . 255
5.209.1.4 SetArg() . . . 255
5.209.1.5 SetNameArg() . . . 255
5.210 rxdock::NegChargeConstraint Class Reference . . . 255
5.210.1 Member Function Documentation . . . 256
5.210.1.1 AddAtomList() . . . 256
5.211 rxdock::NmrRestraintFileSource Class Reference . . . 257
5.212 rxdock::NmrSF Class Reference . . . 258
5.212.1 Member Function Documentation . . . 261
5.212.1.1 ParameterUpdated() . . . 261
5.212.1.2 RawScore() . . . 261
5.212.1.3 SetupLigand() . . . 261
5.212.1.4 SetupReceptor() . . . 261
5.212.1.5 SetupScore() . . . 262
5.213 rxdock::NoeEndAtoms Class Reference . . . 262
5.214 rxdock::NoeEndNames Class Reference . . . 262
5.215 rxdock::NoeRestraintAtoms Class Reference . . . 263
5.216 rxdock::NoeRestraintNames Class Reference . . . 263
5.217 rxdock::NoFileInDir Class Reference . . . 264
5.218 rxdock::NonBondedGrid Class Reference . . . 265
5.218.1 Member Function Documentation . . . 267
5.218.1.1 Print() . . . 267
5.219 rxdock::NonBondedHHSGrid Class Reference . . . 267
5.219.1 Member Function Documentation . . . 269
5.219.1.1 Print() . . . 269

5.220 rxdock::NullTransform Class Reference . . . 269
5.220.1 Member Function Documentation . . . 271
5.220.1.1 Execute() . . . 271
5.220.1.2 Update() . . . 272
5.221 rxdock::Observer Class Reference . . . 272
5.221.1 Member Function Documentation . . . 272
5.221.1.1 Update() . . . 272
5.222 rxdock::unittest::OccupancyTest Class Reference . . . 273
5.223 rxdock::OverlapHHS Class Reference . . . 273
5.224 rxdock::OverlapVariableHHS Class Reference . . . 274
5.225 rxdock::ParameterFileSource Class Reference . . . 274
5.226 rxdock::ParamHandler Class Reference . . . 275
5.227 rxdock::Parser Class Reference . . . 276
5.228 rxdock::PartitionHHS Class Reference . . . 276
5.229 rxdock::PdbFileSource Class Reference . . . 277
5.229.1 Member Function Documentation . . . 279
5.229.1.1 isAtomListSupported() . . . 279
5.229.1.2 isBondListSupported() . . . 279
5.229.1.3 isCoordinatesSupported() . . . 279
5.229.1.4 isTitleListSupported() . . . 279
5.229.1.5 Parse() . . . 280
5.230 rxdock::PharmaSF Class Reference . . . 280
5.230.1 Member Function Documentation . . . 283
5.230.1.1 ParameterUpdated() . . . 283
5.230.1.2 RawScore() . . . 283
5.230.1.3 ScoreMap() . . . 283
5.230.1.4 SetupLigand() . . . 283
5.230.1.5 SetupReceptor() . . . 283
5.230.1.6 SetupScore() . . . 283
5.231 rxdock::Plane Class Reference . . . 284
5.232 rxdock::PMFDirSource Class Reference . . . 284
5.233 rxdock::PMFGridSF Class Reference . . . 285
5.233.1 Member Function Documentation . . . 288
5.233.1.1 RawScore() . . . 288
5.233.1.2 SetupLigand() . . . 288
5.233.1.3 SetupReceptor() . . . 288
5.233.1.4 SetupScore() . . . 288
5.234 rxdock::PMFIdxSF Class Reference . . . 289
5.234.1 Detailed Description . . . 292
5.234.2 Constructor & Destructor Documentation . . . 292
5.234.2.1 PMFIdxSF() . . . 292
5.234.2.2 ~PMFIdxSF() . . . 292

5.234.3 Member Function Documentation . . . 293
5.234.3.1 GetLinearCloseRangeValue() . . . 293
5.234.3.2 RawScore() . . . 293
5.234.3.3 SetupLigand() . . . 293
5.234.3.4 SetupReceptor() . . . 293
5.234.3.5 SetupScore() . . . 293
5.234.3.6 Update() . . . 294
5.234.4 Member Data Documentation . . . 294
5.234.4.1 _CT . . . 294
5.234.4.2 _PMFDIR . . . 294
5.235 rxdock::PMFValue Struct Reference . . . 294
5.235.1 Detailed Description . . . 294
5.236 rxdock::PolarIdxSF Class Reference . . . 295
5.236.1 Member Function Documentation . . . 299
5.236.1.1 ParameterUpdated() . . . 299
5.236.1.2 RawScore() . . . 299
5.236.1.3 ScoreMap() . . . 300
5.236.1.4 SetupLigand() . . . 300
5.236.1.5 SetupReceptor() . . . 300
5.236.1.6 SetupScore() . . . 300
5.236.1.7 SetupSolvent() . . . 300
5.237 rxdock::PolarIntraSF Class Reference . . . 300
5.237.1 Member Function Documentation . . . 304
5.237.1.1 HandleRequest() . . . 304
5.237.1.2 ParameterUpdated() . . . 304
5.237.1.3 RawScore() . . . 304
5.237.1.4 SetupScore() . . . 304
5.238 rxdock::PolarSF Class Reference . . . 305
5.239 rxdock::geneticprogram::PopCmp Class Reference . . . 308
5.240 rxdock::Population Class Reference . . . 309
5.241 rxdock::PosChargeConstraint Class Reference . . . 309
5.241.1 Member Function Documentation . . . 310
5.241.1.1 AddAtomList() . . . 310
5.242 rxdock::PrettyPrintVisitor Class Reference . . . 311
5.242.1 Member Function Documentation . . . 311
5.242.1.1 VisitAddExp() . . . 311
5.242.1.2 VisitAndExp() . . . 311
5.242.1.3 VisitDivExp() . . . 312
5.242.1.4 VisitExpExp() . . . 312
5.242.1.5 VisitIfExp() . . . 312
5.242.1.6 VisitLogExp() . . . 312
5.242.1.7 VisitMulExp() . . . 312

5.242.1.8 VisitSubExp() . . . 312
5.242.1.9 VisitVbleExp() . . . 313
5.243 rxdock::PrincipalAxes Class Reference . . . 313
5.244 rxdock::PrintVisitor Class Reference . . . 313
5.244.1 Member Function Documentation . . . 314
5.244.1.1 VisitAddExp() . . . 314
5.244.1.2 VisitAndExp() . . . 314
5.244.1.3 VisitDivExp() . . . 314
5.244.1.4 VisitExpExp() . . . 315
5.244.1.5 VisitIfExp() . . . 315
5.244.1.6 VisitLogExp() . . . 315
5.244.1.7 VisitMulExp() . . . 315
5.244.1.8 VisitSubExp() . . . 315
5.244.1.9 VisitVbleExp() . . . 315
5.245 rxdock::PRMFactory Class Reference . . . 316
5.245.1 Member Data Documentation . . . 316
5.245.1.1 _REC_DIHEDRAL_STEP . . . 316
5.245.1.2 _REC_NUM_COORD_FILES . . . 316
5.246 rxdock::DihedralElement::prms Class Reference . . . 317
5.247 rxdock::PseudoAtom Class Reference . . . 317
5.247.1 Member Function Documentation . . . 319
5.247.1.1 Print() . . . 319
5.248 rxdock::PsfFileSink Class Reference . . . 319
5.248.1 Member Function Documentation . . . 321
5.248.1.1 Render() . . . 321
5.249 rxdock::PsfFileSource Class Reference . . . 321
5.249.1 Member Function Documentation . . . 323
5.249.1.1 isAtomListSupported() . . . 323
5.249.1.2 isBondListSupported() . . . 323
5.249.1.3 isCoordinatesSupported() . . . 323
5.249.1.4 isTitleListSupported() . . . 323
5.249.1.5 Parse() . . . 324
5.250 rxdock::Quat Class Reference . . . 324
5.251 rxdock::Rand Class Reference . . . 325
5.252 rxdock::RandLigTransform Class Reference . . . 325
5.252.1 Member Function Documentation . . . 328
5.252.1.1 Execute() . . . 328
5.252.1.2 SetupTransform() . . . 328
5.253 rxdock::RandPopTransform Class Reference . . . 328
5.253.1 Member Function Documentation . . . 331
5.253.1.1 Execute() . . . 331
5.253.1.2 SetupLigand() . . . 331

5.253.1.3 SetupReceptor() . . . 331
5.253.1.4 SetupSolvent() . . . 331
5.253.1.5 SetupTransform() . . . 331
5.253.2 Member Data Documentation . . . 331
5.253.2.1 _SCALE_CHROM_LENGTH . . . 331
5.254 rxdock::RealGrid Class Reference . . . 332
5.254.1 Member Function Documentation . . . 334
5.254.1.1 Print() . . . 334
5.255 rxdock::ReceptorFlexData Class Reference . . . 334
5.255.1 Member Function Documentation . . . 335
5.255.1.1 Accept() . . . 335
5.256 rxdock::neldermead::RelativeValueCriterion< DataType > Class Template Reference . . . 336
5.257 rxdock::Request Class Reference . . . 336
5.258 rxdock::RequestHandler Class Reference . . . 337
5.259 rxdock::RestoreHHS Class Reference . . . 337
5.260 rxdock::RingAromaticConstraint Class Reference . . . 337
5.260.1 Member Function Documentation . . . 338
5.260.1.1 AddAtomList() . . . 338
5.261 rxdock::RotateAtomUsingQuat Class Reference . . . 339
5.262 rxdock::RotateAtomUsingQuatIfSelected Class Reference . . . 339
5.263 rxdock::RotSF Class Reference . . . 339
5.263.1 Member Function Documentation . . . 342
5.263.1.1 ParameterUpdated() . . . 342
5.263.1.2 RawScore() . . . 342
5.263.1.3 SetupLigand() . . . 342
5.263.1.4 SetupReceptor() . . . 342
5.263.1.5 SetupScore() . . . 343
5.264 rxdock::SAIdxSF Class Reference . . . 343
5.264.1 Member Function Documentation . . . 346
5.264.1.1 HandleRequest() . . . 346
5.264.1.2 RawScore() . . . 346
5.264.1.3 ScoreMap() . . . 347
5.264.1.4 SetupLigand() . . . 347
5.264.1.5 SetupReceptor() . . . 347
5.264.1.6 SetupScore() . . . 347
5.264.1.7 SetupSolvent() . . . 347
5.265 rxdock::SaveHHS Class Reference . . . 347
5.266 rxdock::unittest::SearchTest Class Reference . . . 348
5.267 rxdock::SelectAtom Class Reference . . . 348
5.268 rxdock::SelectFlexAtoms Class Reference . . . 348
5.269 rxdock::SelectInteractionCenter Class Reference . . . 349
5.270 rxdock::SetupPMFSF Class Reference . . . 349

5.270.1 Member Function Documentation . . . 352
5.270.1.1 GetPMFfor_IC() . . . 352
5.270.1.2 RawScore() . . . 352
5.270.1.3 SetupLigand() . . . 352
5.270.1.4 SetupReceptor() . . . 352
5.270.1.5 SetupScore() . . . 352
5.271 rxdock::SetupPolarSF Class Reference . . . 353
5.271.1 Member Function Documentation . . . 355
5.271.1.1 RawScore() . . . 355
5.271.1.2 SetupLigand() . . . 356
5.271.1.3 SetupReceptor() . . . 356
5.271.1.4 SetupScore() . . . 356
5.271.1.5 SetupSolvent() . . . 356
5.272 rxdock::SetupSASF Class Reference . . . 356
5.272.1 Member Function Documentation . . . 359
5.272.1.1 RawScore() . . . 359
5.272.1.2 SetupLigand() . . . 359
5.272.1.3 SetupReceptor() . . . 360
5.272.1.4 SetupScore() . . . 360
5.273 rxdock::SFAgg Class Reference . . . 360
5.273.1 Member Function Documentation . . . 362
5.273.1.1 Add() . . . 362
5.273.1.2 GetNumSF() . . . 363
5.273.1.3 GetSF() . . . 363
5.273.1.4 HandleRequest() . . . 363
5.273.1.5 isAgg() . . . 363
5.273.1.6 Print() . . . 363
5.273.1.7 RawScore() . . . 363
5.273.1.8 Register() . . . 363
5.273.1.9 Remove() . . . 364
5.273.1.10 ScoreMap() . . . 364
5.273.1.11 Unregister() . . . 364
5.273.1.12 Update() . . . 364
5.274 rxdock::SFDisableRequest Class Reference . . . 364
5.275 rxdock::SFEnableRequest Class Reference . . . 365
5.276 rxdock::SFFactory Class Reference . . . 365
5.277 rxdock::SFPartitionRequest Class Reference . . . 366
5.278 rxdock::SFSetParamRequest Class Reference . . . 366
5.279 rxdock::SimAnnTransform Class Reference . . . 367
5.279.1 Member Function Documentation . . . 370
5.279.1.1 Execute() . . . 370
5.279.1.2 SetupLigand() . . . 370

5.279.1.3 SetupReceptor() . . . 370
5.279.1.4 SetupTransform() . . . 370
5.279.2 Member Data Documentation . . . 370
5.279.2.1 _MIN_ACC_RATE . . . 370
5.279.2.2 _SCALE_CHROM_LENGTH . . . 371
5.280 rxdock::SimpleSolvationParameters Struct Reference . . . 371
5.281 rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion > Class Template Reference . . . 371
5.281.1 Detailed Description . . . 371
5.281.2 Member Function Documentation . . . 372
5.281.2.1 GetBestParameters() . . . 372
5.281.2.2 GetBestValue() . . . 372
5.282 rxdock::SimplexCostFunction Class Reference . . . 372
5.283 rxdock::SimplexTransform Class Reference . . . 373
5.283.1 Member Function Documentation . . . 375
5.283.1.1 Execute() . . . 375
5.283.1.2 SetupLigand() . . . 376
5.283.1.3 SetupReceptor() . . . 376
5.283.1.4 SetupSolvent() . . . 376
5.283.1.5 SetupTransform() . . . 376
5.283.2 Member Data Documentation . . . 376
5.283.2.1 _STOPPING_STEP_LENGTH . . . 376
5.284 rxdock::Singleton< singleton_type > Class Template Reference . . . 376
5.285 rxdock::SiteMapper Class Reference . . . 377
5.285.1 Member Function Documentation . . . 379
5.285.1.1 Update() . . . 379
5.286 rxdock::SiteMapperFactory Class Reference . . . 379
5.287 rxdock::SmartPtr< T > Class Template Reference . . . 379
5.288 rxdock::SolventFlexData Class Reference . . . 380
5.288.1 Member Function Documentation . . . 381
5.288.1.1 Accept() . . . 381
5.289 rxdock::SphereSiteMapper Class Reference . . . 382
5.289.1 Member Function Documentation . . . 383
5.289.1.1 operator()() . . . 383
5.289.2 Member Data Documentation . . . 384
5.289.2.1 _VOL_INCR . . . 384
5.290 rxdock::neldermead::State< DataType, ParameterType > Struct Template Reference . . . 384
5.290.1 Detailed Description . . . 384
5.291 rxdock::StdRestraintAtoms Class Reference . . . 385
5.292 rxdock::StdRestraintNames Class Reference . . . 385
5.293 rxdock::StringContext Class Reference . . . 386
5.293.1 Member Function Documentation . . . 386

5.293.1.1 Assign() [1/2] . . . 386
5.293.1.2 Assign() [2/2] . . . 387
5.293.1.3 GetVble() [1/2] . . . 387
5.293.1.4 GetVble() [2/2] . . . 387
5.293.1.5 SetVble() . . . 387
5.294 rxdock::StringTokenIter Class Reference . . . 387
5.294.1 Member Function Documentation . . . 388
5.294.1.1 Current() . . . 388
5.294.1.2 Next() . . . 388
5.295 rxdock::StringTooLong Class Reference . . . 388
5.296 rxdock::SubCommand Class Reference . . . 389
5.296.1 Member Function Documentation . . . 390
5.296.1.1 Execute() . . . 390
5.296.1.2 GetName() . . . 390
5.296.1.3 GetNArgs() . . . 390
5.296.1.4 SetArg() . . . 391
5.296.1.5 SetNameArg() . . . 391
5.297 rxdock::Subject Class Reference . . . 391
5.298 rxdock::SymBond Class Reference . . . 391
5.299 rxdock::TetherSF Class Reference . . . 392
5.299.1 Member Function Documentation . . . 395
5.299.1.1 ParameterUpdated() . . . 395
5.299.1.2 RawScore() . . . 395
5.299.1.3 SetupLigand() . . . 395
5.299.1.4 SetupReceptor() . . . 395
5.299.1.5 SetupScore() . . . 395
5.300 rxdock::Token Class Reference . . . 396
5.301 rxdock::TokenIter Class Reference . . . 396
5.302 rxdock::TransformAgg Class Reference . . . 397
5.302.1 Member Function Documentation . . . 399
5.302.1.1 Add() . . . 399
5.302.1.2 Execute() . . . 399
5.302.1.3 GetNumTransforms() . . . 399
5.302.1.4 GetTransform() . . . 399
5.302.1.5 HandleRequest() . . . 400
5.302.1.6 isAgg() . . . 400
5.302.1.7 Print() . . . 400
5.302.1.8 Register() . . . 400
5.302.1.9 Remove() . . . 400
5.302.1.10 Unregister() . . . 400
5.302.1.11 Update() . . . 400
5.303 rxdock::TransformFactory Class Reference . . . 401

5.304 rxdock::TranslateAtom Class Reference . . . 401
5.305 rxdock::TranslateAtomIfSelected Class Reference . . . 401
5.306 rxdock::TriposAtomType Class Reference . . . 401
5.307 rxdock::Variant Class Reference . . . 402
5.308 rxdock::Vble Class Reference . . . 403
5.309 rxdock::VdwGridSF Class Reference . . . 404
5.309.1 Member Function Documentation . . . 407
5.309.1.1 ParameterUpdated() . . . 407
5.309.1.2 RawScore() . . . 407
5.309.1.3 SetupLigand() . . . 407
5.309.1.4 SetupReceptor() . . . 407
5.309.1.5 SetupScore() . . . 407
5.309.1.6 SetupSolvent() . . . 407
5.310 rxdock::VdwIdxSF Class Reference . . . 408
5.310.1 Member Function Documentation . . . 412
5.310.1.1 ParameterUpdated() . . . 412
5.310.1.2 RawScore() . . . 412
5.310.1.3 ScoreMap() . . . 412
5.310.1.4 SetupLigand() . . . 412
5.310.1.5 SetupReceptor() . . . 412
5.310.1.6 SetupScore() . . . 412
5.310.1.7 SetupSolvent() . . . 413
5.310.2 Member Data Documentation . . . 413
5.310.2.1 _ANNOTATION_LIPO . . . 413
5.311 rxdock::VdwIntraSF Class Reference . . . 413
5.311.1 Member Function Documentation . . . 417
5.311.1.1 HandleRequest() . . . 417
5.311.1.2 ParameterUpdated() . . . 417
5.311.1.3 RawScore() . . . 417
5.311.1.4 SetupScore() . . . 417
5.312 rxdock::VdwSF Class Reference . . . 417
5.313 rxdock::WorkSpace Class Reference . . . 421

6 File Documentation **423**

6.1 AlignTransform.h . . . 423
6.2 Annotation.h . . . 424
6.3 AnnotationHandler.h . . . 425
6.4 AromIdxSF.h . . . 426
6.5 Atom.h . . . 428
6.6 AtomFuncs.h . . . 442
6.7 BaseBiMolTransform.h . . . 443
6.8 BaseFileSink.h . . . 444

6.9 BaseFileSource.h . . . 445
6.10 BaseGrid.h . . . 446
6.11 BaseIdxSF.h . . . 449
6.12 BaseInterSF.h . . . 450
6.13 BaseIntraSF.h . . . 451
6.14 BaseMolecularDataSource.h . . . 452
6.15 BaseMolecularFileSink.h . . . 453
6.16 BaseMolecularFileSource.h . . . 455
6.17 BaseObject.h . . . 456
6.18 BaseSF.h . . . 458
6.19 BaseTransform.h . . . 459
6.20 BaseUniMolTransform.h . . . 460
6.21 BiMolWorkSpace.h . . . 461
6.22 Bond.h . . . 462
6.23 Cavity.h . . . 465
6.24 CavityFillSF.h . . . 467
6.25 CavityGridSF.h . . . 468
6.26 CellTokenIter.h . . . 469
6.27 CharmmDataSource.h . . . 470
6.28 CharmmTypesFileSource.h . . . 471
6.29 Chrom.h . . . 472
6.30 ChromDihedralElement.h . . . 473
6.31 ChromDihedralRefData.h . . . 474
6.32 ChromElement.h . . . 475
6.33 ChromFactory.h . . . 477
6.34 ChromOccupancyElement.h . . . 478
6.35 ChromOccupancyRefData.h . . . 479
6.36 ChromPositionElement.h . . . 479
6.37 ChromPositionRefData.h . . . 480
6.38 Commands.h . . . 482
6.39 Config.h . . . 483
6.40 Constraint.h . . . 483
6.41 ConstSF.h . . . 485
6.42 Context.h . . . 486
6.43 Coord.h . . . 488
6.44 CrdFileSink.h . . . 493
6.45 CrdFileSource.h . . . 494
6.46 CSVFileSink.h File Reference . . . 494
6.46.1 Detailed Description . . . 495
6.47 CSVFileSink.h . . . 495
6.48 Smarts.h . . . 495
6.49 Debug.h . . . 496

6.50 DihedralIntraSF.h . . . 497
6.51 DihedralSF.h . . . 497
6.52 DihedralTargetSF.h . . . 499
6.53 DirectorySource.h . . . 500
6.54 DockingError.h . . . 500
6.55 DockingSite.h . . . 501
6.56 ElementFileSource.h . . . 503
6.57 Error.h . . . 504
6.58 Euler.h . . . 506
6.59 FFTGrid.h . . . 507
6.60 FileError.h . . . 509
6.61 Filter.h . . . 510
6.62 FilterExpression.h . . . 511
6.63 FilterExpressionVisitor.h . . . 514
6.64 FlexAtomFactory.h . . . 516
6.65 FlexData.h . . . 517
6.66 FlexDataVisitor.h . . . 517
6.67 GATransform.h . . . 518
6.68 Cell.h . . . 519
6.69 Command.h . . . 520
6.70 GPChromosome.h . . . 522
6.71 GPFFCHK1.h . . . 523
6.72 GPFFCrossDock.h . . . 524
6.73 GPFFGold.h . . . 525
6.74 GPFFHSP90.h . . . 525
6.75 GPFFSpike.h . . . 526
6.76 GPFitnessFunction.h . . . 527
6.77 GPGenome.h . . . 528
6.78 GPParser.h . . . 529
6.79 GPPopulation.h . . . 530
6.80 GPTypes.h . . . 531
6.81 Genome.h . . . 531
6.82 InteractionGrid.h . . . 533
6.83 InteractionTemplate.h . . . 535
6.84 LigandError.h . . . 537
6.85 LigandFlexData.h . . . 538
6.86 LigandSiteMapper.h . . . 539
6.87 MdlFileSink.h . . . 539
6.88 MdlFileSource.h . . . 541
6.89 Model.h . . . 542
6.90 ModelError.h . . . 546
6.91 ModelMutator.h . . . 547

6.92 MOEGrid.h . . . 548
6.93 MOL2FileSource.h . . . 551
6.94 NMCriteria.h . . . 552
6.95 NmrRestraintFileSource.h . . . 553
6.96 NmrSF.h . . . 555
6.97 NMSimplex.h . . . 556
6.98 NMState.h . . . 558
6.99 NoeRestraint.h . . . 559
6.100 NonBondedGrid.h . . . 561
6.101 NonBondedHHSGrid.h . . . 563
6.102 NullTransform.h . . . 564
6.103 Observer.h . . . 564
6.104 CavitySearch.h File Reference . . . 565
6.104.1 Detailed Description . . . 566
6.105 CavitySearch.h . . . 566
6.106 Dock.h File Reference . . . 566
6.106.1 Detailed Description . . . 566
6.107 Dock.h . . . 567
6.108 Tabularize.h File Reference . . . 567
6.108.1 Detailed Description . . . 567
6.109 Tabularize.h . . . 568
6.110 Transform.h File Reference . . . 568
6.110.1 Detailed Description . . . 568
6.111 Transform.h . . . 569
6.112 ParameterFileSource.h . . . 569
6.113 ParamHandler.h . . . 570
6.114 Parser.h . . . 572
6.115 PdbFileSource.h . . . 572
6.116 PharmaSF.h . . . 573
6.117 Plane.h . . . 574
6.118 PMF.h . . . 575
6.119 PMFDirSource.h . . . 576
6.120 PMFGridSF.h . . . 577
6.121 PMFIdxSF.h . . . 578
6.122 PolarIdxSF.h . . . 579
6.123 PolarIntraSF.h . . . 580
6.124 PolarSF.h . . . 581
6.125 Population.h . . . 583
6.126 PrincipalAxes.h . . . 584
6.127 PRMFactory.h . . . 585
6.128 PseudoAtom.h . . . 586
6.129 PsfFileSink.h . . . 587

6.130 PsfFileSource.h . . . 588
6.131 Quat.h . . . 589
6.132 Rand.h . . . 592
6.133 RandLigTransform.h . . . 592
6.134 RandPopTransform.h . . . 593
6.135 Rbt.h . . . 594
6.136 RealGrid.h . . . 596
6.137 ReceptorFlexData.h . . . 599
6.138 Request.h . . . 599
6.139 RequestHandler.h . . . 600
6.140 Resources.h . . . 601
6.141 RotSF.h . . . 601
6.142 SAIdxSF.h . . . 602
6.143 SATypes.h . . . 603
6.144 SetupPMFSF.h . . . 607
6.145 SetupPolarSF.h . . . 608
6.146 SetupSASF.h . . . 608
6.147 SFAgg.h . . . 609
6.148 SFFactory.h . . . 611
6.149 SFRequest.h . . . 612
6.150 SimAnnTransform.h . . . 613
6.151 SimplexCostFunction.h . . . 614
6.152 SimplexTransform.h . . . 615
6.153 Singleton.h . . . 616
6.154 SiteMapper.h . . . 616
6.155 SiteMapperFactory.h . . . 617
6.156 SmartPointer.h . . . 618
6.157 SolventFlexData.h . . . 621
6.158 SphereSiteMapper.h . . . 621
6.159 StringTokenIter.h . . . 622
6.160 Subject.h . . . 623
6.161 Export.h File Reference . . . 623
6.161.1 Detailed Description . . . 623
6.162 Export.h . . . 624
6.163 Number.h File Reference . . . 624
6.163.1 Detailed Description . . . 624
6.163.2 Function Documentation . . . 624
6.163.2.1 getRomanNumeral() . . . 624
6.164 Number.h . . . 625
6.165 Quote.h File Reference . . . 625
6.165.1 Detailed Description . . . 625
6.165.2 Function Documentation . . . 626

6.165.2.1 getRandomQuote() . . . 626
6.166 Quote.h . . . 626
6.167 TetherSF.h . . . 626
6.168 Token.h . . . 627
6.169 TokenIter.h . . . 628
6.170 TransformAgg.h . . . 629
6.171 TransformFactory.h . . . 630
6.172 TriposAtomType.h . . . 631
6.173 Variant.h . . . 633
6.174 Vble.h . . . 635
6.175 VdwGridSF.h . . . 636
6.176 VdwIdxSF.h . . . 637
6.177 VdwIntraSF.h . . . 639
6.178 VdwSF.h . . . 639
6.179 WorkSpace.h . . . 642
6.180 CSVFileSink.cxx File Reference . . . 643
6.180.1 Detailed Description . . . 644
6.181 CavitySearch.cxx File Reference . . . 644
6.181.1 Detailed Description . . . 644
6.182 Dock.cxx File Reference . . . 644
6.182.1 Detailed Description . . . 644
6.183 Tabularize.cxx File Reference . . . 645
6.183.1 Detailed Description . . . 645
6.184 Transform.cxx File Reference . . . 645
6.184.1 Detailed Description . . . 645
6.185 Number.cxx File Reference . . . 645
6.185.1 Detailed Description . . . 645
6.186 Quote.cxx File Reference . . . 645
6.186.1 Detailed Description . . . 646
6.187 ChromTest.h . . . 646
6.188 OccupancyTest.h . . . 646
6.189 SearchTest.h . . . 647
6.190 ParseCavitySearch.cxx File Reference . . . 648
6.190.1 Detailed Description . . . 648
6.191 ParseCavitySearch.h File Reference . . . 648
6.191.1 Detailed Description . . . 648
6.192 ParseCavitySearch.h . . . 648
6.193 ParseDock.cxx File Reference . . . 649
6.193.1 Detailed Description . . . 649
6.194 ParseDock.h File Reference . . . 649
6.194.1 Detailed Description . . . 649
6.195 ParseDock.h . . . 649

6.196 ParseTabularize.cxx File Reference 649
6.196.1 Detailed Description 650
6.197 ParseTabularize.h File Reference 650
6.197.1 Detailed Description 650
6.198 ParseTabularize.h 650
6.199 ParseTransform.cxx File Reference 650
6.199.1 Detailed Description 650
6.200 ParseTransform.h File Reference 651
6.200.1 Detailed Description 651
6.201 ParseTransform.h 651
6.202 rxcmd.cxx File Reference 651
6.202.1 Detailed Description 652

Index **653**

Chapter 1

Main Page

Figure 1.1 RxDock logo

1.1 Introduction

RxDock is a fast and versatile **open-source docking program** that can be used to dock **small molecules** against **proteins** and **nucleic acids**. It is designed for high-throughput virtual screening (HTVS) campaigns and binding mode prediction studies.

RxDock is mainly written in C++ and accessory scripts and programs are written in C++, Perl or Python languages.

1.2 Documentation

This documentation describes the code that makes up RxDock, a part of which is the application programming interface (API).

This documentation does *not* describe the use of RxDock's command-line interface (CLI), the preparation of the input files, or the post-processing options available for the output files. For usage instructions, please see the **Getting started guide**, the **User guide**, and the **Reference manual**.

While intended for developers, this documentation also does *not* describe how to format the code, customize the build system, use the version control software, write the documentation and tests for the code, or test the code changes using existing tests. These topics are, among others, covered in the **Developer guide**.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

rxdock::neldermead::AndCriteria< Criteria1, Criteria2 > . . . 26
rxdock::Ann_Cmp_AtomId2 . . . 26
rxdock::Annotation . . . 27
rxdock::AnnotationHandler . . . 27
  rxdock::AromIdxSF . . . 28
  rxdock::PMFIdxSF . . . 289
  rxdock::PolarSF . . . 305
    rxdock::PolarIdxSF . . . 295
    rxdock::PolarIntraSF . . . 300
  rxdock::SAIdxSF . . . 343
  rxdock::VdwSF . . . 417
    rxdock::VdwIdxSF . . . 408
    rxdock::VdwIntraSF . . . 413
rxdock::Atom . . . 33
  rxdock::PseudoAtom . . . 317
rxdock::AtomPtrCmp_AtomicNo . . . 35
rxdock::AtomPtrCmp_AtomId . . . 36
rxdock::AtomPtrCmp_AtomName . . . 36
rxdock::AtomPtrCmp_Ptr . . . 36
rxdock::BaseFileSink . . . 41
  rxdock::BaseMolecularFileSink . . . 54
    rxdock::CSVFileSink . . . 120
    rxdock::CrdFileSink . . . 115
    rxdock::MdlFileSink . . . 240
    rxdock::PsfFileSink . . . 319
rxdock::BaseFileSource . . . 41
  rxdock::BaseMolecularFileSource . . . 55
    rxdock::CrdFileSource . . . 117
    rxdock::MOL2FileSource . . . 250
    rxdock::MdlFileSource . . . 242
    rxdock::PdbFileSource . . . 277
    rxdock::PsfFileSource . . . 321
  rxdock::CharmmTypesFileSource . . . 85
  rxdock::ElementFileSource . . . 137

rxdock::NmrRestraintFileSource . . . 257
rxdock::ParameterFileSource . . . 274
rxdock::BaseGrid . . . 42
rxdock::InteractionGrid . . . 204
rxdock::InteractionGridTemplate< T > . . . 207
rxdock::NonBondedGrid . . . 265
rxdock::NonBondedHHSGrid . . . 267
rxdock::RealGrid . . . 332
rxdock::FFTGrid . . . 146
rxdock::BaseMolecularDataSource . . . 53
rxdock::BaseMolecularFileSource . . . 55
rxdock::Bond . . . 71
rxdock::BondPCmp_BondId . . . 72
rxdock::Cavity . . . 72
rxdock::CavityPtrCmp_Distance . . . 81
rxdock::CavityPtrCmp_Volume . . . 81
rxdock::Cell . . . 81
rxdock::CharmmDataSource . . . 84
rxdock::CharmmType . . . 85
rxdock::ChromDihedralRefData . . . 94
rxdock::ChromElement . . . 95
rxdock::Chrom . . . 86
rxdock::ChromDihedralElement . . . 90
rxdock::ChromOccupancyElement . . . 97
rxdock::ChromPositionElement . . . 102
rxdock::ChromOccupancyRefData . . . 101
rxdock::ChromPositionRefData . . . 106
rxdock::Command . . . 107
rxdock::AddCommand . . . 21
rxdock::DivCommand . . . 134
rxdock::ExpCommand . . . 144
rxdock::IfCommand . . . 201
rxdock::LogCommand . . . 238
rxdock::MulCommand . . . 254
rxdock::SubCommand . . . 389
rxdock::Commands . . . 108
std::vector< T >::const_iterator
rxdock::InteractionNodeListMapConstIter< T > . . . 211
rxdock::Constraint . . . 109
rxdock::HBAConstraint . . . 190
rxdock::HBDConstraint . . . 192
rxdock::HeavyConstraint . . . 194
rxdock::HydroAliphaticConstraint . . . 197
rxdock::HydroAromaticConstraint . . . 199
rxdock::HydroConstraint . . . 200
rxdock::NegChargeConstraint . . . 255
rxdock::PosChargeConstraint . . . 309
rxdock::RingAromaticConstraint . . . 337
rxdock::Context . . . 114
rxdock::CellContext . . . 82
rxdock::StringContext . . . 386
rxdock::Coord . . . 114
rxdock::CoordCmp . . . 115
rxdock::CyclicAtom . . . 122
rxdock::DihedralElement . . . 122
rxdock::DirectorySource . . . 132

rxdock::PMFDirSource . . . 284
rxdock::DockingSite . . . 136
rxdock::ElementData . . . 137
rxdock::EnumerateSymCoords . . . 139
rxdock::Euler . . . 141
std::exception
rxdock::Error . . . 140
rxdock::Assertion . . . 32
rxdock::BadArgument . . . 36
rxdock::BadReceptorFile . . . 37
rxdock::DockingError . . . 135
rxdock::FileError . . . 149
rxdock::DirIsNotAccessible . . . 133
rxdock::EnvNotDefined . . . 139
rxdock::FileMissingParameter . . . 150
rxdock::FileParseError . . . 151
rxdock::FileReadError . . . 152
rxdock::FileWriteError . . . 153
rxdock::NoFileInDir . . . 264
rxdock::StringTooLong . . . 388
rxdock::InvalidRequest . . . 211
rxdock::LigandError . . . 233
rxdock::ModelError . . . 247
rxdock::PolarSF::f1prms . . . 145
rxdock::FFTPeak . . . 149
rxdock::FilterExpression . . . 162
rxdock::FilterAddExp . . . 156
rxdock::FilterAndExp . . . 158
rxdock::FilterDivExp . . . 159
rxdock::FilterExpExp . . . 161
rxdock::FilterIfExp . . . 164
rxdock::FilterLogExp . . . 165
rxdock::FilterMulExp . . . 167
rxdock::FilterSubExp . . . 168
rxdock::FilterVbleExp . . . 170
rxdock::FilterExpressionVisitor . . . 163
rxdock::EvaluateVisitor . . . 141
rxdock::PrettyPrintVisitor . . . 311
rxdock::PrintVisitor . . . 313
rxdock::FlexDataVisitor . . . 173
rxdock::CavityGridSF::HeavyAtomFactory . . . 193
rxdock::ChromFactory . . . 96
rxdock::FlexAtomFactory . . . 171
std::function
rxdock::DockingSite::isAtomInRange . . . 218
rxdock::isAtomAnionic . . . 214
rxdock::isAtomBridgehead . . . 214
rxdock::isAtomCationic . . . 214
rxdock::isAtomCharged . . . 215
rxdock::isAtomCyclic . . . 215
rxdock::isAtomEnabled . . . 216
rxdock::isAtomExtended . . . 216
rxdock::isAtomGuanidiniumCarbon . . . 216
rxdock::isAtomHBondAcceptor . . . 217
rxdock::isAtomHBondDonor . . . 217
rxdock::isAtomId_eq . . . 218
rxdock::isAtomInsideCuboid . . . 219

rxdock::isAtomInsideSphere . . . 219
rxdock::isAtomIonic . . . 220
rxdock::isAtomLipophilic . . . 220
rxdock::isAtomMetal . . . 220
rxdock::isAtomName_eq . . . 221
rxdock::isAtomNearCoordList . . . 221
rxdock::isAtomNegCharged . . . 222
rxdock::isAtomNitroN . . . 222
rxdock::isAtomPlanar . . . 222
rxdock::isAtomPosCharged . . . 223
rxdock::isAtomPtr_eq . . . 223
rxdock::isAtomRNA . . . 224
rxdock::isAtomSelected . . . 224
rxdock::isAtom_12Connected . . . 212
rxdock::isAtom_13Connected . . . 213
rxdock::isAtom_eq . . . 213
rxdock::isAtomicNo_eq . . . 218
rxdock::isBondAmide . . . 225
rxdock::isBondCyclic . . . 225
rxdock::isBondPtr_eq . . . 226
rxdock::isBondRotatable . . . 226
rxdock::isBondSelected . . . 226
rxdock::isBondToNH3 . . . 227
rxdock::isBondToOH . . . 227
rxdock::isBond_eq . . . 224
rxdock::isCavityNearCoord . . . 228
rxdock::isCoordinationNumber_eq . . . 228
rxdock::isFFType_eq . . . 229
rxdock::isGenome_eq . . . 229
rxdock::isHHSSelected . . . 229
rxdock::isHHSType_eq . . . 230
rxdock::isHybridState_eq . . . 230
rxdock::isInteractionCenterSelected . . . 231
rxdock::isInteractionD_lt . . . 231
rxdock::isPiAtom . . . 231
rxdock::isSegmentName_eq . . . 232
rxdock::isSubunitId_eq . . . 232
rxdock::isSubunitName_eq . . . 233
rxdock::Genome . . . 177
rxdock::GenomeCmp_Score . . . 178
rxdock::geneticprogram::GPChromosome . . . 178
rxdock::geneticprogram::GPFitnessFunction . . . 187
rxdock::geneticprogram::GPFFCHK1 . . . 179
rxdock::geneticprogram::GPFFCrossDock . . . 181
rxdock::geneticprogram::GPFFGold . . . 182
rxdock::geneticprogram::GPFFHSP90 . . . 184
rxdock::geneticprogram::GPFFSpike . . . 186
rxdock::geneticprogram::GPGenome . . . 188
rxdock::geneticprogram::GPParser . . . 189
rxdock::geneticprogram::GPPopulation . . . 190
rxdock::HHS_Solvation . . . 196
rxdock::HHSType . . . 197
rxdock::TriposAtomType::info . . . 203
rxdock::InitHHS . . . 203
rxdock::InteractionCenter . . . 203
rxdock::InteractionCenterCmp . . . 204
rxdock::InteractionNode< T > . . . 210
rxdock::InvertSelectAtom . . . 212

rxdock::neldermead::IterationCriterion . . . 233
std::vector< T >::iterator
rxdock::InteractionNodeListMapIter< T > . . . 211
rxdock::MCStats . . . 240
rxdock::Model . . . 245
rxdock::ModelMutator . . . 248
rxdock::MOEGrid . . . 249
rxdock::MOEGridPoint . . . 249
rxdock::MOEGridShape . . . 249
rxdock::MOL2Substructure . . . 253
rxdock::NoeEndAtoms . . . 262
rxdock::NoeEndNames . . . 262
rxdock::NoeRestraintAtoms . . . 263
rxdock::NoeRestraintNames . . . 263
rxdock::Observer . . . 272
rxdock::BaseObject . . . 59
rxdock::BaseSF . . . 61
rxdock::BaseIdxSF . . . 44
rxdock::AromIdxSF . . . 28
rxdock::NmrSF . . . 258
rxdock::PMFIdxSF . . . 289
rxdock::PolarIdxSF . . . 295
rxdock::SAIdxSF . . . 343
rxdock::VdwIdxSF . . . 408
rxdock::BaseInterSF . . . 47
rxdock::AromIdxSF . . . 28
rxdock::CavityFillSF . . . 73
rxdock::CavityGridSF . . . 77
rxdock::ConstSF . . . 110
rxdock::DihedralTargetSF . . . 128
rxdock::NmrSF . . . 258
rxdock::PMFGridSF . . . 285
rxdock::PMFIdxSF . . . 289
rxdock::PharmaSF . . . 280
rxdock::PolarIdxSF . . . 295
rxdock::RotSF . . . 339
rxdock::SAIdxSF . . . 343
rxdock::SetupPMFSF . . . 349
rxdock::SetupPolarSF . . . 353
rxdock::SetupSASF . . . 356
rxdock::TetherSF . . . 392
rxdock::VdwGridSF . . . 404
rxdock::VdwIdxSF . . . 408
rxdock::BaseIntraSF . . . 50
rxdock::DihedralIntraSF . . . 123
rxdock::PolarIntraSF . . . 300
rxdock::VdwIntraSF . . . 413
rxdock::DihedralSF . . . 126
rxdock::DihedralIntraSF . . . 123
rxdock::DihedralTargetSF . . . 128
rxdock::PolarSF . . . 305
rxdock::SFAgg . . . 360
rxdock::VdwSF . . . 417
rxdock::BaseTransform . . . 64
rxdock::BaseBiMolTransform . . . 38
rxdock::AlignTransform . . . 23
rxdock::GATransform . . . 174
rxdock::RandPopTransform . . . 328

rxdock::SimAnnTransform . . . 367
rxdock::SimplexTransform . . . 373
rxdock::BaseUniMolTransform . . . 66
rxdock::RandLigTransform . . . 325
rxdock::NullTransform . . . 269
rxdock::TransformAgg . . . 397
rxdock::Filter . . . 154
rxdock::SiteMapper . . . 377
rxdock::LigandSiteMapper . . . 236
rxdock::SphereSiteMapper . . . 382
rxdock::OverlapHHS . . . 273
rxdock::OverlapVariableHHS . . . 274
rxdock::ParamHandler . . . 275
rxdock::BaseObject . . . 59
rxdock::FlexData . . . 172
rxdock::LigandFlexData . . . 234
rxdock::SolventFlexData . . . 380
rxdock::ReceptorFlexData . . . 334
rxdock::WorkSpace . . . 421
rxdock::BiMolWorkSpace . . . 69
rxdock::Parser . . . 276
rxdock::PartitionHHS . . . 276
rxdock::Plane . . . 284
rxdock::PMFValue . . . 294
rxdock::geneticprogram::PopCmp . . . 308
rxdock::Population . . . 309
rxdock::PrincipalAxes . . . 313
rxdock::PRMFactory . . . 316
rxdock::DihedralElement::prms . . . 317
rxdock::Quat . . . 324
rxdock::Rand . . . 325
rxdock::neldermead::RelativeValueCriterion< DataType > . . . 336
rxdock::Request . . . 336
rxdock::SFDisableRequest . . . 364
rxdock::SFEnableRequest . . . 365
rxdock::SFPartitionRequest . . . 366
rxdock::SFSetParamRequest . . . 366
rxdock::RequestHandler . . . 337
rxdock::BaseObject . . . 59
rxdock::RestoreHHS . . . 337
rxdock::RotateAtomUsingQuat . . . 339
rxdock::RotateAtomUsingQuatIfSelected . . . 339
rxdock::SaveHHS . . . 347
rxdock::SelectAtom . . . 348
rxdock::SelectFlexAtoms . . . 348
rxdock::SelectInteractionCenter . . . 349
rxdock::SFFactory . . . 365
rxdock::SimpleSolvationParameters . . . 371
rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion > . . . 371
rxdock::SimplexCostFunction . . . 372
rxdock::Singleton< singleton_type > . . . 376
rxdock::SiteMapperFactory . . . 379
rxdock::SmartPtr< T > . . . 379
rxdock::SmartPtr< Atom > . . . 379
rxdock::SmartPtr< BaseMolecularFileSink > . . . 379
rxdock::SmartPtr< BiMolWorkSpace > . . . 379
rxdock::SmartPtr< Bond > . . . 379

rxdock::SmartPtr< CellContext > . . . 379
rxdock::SmartPtr< CharmmDataSource > . . . 379
rxdock::SmartPtr< ChromDihedralRefData > . . . 379
rxdock::SmartPtr< ChromElement > . . . 379
rxdock::SmartPtr< ChromOccupancyRefData > . . . 379
rxdock::SmartPtr< ChromPositionRefData > . . . 379
rxdock::SmartPtr< Context > . . . 379
rxdock::SmartPtr< DockingSite > . . . 379
rxdock::SmartPtr< ElementFileSource > . . . 379
rxdock::SmartPtr< FFTGrid > . . . 379
rxdock::SmartPtr< Filter > . . . 379
rxdock::SmartPtr< FilterExpression > . . . 379
rxdock::SmartPtr< GPChromosome > . . . 379
rxdock::SmartPtr< GPFitnessFunction > . . . 379
rxdock::SmartPtr< InteractionGrid > . . . 379
rxdock::SmartPtr< InteractionGridTemplate< T > > . . . 379
rxdock::InteractionGridTemplatePtr< T > . . . 209
rxdock::SmartPtr< MCStats > . . . 379
rxdock::SmartPtr< Model > . . . 379
rxdock::SmartPtr< ModelMutator > . . . 379
rxdock::SmartPtr< NonBondedGrid > . . . 379
rxdock::SmartPtr< NonBondedHHSGrid > . . . 379
rxdock::SmartPtr< ParameterFileSource > . . . 379
rxdock::SmartPtr< Population > . . . 379
rxdock::SmartPtr< RealGrid > . . . 379
rxdock::SmartPtr< Request > . . . 379
rxdock::SmartPtr< SFAgg > . . . 379
rxdock::SmartPtr< std::istream > . . . 379
rxdock::SmartPtr< Token > . . . 379
rxdock::neldermead::State< DataType, ParameterType > . . . 384
rxdock::StdRestraintAtoms . . . 385
rxdock::StdRestraintNames . . . 385
rxdock::Subject . . . 391
rxdock::WorkSpace . . . 421
rxdock::SymBond . . . 391
testing::Test
rxdock::unittest::ChromTest . . . 106
rxdock::unittest::OccupancyTest . . . 273
rxdock::unittest::SearchTest . . . 348
rxdock::Token . . . 396
rxdock::TokenIter . . . 396
rxdock::CellTokenIter . . . 83
rxdock::StringTokenIter . . . 387
rxdock::TransformFactory . . . 401
rxdock::TranslateAtom . . . 401
rxdock::TranslateAtomIfSelected . . . 401
rxdock::TriposAtomType . . . 401
rxdock::Variant . . . 402
rxdock::Vble . . . 403
std::vector< T >
rxdock::InteractionNodeList< T > . . . 210
rxdock::InteractionNodeListMap< T > . . . 210

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

rxdock::AddCommand . . . 21
rxdock::AlignTransform . . . 23
rxdock::neldermead::AndCriteria< Criteria1, Criteria2 > . . . 26
rxdock::Ann_Cmp_AtomId2 . . . 26
rxdock::Annotation . . . 27
rxdock::AnnotationHandler . . . 27
rxdock::AromIdxSF . . . 28
rxdock::Assertion . . . 32
rxdock::Atom . . . 33
rxdock::AtomPtrCmp_AtomicNo . . . 35
rxdock::AtomPtrCmp_AtomId . . . 36
rxdock::AtomPtrCmp_AtomName . . . 36
rxdock::AtomPtrCmp_Ptr . . . 36
rxdock::BadArgument . . . 36
rxdock::BadReceptorFile . . . 37
rxdock::BaseBiMolTransform . . . 38
rxdock::BaseFileSink . . . 41
rxdock::BaseFileSource . . . 41
rxdock::BaseGrid . . . 42
rxdock::BaseIdxSF . . . 44
rxdock::BaseInterSF . . . 47
rxdock::BaseIntraSF . . . 50
rxdock::BaseMolecularDataSource . . . 53
rxdock::BaseMolecularFileSink . . . 54
rxdock::BaseMolecularFileSource . . . 55
rxdock::BaseObject . . . 59
rxdock::BaseSF . . . 61
rxdock::BaseTransform . . . 64
rxdock::BaseUniMolTransform . . . 66
rxdock::BiMolWorkSpace . . . 69
rxdock::Bond . . . 71
rxdock::BondPCmp_BondId . . . 72
rxdock::Cavity . . . 72
rxdock::CavityFillSF . . . 73
rxdock::CavityGridSF . . . 77

rxdock::CavityPtrCmp_Distance . . . 81
rxdock::CavityPtrCmp_Volume . . . 81
rxdock::Cell . . . 81
rxdock::CellContext . . . 82
rxdock::CellTokenIter . . . 83
rxdock::CharmmDataSource . . . 84
rxdock::CharmmType . . . 85
rxdock::CharmmTypesFileSource . . . 85
rxdock::Chrom . . . 86
rxdock::ChromDihedralElement . . . 90
rxdock::ChromDihedralRefData . . . 94
rxdock::ChromElement . . . 95
rxdock::ChromFactory . . . 96
rxdock::ChromOccupancyElement . . . 97
rxdock::ChromOccupancyRefData . . . 101
rxdock::ChromPositionElement . . . 102
rxdock::ChromPositionRefData . . . 106
rxdock::unittest::ChromTest . . . 106
rxdock::Command . . . 107
rxdock::Commands . . . 108
rxdock::Constraint . . . 109
rxdock::ConstSF . . . 110
rxdock::Context . . . 114
rxdock::Coord . . . 114
rxdock::CoordCmp . . . 115
rxdock::CrdFileSink . . . 115
rxdock::CrdFileSource . . . 117
rxdock::CSVFileSink . . . 120
rxdock::CyclicAtom . . . 122
rxdock::DihedralElement . . . 122
rxdock::DihedralIntraSF . . . 123
rxdock::DihedralSF . . . 126
rxdock::DihedralTargetSF . . . 128
rxdock::DirectorySource . . . 132
rxdock::DirIsNotAccessible . . . 133
rxdock::DivCommand . . . 134
rxdock::DockingError . . . 135
rxdock::DockingSite . . . 136
rxdock::ElementData . . . 137
rxdock::ElementFileSource . . . 137
rxdock::EnumerateSymCoords . . . 139
rxdock::EnvNotDefined . . . 139
rxdock::Error . . . 140
rxdock::Euler . . . 141
rxdock::EvaluateVisitor . . . 141
rxdock::ExpCommand . . . 144
rxdock::PolarSF::f1prms . . . 145
rxdock::FFTGrid . . . 146
rxdock::FFTPeak . . . 149
rxdock::FileError . . . 149
rxdock::FileMissingParameter . . . 150
rxdock::FileParseError . . . 151
rxdock::FileReadError . . . 152
rxdock::FileWriteError . . . 153
rxdock::Filter . . . 154
rxdock::FilterAddExp . . . 156
rxdock::FilterAndExp . . . 158
rxdock::FilterDivExp . . . 159

rxdock::FilterExpExp . . . 161
rxdock::FilterExpression . . . 162
rxdock::FilterExpressionVisitor . . . 163
rxdock::FilterIfExp . . . 164
rxdock::FilterLogExp . . . 165
rxdock::FilterMulExp . . . 167
rxdock::FilterSubExp . . . 168
rxdock::FilterVbleExp . . . 170
rxdock::FlexAtomFactory . . . 171
rxdock::FlexData . . . 172
rxdock::FlexDataVisitor . . . 173
rxdock::GATransform . . . 174
rxdock::Genome . . . 177
rxdock::GenomeCmp_Score . . . 178
rxdock::geneticprogram::GPChromosome . . . 178
rxdock::geneticprogram::GPFFCHK1 . . . 179
rxdock::geneticprogram::GPFFCrossDock . . . 181
rxdock::geneticprogram::GPFFGold . . . 182
rxdock::geneticprogram::GPFFHSP90 . . . 184
rxdock::geneticprogram::GPFFSpike . . . 186
rxdock::geneticprogram::GPFitnessFunction . . . 187
rxdock::geneticprogram::GPGenome . . . 188
rxdock::geneticprogram::GPParser . . . 189
rxdock::geneticprogram::GPPopulation . . . 190
rxdock::HBAConstraint . . . 190
rxdock::HBDConstraint . . . 192
rxdock::CavityGridSF::HeavyAtomFactory . . . 193
rxdock::HeavyConstraint . . . 194
rxdock::HHS_Solvation . . . 196
rxdock::HHSType . . . 197
rxdock::HydroAliphaticConstraint . . . 197
rxdock::HydroAromaticConstraint . . . 199
rxdock::HydroConstraint . . . 200
rxdock::IfCommand . . . 201
rxdock::TriposAtomType::info . . . 203
rxdock::InitHHS . . . 203
rxdock::InteractionCenter . . . 203
rxdock::InteractionCenterCmp . . . 204
rxdock::InteractionGrid . . . 204
rxdock::InteractionGridTemplate< T > . . . 207
rxdock::InteractionGridTemplatePtr< T > . . . 209
rxdock::InteractionNode< T > . . . 210
rxdock::InteractionNodeList< T > . . . 210
rxdock::InteractionNodeListMap< T > . . . 210
rxdock::InteractionNodeListMapConstIter< T > . . . 211
rxdock::InteractionNodeListMapIter< T > . . . 211
rxdock::InvalidRequest . . . 211
rxdock::InvertSelectAtom . . . 212
rxdock::isAtom_12Connected . . . 212
rxdock::isAtom_13Connected . . . 213
rxdock::isAtom_eq . . . 213
rxdock::isAtomAnionic . . . 214
rxdock::isAtomBridgehead . . . 214
rxdock::isAtomCationic . . . 214
rxdock::isAtomCharged . . . 215
rxdock::isAtomCyclic . . . 215
rxdock::isAtomEnabled . . . 216
rxdock::isAtomExtended . . . 216

rxdock::isAtomGuanidiniumCarbon . . . 216
rxdock::isAtomHBondAcceptor . . . 217
rxdock::isAtomHBondDonor . . . 217
rxdock::isAtomicNo_eq . . . 218
rxdock::isAtomId_eq . . . 218
rxdock::DockingSite::isAtomInRange . . . 218
rxdock::isAtomInsideCuboid . . . 219
rxdock::isAtomInsideSphere . . . 219
rxdock::isAtomIonic . . . 220
rxdock::isAtomLipophilic . . . 220
rxdock::isAtomMetal . . . 220
rxdock::isAtomName_eq . . . 221
rxdock::isAtomNearCoordList . . . 221
rxdock::isAtomNegCharged . . . 222
rxdock::isAtomNitroN . . . 222
rxdock::isAtomPlanar . . . 222
rxdock::isAtomPosCharged . . . 223
rxdock::isAtomPtr_eq . . . 223
rxdock::isAtomRNA . . . 224
rxdock::isAtomSelected . . . 224
rxdock::isBond_eq . . . 224
rxdock::isBondAmide . . . 225
rxdock::isBondCyclic . . . 225
rxdock::isBondPtr_eq . . . 226
rxdock::isBondRotatable . . . 226
rxdock::isBondSelected . . . 226
rxdock::isBondToNH3 . . . 227
rxdock::isBondToOH . . . 227
rxdock::isCavityNearCoord . . . 228
rxdock::isCoordinationNumber_eq . . . 228
rxdock::isFFType_eq . . . 229
rxdock::isGenome_eq . . . 229
rxdock::isHHSSelected . . . 229
rxdock::isHHSType_eq . . . 230
rxdock::isHybridState_eq . . . 230
rxdock::isInteractionCenterSelected . . . 231
rxdock::isInteractionD_lt . . . 231
rxdock::isPiAtom . . . 231
rxdock::isSegmentName_eq . . . 232
rxdock::isSubunitId_eq . . . 232
rxdock::isSubunitName_eq . . . 233
rxdock::neldermead::IterationCriterion . . . 233
rxdock::LigandError . . . 233
rxdock::LigandFlexData . . . 234
rxdock::LigandSiteMapper . . . 236
rxdock::LogCommand . . . 238
rxdock::MCStats . . . 240
rxdock::MdlFileSink . . . 240
rxdock::MdlFileSource . . . 242
rxdock::Model . . . 245
rxdock::ModelError . . . 247
rxdock::ModelMutator . . . 248
rxdock::MOEGrid . . . 249
rxdock::MOEGridPoint . . . 249
rxdock::MOEGridShape . . . 249
rxdock::MOL2FileSource . . . 250
rxdock::MOL2Substructure . . . 253
rxdock::MulCommand . . . 254

rxdock::NegChargeConstraint . . . 255
rxdock::NmrRestraintFileSource . . . 257
rxdock::NmrSF . . . 258
rxdock::NoeEndAtoms . . . 262
rxdock::NoeEndNames . . . 262
rxdock::NoeRestraintAtoms . . . 263
rxdock::NoeRestraintNames . . . 263
rxdock::NoFileInDir . . . 264
rxdock::NonBondedGrid . . . 265
rxdock::NonBondedHHSGrid . . . 267
rxdock::NullTransform . . . 269
rxdock::Observer . . . 272
rxdock::unittest::OccupancyTest . . . 273
rxdock::OverlapHHS . . . 273
rxdock::OverlapVariableHHS . . . 274
rxdock::ParameterFileSource . . . 274
rxdock::ParamHandler . . . 275
rxdock::Parser . . . 276
rxdock::PartitionHHS . . . 276
rxdock::PdbFileSource . . . 277
rxdock::PharmaSF . . . 280
rxdock::Plane . . . 284
rxdock::PMFDirSource . . . 284
rxdock::PMFGridSF . . . 285
rxdock::PMFIdxSF . . . 289
rxdock::PMFValue . . . 294
rxdock::PolarIdxSF . . . 295
rxdock::PolarIntraSF . . . 300
rxdock::PolarSF . . . 305
rxdock::geneticprogram::PopCmp . . . 308
rxdock::Population . . . 309
rxdock::PosChargeConstraint . . . 309
rxdock::PrettyPrintVisitor . . . 311
rxdock::PrincipalAxes . . . 313
rxdock::PrintVisitor . . . 313
rxdock::PRMFactory . . . 316
rxdock::DihedralElement::prms . . . 317
rxdock::PseudoAtom . . . 317
rxdock::PsfFileSink . . . 319
rxdock::PsfFileSource . . . 321
rxdock::Quat . . . 324
rxdock::Rand . . . 325
rxdock::RandLigTransform . . . 325
rxdock::RandPopTransform . . . 328
rxdock::RealGrid . . . 332
rxdock::ReceptorFlexData . . . 334
rxdock::neldermead::RelativeValueCriterion< DataType > . . . 336
rxdock::Request . . . 336
rxdock::RequestHandler . . . 337
rxdock::RestoreHHS . . . 337
rxdock::RingAromaticConstraint . . . 337
rxdock::RotateAtomUsingQuat . . . 339
rxdock::RotateAtomUsingQuatIfSelected . . . 339
rxdock::RotSF . . . 339
rxdock::SAIdxSF . . . 343
rxdock::SaveHHS . . . 347
rxdock::unittest::SearchTest . . . 348
rxdock::SelectAtom . . . 348

rxdock::SelectFlexAtoms . . . 348
rxdock::SelectInteractionCenter . . . 349
rxdock::SetupPMFSF . . . 349
rxdock::SetupPolarSF . . . 353
rxdock::SetupSASF . . . 356
rxdock::SFAgg . . . 360
rxdock::SFDisableRequest . . . 364
rxdock::SFEnableRequest . . . 365
rxdock::SFFactory . . . 365
rxdock::SFPartitionRequest . . . 366
rxdock::SFSetParamRequest . . . 366
rxdock::SimAnnTransform . . . 367
rxdock::SimpleSolvationParameters . . . 371
rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion > . . . 371
rxdock::SimplexCostFunction . . . 372
rxdock::SimplexTransform . . . 373
rxdock::Singleton< singleton_type > . . . 376
rxdock::SiteMapper . . . 377
rxdock::SiteMapperFactory . . . 379
rxdock::SmartPtr< T > . . . 379
rxdock::SolventFlexData . . . 380
rxdock::SphereSiteMapper . . . 382
rxdock::neldermead::State< DataType, ParameterType > . . . 384
rxdock::StdRestraintAtoms . . . 385
rxdock::StdRestraintNames . . . 385
rxdock::StringContext . . . 386
rxdock::StringTokenIter . . . 387
rxdock::StringTooLong . . . 388
rxdock::SubCommand . . . 389
rxdock::Subject . . . 391
rxdock::SymBond . . . 391
rxdock::TetherSF . . . 392
rxdock::Token . . . 396
rxdock::TokenIter . . . 396
rxdock::TransformAgg . . . 397
rxdock::TransformFactory . . . 401
rxdock::TranslateAtom . . . 401
rxdock::TranslateAtomIfSelected . . . 401
rxdock::TriposAtomType . . . 401
rxdock::Variant . . . 402
rxdock::Vble . . . 403
rxdock::VdwGridSF . . . 404
rxdock::VdwIdxSF . . . 408
rxdock::VdwIntraSF . . . 413
rxdock::VdwSF . . . 417
rxdock::WorkSpace . . . 421

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

AlignTransform.h . . . 423
Annotation.h . . . 424
AnnotationHandler.h . . . 425
AromIdxSF.h . . . 426
Atom.h . . . 428
AtomFuncs.h . . . 442
BaseBiMolTransform.h . . . 443
BaseFileSink.h . . . 444
BaseFileSource.h . . . 445
BaseGrid.h . . . 446
BaseIdxSF.h . . . 449
BaseInterSF.h . . . 450
BaseIntraSF.h . . . 451
BaseMolecularDataSource.h . . . 452
BaseMolecularFileSink.h . . . 453
BaseMolecularFileSource.h . . . 455
BaseObject.h . . . 456
BaseSF.h . . . 458
BaseTransform.h . . . 459
BaseUniMolTransform.h . . . 460
BiMolWorkSpace.h . . . 461
Bond.h . . . 462
Cavity.h . . . 465
CavityFillSF.h . . . 467
CavityGridSF.h . . . 468
CellTokenIter.h . . . 469
CharmmDataSource.h . . . 470
CharmmTypesFileSource.h . . . 471
Chrom.h . . . 472
ChromDihedralElement.h . . . 473
ChromDihedralRefData.h . . . 474
ChromElement.h . . . 475
ChromFactory.h . . . 477
ChromOccupancyElement.h . . . 478
ChromOccupancyRefData.h . . . 479

ChromPositionElement.h . . . 479
ChromPositionRefData.h . . . 480
Commands.h . . . 482
Config.h . . . 483
Constraint.h . . . 483
ConstSF.h . . . 485
Context.h . . . 486
Coord.h . . . 488
CrdFileSink.h . . . 493
CrdFileSource.h . . . 494
CSVFileSink.h . . . 494
Smarts.h . . . 495
Debug.h . . . 496
DihedralIntraSF.h . . . 497
DihedralSF.h . . . 497
DihedralTargetSF.h . . . 499
DirectorySource.h . . . 500
DockingError.h . . . 500
DockingSite.h . . . 501
ElementFileSource.h . . . 503
Error.h . . . 504
Euler.h . . . 506
FFTGrid.h . . . 507
FileError.h . . . 509
Filter.h . . . 510
FilterExpression.h . . . 511
FilterExpressionVisitor.h . . . 514
FlexAtomFactory.h . . . 516
FlexData.h . . . 517
FlexDataVisitor.h . . . 517
GATransform.h . . . 518
Cell.h . . . 519
Command.h . . . 520
GPChromosome.h . . . 522
GPFFCHK1.h . . . 523
GPFFCrossDock.h . . . 524
GPFFGold.h . . . 525
GPFFHSP90.h . . . 525
GPFFSpike.h . . . 526
GPFitnessFunction.h . . . 527
GPGenome.h . . . 528
GPParser.h . . . 529
GPPopulation.h . . . 530
GPTypes.h . . . 531
Genome.h . . . 531
InteractionGrid.h . . . 533
InteractionTemplate.h . . . 535
LigandError.h . . . 537
LigandFlexData.h . . . 538
LigandSiteMapper.h . . . 539
MdlFileSink.h . . . 539
MdlFileSource.h . . . 541
Model.h . . . 542
ModelError.h . . . 546
ModelMutator.h . . . 547
MOEGrid.h . . . 548
MOL2FileSource.h . . . 551
NMCriteria.h . . . 552

NmrRestraintFileSource.h . . . 553
NmrSF.h . . . 555
NMSimplex.h . . . 556
NMState.h . . . 558
NoeRestraint.h . . . 559
NonBondedGrid.h . . . 561
NonBondedHHSGrid.h . . . 563
NullTransform.h . . . 564
Observer.h . . . 564
CavitySearch.h . . . 565
Dock.h . . . 566
Tabularize.h . . . 567
Transform.h . . . 568
ParameterFileSource.h . . . 569
ParamHandler.h . . . 570
Parser.h . . . 572
PdbFileSource.h . . . 572
PharmaSF.h . . . 573
Plane.h . . . 574
PMF.h . . . 575
PMFDirSource.h . . . 576
PMFGridSF.h . . . 577
PMFIdxSF.h . . . 578
PolarIdxSF.h . . . 579
PolarIntraSF.h . . . 580
PolarSF.h . . . 581
Population.h . . . 583
PrincipalAxes.h . . . 584
PRMFactory.h . . . 585
PseudoAtom.h . . . 586
PsfFileSink.h . . . 587
PsfFileSource.h . . . 588
Quat.h . . . 589
Rand.h . . . 592
RandLigTransform.h . . . 592
RandPopTransform.h . . . 593
Rbt.h . . . 594
RealGrid.h . . . 596
ReceptorFlexData.h . . . 599
Request.h . . . 599
RequestHandler.h . . . 600
Resources.h . . . 601
RotSF.h . . . 601
SAIdxSF.h . . . 602
SATypes.h . . . 603
SetupPMFSF.h . . . 607
SetupPolarSF.h . . . 608
SetupSASF.h . . . 608
SFAgg.h . . . 609
SFFactory.h . . . 611
SFRequest.h . . . 612
SimAnnTransform.h . . . 613
SimplexCostFunction.h . . . 614
SimplexTransform.h . . . 615
Singleton.h . . . 616
SiteMapper.h . . . 616
SiteMapperFactory.h . . . 617
SmartPointer.h . . . 618

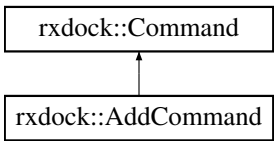
SolventFlexData.h . . . 621
SphereSiteMapper.h . . . 621
StringTokenIter.h . . . 622
Subject.h . . . 623
Export.h . . . 623
Number.h . . . 624
Quote.h . . . 625
TetherSF.h . . . 626
Token.h . . . 627
TokenIter.h . . . 628
TransformAgg.h . . . 629
TransformFactory.h . . . 630
TriposAtomType.h . . . 631
Variant.h . . . 633
Vble.h . . . 635
VdwGridSF.h . . . 636
VdwIdxSF.h . . . 637
VdwIntraSF.h . . . 639
VdwSF.h . . . 639
WorkSpace.h . . . 642
CSVFileSink.cxx . . . 643
CavitySearch.cxx . . . 644
Dock.cxx . . . 644
Tabularize.cxx . . . 645
Transform.cxx . . . 645
Number.cxx . . . 645
Quote.cxx . . . 645
ChromTest.h . . . 646
OccupancyTest.h . . . 646
SearchTest.h . . . 647
ParseCavitySearch.cxx . . . 648
ParseCavitySearch.h . . . 648
ParseDock.cxx . . . 649
ParseDock.h . . . 649
ParseTabularize.cxx . . . 649
ParseTabularize.h . . . 650
ParseTransform.cxx . . . 650
ParseTransform.h . . . 651
rxcmd.cxx . . . 651

Chapter 5

Class Documentation

5.1 rxdock::AddCommand Class Reference

Inheritance diagram for rxdock::AddCommand:



Public Member Functions

- int GetNArgs ()
- std::string GetName ()
- void SetArg (int i, ReturnType f)
- void SetNameArg (int i, std::string &n)
- ReturnType Execute ()

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Additional Inherited Members

Static Public Member Functions inherited from rxdock::Command

- static void **Clear** ()

Static Public Attributes inherited from rxdock::Command

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions inherited from rxdock::Command

- std::string **tabs** (int n)

5.1.1 Member Function Documentation

5.1.1.1 Execute()

```
ReturnType rxdock::AddCommand::Execute ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.1.1.2 GetName()

```
std::string rxdock::AddCommand::GetName ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.1.1.3 GetNArgs()

```
int rxdock::AddCommand::GetNArgs ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.1.1.4 SetArg()

```
void rxdock::AddCommand::SetArg (
            int i,
            ReturnType f )  [inline], [virtual]
```

Implements rxdock::Command.

5.1.1.5 SetNameArg()

```
void rxdock::AddCommand::SetNameArg (
            int i,
            std::string & n )  [inline], [virtual]
```

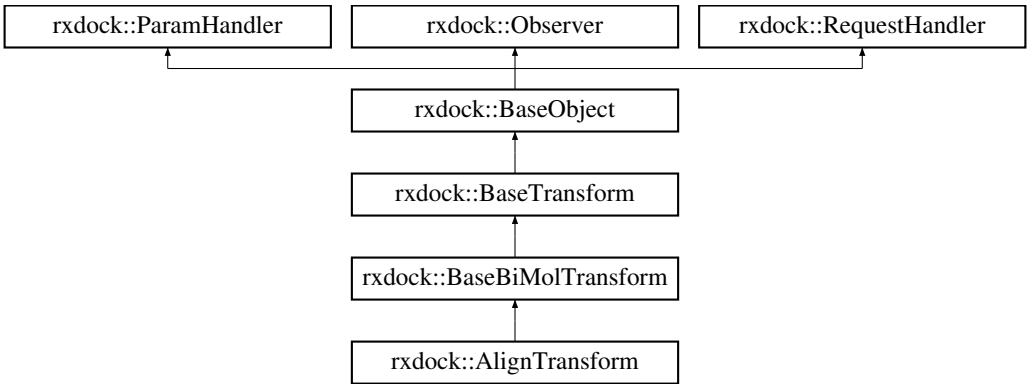
Implements rxdock::Command.

The documentation for this class was generated from the following file:

- Command.h

5.2 rxdock::AlignTransform Class Reference

Inheritance diagram for rxdock::AlignTransform:



Public Member Functions

- **AlignTransform** (const std::string &strName="ALIGN")

Public Member Functions inherited from rxdock::BaseBiMolTransform

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform *)
- virtual void **Remove** (BaseTransform *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform * **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform * **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨AlignTransform¨
- static const std::string **_COM** = ¨center-of-mass¨
- static const std::string **_AXES** = ¨axes¨

Static Public Attributes inherited from rxdock::BaseBiMolTransform

- static const std::string **_CT** = ¨BaseBiMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupTransform ()
- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseBiMolTransform

- **BaseBiMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupReceptor** ()=0
- virtual void **SetupLigand** ()=0
- virtual void **SetupSolvent** ()
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const AlignTransform &alignTransform)
- void **from_json** (const json &j, AlignTransform &alignTransform)

5.2.1 Member Function Documentation

5.2.1.1 Execute()

```
void AlignTransform::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.2.1.2 SetupLigand()

```
void AlignTransform::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.2.1.3 SetupReceptor()

```
void AlignTransform::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.2.1.4 SetupTransform()

```
void AlignTransform::SetupTransform ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

The documentation for this class was generated from the following files:

- AlignTransform.h
- AlignTransform.cxx

5.3 rxdock::neldermead::AndCriteria< Criteria1, Criteria2 > Class Template Reference

Public Member Functions

- **AndCriteria** (const Criteria1 &criteria1, const Criteria2 &criteria2)
- template<class State >
  bool **operator()** (const State &state)

The documentation for this class was generated from the following file:

- NMCriteria.h

5.4 rxdock::Ann_Cmp_AtomId2 Class Reference

Public Member Functions

- bool **operator()** (Annotation *pAnn1, Annotation *pAnn2) const

The documentation for this class was generated from the following file:

- Annotation.h

5.5 rxdock::Annotation Class Reference

Public Member Functions

- **Annotation** (const Atom ∗pAtom1, const Atom ∗pAtom2, double dist, double score)
- **Annotation** (json j)
- const Atom ∗ **GetAtom1Ptr** () const
- const Atom ∗ **GetAtom2Ptr** () const
- double **GetDistance** () const
- double **GetScore** () const
- std::string **GetFQResName** () const
- void **SetAtom1Ptr** (const Atom ∗pAt1)
- void **SetAtom2Ptr** (const Atom ∗pAt2)
- void **SetDistance** (double d)
- void **SetScore** (double s)
- std::string **Render** () const
- void **operator+=** (const Annotation &ann)

Friends

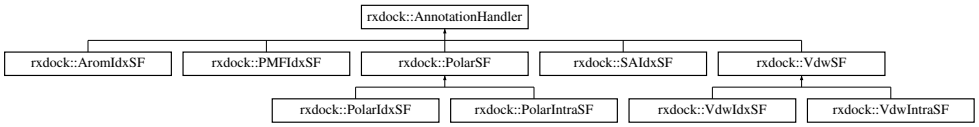
- void **to_json** (json &j, const Annotation &annotation)
- void **from_json** (const json &j, Annotation &annotation)

The documentation for this class was generated from the following files:

- Annotation.h
- Annotation.cxx

5.6 rxdock::AnnotationHandler Class Reference

Inheritance diagram for rxdock::AnnotationHandler:



Public Member Functions

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Attributes

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

Friends

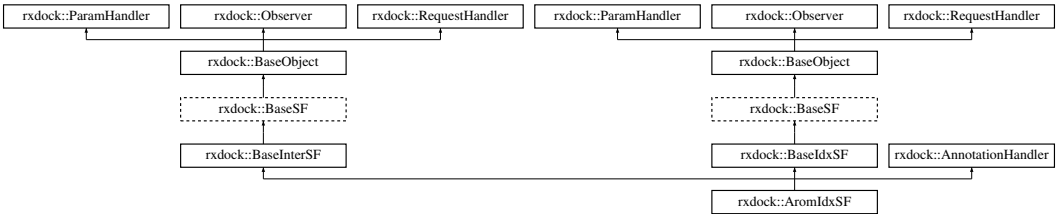
- void **to_json** (json &j, const AnnotationHandler &annotationHandler)
- void **from_json** (const json &j, AnnotationHandler &annotationHandler)

The documentation for this class was generated from the following files:

- AnnotationHandler.h
- AnnotationHandler.cxx

5.7 rxdock::AromIdxSF Class Reference

Inheritance diagram for rxdock::AromIdxSF:



Public Member Functions

- **AromIdxSF** (const std::string &strName="arom")

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseIdxSF

- double **GetGridStep** () const
- void **SetGridStep** (double step)
- double **GetBorder** () const
- void **SetBorder** (double border)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Attributes

- static const std::string **_CT** = ¨AromIdxSF¨
- static const std::string **_INCR** = ¨increment¨
- static const std::string **_R12** = ¨r12¨
- static const std::string **_DR12MIN** = ¨dr12-minimum¨
- static const std::string **_DR12MAX** = ¨dr12-maximum¨
- static const std::string **_DAMIN** = ¨da-minimum¨
- static const std::string **_DAMAX** = ¨da-maximum¨
- static const std::string **_THRESHOLD** = ¨threshold¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::BaseIdxSF

- static const std::string **_CT** = ¨BaseIdxSF¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_BORDER** = ¨border¨

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void **ClearReceptor** ()
- void **ClearLigand** ()
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseIdxSF

- InteractionGridPtr **CreateInteractionGrid** () const
- NonBondedGridPtr **CreateNonBondedGrid** () const
- NonBondedHHSGridPtr **CreateNonBondedHHSGrid** () const
- double **GetMaxError** () const
- double **GetCorrectedRange** () const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

Friends

- void **to_json** (json &j, const AromIdxSF &aromIdxSF)
- void **from_json** (const json &j, AromIdxSF &aromIdxSF)

5.7.1 Member Function Documentation

5.7.1.1 ParameterUpdated()

```
void AromIdxSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::ParamHandler.

5.7.1.2 RawScore()

```
double AromIdxSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.7.1.3 SetupLigand()

```
void AromIdxSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.7.1.4 SetupReceptor()

```
void AromIdxSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.7.1.5 SetupScore()

```
void AromIdxSF::SetupScore ( )  [protected], [virtual]
```

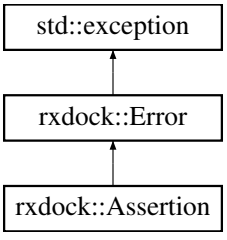
Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- AromIdxSF.h
- AromIdxSF.cxx

5.8 rxdock::Assertion Class Reference

Inheritance diagram for rxdock::Assertion:



Additional Inherited Members

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Protected Member Functions inherited from rxdock::Error

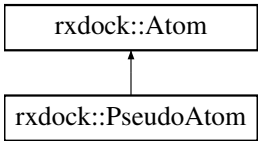
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- Error.h

5.9 rxdock::Atom Class Reference

Inheritance diagram for rxdock::Atom:



Public Types

- enum **eHybridState** {
  UNDEFINED = 0 , **SP** = 1 , **SP2** = 2 , **SP3** = 3 ,
  AROM = 4 , **TRI** = 5 }

Public Member Functions

- RBTDLL_EXPORT **Atom** (int nAtomId, int nAtomicNo=6, std::string strAtomName="C", std::string strSubunitId="1", std::string strSubunitName="RES", std::string strSegmentName="SEG1", eHybridState eState=UNDEFINED, unsigned int nHydrogens=0, int nFormalCharge=0)
- **Atom** (json j)
- **Atom** (const Atom &atom)
- Atom & **operator=** (const Atom &atom)
- virtual std::ostream & **Print** (std::ostream &s) const
- RBTDLL_EXPORT bool **GetEnabled** () const
- int **GetAtomId** () const
- void **SetAtomId** (const int nAtomId)
- int **GetAtomicNo** () const
- void **SetAtomicNo** (const int nAtomicNo)
- std::string **GetName** () const
- void **SetAtomName** (const std::string &strAtomName)
- RBTDLL_EXPORT std::string **GetFullAtomName** () const
- std::string **GetSubunitId** () const
- void **SetSubunitId** (const std::string &strSubunitId)
- std::string **GetSubunitName** () const
- void **SetSubunitName** (const std::string &strSubunitName)
- std::string **GetSegmentName** () const
- void **SetSegmentName** (const std::string &strSegmentName)
- eHybridState **GetHybridState** () const
- void **SetHybridState** (const eHybridState eState)
- unsigned int **GetNumImplicitHydrogens** () const
- void **SetNumImplicitHydrogens** (const unsigned int nHydrogens)
- int **GetFormalCharge** () const
- void **SetFormalCharge** (const int nFormalCharge)
- bool **GetCyclicFlag** () const
- void **SetCyclicFlag** (bool bCyclic=true)
- bool **GetSelectionFlag** () const
- void **SetSelectionFlag** (bool bSelected=true)
- void **InvertSelectionFlag** ()
- bool **GetUser1Flag** () const
- void **SetUser1Flag** (bool bUser1=true)
- double **GetUser1Value** () const
- void **SetUser1Value** (double dUser1=0.0)
- double **GetUser2Value** () const
- void **SetUser2Value** (double dUser2=0.0)
- Model * **GetModelPtr** () const
- void **SetModelPtr** (Model *pModel=nullptr)
- unsigned int **GetNumBonds** () const
- const BondMap & **GetBondMap** () const
- bool **AddBond** (Bond *pBond)
- bool **RemoveBond** (Bond *pBond)
- unsigned int **GetNumCyclicBonds** () const
- BondMap **GetCyclicBondMap** () const
- int **GetTotalFormalBondOrder** () const
- int **GetMaxFormalBondOrder** () const
- const Coord & **GetCoords** () const
- double **GetX** () const
- double **GetY** () const
- double **GetZ** () const
- void **SetCoords** (const Coord &coord)

- void **SetCoords** (const double x, const double y, const double z)
- void **SetX** (const double x)
- void **SetY** (const double y)
- void **SetZ** (const double z)
- double **GetPartialCharge** () const
- void **SetPartialCharge** (const double dPartialCharge)
- double **GetGroupCharge** () const
- void **SetGroupCharge** (const double dGroupCharge)
- double **GetAtomicMass** () const
- void **SetAtomicMass** (const double dAtomicMass)
- double **GetVdwRadius** () const
- void **SetVdwRadius** (const double dVdwRadius)
- std::string **GetFFType** () const
- void **SetFFType** (const std::string &strFFType)
- PMFType **GetPMFType** () const
- void **SetPMFType** (PMFType aType)
- TriposAtomType::eType **GetTriposType** () const
- void **SetTriposType** (TriposAtomType::eType aType)
- void **SaveCoords** (unsigned int coordNum=0)
- void **RevertCoords** (unsigned int coordNum=0)
- void **Translate** (const Vector &vector)
- void **Translate** (const double vx, const double vy, const double vz)
- void **RotateUsingQuat** (const Quat &q)
- unsigned int **GetCoordinationNumber** () const
- unsigned int **GetCoordinationNumber** (int nAtomicNo) const
- unsigned int **GetCoordinationNumber** (const std::string &strFFType) const
- unsigned int **GetCoordinationNumber** (eHybridState e) const

Friends

- RBTDLL_EXPORT friend std::ostream & **operator**<< (std::ostream &s, const Atom &atom)
- void **to_json** (json &j, const Atom &atom)
- void **from_json** (const json &j, Atom &atom)

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.10 rxdock::AtomPtrCmp_AtomicNo Class Reference

Public Member Functions

- bool **operator()** (Atom *pAtom1, Atom *pAtom2) const

The documentation for this class was generated from the following file:

- Atom.h

5.11 rxdock::AtomPtrCmp_AtomId Class Reference

Public Member Functions

- bool **operator()** (Atom ∗pAtom1, Atom ∗pAtom2) const

The documentation for this class was generated from the following file:

- Atom.h

5.12 rxdock::AtomPtrCmp_AtomName Class Reference

Public Member Functions

- bool **operator()** (Atom ∗pAtom1, Atom ∗pAtom2) const

The documentation for this class was generated from the following file:

- Atom.h

5.13 rxdock::AtomPtrCmp_Ptr Class Reference

Public Member Functions

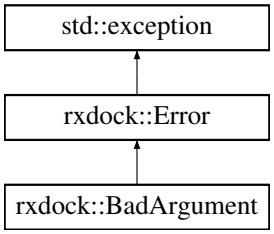
- bool **operator()** (Atom ∗pAtom1, Atom ∗pAtom2) const

The documentation for this class was generated from the following file:

- Atom.h

5.14 rxdock::BadArgument Class Reference

Inheritance diagram for rxdock::BadArgument:



Public Member Functions

- **BadArgument** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::Error

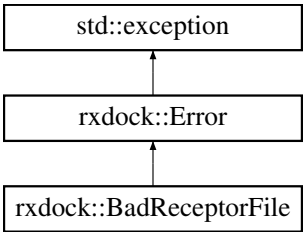
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- Error.h

5.15 rxdock::BadReceptorFile Class Reference

Inheritance diagram for rxdock::BadReceptorFile:



Public Member Functions

- **BadReceptorFile** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::Error

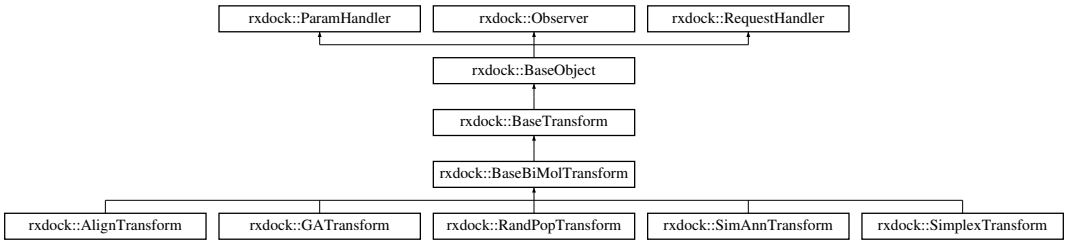
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- Error.h

5.16 rxdock::BaseBiMolTransform Class Reference

Inheritance diagram for rxdock::BaseBiMolTransform:



Public Member Functions

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform *)
- virtual void **Remove** (BaseTransform *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform * **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform * **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨BaseBiMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- **BaseBiMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupReceptor** ()=0
- virtual void **SetupLigand** ()=0
- virtual void **SetupSolvent** ()
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const BaseBiMolTransform &baseBiMolTrans)
- void **from_json** (const json &j, BaseBiMolTransform &baseBiMolTrans)

5.16.1 Member Function Documentation

5.16.1.1 Update()

```
void BaseBiMolTransform::Update (
            Subject * theChangedSubject )  [virtual]
```

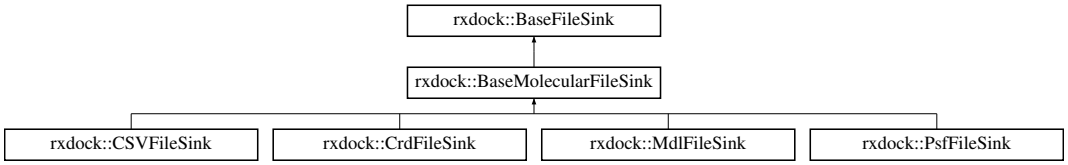
Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- BaseBiMolTransform.h
- BaseBiMolTransform.cxx

5.17 rxdock::BaseFileSink Class Reference

Inheritance diagram for rxdock::BaseFileSink:



Public Member Functions

- **BaseFileSink** (const std::string &fileName)
- std::string **GetFileName** () const
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- Error **Status** ()
- virtual void **Render** ()=0

Protected Member Functions

- void **Write** (bool bClearCache=true)
- void **AddLine** (const std::string &fileRec)
- void **ReplaceLine** (const std::string &fileRec, unsigned int nRec)
- bool **isCacheEmpty** () const
- bool **GetAppend** () const
- void **SetAppend** (bool bAppend)

Friends

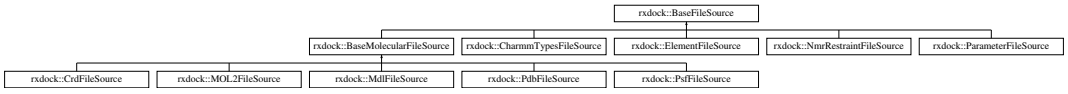
- void **to_json** (json &j, const BaseFileSink &baseFileSink)
- void **from_json** (const json &j, BaseFileSink &baseFileSink)

The documentation for this class was generated from the following files:

- BaseFileSink.h
- BaseFileSink.cxx

5.18 rxdock::BaseFileSource Class Reference

Inheritance diagram for rxdock::BaseFileSource:



Public Member Functions

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Protected Member Functions

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Protected Attributes

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

Friends

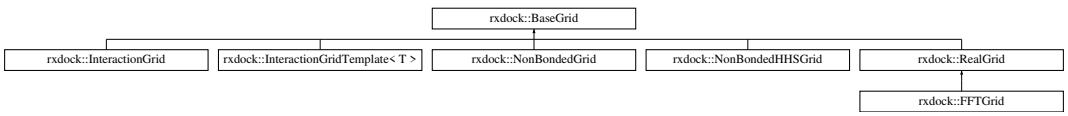
- void **to_json** (json &j, const BaseFileSource &baseFileSource)
- void **from_json** (const json &j, BaseFileSource &baseFileSource)

The documentation for this class was generated from the following files:

- BaseFileSource.h
- BaseFileSource.cxx

5.19 rxdock::BaseGrid Class Reference

Inheritance diagram for rxdock::BaseGrid:



Public Member Functions

- **BaseGrid** (const Coord &gridMin, const Vector &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **BaseGrid** (json j)
- **BaseGrid** (const BaseGrid &)
- BaseGrid & **operator=** (const BaseGrid &)
- virtual void **Print** (std::ostream &ostr) const
- unsigned int **GetNX** () const
- unsigned int **GetNY** () const
- unsigned int **GetNZ** () const
- unsigned int **GetN** () const
- unsigned int **GetStrideX** () const
- unsigned int **GetStrideY** () const
- unsigned int **GetStrideZ** () const
- int **GetnXMin** () const
- int **GetnYMin** () const
- int **GetnZMin** () const
- int **GetnXMax** () const
- int **GetnYMax** () const
- int **GetnZMax** () const
- const Coord & **GetGridMin** () const
- const Coord & **GetGridMax** () const
- const Vector & **GetGridStep** () const
- Coord **GetGridCenter** () const
- Coord **GetGridSize** () const
- unsigned int **GetPad** () const
- const Coord & **GetPadMin** () const
- const Coord & **GetPadMax** () const
- void **SetGridMin** (const Coord &gridMin)
- void **TranslateGrid** (const Vector &vec)
- void **SetGridCenter** (const Coord &gridCenter)
- void **SetGridStep** (const Vector &gridStep)
- void **SetPad** (unsigned int NPad=0)
- bool **isValid** (const Coord &c) const
- bool **isValid** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- bool **isValid** (unsigned int iXYZ) const
- unsigned int **GetIX** (const Coord &c) const
- unsigned int **GetIY** (const Coord &c) const
- unsigned int **GetIZ** (const Coord &c) const
- unsigned int **GetIX** (double x) const
- unsigned int **GetIY** (double y) const
- unsigned int **GetIZ** (double z) const
- unsigned int **GetIX** (unsigned int iXYZ) const
- unsigned int **GetIY** (unsigned int iXYZ) const
- unsigned int **GetIZ** (unsigned int iXYZ) const
- unsigned int **GetIXYZ** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- unsigned int **GetIXYZ** (const Coord &c) const
- Coord **GetCoord** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetXCoord** (unsigned int iX) const
- double **GetYCoord** (unsigned int iY) const
- double **GetZCoord** (unsigned int iZ) const
- Coord **GetCoord** (unsigned int iXYZ) const
- CoordList **GetCoordList** (const std::set< unsigned int > &iXYZSet) const
- void **GetSphereIndices** (const Coord &c, double radius, std::vector< unsigned int > &sIndices) const

Static Public Attributes

- static const std::string **_CT** = ¨BaseGrid¨

Protected Member Functions

- void **OwnPrint** (std::ostream &ostr) const

Friends

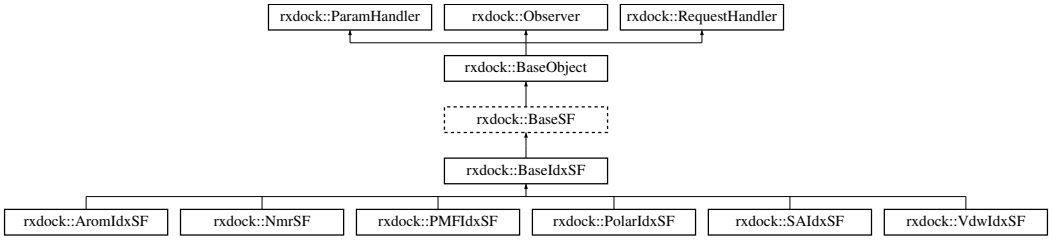
- std::ostream & **operator<<** (std::ostream &s, const BaseGrid &grid)
- void **to_json** (json &j, const BaseGrid &grid)
- void **from_json** (const json &j, BaseGrid &grid)

The documentation for this class was generated from the following files:

- BaseGrid.h
- BaseGrid.cxx

5.20 rxdock::BaseIdxSF Class Reference

Inheritance diagram for rxdock::BaseIdxSF:



Public Member Functions

- double **GetGridStep** () const
- void **SetGridStep** (double step)
- double **GetBorder** () const
- void **SetBorder** (double border)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Public Member Functions inherited from rxdock::Observer

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨BaseIdxSF¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_BORDER** = ¨border¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- InteractionGridPtr **CreateInteractionGrid** () const
- NonBondedGridPtr **CreateNonBondedGrid** () const
- NonBondedHHSGridPtr **CreateNonBondedHHSGrid** () const
- double **GetMaxError** () const
- double **GetCorrectedRange** () const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

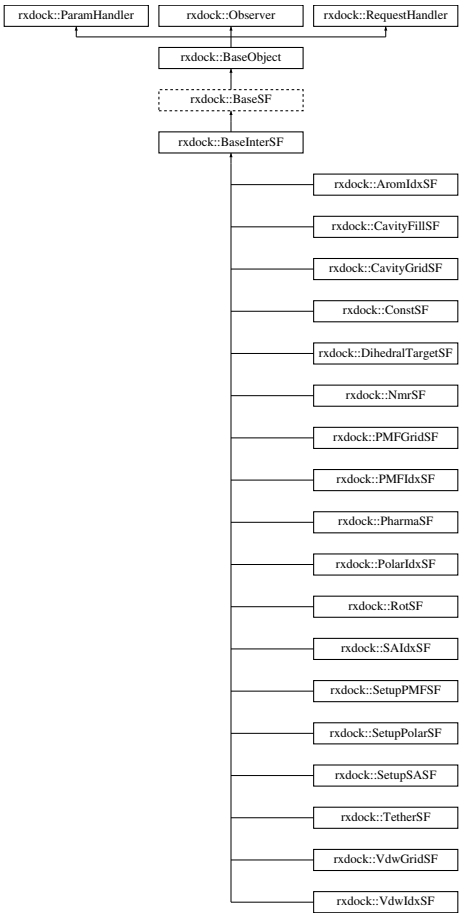
- void **to_json** (json &j, const BaseIdxSF &baseIdxSF)
- void **from_json** (const json &j, BaseIdxSF &baseIdxSF)

The documentation for this class was generated from the following files:

- BaseIdxSF.h
- BaseIdxSF.cxx

5.21 rxdock::BaseInterSF Class Reference

Inheritance diagram for rxdock::BaseInterSF:



Public Member Functions

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const BaseInterSF &baseInterSF)
- void **from_json** (const json &j, BaseInterSF &baseInterSF)

5.21.1 Member Function Documentation

5.21.1.1 SetupLigand()

```
virtual void rxdock::BaseInterSF::SetupLigand ( )  [protected], [pure virtual]
```

Implemented in rxdock::PMFIdxSF.

5.21.1.2 SetupReceptor()

```
virtual void rxdock::BaseInterSF::SetupReceptor ( )  [protected], [pure virtual]
```

Implemented in rxdock::PMFIdxSF.

5.21.1.3 SetupScore()

```
virtual void rxdock::BaseInterSF::SetupScore ( )  [protected], [pure virtual]
```

Implemented in rxdock::PMFIdxSF.

5.21.1.4 Update()

```
void BaseInterSF::Update (
            Subject * theChangedSubject )  [virtual]
```

Implements rxdock::Observer.

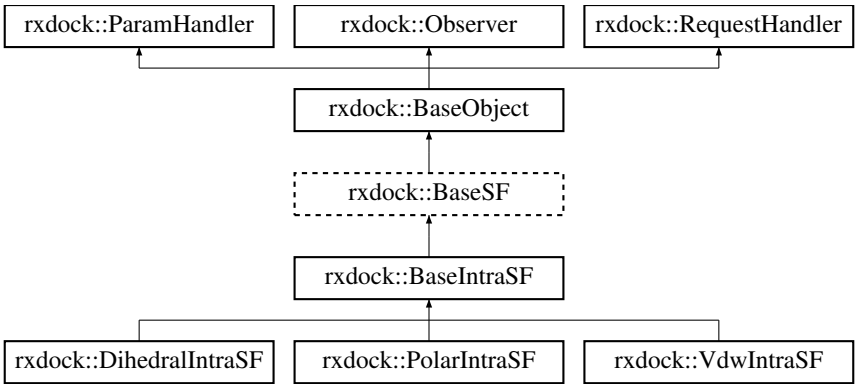
Reimplemented in rxdock::PMFIdxSF.

The documentation for this class was generated from the following files:

- BaseInterSF.h
- BaseInterSF.cxx

5.22 rxdock::BaseIntraSF Class Reference

Inheritance diagram for rxdock::BaseIntraSF:



Public Member Functions

- ModelPtr **GetLigand** () const
- virtual void Update (Subject *theChangedSubject)
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨BaseIntraSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void **SetupScore** ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const BaseIntraSF &baseIntraSF)
- void **from_json** (const json &j, BaseIntraSF &baseIntraSF)

5.22.1 Member Function Documentation

5.22.1.1 ScoreMap()

```
void BaseIntraSF::ScoreMap (
            StringVariantMap & scoreMap ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.22.1.2 Update()

```
void BaseIntraSF::Update (
            Subject * theChangedSubject )  [virtual]
```

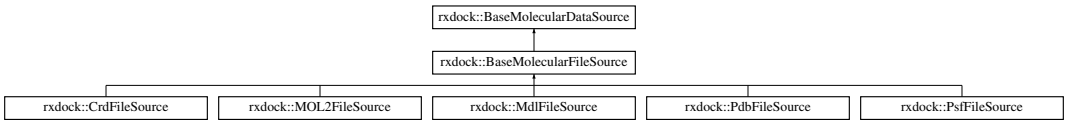
Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- BaseIntraSF.h
- BaseIntraSF.cxx

5.23 rxdock::BaseMolecularDataSource Class Reference

Inheritance diagram for rxdock::BaseMolecularDataSource:



Public Member Functions

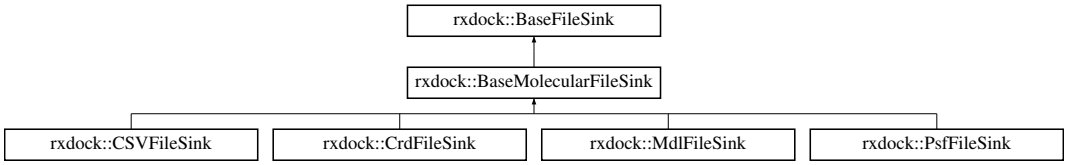
- **BaseMolecularDataSource** (std::string strName)
- std::string **GetName** ()
- virtual bool **isTitleListSupported** ()
- virtual bool **isAtomListSupported** ()
- virtual bool **isCoordinatesSupported** ()
- virtual bool **isBondListSupported** ()
- virtual void **Reset** ()=0
- virtual int **GetNumTitles** ()=0
- virtual int **GetNumAtoms** ()=0
- virtual int **GetNumBonds** ()=0
- virtual int **GetNumSegments** ()=0
- virtual std::vector< std::string > **GetTitleList** ()=0
- virtual AtomList **GetAtomList** ()=0
- virtual BondList **GetBondList** ()=0
- virtual SegmentMap **GetSegmentMap** ()=0
- virtual bool **isDataSupported** ()
- virtual int **GetNumData** ()=0
- virtual std::vector< std::string > **GetDataFieldList** ()=0
- virtual StringVariantMap **GetDataMap** ()=0
- virtual bool **isDataFieldPresent** (const std::string &strDataField)=0
- virtual Variant **GetDataValue** (const std::string &strDataField)=0

The documentation for this class was generated from the following file:

- BaseMolecularDataSource.h

5.24 rxdock::BaseMolecularFileSink Class Reference

Inheritance diagram for rxdock::BaseMolecularFileSink:



Public Member Functions

- **BaseMolecularFileSink** (const std::string &fileName, ModelPtr spModel, bool bUseModelSegment↩ Names=true)
- virtual bool **isMultiConfSupported** ()
- ModelPtr **GetModel** () const
- RBTDLL_EXPORT void **SetModel** (ModelPtr spModel, bool bUseModelSegmentNames=false)
- void **SetNextSegmentId** (int nSegmentId)
- bool **GetMultiConf** () const
- void **SetMultiConf** (bool bMultiConf)
- void **WriteMultiConf** ()
- ModelList **GetSolvent** () const
- void **SetSolvent** (ModelList solventList)

Public Member Functions inherited from rxdock::BaseFileSink

- **BaseFileSink** (const std::string &fileName)
- std::string **GetFileName** () const
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- Error **Status** ()
- virtual void **Render** ()=0

Protected Member Functions

- void **Reset** ()

Protected Member Functions inherited from rxdock::BaseFileSink

- void **Write** (bool bClearCache=true)
- void **AddLine** (const std::string &fileRec)
- void **ReplaceLine** (const std::string &fileRec, unsigned int nRec)
- bool **isCacheEmpty** () const
- bool **GetAppend** () const
- void **SetAppend** (bool bAppend)

Protected Attributes

- int **m_nAtomId**
- int **m_nSubunitId**
- int **m_nSegmentId**
- bool **m_bUseModelSegmentNames**

Friends

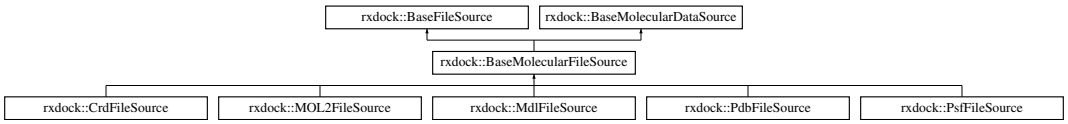
- void **to_json** (json &j, const BaseMolecularFileSink &baseBiMolFSink)
- void **from_json** (const json &j, BaseMolecularFileSink &baseBiMolFSink)

The documentation for this class was generated from the following files:

- BaseMolecularFileSink.h
- BaseMolecularFileSink.cxx

5.25 rxdock::BaseMolecularFileSource Class Reference

Inheritance diagram for rxdock::BaseMolecularFileSource:



Public Member Functions

- RBTDLL_EXPORT **BaseMolecularFileSource** (const std::string &fileName, const std::string &sourceName)
- **BaseMolecularFileSource** (const std::string &fileName, const std::string &strRecDelim, const std::string &sourceName)
- void Reset ()
- int GetNumTitles ()
- int GetNumAtoms ()
- int GetNumBonds ()
- int GetNumSegments ()
- std::vector< std::string > GetTitleList ()
- AtomList GetAtomList ()
- BondList GetBondList ()
- SegmentMap GetSegmentMap ()
- virtual int GetNumData ()
- virtual std::vector< std::string > GetDataFieldList ()
- virtual StringVariantMap GetDataMap ()
- virtual bool isDataFieldPresent (const std::string &strDataField)
- virtual Variant GetDataValue (const std::string &strDataField)
- SegmentMap **GetSegmentFilterMap** ()
- RBTDLL_EXPORT void **SetSegmentFilterMap** (const SegmentMap &segmentFilterMap)
- void **ClearSegmentFilterMap** ()
- bool **isSegmentFilterMapDefined** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Public Member Functions inherited from rxdock::BaseMolecularDataSource

- **BaseMolecularDataSource** (std::string strName)
- std::string **GetName** ()
- virtual bool **isTitleListSupported** ()
- virtual bool **isAtomListSupported** ()
- virtual bool **isCoordinatesSupported** ()
- virtual bool **isBondListSupported** ()
- virtual void **Reset** ()=0
- virtual int **GetNumTitles** ()=0
- virtual int **GetNumAtoms** ()=0
- virtual int **GetNumBonds** ()=0
- virtual int **GetNumSegments** ()=0
- virtual std::vector< std::string > **GetTitleList** ()=0
- virtual AtomList **GetAtomList** ()=0
- virtual BondList **GetBondList** ()=0
- virtual SegmentMap **GetSegmentMap** ()=0
- virtual bool **isDataSupported** ()
- virtual int **GetNumData** ()=0
- virtual std::vector< std::string > **GetDataFieldList** ()=0
- virtual StringVariantMap **GetDataMap** ()=0
- virtual bool **isDataFieldPresent** (const std::string &strDataField)=0
- virtual Variant **GetDataValue** (const std::string &strDataField)=0

Protected Member Functions

- void **ClearMolCache** ()
- void **RemoveAtom** (AtomPtr spAtom)
- void **RenumberAtomsAndBonds** ()
- void **SetupPartiallonicGroups** (AtomList &atoms, ParameterFileSourcePtr spParamSource)

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Protected Attributes

- std::vector< std::string > **m_titleList**
- AtomList **m_atomList**
- BondList **m_bondList**
- SegmentMap **m_segmentMap**
- StringVariantMap **m_dataMap**

Protected Attributes inherited from rxdock::BaseFileSource

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

Friends

- void **to_json** (json &j, const BaseMolecularFileSource &baseMolFilSrc)
- void **from_json** (const json &j, BaseMolecularFileSource &baseMolFilSrc)

5.25.1 Member Function Documentation

5.25.1.1 GetAtomList()

```
AtomList BaseMolecularFileSource::GetAtomList ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.2 GetBondList()

```
BondList BaseMolecularFileSource::GetBondList ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.3 GetDataFieldList()

```
std::vector< std::string > BaseMolecularFileSource::GetDataFieldList ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.4 GetDataMap()

```
StringVariantMap BaseMolecularFileSource::GetDataMap ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.5 GetDataValue()

```
Variant BaseMolecularFileSource::GetDataValue (
            const std::string & strDataField )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.6 GetNumAtoms()

```
int BaseMolecularFileSource::GetNumAtoms ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.7 GetNumBonds()

```
int BaseMolecularFileSource::GetNumBonds ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.8 GetNumData()

```
int BaseMolecularFileSource::GetNumData ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.9 GetNumSegments()

```
int BaseMolecularFileSource::GetNumSegments ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.10 GetNumTitles()

```
int BaseMolecularFileSource::GetNumTitles ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.11 GetSegmentMap()

```
SegmentMap BaseMolecularFileSource::GetSegmentMap ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.12 GetTitleList()

```
std::vector< std::string > BaseMolecularFileSource::GetTitleList ( )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.13 isDataFieldPresent()

```
bool BaseMolecularFileSource::isDataFieldPresent (
            const std::string & strDataField )  [virtual]
```

Implements rxdock::BaseMolecularDataSource.

5.25.1.14 Reset()

```
void BaseMolecularFileSource::Reset ( )  [virtual]
```

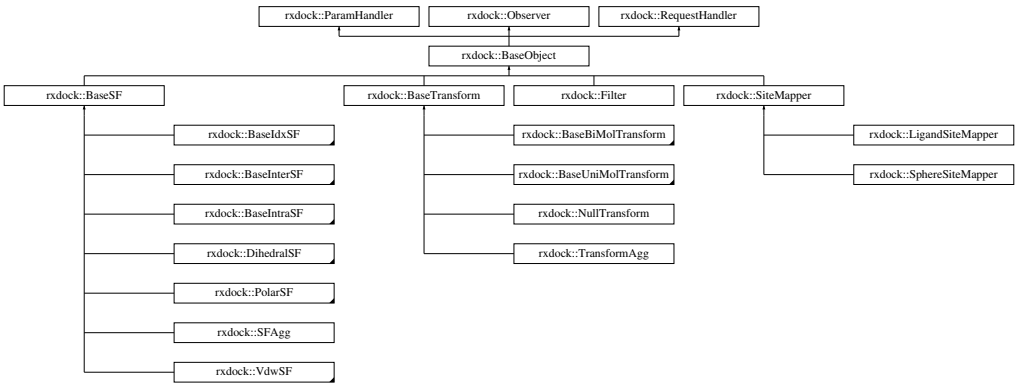
Implements rxdock::BaseMolecularDataSource.

The documentation for this class was generated from the following files:

- BaseMolecularFileSource.h
- BaseMolecularFileSource.cxx

5.26 rxdock::BaseObject Class Reference

Inheritance diagram for rxdock::BaseObject:



Public Member Functions

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Public Member Functions inherited from rxdock::Observer

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = "BaseObject"
- static const std::string **_CLASS** = "CLASS"
- static const std::string **_NAME** = "NAME"
- static const std::string **_ENABLED** = "ENABLED"

Protected Member Functions

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const BaseObject &baseObject)
- void **from_json** (const json &j, BaseObject &baseObject)

5.26.1 Member Function Documentation

5.26.1.1 Deleted()

```
void BaseObject::Deleted (
            Subject * theDeletedSubject )  [virtual]
```

Implements rxdock::Observer.

5.26.1.2 HandleRequest()

```
void BaseObject::HandleRequest (
            RequestPtr spRequest )  [virtual]
```

Reimplemented from rxdock::RequestHandler.

5.26.1.3 ParameterUpdated()

```
void BaseObject::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::ParamHandler.

5.26.1.4 Print()

```
void BaseObject::Print (
            std::ostream & s ) const  [virtual]
```

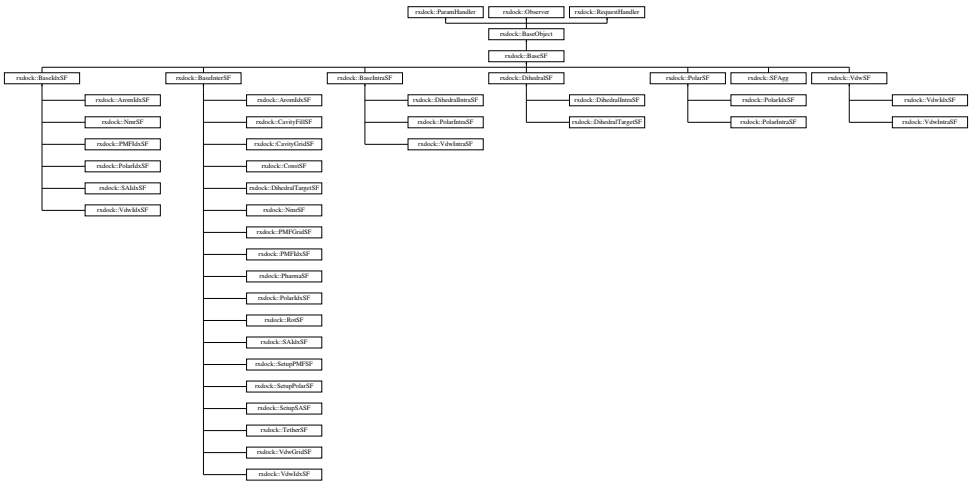
Reimplemented from rxdock::ParamHandler.

The documentation for this class was generated from the following files:

- BaseObject.h
- BaseObject.cxx

5.27 rxdock::BaseSF Class Reference

Inheritance diagram for rxdock::BaseSF:



Public Member Functions

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Public Member Functions inherited from rxdock::Observer

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- class **SFAgg**
- void **to_json** (json &j, const BaseSF &baseSF)
- void **from_json** (const json &j, BaseSF &baseSF)

5.27.1 Member Function Documentation

5.27.1.1 GetFullName()

```
std::string BaseSF::GetFullName ( ) const  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.27.1.2 ParameterUpdated()

```
void BaseSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseObject.

5.27.1.3 RawScore()

```
virtual double rxdock::BaseSF::RawScore ( ) const  [protected], [pure virtual]
```

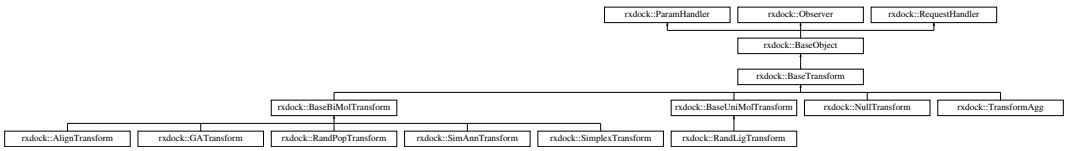
Implemented in rxdock::PMFIdxSF.

The documentation for this class was generated from the following files:

- BaseSF.h
- BaseSF.cxx

5.28 rxdock::BaseTransform Class Reference

Inheritance diagram for rxdock::BaseTransform:



Public Member Functions

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform ∗)
- virtual void **Remove** (BaseTransform ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform ∗ **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform ∗ **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Public Member Functions inherited from rxdock::Observer

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- class **TransformAgg**
- void **to_json** (json &j, const BaseTransform &bt)
- void **from_json** (const json &j, BaseTransform &bt)

5.28.1 Member Function Documentation

5.28.1.1 GetFullName()

```
std::string BaseTransform::GetFullName ( ) const  [virtual]
```

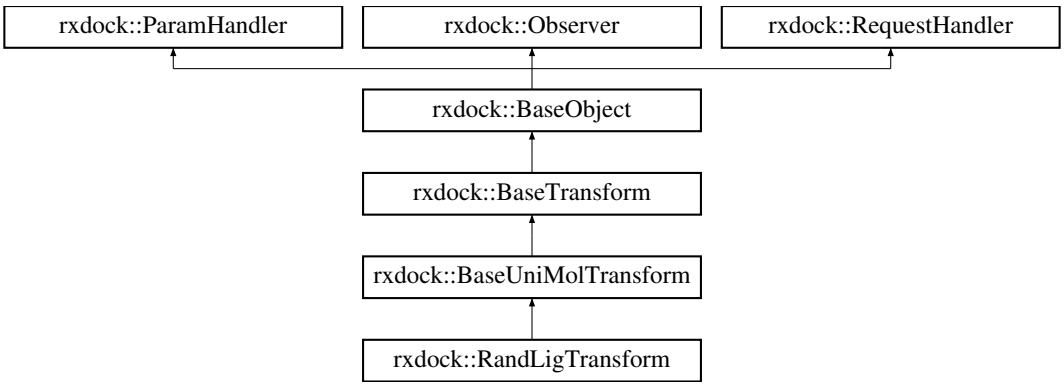
Reimplemented from rxdock::BaseObject.

The documentation for this class was generated from the following files:

- BaseTransform.h
- BaseTransform.cxx

5.29 rxdock::BaseUniMolTransform Class Reference

Inheritance diagram for rxdock::BaseUniMolTransform:



Public Member Functions

- ModelPtr **GetLigand** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform *)
- virtual void **Remove** (BaseTransform *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform * **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform * **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨BaseUniMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- **BaseUniMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const BaseUniMolTransform &baseUniMolTrans)
- void **from_json** (const json &j, BaseUniMolTransform &baseUniMolTrans)

5.29.1 Member Function Documentation

5.29.1.1 Update()

```
void BaseUniMolTransform::Update (
            Subject * theChangedSubject )  [virtual]
```

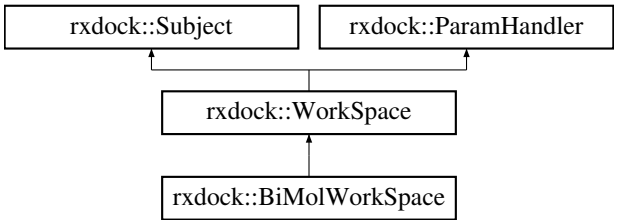
Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- BaseUniMolTransform.h
- BaseUniMolTransform.cxx

5.30 rxdock::BiMolWorkSpace Class Reference

Inheritance diagram for rxdock::BiMolWorkSpace:



Public Member Functions

- RBTDLL_EXPORT ModelPtr **GetReceptor** () const
- RBTDLL_EXPORT ModelPtr **GetLigand** () const
- RBTDLL_EXPORT ModelList **GetSolvent** () const
- RBTDLL_EXPORT bool **hasSolvent** () const
- RBTDLL_EXPORT void **SetReceptor** (ModelPtr spReceptor)
- RBTDLL_EXPORT void **SetLigand** (ModelPtr spLigand)
- RBTDLL_EXPORT void **SetSolvent** (ModelList solventList)
- RBTDLL_EXPORT void **RemoveSolvent** ()
- RBTDLL_EXPORT void **UpdateModelCoordsFromChromRecords** (BaseMolecularFileSource *pSource)
- virtual void Save (bool bSaveScores=true)
- virtual void SaveHistory (bool bSaveScores=true)

Public Member Functions inherited from rxdock::WorkSpace

- **WorkSpace** (unsigned int nModels=2)
- RBTDLL_EXPORT std::string **GetName** () const
- RBTDLL_EXPORT void **SetName** (const std::string &)
- RBTDLL_EXPORT unsigned int **GetNumModels** () const
- RBTDLL_EXPORT ModelList **GetModels** () const
- RBTDLL_EXPORT ModelPtr **GetModel** (unsigned int iModel) const
- void **SetModel** (unsigned int iModel, ModelPtr spModel)
- ModelList **GetModels** (unsigned int iModel) const
- void **AddModels** (ModelList modelList)
- void **SetModels** (unsigned int iModel, ModelList modelList)
- void **RemoveModels** (unsigned int iModel)
- MolecularFileSinkPtr **GetSink** () const
- RBTDLL_EXPORT void **SetSink** (MolecularFileSinkPtr)
- virtual void **Save** (bool bSaveScores=true)
- MolecularFileSinkPtr **GetHistorySink** () const
- RBTDLL_EXPORT void **SetHistorySink** (MolecularFileSinkPtr)
- virtual void **SaveHistory** (bool bSaveScores=true)
- RBTDLL_EXPORT BaseSF * **GetSF** () const
- RBTDLL_EXPORT void **SetSF** (BaseSF *)
- BaseTransform * **GetTransform** () const
- RBTDLL_EXPORT void **SetTransform** (BaseTransform *)
- virtual void **Run** ()
- void **SetPopulation** (PopulationPtr population)
- PopulationPtr **GetPopulation** () const
- void **ClearPopulation** ()
- RBTDLL_EXPORT DockingSitePtr **GetDockingSite** () const
- RBTDLL_EXPORT void **SetDockingSite** (DockingSitePtr spDockSite)
- FilterPtr **GetFilter** () const
- RBTDLL_EXPORT void **SetFilter** (FilterPtr spFilter)

Public Member Functions inherited from rxdock::Subject

- virtual void **Attach** (Observer *)
- virtual void **Detach** (Observer *)
- virtual void **Notify** ()

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Additional Inherited Members

Static Public Attributes inherited from rxdock::WorkSpace

- static const std::string **_CT** = ¨WorkSpace¨
- static const std::string **_NAME** = ¨NAME¨

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.30.1 Member Function Documentation

5.30.1.1 Save()

```
void BiMolWorkSpace::Save (
            bool bSaveScores = true )  [virtual]
```

Reimplemented from rxdock::WorkSpace.

5.30.1.2 SaveHistory()

```
void BiMolWorkSpace::SaveHistory (
            bool bSaveScores = true )  [virtual]
```

Reimplemented from rxdock::WorkSpace.

The documentation for this class was generated from the following files:

- BiMolWorkSpace.h
- BiMolWorkSpace.cxx

5.31 rxdock::Bond Class Reference

Public Member Functions

- **Bond** (int nBondId, AtomPtr &spAtom1, AtomPtr &spAtom2, int nFormalBondOrder=1)
- **Bond** (json j)
- **Bond** (const Bond &bond)
- Bond & **operator=** (const Bond &bond)
- int **GetBondId** () const
- void **SetBondId** (const int nBondId)
- AtomPtr **GetAtom1Ptr** () const
- AtomPtr **GetAtom2Ptr** () const
- void **SetAtom1Ptr** (AtomPtr &spAtom1)
- void **SetAtom2Ptr** (AtomPtr &spAtom2)
- int **GetFormalBondOrder** () const
- void **SetFormalBondOrder** (const int nFormalBondOrder)
- double **GetPartialBondOrder** () const
- void **SetPartialBondOrder** (const double dPartialBondOrder)
- bool **GetCyclicFlag** () const
- void **SetCyclicFlag** (bool bCyclic=true)
- bool **GetSelectionFlag** () const
- void **SetSelectionFlag** (bool bSelected=true)
- double **Length** () const

Friends

- RBTDLL_EXPORT friend std::ostream & **operator<<** (std::ostream &s, const Bond &bond)
- void **to_json** (json &j, const Bond &bond)
- void **from_json** (const json &j, Bond &bond)

The documentation for this class was generated from the following files:

- Bond.h
- Bond.cxx

5.32 rxdock::BondPCmp_BondId Class Reference

Public Member Functions

- bool **operator()** (Bond *pBond1, Bond *pBond2) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.33 rxdock::Cavity Class Reference

Public Member Functions

- **Cavity** (const CoordList &coordList, const Vector gridStep)
- **Cavity** (json j)
- virtual void **Print** (std::ostream &s) const
- RealGridPtr **GetGrid** () const
- const Coord & **GetCenterOfMass** () const
- const PrincipalAxes & **GetPrincipalAxes** () const
- int **GetNumCoords** () const
- const CoordList & **GetCoordList** () const
- const Coord & **GetMinCoord** () const
- const Coord & **GetMaxCoord** () const
- const Vector & **GetGridStep** () const
- double **GetVolume** () const

Friends

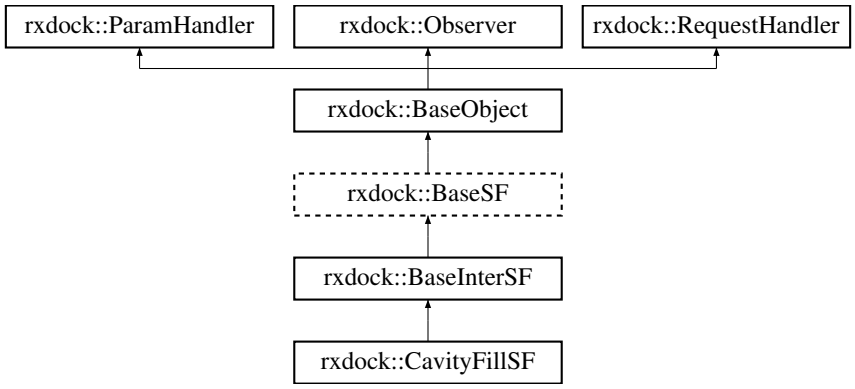
- std::ostream & **operator<<** (std::ostream &s, const Cavity &cavity)
- void **to_json** (json &j, const Cavity &c)
- void **from_json** (const json &j, Cavity &c)

The documentation for this class was generated from the following file:

- Cavity.h

5.34 rxdock::CavityFillSF Class Reference

Inheritance diagram for rxdock::CavityFillSF:



Public Member Functions

- **CavityFillSF** (const std::string &strName="cavity-fill")

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨CavityFillSF¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const CavityFillSF &sf)
- void **from_json** (const json &j, CavityFillSF &sf)

5.34.1 Member Function Documentation

5.34.1.1 ParameterUpdated()

```
void CavityFillSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.34.1.2 RawScore()

```
double CavityFillSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.34.1.3 SetupLigand()

```
void CavityFillSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.34.1.4 SetupReceptor()

```
void CavityFillSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.34.1.5 SetupScore()

```
void CavityFillSF::SetupScore ( )  [protected], [virtual]
```

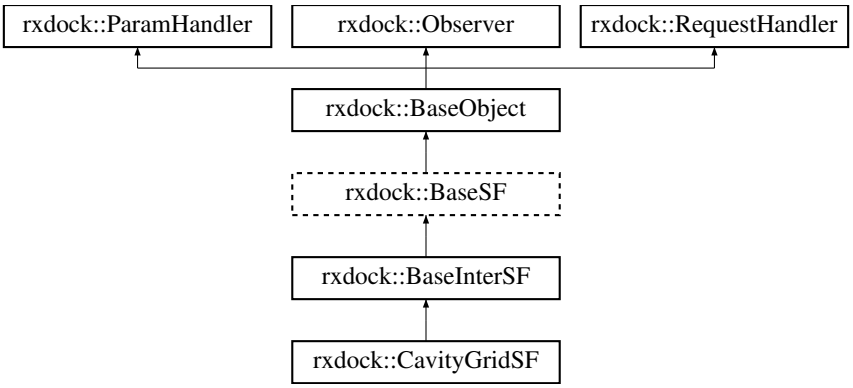
Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- CavityFillSF.h
- CavityFillSF.cxx

5.35 rxdock::CavityGridSF Class Reference

Inheritance diagram for rxdock::CavityGridSF:



Classes

- class HeavyAtomFactory

Public Member Functions

- void **to_json** (json &j, const HeavyAtomFactory &heavyAtomFactory)
- void **from_json** (const json &j, HeavyAtomFactory &heavyAtomFactory)
- **CavityGridSF** (const std::string &strName=¨cavity¨)

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨CavityGridSF¨
- static const std::string **_RMAX** = ¨rmax¨
- static const std::string **_QUADRATIC** = ¨quadratic¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const CavityGridSF &cavGridSF)
- void **from_json** (const json &j, CavityGridSF &cavGridSF)

5.35.1 Member Function Documentation

5.35.1.1 ParameterUpdated()

```
void CavityGridSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.35.1.2 RawScore()

```
double CavityGridSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.35.1.3 SetupLigand()

```
void CavityGridSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.35.1.4 SetupReceptor()

```
void CavityGridSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.35.1.5 SetupScore()

```
void CavityGridSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.35.1.6 SetupSolvent()

```
void CavityGridSF::SetupSolvent ( )  [protected], [virtual]
```

Reimplemented from rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- CavityGridSF.h
- CavityGridSF.cxx

5.36 rxdock::CavityPtrCmp_Distance Class Reference

Public Member Functions

- **CavityPtrCmp_Distance** (const Coord &cc)
- bool **operator()** (CavityPtr spCav1, CavityPtr spCav2) const

The documentation for this class was generated from the following file:

- Cavity.h

5.37 rxdock::CavityPtrCmp_Volume Class Reference

Public Member Functions

- bool **operator()** (CavityPtr spCav1, CavityPtr spCav2) const

The documentation for this class was generated from the following file:

- Cavity.h

5.38 rxdock::Cell Class Reference

Public Member Functions

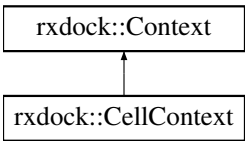
- **Cell** (const Cell &c)
- Cell & **operator=** (const Cell &c)
- bool **Evaluated** () const
- bool **Named** () const
- bool **Constant** () const
- void **Clear** ()
- ReturnType **GetResult** () const
- std::string **GetName** () const
- void **SetName** (std::string s)
- void **SetResult** (ReturnType r)
- void **SetConstant** (ReturnType r)
- void **ResetConstant** ()

The documentation for this class was generated from the following file:

- Cell.h

5.39 rxdock::CellContext Class Reference

Inheritance diagram for rxdock::CellContext:



Public Member Functions

- **CellContext** (std::ifstream &ifile)
- **CellContext** (const CellContext &c)
- void Assign (int key, ReturnType val)
- void Assign (std::string s, ReturnType val)
- const Vble & GetVble (int key)
- void SetVble (int key, const Vble &v)
- const Vble & GetVble (std::string key)

Public Member Functions inherited from rxdock::Context

- **Context** (const Context &c)
- virtual void **Assign** (std::string, ReturnType)=0
- virtual void **Assign** (int, ReturnType)=0
- virtual const Vble & **GetVble** (int)=0
- virtual const Vble & **GetVble** (std::string)=0
- virtual void **SetVble** (int key, const Vble &v)=0

Additional Inherited Members

Static Public Attributes inherited from rxdock::Context

- static const std::string **_CT** = ¨Context¨

5.39.1 Member Function Documentation

5.39.1.1 Assign() [1/2]

```
void rxdock::CellContext::Assign (
            int key,
            ReturnType val )  [inline], [virtual]
```

Implements rxdock::Context.

5.39.1.2 Assign() [2/2]

```
void rxdock::CellContext::Assign (
            std::string s,
            ReturnType val )  [inline], [virtual]
```

Implements rxdock::Context.

5.39.1.3 GetVble() [1/2]

```
const Vble & rxdock::CellContext::GetVble (
            int key )  [inline], [virtual]
```

Implements rxdock::Context.

5.39.1.4 GetVble() [2/2]

```
const Vble & rxdock::CellContext::GetVble (
            std::string key )  [inline], [virtual]
```

Implements rxdock::Context.

5.39.1.5 SetVble()

```
void rxdock::CellContext::SetVble (
            int key,
            const Vble & v )  [inline], [virtual]
```

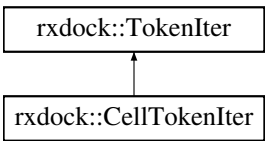
Implements rxdock::Context.

The documentation for this class was generated from the following files:

- Context.h
- Context.cxx

5.40 rxdock::CellTokenIter Class Reference

Inheritance diagram for rxdock::CellTokenIter:



Public Member Functions

- **CellTokenIter** (const CellTokenIter &)
- **CellTokenIter** (const geneticprogram::GPChromosomePtr, ContextPtr)
- virtual void **copy** (const CellTokenIter &)
- void Next (ContextPtr)
- TokenPtr Current ()

- virtual void **Next** (ContextPtr)=0
- virtual TokenPtr **Current** ()=0

Static Public Attributes

- static const std::string **_CT** = ¨CellTokenIter¨

Friends

- void **to_json** (json &j, const CellTokenIter &callTokIt)
- void **from_json** (const json &j, CellTokenIter &callTokIt)

5.40.1 Member Function Documentation

5.40.1.1 Current()

```
TokenPtr CellTokenIter::Current ( )  [virtual]
```

Implements rxdock::TokenIter.

5.40.1.2 Next()

```
void CellTokenIter::Next (
            ContextPtr contextp )  [virtual]
```

Implements rxdock::TokenIter.

The documentation for this class was generated from the following files:

- CellTokenIter.h
- CellTokenIter.cxx

5.41 rxdock::CharmmDataSource Class Reference

Public Member Functions

- **CharmmDataSource** (const std::string &strMassesFile=GetDataFileName(¨data¨, ¨masses.rtf¨))
- std::string **AtomTypeString** (int nAtomType)
- int **ImplicitHydrogens** (const std::string &strFFType)
- int **AtomicNumber** (const std::string &strFFType)
- int **FormalCharge** (const std::string &strFFType)
- Atom::eHybridState **HybridState** (const std::string &strFFType)

Friends

- void **to_json** (json &j, const CharmmDataSource &charDatSrc)
- void **from_json** (const json &j, CharmmDataSource &charDatSrc)

The documentation for this class was generated from the following files:

- CharmmDataSource.h
- CharmmDataSource.cxx

5.42 rxdock::CharmmType Class Reference

Public Attributes

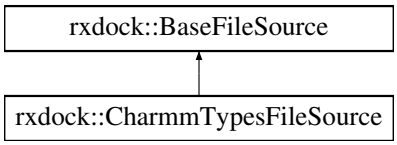
- int **nAtomType**
- std::string **strAtomType**
- double **mass**
- std::string **element**
- std::string **comment**

The documentation for this class was generated from the following file:

- CharmmTypesFileSource.h

5.43 rxdock::CharmmTypesFileSource Class Reference

Inheritance diagram for rxdock::CharmmTypesFileSource:



Public Member Functions

- **CharmmTypesFileSource** (const char *fileName)
- **CharmmTypesFileSource** (const std::string fileName)
- int **GetNumTypes** ()
- CharmmTypeList **GetTypeList** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Friends

- void **to_json** (json &j, const CharmmTypesFileSource &charmmTypesFileSource)
- void **from_json** (const json &j, CharmmTypesFileSource &charmmTypesFileSource)

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Protected Attributes inherited from rxdock::BaseFileSource

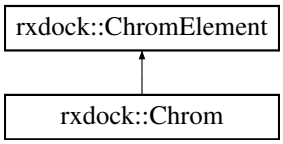
- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

The documentation for this class was generated from the following files:

- CharmmTypesFileSource.h
- CharmmTypesFileSource.cxx

5.44 rxdock::Chrom Class Reference

Inheritance diagram for rxdock::Chrom:



Public Member Functions

- **Chrom** (const ModelList &modelList)
- virtual void Reset ()
- virtual void Randomise ()
- virtual void Mutate (double relStepSize)
- virtual void SyncFromModel ()
- virtual void SyncToModel ()
- virtual ChromElement * clone () const
- virtual int GetLength () const
- virtual int GetXOverLength () const
- virtual void GetVector (std::vector< double > &v) const
- virtual void GetVector (XOverList &v) const
- virtual void SetVector (const std::vector< double > &v, int &i)
- virtual void SetVector (const XOverList &v, int &i)
- virtual void GetStepVector (std::vector< double > &v) const
- virtual double CompareVector (const std::vector< double > &v, int &i) const
- virtual void Print (std::ostream &s) const
- virtual void Add (ChromElement *pChromElement)

Public Member Functions inherited from rxdock::ChromElement

- virtual void **Reset** ()=0
- virtual void **Randomise** ()=0
- virtual void **Mutate** (double relStepSize)=0
- virtual void **SyncFromModel** ()=0
- virtual void **SyncToModel** ()=0
- virtual ChromElement * **clone** () const =0
- virtual int **GetLength** () const =0
- virtual int **GetXOverLength** () const =0
- virtual void **GetVector** (std::vector< double > &v) const =0
- virtual void **GetVector** (XOverList &v) const =0
- virtual void **SetVector** (const std::vector< double > &v, int &i)=0
- virtual void **SetVector** (const XOverList &v, int &i)=0
- virtual void **GetStepVector** (std::vector< double > &v) const =0
- virtual double **CompareVector** (const std::vector< double > &v, int &i) const =0
- virtual void **Add** (ChromElement *pChromElement)
- virtual void **Print** (std::ostream &s) const
- Rand & **GetRand** () const
- void **CauchyMutate** (double mean, double variance)
- RBTDLL_EXPORT double **Compare** (const ChromElement &c) const
- bool **Equals** (const ChromElement &c, double threshold) const
- void **SetVector** (const std::vector< double > &v)
- void **SetVector** (const XOverList &v)

Static Public Attributes

- static const std::string **_CT** = "Chrom"

Static Public Attributes inherited from rxdock::ChromElement

- static const std::string **_CT** = "ChromElement"
- static double **_THRESHOLD** = 1E-4

Additional Inherited Members

Public Types inherited from rxdock::ChromElement

- enum **eMode** { **FIXED** = 0 , **TETHERED** = 1 , **FREE** = 2 }

Static Public Member Functions inherited from rxdock::ChromElement

- static RBTDLL_EXPORT double & **GetThreshold** ()
- static eMode **StrToMode** (const std::string &modeStr)
- static std::string RBTDLL_EXPORT **ModeToStr** (eMode mode)

Protected Member Functions inherited from rxdock::ChromElement

- RBTDLL_EXPORT **ChromElement** (const ChromElement &c)
- RBTDLL_EXPORT ChromElement & **operator=** (const ChromElement &c)
- bool **VectorOK** (const std::vector< double > &v, unsigned int i) const
- bool **VectorOK** (const XOverList &v, unsigned int i) const

5.44.1 Member Function Documentation

5.44.1.1 Add()

```
void Chrom::Add (
            ChromElement * pChromElement )  [virtual]
```

Reimplemented from rxdock::ChromElement.

5.44.1.2 clone()

```
ChromElement * Chrom::clone ( ) const  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.3 CompareVector()

```
double Chrom::CompareVector (
            const std::vector< double > & v,
            int & i ) const  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.4 GetLength()

```
int Chrom::GetLength ( ) const  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.5 GetStepVector()

```
void Chrom::GetStepVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.6 GetVector() [1/2]

```
void Chrom::GetVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.7 GetVector() [2/2]

```
void Chrom::GetVector (
            XOverList & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.8 GetXOverLength()

```
int Chrom::GetXOverLength ( ) const  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.9 Mutate()

```
void Chrom::Mutate (
            double relStepSize )  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.10 Print()

```
void Chrom::Print (
            std::ostream & s ) const  [virtual]
```

Reimplemented from rxdock::ChromElement.

5.44.1.11 Randomise()

```
void Chrom::Randomise ( )  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.12 Reset()

```
void Chrom::Reset ( )  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.13 SetVector() [1/2]

```
void Chrom::SetVector (
            const std::vector< double > & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.14 SetVector() [2/2]

```
void Chrom::SetVector (
            const XOverList & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.15 SyncFromModel()

```
void Chrom::SyncFromModel ( )  [virtual]
```

Implements rxdock::ChromElement.

5.44.1.16 SyncToModel()

```
void Chrom::SyncToModel ( )  [virtual]
```

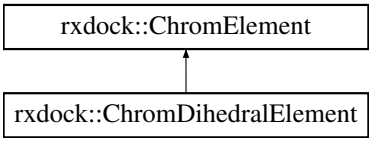
Implements rxdock::ChromElement.

The documentation for this class was generated from the following files:

- Chrom.h
- Chrom.cxx

5.45 rxdock::ChromDihedralElement Class Reference

Inheritance diagram for rxdock::ChromDihedralElement:



Public Member Functions

- RBTDLL_EXPORT **ChromDihedralElement** (BondPtr spBond, AtomList tetheredAtoms, double stepSize, ChromElement::eMode mode=ChromElement::FREE, double maxDihedral=0.0)
- virtual void Reset ()
- virtual void Randomise ()
- virtual void Mutate (double relStepSize)
- virtual void SyncFromModel ()
- virtual void SyncToModel ()
- virtual ChromElement * clone () const
- virtual int GetLength () const
- virtual int GetXOverLength () const
- virtual void GetVector (std::vector< double > &v) const
- virtual void GetVector (XOverList &v) const
- virtual void SetVector (const std::vector< double > &v, int &i)
- virtual void SetVector (const XOverList &v, int &i)
- virtual void GetStepVector (std::vector< double > &v) const
- virtual double CompareVector (const std::vector< double > &v, int &i) const
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ChromElement

- virtual void **Reset** ()=0
- virtual void **Randomise** ()=0
- virtual void **Mutate** (double relStepSize)=0
- virtual void **SyncFromModel** ()=0
- virtual void **SyncToModel** ()=0
- virtual ChromElement * **clone** () const =0
- virtual int **GetLength** () const =0
- virtual int **GetXOverLength** () const =0
- virtual void **GetVector** (std::vector< double > &v) const =0
- virtual void **GetVector** (XOverList &v) const =0
- virtual void **SetVector** (const std::vector< double > &v, int &i)=0
- virtual void **SetVector** (const XOverList &v, int &i)=0
- virtual void **GetStepVector** (std::vector< double > &v) const =0
- virtual double **CompareVector** (const std::vector< double > &v, int &i) const =0
- virtual void **Add** (ChromElement *pChromElement)
- virtual void **Print** (std::ostream &s) const
- Rand & **GetRand** () const
- void **CauchyMutate** (double mean, double variance)
- RBTDLL_EXPORT double **Compare** (const ChromElement &c) const
- bool **Equals** (const ChromElement &c, double threshold) const
- void **SetVector** (const std::vector< double > &v)
- void **SetVector** (const XOverList &v)

Static Public Member Functions

- static double **StandardisedValue** (double dihedralAngle)

Static Public Member Functions inherited from rxdock::ChromElement

- static RBTDLL_EXPORT double & **GetThreshold** ()
- static eMode **StrToMode** (const std::string &modeStr)
- static std::string RBTDLL_EXPORT **ModeToStr** (eMode mode)

Static Public Attributes

- static const std::string **_CT** = ¨ChromDihedralElement¨

Static Public Attributes inherited from rxdock::ChromElement

- static const std::string **_CT** = ¨ChromElement¨
- static double **_THRESHOLD** = 1E-4

Protected Member Functions

- **ChromDihedralElement** (ChromDihedralRefDataPtr spRefData, double value)
- void **CorrectTetheredDihedral** ()

Protected Member Functions inherited from rxdock::ChromElement

- RBTDLL_EXPORT **ChromElement** (const ChromElement &c)
- RBTDLL_EXPORT ChromElement & **operator=** (const ChromElement &c)
- bool **VectorOK** (const std::vector< double > &v, unsigned int i) const
- bool **VectorOK** (const XOverList &v, unsigned int i) const

Additional Inherited Members

Public Types inherited from rxdock::ChromElement

- enum **eMode** { **FIXED** = 0 , **TETHERED** = 1 , **FREE** = 2 }

5.45.1 Member Function Documentation

5.45.1.1 clone()

```
ChromElement * ChromDihedralElement::clone ( ) const  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.2 CompareVector()

```
double ChromDihedralElement::CompareVector (
            const std::vector< double > & v,
            int & i ) const  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.3 GetLength()

```
virtual int rxdock::ChromDihedralElement::GetLength ( ) const  [inline], [virtual]
```

Implements rxdock::ChromElement.

5.45.1.4 GetStepVector()

```
void ChromDihedralElement::GetStepVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.5 GetVector() [1/2]

```
void ChromDihedralElement::GetVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.6 GetVector() [2/2]

```
void ChromDihedralElement::GetVector (
            XOverList & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.7 GetXOverLength()

```
virtual int rxdock::ChromDihedralElement::GetXOverLength ( ) const  [inline], [virtual]
```

Implements rxdock::ChromElement.

5.45.1.8 Mutate()

```
void ChromDihedralElement::Mutate (
            double relStepSize )  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.9 Print()

```
void ChromDihedralElement::Print (
            std::ostream & s ) const  [virtual]
```

Reimplemented from rxdock::ChromElement.

5.45.1.10 Randomise()

```
void ChromDihedralElement::Randomise ( )  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.11 Reset()

```
void ChromDihedralElement::Reset ( )  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.12 SetVector() [1/2]

```
void ChromDihedralElement::SetVector (
            const std::vector< double > & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.13 SetVector() [2/2]

```
void ChromDihedralElement::SetVector (
            const XOverList & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.14 SyncFromModel()

```
void ChromDihedralElement::SyncFromModel ( )  [virtual]
```

Implements rxdock::ChromElement.

5.45.1.15 SyncToModel()

```
void ChromDihedralElement::SyncToModel ( )  [virtual]
```

Implements rxdock::ChromElement.

The documentation for this class was generated from the following files:

- ChromDihedralElement.h
- ChromDihedralElement.cxx

5.46 rxdock::ChromDihedralRefData Class Reference

Public Member Functions

- **ChromDihedralRefData** (BondPtr spBond, AtomList tetheredAtoms, double stepSize, ChromElement::eMode mode=ChromElement::FREE, double maxDihedral=0.0)
- double **GetStepSize** () const
- ChromElement::eMode **GetMode** () const
- double **GetMaxDihedral** () const
- double **GetModelValue** () const
- void **SetModelValue** (double dihedralAngle)
- double **GetInitialValue** () const

Static Public Attributes

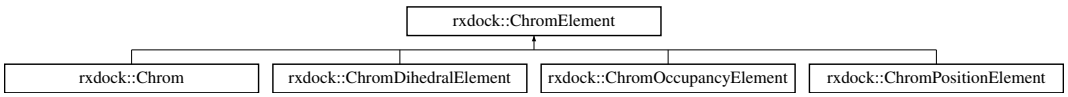
- static const std::string **_CT** = ¨ChromDihedralRefData¨

The documentation for this class was generated from the following files:

- ChromDihedralRefData.h
- ChromDihedralRefData.cxx

5.47 rxdock::ChromElement Class Reference

Inheritance diagram for rxdock::ChromElement:



Public Types

- enum **eMode** { **FIXED** = 0 , **TETHERED** = 1 , **FREE** = 2 }

Public Member Functions

- virtual void **Reset** ()=0
- virtual void **Randomise** ()=0
- virtual void **Mutate** (double relStepSize)=0
- virtual void **SyncFromModel** ()=0
- virtual void **SyncToModel** ()=0
- virtual ChromElement ∗ **clone** () const =0
- virtual int **GetLength** () const =0
- virtual int **GetXOverLength** () const =0
- virtual void **GetVector** (std::vector< double > &v) const =0
- virtual void **GetVector** (XOverList &v) const =0
- virtual void **SetVector** (const std::vector< double > &v, int &i)=0
- virtual void **SetVector** (const XOverList &v, int &i)=0
- virtual void **GetStepVector** (std::vector< double > &v) const =0
- virtual double **CompareVector** (const std::vector< double > &v, int &i) const =0
- virtual void **Add** (ChromElement ∗pChromElement)
- virtual void **Print** (std::ostream &s) const
- Rand & **GetRand** () const
- void **CauchyMutate** (double mean, double variance)
- RBTDLL_EXPORT double **Compare** (const ChromElement &c) const
- bool **Equals** (const ChromElement &c, double threshold) const
- void **SetVector** (const std::vector< double > &v)
- void **SetVector** (const XOverList &v)

Static Public Member Functions

- static RBTDLL_EXPORT double & **GetThreshold** ()
- static eMode **StrToMode** (const std::string &modeStr)
- static std::string RBTDLL_EXPORT **ModeToStr** (eMode mode)

Static Public Attributes

- static const std::string **_CT** = ¨ChromElement¨
- static double **_THRESHOLD** = 1E-4

Protected Member Functions

- RBTDLL_EXPORT **ChromElement** (const ChromElement &c)
- RBTDLL_EXPORT ChromElement & **operator=** (const ChromElement &c)
- bool **VectorOK** (const std::vector< double > &v, unsigned int i) const
- bool **VectorOK** (const XOverList &v, unsigned int i) const

Friends

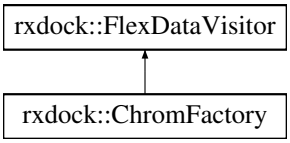
- RBTDLL_EXPORT friend bool **operator==** (const ChromElement &c1, const ChromElement &c2)
- RBTDLL_EXPORT friend bool **operator!=** (const ChromElement &c1, const ChromElement &c2)
- RBTDLL_EXPORT friend std::ostream & **operator**<< (std::ostream &s, const ChromElement &c)

The documentation for this class was generated from the following files:

- ChromElement.h
- ChromElement.cxx

5.48 rxdock::ChromFactory Class Reference

Inheritance diagram for rxdock::ChromFactory:



Public Member Functions

- virtual void VisitReceptorFlexData (ReceptorFlexData *)
- virtual void VisitLigandFlexData (LigandFlexData *)
- virtual void VisitSolventFlexData (SolventFlexData *)
- ChromElement * **GetChrom** () const
- ModelMutatorPtr **GetModelMutator** () const

- virtual void **VisitReceptorFlexData** (ReceptorFlexData *)=0
- virtual void **VisitLigandFlexData** (LigandFlexData *)=0
- virtual void **VisitSolventFlexData** (SolventFlexData *)=0

5.48.1 Member Function Documentation

5.48.1.1 VisitLigandFlexData()

```
void ChromFactory::VisitLigandFlexData (
            LigandFlexData * pFlexData )  [virtual]
```

Implements rxdock::FlexDataVisitor.

5.48.1.2 VisitReceptorFlexData()

```
void ChromFactory::VisitReceptorFlexData (
            ReceptorFlexData * pFlexData )  [virtual]
```

Implements rxdock::FlexDataVisitor.

5.48.1.3 VisitSolventFlexData()

```
void ChromFactory::VisitSolventFlexData (
            SolventFlexData * pFlexData )  [virtual]
```

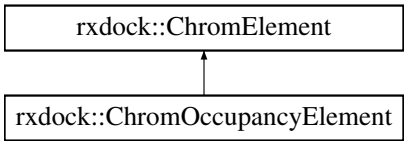
Implements rxdock::FlexDataVisitor.

The documentation for this class was generated from the following files:

- ChromFactory.h
- ChromFactory.cxx

5.49 rxdock::ChromOccupancyElement Class Reference

Inheritance diagram for rxdock::ChromOccupancyElement:



Public Member Functions

- RBTDLL_EXPORT **ChromOccupancyElement** (Model *pModel, double stepSize, double threshold)
- virtual void Reset ()
- virtual void Randomise ()
- virtual void Mutate (double relStepSize)
- virtual void SyncFromModel ()
- virtual void SyncToModel ()
- virtual ChromElement * clone () const
- virtual int GetLength () const
- virtual int GetXOverLength () const
- virtual void GetVector (std::vector< double > &v) const
- virtual void GetVector (XOverList &v) const
- virtual void SetVector (const std::vector< double > &v, int &i)
- virtual void SetVector (const XOverList &v, int &i)
- virtual void GetStepVector (std::vector< double > &v) const
- virtual double CompareVector (const std::vector< double > &v, int &i) const
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ChromElement

- virtual void **Reset** ()=0
- virtual void **Randomise** ()=0
- virtual void **Mutate** (double relStepSize)=0
- virtual void **SyncFromModel** ()=0
- virtual void **SyncToModel** ()=0
- virtual ChromElement * **clone** () const =0
- virtual int **GetLength** () const =0
- virtual int **GetXOverLength** () const =0
- virtual void **GetVector** (std::vector< double > &v) const =0
- virtual void **GetVector** (XOverList &v) const =0
- virtual void **SetVector** (const std::vector< double > &v, int &i)=0
- virtual void **SetVector** (const XOverList &v, int &i)=0
- virtual void **GetStepVector** (std::vector< double > &v) const =0
- virtual double **CompareVector** (const std::vector< double > &v, int &i) const =0
- virtual void **Add** (ChromElement *pChromElement)
- virtual void **Print** (std::ostream &s) const
- Rand & **GetRand** () const
- void **CauchyMutate** (double mean, double variance)
- RBTDLL_EXPORT double **Compare** (const ChromElement &c) const
- bool **Equals** (const ChromElement &c, double threshold) const
- void **SetVector** (const std::vector< double > &v)
- void **SetVector** (const XOverList &v)

Static Public Member Functions

- static double **StandardisedValue** (double occupancy)

Static Public Member Functions inherited from rxdock::ChromElement

- static RBTDLL_EXPORT double & **GetThreshold** ()
- static eMode **StrToMode** (const std::string &modeStr)
- static std::string RBTDLL_EXPORT **ModeToStr** (eMode mode)

Static Public Attributes

- static const std::string **_CT** = ¨ChromOccupancyElement¨

Static Public Attributes inherited from rxdock::ChromElement

- static const std::string **_CT** = ¨ChromElement¨
- static double **_THRESHOLD** = 1E-4

Protected Member Functions

- **ChromOccupancyElement** (ChromOccupancyRefDataPtr spRefData, double value)

Protected Member Functions inherited from rxdock::ChromElement

- RBTDLL_EXPORT **ChromElement** (const ChromElement &c)
- RBTDLL_EXPORT ChromElement & **operator=** (const ChromElement &c)
- bool **VectorOK** (const std::vector< double > &v, unsigned int i) const
- bool **VectorOK** (const XOverList &v, unsigned int i) const

Additional Inherited Members

Public Types inherited from rxdock::ChromElement

- enum **eMode** { **FIXED** = 0 , **TETHERED** = 1 , **FREE** = 2 }

5.49.1 Member Function Documentation

5.49.1.1 clone()

```
ChromElement * ChromOccupancyElement::clone ( ) const  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.2 CompareVector()

```
double ChromOccupancyElement::CompareVector (
            const std::vector< double > & v,
            int & i ) const  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.3 GetLength()

```
virtual int rxdock::ChromOccupancyElement::GetLength ( ) const  [inline], [virtual]
```

Implements rxdock::ChromElement.

5.49.1.4 GetStepVector()

```
void ChromOccupancyElement::GetStepVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.5 GetVector() [1/2]

```
void ChromOccupancyElement::GetVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.6 GetVector() [2/2]

```
void ChromOccupancyElement::GetVector (
            XOverList & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.7 GetXOverLength()

```
virtual int rxdock::ChromOccupancyElement::GetXOverLength ( ) const  [inline], [virtual]
```

Implements rxdock::ChromElement.

5.49.1.8 Mutate()

```
void ChromOccupancyElement::Mutate (
            double relStepSize )  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.9 Print()

```
void ChromOccupancyElement::Print (
            std::ostream & s ) const  [virtual]
```

Reimplemented from rxdock::ChromElement.

5.49.1.10 Randomise()

```
void ChromOccupancyElement::Randomise ( )  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.11 Reset()

```
void ChromOccupancyElement::Reset ( )  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.12 SetVector() [1/2]

```
void ChromOccupancyElement::SetVector (
            const std::vector< double > & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.13 SetVector() [2/2]

```
void ChromOccupancyElement::SetVector (
            const XOverList & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.14 SyncFromModel()

```
void ChromOccupancyElement::SyncFromModel ( )  [virtual]
```

Implements rxdock::ChromElement.

5.49.1.15 SyncToModel()

```
void ChromOccupancyElement::SyncToModel ( )  [virtual]
```

Implements rxdock::ChromElement.

The documentation for this class was generated from the following files:

- ChromOccupancyElement.h
- ChromOccupancyElement.cxx

5.50 rxdock::ChromOccupancyRefData Class Reference

Public Member Functions

- **ChromOccupancyRefData** (Model *pModel, double stepSize, double threshold)
- double **GetStepSize** () const
- double **GetModelValue** () const
- void **SetModelValue** (double occupancy)
- double **GetInitialValue** () const

Static Public Attributes

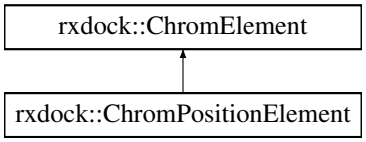
- static const std::string **_CT** = ¨ChromOccupancyRefData¨

The documentation for this class was generated from the following files:

- ChromOccupancyRefData.h
- ChromOccupancyRefData.cxx

5.51 rxdock::ChromPositionElement Class Reference

Inheritance diagram for rxdock::ChromPositionElement:



Public Member Functions

- RBTDLL_EXPORT **ChromPositionElement** (const Model *pModel, const DockingSite *pDockSite, double transStepSize, double rotStepSize, ChromElement::eMode transMode=ChromElement::FREE, ChromElement::eMode rotMode=ChromElement::FREE, double maxTrans=0.0, double maxRot=0.0)
- virtual void Reset ()
- virtual void Randomise ()
- virtual void Mutate (double relStepSize)
- virtual void SyncFromModel ()
- virtual void SyncToModel ()
- virtual ChromElement * clone () const
- virtual int GetLength () const
- virtual int GetXOverLength () const
- virtual void GetVector (std::vector< double > &v) const
- virtual void GetVector (XOverList &v) const
- virtual void SetVector (const std::vector< double > &v, int &i)
- virtual void SetVector (const XOverList &v, int &i)
- virtual void GetStepVector (std::vector< double > &v) const
- virtual double CompareVector (const std::vector< double > &v, int &i) const
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ChromElement

- virtual void **Reset** ()=0
- virtual void **Randomise** ()=0
- virtual void **Mutate** (double relStepSize)=0
- virtual void **SyncFromModel** ()=0
- virtual void **SyncToModel** ()=0
- virtual ChromElement * **clone** () const =0
- virtual int **GetLength** () const =0
- virtual int **GetXOverLength** () const =0
- virtual void **GetVector** (std::vector< double > &v) const =0
- virtual void **GetVector** (XOverList &v) const =0
- virtual void **SetVector** (const std::vector< double > &v, int &i)=0
- virtual void **SetVector** (const XOverList &v, int &i)=0
- virtual void **GetStepVector** (std::vector< double > &v) const =0
- virtual double **CompareVector** (const std::vector< double > &v, int &i) const =0
- virtual void **Add** (ChromElement *pChromElement)
- virtual void **Print** (std::ostream &s) const
- Rand & **GetRand** () const
- void **CauchyMutate** (double mean, double variance)
- RBTDLL_EXPORT double **Compare** (const ChromElement &c) const
- bool **Equals** (const ChromElement &c, double threshold) const
- void **SetVector** (const std::vector< double > &v)
- void **SetVector** (const XOverList &v)

Static Public Member Functions

- static double **StandardisedValue** (double rotationAngle)

Static Public Member Functions inherited from rxdock::ChromElement

- static RBTDLL_EXPORT double & **GetThreshold** ()
- static eMode **StrToMode** (const std::string &modeStr)
- static std::string RBTDLL_EXPORT **ModeToStr** (eMode mode)

Static Public Attributes

- static const std::string **_CT** = ¨ChromPositionElement¨

Static Public Attributes inherited from rxdock::ChromElement

- static const std::string **_CT** = ¨ChromElement¨
- static double **_THRESHOLD** = 1E-4

Protected Member Functions

- **ChromPositionElement** (ChromPositionRefDataPtr spRefData, const Coord &com, const Euler &orientation)
- void **RandomiseCOM** ()
- void **RandomiseOrientation** ()
- void **MutateCOM** (double relStepSize)
- void **MutateOrientation** (double relStepSize)
- void **CorrectTetheredCOM** ()
- void **CorrectTetheredOrientation** ()

Protected Member Functions inherited from rxdock::ChromElement

- RBTDLL_EXPORT **ChromElement** (const ChromElement &c)
- RBTDLL_EXPORT ChromElement & **operator=** (const ChromElement &c)
- bool **VectorOK** (const std::vector< double > &v, unsigned int i) const
- bool **VectorOK** (const XOverList &v, unsigned int i) const

Additional Inherited Members

Public Types inherited from rxdock::ChromElement

- enum **eMode** { **FIXED** = 0 , **TETHERED** = 1 , **FREE** = 2 }

5.51.1 Member Function Documentation

5.51.1.1 clone()

```
ChromElement * ChromPositionElement::clone ( ) const  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.2 CompareVector()

```
double ChromPositionElement::CompareVector (
            const std::vector< double > & v,
            int & i ) const  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.3 GetLength()

```
virtual int rxdock::ChromPositionElement::GetLength ( ) const  [inline], [virtual]
```

Implements rxdock::ChromElement.

5.51.1.4 GetStepVector()

```
void ChromPositionElement::GetStepVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.5 GetVector() [1/2]

```
void ChromPositionElement::GetVector (
            std::vector< double > & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.6 GetVector() [2/2]

```
void ChromPositionElement::GetVector (
            XOverList & v ) const  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.7 GetXOverLength()

```
virtual int rxdock::ChromPositionElement::GetXOverLength ( ) const  [inline], [virtual]
```

Implements rxdock::ChromElement.

5.51.1.8 Mutate()

```
void ChromPositionElement::Mutate (
            double relStepSize )  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.9 Print()

```
void ChromPositionElement::Print (
            std::ostream & s ) const  [virtual]
```

Reimplemented from rxdock::ChromElement.

5.51.1.10 Randomise()

```
void ChromPositionElement::Randomise ( )  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.11 Reset()

```
void ChromPositionElement::Reset ( )  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.12 SetVector() [1/2]

```
void ChromPositionElement::SetVector (
            const std::vector< double > & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.13 SetVector() [2/2]

```
void ChromPositionElement::SetVector (
            const XOverList & v,
            int & i )  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.14 SyncFromModel()

```
void ChromPositionElement::SyncFromModel ( )  [virtual]
```

Implements rxdock::ChromElement.

5.51.1.15 SyncToModel()

```
void ChromPositionElement::SyncToModel ( )  [virtual]
```

Implements rxdock::ChromElement.

The documentation for this class was generated from the following files:

- ChromPositionElement.h
- ChromPositionElement.cxx

5.52 rxdock::ChromPositionRefData Class Reference

Public Member Functions

- **ChromPositionRefData** (const Model *pModel, const DockingSite *pDockSite, double transStepSize, double rotStepSize, ChromElement::eMode transMode=ChromElement::FREE, ChromElement::eMode rotMode=ChromElement::FREE, double maxTrans=0.0, double maxRot=0.0)
- int **GetNumStartCoords** () const
- const Coord & **GetStartCoord** (int iCoord) const
- double **GetTransStepSize** () const
- double **GetRotStepSize** () const
- ChromElement::eMode **GetTransMode** () const
- ChromElement::eMode **GetRotMode** () const
- int **GetLength** () const
- int **GetXOverLength** () const
- bool **IsTransFixed** () const
- bool **IsRotFixed** () const
- double **GetMaxTrans** () const
- double **GetMaxRot** () const
- const Coord & **GetInitialCOM** () const
- const Euler & **GetInitialOrientation** () const
- const Quat & **GetInitialQuat** () const
- void **GetModelValue** (Coord &com, Euler &orientation) const
- void **SetModelValue** (const Coord &com, const Euler &orientation)

Static Public Attributes

- static const std::string **_CT** = ¨ChromPositionRefData¨
- static const PrincipalAxes **CARTESIAN_AXES**

Friends

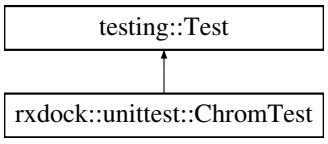
- void **to_json** (json &j, const ChromPositionRefData &chrposrdata)
- void **from_json** (const json &j, ChromPositionRefData &chrposrdata)

The documentation for this class was generated from the following files:

- ChromPositionRefData.h
- ChromPositionRefData.cxx

5.53 rxdock::unittest::ChromTest Class Reference

Inheritance diagram for rxdock::unittest::ChromTest:



Protected Member Functions

- void **SetUp** () override
- void **TearDown** () override
- void **setupWorkSpace** ()
- double **rmsd** (const CoordList &rc, const CoordList &c)
- void **measureRandOrMutateDiff** (ChromElement ∗chrom, int nTrials, bool bMutate, double &meanDiff, double &minDiff, double &maxDiff)
- void **measureCrossoverDiff** (ChromElement ∗chrom, int nTrials, double &meanDiff, double &minDiff, double &maxDiff)

Protected Attributes

- ModelPtr **m_recep_1koc**
- ModelPtr **m_lig_1koc**
- DockingSitePtr **m_site_1koc**
- ChromElementPtr **m_chrom_1koc**
- AtomList **m_atomList**
- BiMolWorkSpacePtr **m_workSpace**
- SFAggPtr **m_SF**

Static Protected Attributes

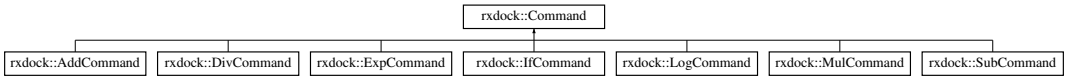
- static double **TINY** = 1E-4

The documentation for this class was generated from the following files:

- ChromTest.h
- ChromTest.cxx

5.54 rxdock::Command Class Reference

Inheritance diagram for rxdock::Command:



Public Member Functions

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Static Public Member Functions

- static void **Clear** ()

Static Public Attributes

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions

- std::string **tabs** (int n)

The documentation for this class was generated from the following files:

- Command.h
- GPParser.cxx

5.55 rxdock::Commands Class Reference

Public Types

- enum **CommName** {
  ADD = 0 , **SUB** = 1 , **MUL** = 2 , **DIV** = 3 ,
  IF = 4 , **LOG** = 5 , **EXP** = 6 , **RCTE** = 7 ,
  AND = 8 , **NULLC** = -1 }

Public Member Functions

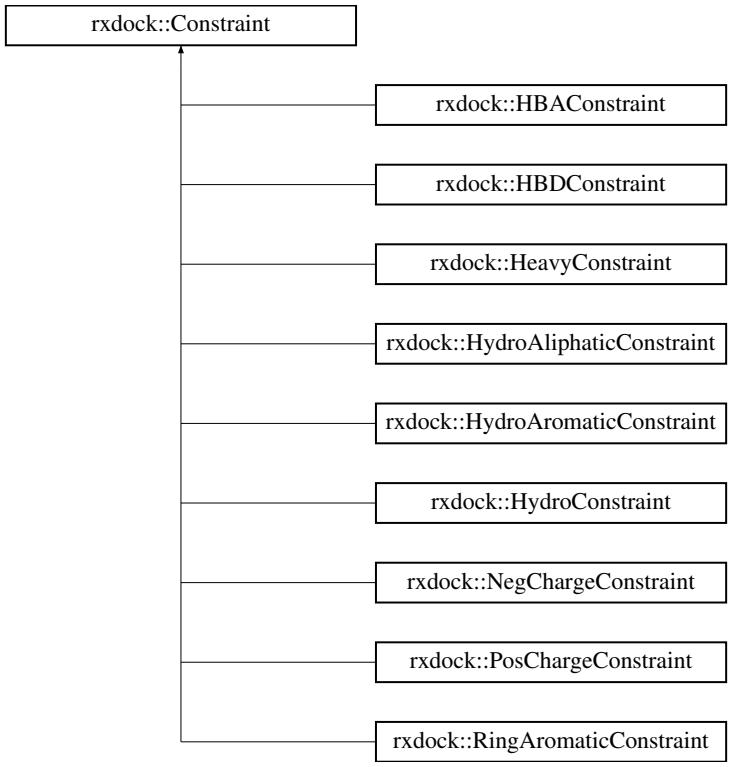
- **Commands** (const Commands &c)
- **Commands** (int ncomm)
- CommName **GetName** () const
- int **GetNArgs** () const
- bool **IsAdd** () const
- bool **IsSub** () const
- bool **IsMul** () const
- bool **IsDiv** () const
- bool **IsIf** () const
- bool **IsLog** () const
- bool **IsExp** () const
- bool **IsRCte** () const
- bool **IsAnd** () const
- virtual void **copy** (const Commands &c)

The documentation for this class was generated from the following file:

- Commands.h

5.56 rxdock::Constraint Class Reference

Inheritance diagram for rxdock::Constraint:



Public Member Functions

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static const std::string **_CT** = ¨Constraint¨

Protected Attributes

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

Friends

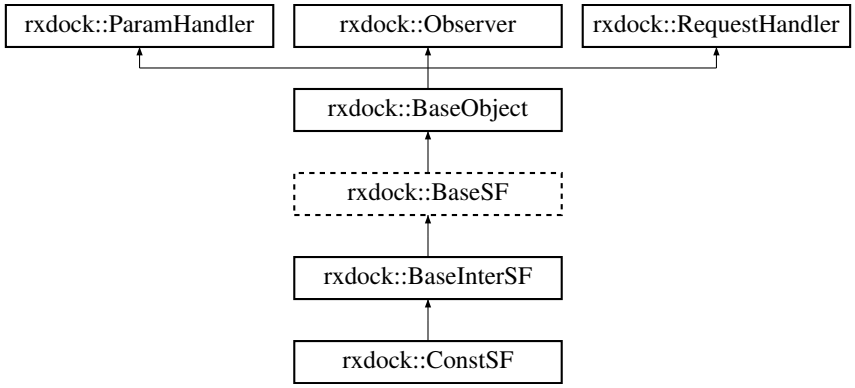
- void **to_json** (json &j, const Constraint &constraint)
- void **from_json** (const json &j, Constraint &constraint)
- std::ostream & **operator**<< (std::ostream &, const Constraint &)

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.57 rxdock::ConstSF Class Reference

Inheritance diagram for rxdock::ConstSF:



Public Member Functions

- **ConstSF** (const std::string &strName="const")
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨ConstSF¨
- static const std::string **_SOLVENT_PENALTY** = ¨solvent-penalty¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const ConstSF &consf)
- void **from_json** (const json &j, ConstSF &consf)

5.57.1 Member Function Documentation

5.57.1.1 ParameterUpdated()

```
void ConstSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.57.1.2 RawScore()

```
double ConstSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.57.1.3 ScoreMap()

```
void ConstSF::ScoreMap (
            StringVariantMap & scoreMap ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.57.1.4 SetupLigand()

```
virtual void rxdock::ConstSF::SetupLigand ( )  [inline], [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.57.1.5 SetupReceptor()

```
virtual void rxdock::ConstSF::SetupReceptor ( )  [inline], [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.57.1.6 SetupScore()

```
virtual void rxdock::ConstSF::SetupScore ( )  [inline], [protected], [virtual]
```

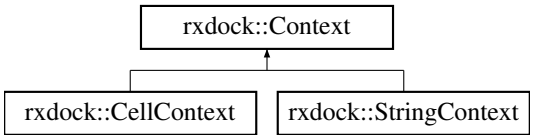
Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- ConstSF.h
- ConstSF.cxx

5.58 rxdock::Context Class Reference

Inheritance diagram for rxdock::Context:



Public Member Functions

- **Context** (const Context &c)
- virtual void **Assign** (std::string, ReturnType)=0
- virtual void **Assign** (int, ReturnType)=0
- virtual const Vble & **GetVble** (int)=0
- virtual const Vble & **GetVble** (std::string)=0
- virtual void **SetVble** (int key, const Vble &v)=0

Static Public Attributes

- static const std::string **_CT** = ¨Context¨

The documentation for this class was generated from the following files:

- Context.h
- Context.cxx

5.59 rxdock::Coord Class Reference

Public Member Functions

- **Coord** (double x1, double y1, double z1)
- **Coord** (Eigen::Vector3d xyz1)
- **Coord** (const Coord &coord)
- Coord & **operator=** (const Coord &coord)
- Coord & **operator=** (const double &d)
- void **operator+=** (const Coord &coord)
- void **operator+=** (const double &d)
- void **operator-=** (const Coord &coord)
- void **operator-=** (const double &d)
- void **operator∗=** (const double &d)
- void **operator/=** (const double &d)
- double **Length2** () const
- double **Length** () const
- Coord **Unit** () const
- Coord **Cross** (const Coord &v2) const
- double **Dot** (const Coord &v2) const

Public Attributes

- Eigen::Vector3d **xyz**

Friends

- std::ostream & **operator**<< (std::ostream &s, const Coord &coord)
- std::istream & **operator**>> (std::istream &s, Coord &coord)
- bool **operator==** (const Coord &coord1, const Coord &coord2)
- bool **operator!=** (const Coord &coord1, const Coord &coord2)
- bool **operator**> (const Coord &coord1, const Coord &coord2)
- bool **operator**>= (const Coord &coord1, const Coord &coord2)
- bool **operator**< (const Coord &coord1, const Coord &coord2)
- bool **operator**<= (const Coord &coord1, const Coord &coord2)
- Coord **operator+** (const Coord &coord1, const Coord &coord2)
- Coord **operator+** (const Coord &coord1, double d)
- Coord **operator+** (double d, const Coord &coord1)
- Coord **operator-** (const Coord &coord1, const Coord &coord2)
- Coord **operator-** (const Coord &coord1, double d)
- Coord **operator-** (double d, const Coord &coord1)
- Coord **operator-** (const Coord &coord)
- Coord **operator**∗ (const Coord &coord, const double &d)
- Coord **operator**∗ (const double &d, const Coord &coord)
- Coord **operator**∗ (const Coord &coord1, const Coord &coord2)
- Coord **operator**/ (const Coord &coord, const double &d)
- void **to_json** (json &j, const Coord &c)
- void **from_json** (const json &j, Coord &c)

The documentation for this class was generated from the following file:

- Coord.h

5.60 rxdock::CoordCmp Class Reference

Public Member Functions

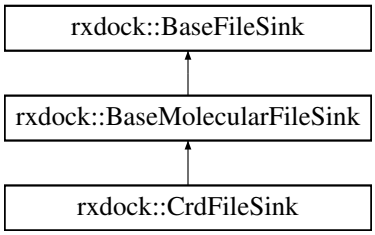
- bool **operator()** (const Coord &c1, const Coord &c2) const

The documentation for this class was generated from the following file:

- DockingSite.cxx

5.61 rxdock::CrdFileSink Class Reference

Inheritance diagram for rxdock::CrdFileSink:



Public Member Functions

- RBTDLL_EXPORT **CrdFileSink** (const std::string &fileName, ModelPtr spModel)
- virtual bool isMultiConfSupported ()
- virtual void Render ()

Public Member Functions inherited from rxdock::BaseMolecularFileSink

- **BaseMolecularFileSink** (const std::string &fileName, ModelPtr spModel, bool bUseModelSegment↩ Names=true)
- virtual bool **isMultiConfSupported** ()
- ModelPtr **GetModel** () const
- RBTDLL_EXPORT void **SetModel** (ModelPtr spModel, bool bUseModelSegmentNames=false)
- void **SetNextSegmentId** (int nSegmentId)
- bool **GetMultiConf** () const
- void **SetMultiConf** (bool bMultiConf)
- void **WriteMultiConf** ()
- ModelList **GetSolvent** () const
- void **SetSolvent** (ModelList solventList)

Public Member Functions inherited from rxdock::BaseFileSink

- **BaseFileSink** (const std::string &fileName)
- std::string **GetFileName** () const
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- Error **Status** ()
- virtual void **Render** ()=0

Friends

- void **to_json** (json &j, const CrdFileSink &crdFileSink)
- void **from_json** (const json &j, CrdFileSink &crdFileSink)

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseMolecularFileSink

- void **Reset** ()

Protected Member Functions inherited from rxdock::BaseFileSink

- void **Write** (bool bClearCache=true)
- void **AddLine** (const std::string &fileRec)
- void **ReplaceLine** (const std::string &fileRec, unsigned int nRec)
- bool **isCacheEmpty** () const
- bool **GetAppend** () const
- void **SetAppend** (bool bAppend)

Protected Attributes inherited from rxdock::BaseMolecularFileSink

- int **m_nAtomId**
- int **m_nSubunitId**
- int **m_nSegmentId**
- bool **m_bUseModelSegmentNames**

5.61.1 Member Function Documentation

5.61.1.1 isMultiConfSupported()

```
virtual bool rxdock::CrdFileSink::isMultiConfSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularFileSink.

5.61.1.2 Render()

```
void CrdFileSink::Render ( )  [virtual]
```

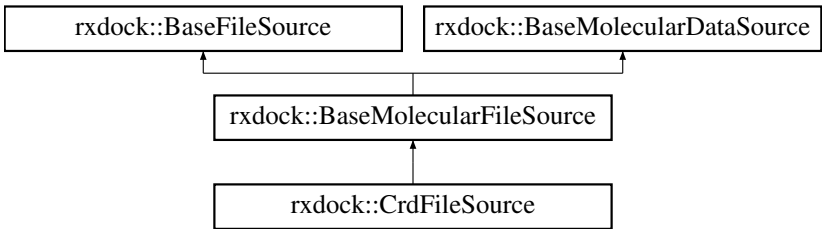
Implements rxdock::BaseFileSink.

The documentation for this class was generated from the following files:

- CrdFileSink.h
- CrdFileSink.cxx

5.62 rxdock::CrdFileSource Class Reference

Inheritance diagram for rxdock::CrdFileSource:



Public Member Functions

- **CrdFileSource** (const char *fileName)
- **CrdFileSource** (const std::string fileName)
- virtual bool isTitleListSupported ()
- virtual bool isAtomListSupported ()
- virtual bool isCoordinatesSupported ()
- virtual bool isBondListSupported ()

Public Member Functions inherited from rxdock::BaseMolecularFileSource

- RBTDLL_EXPORT **BaseMolecularFileSource** (const std::string &fileName, const std::string &sourceName)
- **BaseMolecularFileSource** (const std::string &fileName, const std::string &strRecDelim, const std::string &sourceName)
- void Reset ()
- int GetNumTitles ()
- int GetNumAtoms ()
- int GetNumBonds ()
- int GetNumSegments ()
- std::vector< std::string > GetTitleList ()
- AtomList GetAtomList ()
- BondList GetBondList ()
- SegmentMap GetSegmentMap ()
- virtual int GetNumData ()
- virtual std::vector< std::string > GetDataFieldList ()
- virtual StringVariantMap GetDataMap ()
- virtual bool isDataFieldPresent (const std::string &strDataField)
- virtual Variant GetDataValue (const std::string &strDataField)
- SegmentMap **GetSegmentFilterMap** ()
- RBTDLL_EXPORT void **SetSegmentFilterMap** (const SegmentMap &segmentFilterMap)
- void **ClearSegmentFilterMap** ()
- bool **isSegmentFilterMapDefined** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Public Member Functions inherited from rxdock::BaseMolecularDataSource

- **BaseMolecularDataSource** (std::string strName)
- std::string **GetName** ()
- virtual bool **isTitleListSupported** ()
- virtual bool **isAtomListSupported** ()
- virtual bool **isCoordinatesSupported** ()
- virtual bool **isBondListSupported** ()
- virtual void **Reset** ()=0
- virtual int **GetNumTitles** ()=0
- virtual int **GetNumAtoms** ()=0
- virtual int **GetNumBonds** ()=0
- virtual int **GetNumSegments** ()=0
- virtual std::vector< std::string > **GetTitleList** ()=0

- virtual AtomList **GetAtomList** ()=0
- virtual BondList **GetBondList** ()=0
- virtual SegmentMap **GetSegmentMap** ()=0
- virtual bool **isDataSupported** ()
- virtual int **GetNumData** ()=0
- virtual std::vector< std::string > **GetDataFieldList** ()=0
- virtual StringVariantMap **GetDataMap** ()=0
- virtual bool **isDataFieldPresent** (const std::string &strDataField)=0
- virtual Variant **GetDataValue** (const std::string &strDataField)=0

Protected Member Functions

- virtual void Parse ()

Protected Member Functions inherited from rxdock::BaseMolecularFileSource

- void **ClearMolCache** ()
- void **RemoveAtom** (AtomPtr spAtom)
- void **RenumberAtomsAndBonds** ()
- void **SetupPartiallonicGroups** (AtomList &atoms, ParameterFileSourcePtr spParamSource)

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Additional Inherited Members

Protected Attributes inherited from rxdock::BaseMolecularFileSource

- std::vector< std::string > **m_titleList**
- AtomList **m_atomList**
- BondList **m_bondList**
- SegmentMap **m_segmentMap**
- StringVariantMap **m_dataMap**

Protected Attributes inherited from rxdock::BaseFileSource

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

5.62.1 Member Function Documentation

5.62.1.1 isAtomListSupported()

```
virtual bool rxdock::CrdFileSource::isAtomListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.62.1.2 isBondListSupported()

```
virtual bool rxdock::CrdFileSource::isBondListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.62.1.3 isCoordinatesSupported()

```
virtual bool rxdock::CrdFileSource::isCoordinatesSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.62.1.4 isTitleListSupported()

```
virtual bool rxdock::CrdFileSource::isTitleListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.62.1.5 Parse()

```
void CrdFileSource::Parse (
            void  )  [protected], [virtual]
```

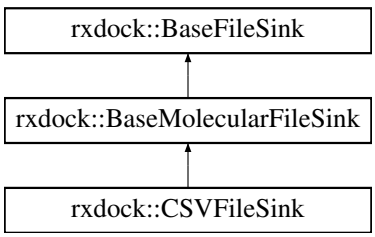
Implements rxdock::BaseFileSource.

The documentation for this class was generated from the following files:

- CrdFileSource.h
- CrdFileSource.cxx

5.63 rxdock::CSVFileSink Class Reference

Inheritance diagram for rxdock::CSVFileSink:



Public Member Functions

- RBTDLL_EXPORT **CSVFileSink** (const std::string &fileName, ModelPtr spModel, std::size_t nAtoms, std↩ ::size_t nDataFields)
- virtual void Render ()

Public Member Functions inherited from rxdock::BaseMolecularFileSink

- **BaseMolecularFileSink** (const std::string &fileName, ModelPtr spModel, bool bUseModelSegmentNames=true)
- virtual bool **isMultiConfSupported** ()
- ModelPtr **GetModel** () const
- RBTDLL_EXPORT void **SetModel** (ModelPtr spModel, bool bUseModelSegmentNames=false)
- void **SetNextSegmentId** (int nSegmentId)
- bool **GetMultiConf** () const
- void **SetMultiConf** (bool bMultiConf)
- void **WriteMultiConf** ()
- ModelList **GetSolvent** () const
- void **SetSolvent** (ModelList solventList)

Public Member Functions inherited from rxdock::BaseFileSink

- **BaseFileSink** (const std::string &fileName)
- std::string **GetFileName** () const
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- Error **Status** ()
- virtual void **Render** ()=0

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseMolecularFileSink

- void **Reset** ()

Protected Member Functions inherited from rxdock::BaseFileSink

- void **Write** (bool bClearCache=true)
- void **AddLine** (const std::string &fileRec)
- void **ReplaceLine** (const std::string &fileRec, unsigned int nRec)
- bool **isCacheEmpty** () const
- bool **GetAppend** () const
- void **SetAppend** (bool bAppend)

Protected Attributes inherited from rxdock::BaseMolecularFileSink

- int **m_nAtomId**
- int **m_nSubunitId**
- int **m_nSegmentId**
- bool **m_bUseModelSegmentNames**

5.63.1 Member Function Documentation

5.63.1.1 Render()

```
void CSVFileSink::Render ( )  [virtual]
```

Implements rxdock::BaseFileSink.

The documentation for this class was generated from the following files:

- CSVFileSink.h
- CSVFileSink.cxx

5.64 rxdock::CyclicAtom Class Reference

Public Member Functions

- **CyclicAtom** (bool bb)
- void **operator()** (Atom *pAtom)

The documentation for this class was generated from the following file:

- Atom.h

5.65 rxdock::DihedralElement Class Reference

Classes

- class prms

Public Member Functions

- **DihedralElement** (Atom *pAtom1, Atom *pAtom2, Atom *pAtom3, Atom *pAtom4, const prms &dihprms)
- double **operator()** () const
- Atom * **GetAtom1Ptr** () const
- Atom * **GetAtom2Ptr** () const
- Atom * **GetAtom3Ptr** () const
- Atom * **GetAtom4Ptr** () const
- void **AddTerm** (const prms &dihprms)

Friends

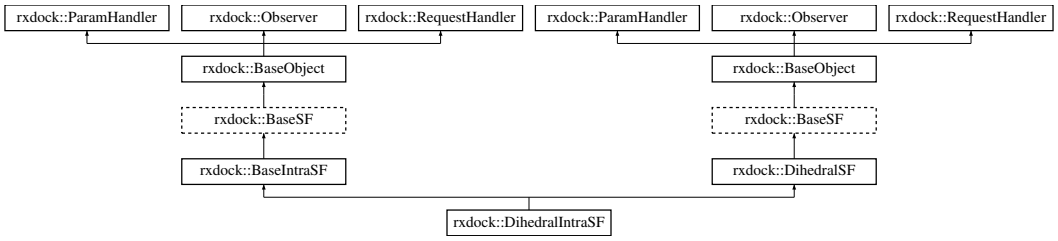
- void **to_json** (json &j, const DihedralElement &dihedralElem)
- void **from_json** (const json &j, DihedralElement &dihedralElem)

The documentation for this class was generated from the following files:

- DihedralSF.h
- DihedralSF.cxx

5.66 rxdock::DihedralIntraSF Class Reference

Inheritance diagram for rxdock::DihedralIntraSF:



Public Member Functions

- **DihedralIntraSF** (const std::string &strName="dihedral")

Public Member Functions inherited from rxdock::BaseIntraSF

- ModelPtr **GetLigand** () const
- virtual void Update (Subject *theChangedSubject)
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = "DihedralIntraSF"

Static Public Attributes inherited from rxdock::BaseIntraSF

- static const std::string **_CT** = "BaseIntraSF"

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = "BaseSF"
- static const std::string **_WEIGHT** = "weight"
- static const std::string **_RANGE** = "range"
- static const std::string **_SYSTEM_SF** = "rxdock.score.system"
- static const std::string **_INTRA_SF** = "rxdock.score.intra"

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = "BaseObject"
- static const std::string **_CLASS** = "CLASS"
- static const std::string **_NAME** = "NAME"
- static const std::string **_ENABLED** = "ENABLED"

Static Public Attributes inherited from rxdock::DihedralSF

- static const std::string **_CT** = "DihedralSF"
- static const std::string **_IMPL_H_CORR** = "implicit-H-correction"

Protected Member Functions

- virtual void SetupScore ()
- virtual double RawScore () const
- void **ClearModel** ()

- virtual void **SetupScore** ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::DihedralSF

- DihedralList **CreateDihedralList** (const BondList &bondList)

Friends

- void **to_json** (json &j, const DihedralIntraSF &dihedralIntraSF)
- void **from_json** (const json &j, DihedralIntraSF &dihedralIntraSF)

5.66.1 Member Function Documentation

5.66.1.1 RawScore()

```
double DihedralIntraSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.66.1.2 SetupScore()

```
void DihedralIntraSF::SetupScore ( )  [protected], [virtual]
```

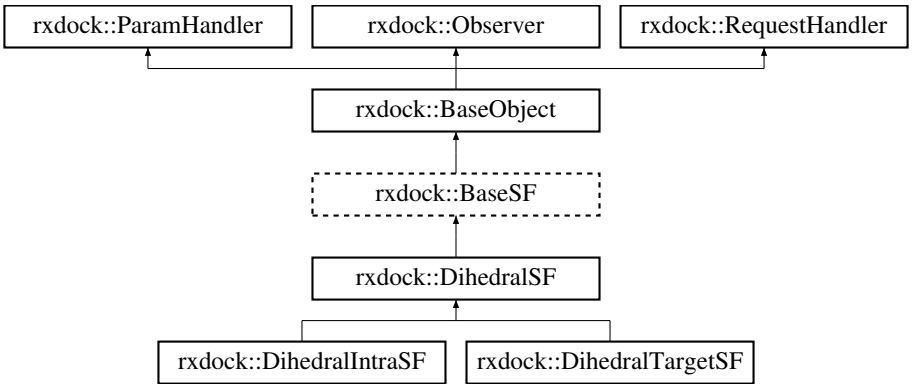
Implements rxdock::BaseIntraSF.

The documentation for this class was generated from the following files:

- DihedralIntraSF.h
- DihedralIntraSF.cxx

5.67 rxdock::DihedralSF Class Reference

Inheritance diagram for rxdock::DihedralSF:



Static Public Attributes

- static const std::string **_CT** = ¨DihedralSF¨
- static const std::string **_IMPL_H_CORR** = ¨implicit-H-correction¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- DihedralList **CreateDihedralList** (const BondList &bondList)

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const DihedralSF &dihedralSF)
- void **from_json** (const json &j, DihedralSF &dihedralSF)

Additional Inherited Members

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Public Member Functions inherited from rxdock::Observer

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

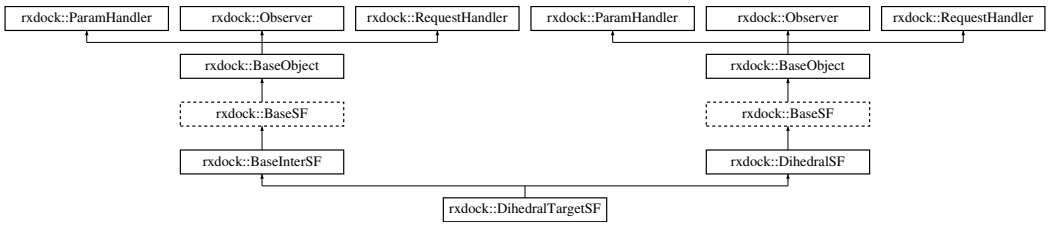
- virtual void **HandleRequest** (RequestPtr spRequest)

The documentation for this class was generated from the following files:

- DihedralSF.h
- DihedralSF.cxx

5.68 rxdock::DihedralTargetSF Class Reference

Inheritance diagram for rxdock::DihedralTargetSF:



Public Member Functions

- **DihedralTargetSF** (const std::string &strName="dihedral")

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨DihedralTargetSF¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::DihedralSF

- static const std::string **_CT** = ¨DihedralSF¨
- static const std::string **_IMPL_H_CORR** = ¨implicit-H-correction¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void **ClearReceptor** ()

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::DihedralSF

- DihedralList **CreateDihedralList** (const BondList &bondList)

5.68.1 Member Function Documentation

5.68.1.1 RawScore()

```
double DihedralTargetSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.68.1.2 SetupLigand()

```
void DihedralTargetSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.68.1.3 SetupReceptor()

```
void DihedralTargetSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.68.1.4 SetupScore()

```
void DihedralTargetSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

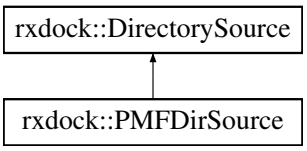
The documentation for this class was generated from the following files:

- DihedralTargetSF.h
- DihedralTargetSF.cxx

5.69 rxdock::DirectorySource Class Reference

`#include <DirectorySource.h>`

Inheritance diagram for rxdock::DirectorySource:



Public Member Functions

- DirectorySource (const std::string &)

Static Public Attributes

- static const std::string **_CT** = ¨DirectorySource¨

Protected Member Functions

- void **CheckDirectory** (std::string &)

Protected Attributes

- struct stat **fStat**
- int **fNum**
- struct dirent ∗∗ **fNameList**
- std::ifstream **inFile**
- std::string **thePath**

5.69.1 Detailed Description

DirectorySource Reads all (or the given number) of files into a <vector> that are in a directory. To extend with other file formasts include other ReadFiles () methods

5.69.2 Constructor & Destructor Documentation

5.69.2.1 DirectorySource()

```
DirectorySource::DirectorySource (
            const std::string & aDirectory )
```

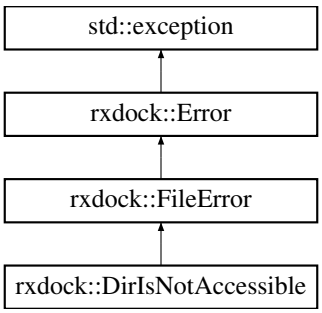
Constructor checks wether the directory exists and gives $RBT_ROOT to the path if required

The documentation for this class was generated from the following files:

- DirectorySource.h
- DirectorySource.cxx

5.70 rxdock::DirIsNotAccessible Class Reference

Inheritance diagram for rxdock::DirIsNotAccessible:



Public Member Functions

- **DirIsNotAccessible** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

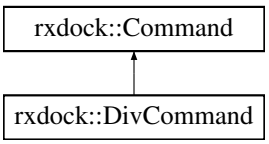
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.71 rxdock::DivCommand Class Reference

Inheritance diagram for rxdock::DivCommand:



Public Member Functions

- int GetNArgs ()
- std::string GetName ()
- void SetArg (int i, ReturnType f)
- void SetNameArg (int i, std::string &n)
- ReturnType Execute ()

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Additional Inherited Members

Static Public Member Functions inherited from rxdock::Command

- static void **Clear** ()

Static Public Attributes inherited from rxdock::Command

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions inherited from rxdock::Command

- std::string **tabs** (int n)

5.71.1 Member Function Documentation

5.71.1.1 Execute()

```
ReturnType rxdock::DivCommand::Execute ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.71.1.2 GetName()

```
std::string rxdock::DivCommand::GetName ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.71.1.3 GetNArgs()

```
int rxdock::DivCommand::GetNArgs ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.71.1.4 SetArg()

```
void rxdock::DivCommand::SetArg (
            int i,
            ReturnType f )  [inline], [virtual]
```

Implements rxdock::Command.

5.71.1.5 SetNameArg()

```
void rxdock::DivCommand::SetNameArg (
            int i,
            std::string & n )  [inline], [virtual]
```

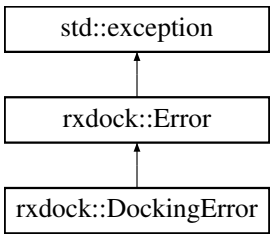
Implements rxdock::Command.

The documentation for this class was generated from the following file:

- Command.h

5.72 rxdock::DockingError Class Reference

Inheritance diagram for rxdock::DockingError:



Public Member Functions

- **DockingError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Protected Member Functions

- **DockingError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- DockingError.h

5.73 rxdock::DockingSite Class Reference

Classes

- class isAtomInRange

Public Member Functions

- RBTDLL_EXPORT **DockingSite** (const CavityList &cavList, double border)
- RBTDLL_EXPORT **DockingSite** (json j)
- virtual void **Print** (std::ostream &s) const
- RBTDLL_EXPORT RealGridPtr **GetGrid** ()
- double **GetBorder** () const
- Coord **GetMinCoord** () const
- Coord **GetMaxCoord** () const
- CavityList **GetCavityList** () const
- int **GetNumCavities** () const
- RBTDLL_EXPORT double **GetVolume** () const
- void **GetCoordList** (CoordList &retVal) const
- RBTDLL_EXPORT AtomList **GetAtomList** (const AtomList &atomList, double minDist, double maxDist)
- RBTDLL_EXPORT AtomList **GetAtomList** (const AtomList &atomList, double maxDist)
- unsigned int **GetNumAtoms** (const AtomList &atomList, double minDist, double maxDist)

Static Public Attributes

- static const std::string **_CT** = ¨DockingSite¨

Friends

- RBTDLL_EXPORT friend std::ostream & **operator**<< (std::ostream &s, const DockingSite &site)
- RBTDLL_EXPORT friend void **to_json** (json &j, const DockingSite &site)
- void **from_json** (const json &j, DockingSite &site)

The documentation for this class was generated from the following files:

- DockingSite.h
- DockingSite.cxx

5.74 rxdock::ElementData Class Reference

Public Member Functions

- **ElementData** (json j)

Public Attributes

- int **atomicNo**
- std::string **element**
- int **minVal**
- int **maxVal**
- int **commonVal**
- double **mass**
- double **vdwRadius**

Friends

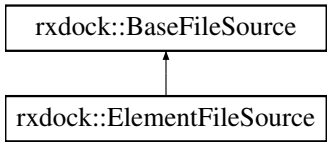
- void **to_json** (json &j, const ElementData &e)
- void **from_json** (const json &j, ElementData &e)

The documentation for this class was generated from the following file:

- ElementFileSource.h

5.75 rxdock::ElementFileSource Class Reference

Inheritance diagram for rxdock::ElementFileSource:



Public Member Functions

- **ElementFileSource** (const std::string &fileName)
- RBTDLL_EXPORT std::string **GetTitle** ()
- std::string **GetVersion** ()
- unsigned int **GetNumElements** ()
- std::vector< std::string > **GetElementNameList** ()
- std::vector< int > **GetAtomicNumberList** ()
- ElementData **GetElementData** (const std::string &strElementName)
- ElementData **GetElementData** (int nAtomicNumber)
- bool **isElementNamePresent** (const std::string &strElementName)
- bool **isAtomicNumberPresent** (int nAtomicNumber)
- double **GetHBondRadiusIncr** ()
- double **GetImplicitRadiusIncr** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Protected Attributes inherited from rxdock::BaseFileSource

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

The documentation for this class was generated from the following files:

- ElementFileSource.h
- ElementFileSource.cxx

5.76 rxdock::EnumerateSymCoords Class Reference

Public Member Functions

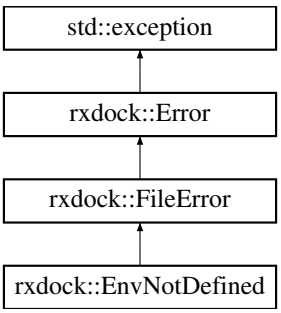
- **EnumerateSymCoords** (ModelPtr spModel)
- void **GetSymCoords** (CoordListList &cll)

The documentation for this class was generated from the following file:

- rbrms.cxx

5.77 rxdock::EnvNotDefined Class Reference

Inheritance diagram for rxdock::EnvNotDefined:



Public Member Functions

- **EnvNotDefined** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

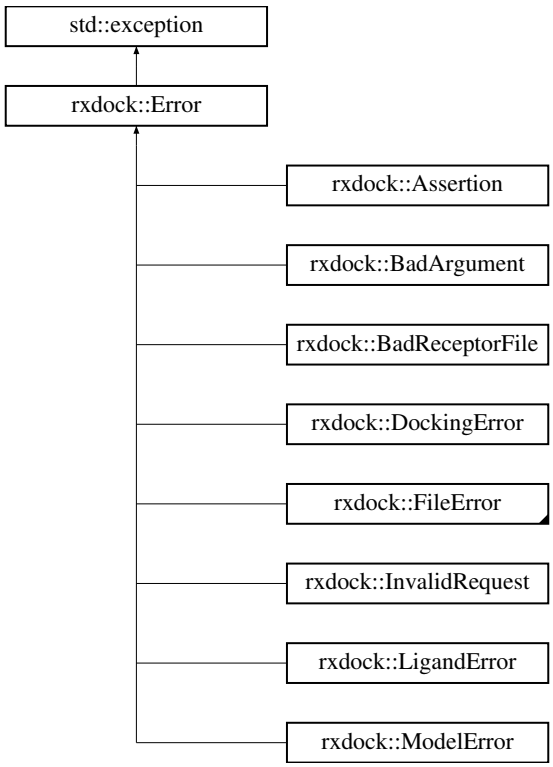
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.78 rxdock::Error Class Reference

Inheritance diagram for rxdock::Error:



Public Member Functions

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Protected Member Functions

- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- Error.h

5.79 rxdock::Euler Class Reference

Public Member Functions

- **Euler** (double heading=0.0, double attitude=0.0, double bank=0.0)
- **Euler** (const Quat &q)
- bool **isStandardised** () const
- void **Standardise** ()
- double **GetHeading** () const
- double **GetAttitude** () const
- double **GetBank** () const
- RBTDLL_EXPORT Quat **ToQuat** () const
- RBTDLL_EXPORT void **FromQuat** (const Quat &q)
- void **Rotate** (const Quat &q)
- void **Rotate** (const Vector &axis, double theta)

Friends

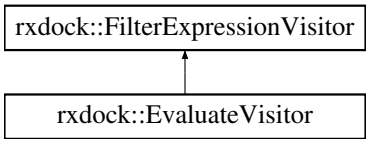
- std::ostream & **operator**<< (std::ostream &s, const Euler &euler)
- void **to_json** (json &j, const Euler &eul)
- void **from_json** (const json &j, Euler &eul)

The documentation for this class was generated from the following files:

- Euler.h
- Euler.cxx

5.80 rxdock::EvaluateVisitor Class Reference

Inheritance diagram for rxdock::EvaluateVisitor:



Public Member Functions

- **EvaluateVisitor** (ContextPtr)
- virtual void VisitVbleExp (FilterVbleExp *)
- virtual void VisitAddExp (FilterAddExp *)
- virtual void VisitSubExp (FilterSubExp *)
- virtual void VisitMulExp (FilterMulExp *)
- virtual void VisitDivExp (FilterDivExp *)
- virtual void VisitAndExp (FilterAndExp *)
- virtual void VisitLogExp (FilterLogExp *)
- virtual void VisitExpExp (FilterExpExp *)
- virtual void VisitIfExp (FilterIfExp *)

- virtual void **VisitVbleExp** (FilterVbleExp *)=0
- virtual void **VisitAddExp** (FilterAddExp *)=0
- virtual void **VisitSubExp** (FilterSubExp *)=0
- virtual void **VisitMulExp** (FilterMulExp *)=0
- virtual void **VisitDivExp** (FilterDivExp *)=0
- virtual void **VisitAndExp** (FilterAndExp *)=0
- virtual void **VisitLogExp** (FilterLogExp *)=0
- virtual void **VisitExpExp** (FilterExpExp *)=0
- virtual void **VisitIfExp** (FilterIfExp *)=0

5.80.1 Member Function Documentation

5.80.1.1 VisitAddExp()

```
void EvaluateVisitor::VisitAddExp (
            FilterAddExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.2 VisitAndExp()

```
void EvaluateVisitor::VisitAndExp (
            FilterAndExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.3 VisitDivExp()

```
void EvaluateVisitor::VisitDivExp (
            FilterDivExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.4 VisitExpExp()

```
void EvaluateVisitor::VisitExpExp (
            FilterExpExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.5 VisitIfExp()

```
void EvaluateVisitor::VisitIfExp (
            FilterIfExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.6 VisitLogExp()

```
void EvaluateVisitor::VisitLogExp (
            FilterLogExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.7 VisitMulExp()

```
void EvaluateVisitor::VisitMulExp (
            FilterMulExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.8 VisitSubExp()

```
void EvaluateVisitor::VisitSubExp (
            FilterSubExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.80.1.9 VisitVbleExp()

```
void EvaluateVisitor::VisitVbleExp (
            FilterVbleExp * fe )  [virtual]
```

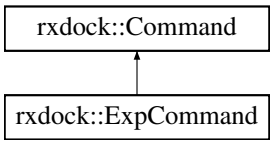
Implements rxdock::FilterExpressionVisitor.

The documentation for this class was generated from the following files:

- FilterExpressionVisitor.h
- FilterExpressionVisitor.cxx

5.81 rxdock::ExpCommand Class Reference

Inheritance diagram for rxdock::ExpCommand:



Public Member Functions

- int GetNArgs ()
- std::string GetName ()
- void SetArg (int i, ReturnType f)
- void SetNameArg (int i, std::string &n)
- ReturnType Execute ()

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Additional Inherited Members

Static Public Member Functions inherited from rxdock::Command

- static void **Clear** ()

Static Public Attributes inherited from rxdock::Command

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions inherited from rxdock::Command

- std::string **tabs** (int n)

5.81.1 Member Function Documentation

5.81.1.1 Execute()

```
ReturnType rxdock::ExpCommand::Execute ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.81.1.2 GetName()

```
std::string rxdock::ExpCommand::GetName ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.81.1.3 GetNArgs()

```
int rxdock::ExpCommand::GetNArgs ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.81.1.4 SetArg()

```
void rxdock::ExpCommand::SetArg (
            int i,
            ReturnType f )  [inline], [virtual]
```

Implements rxdock::Command.

5.81.1.5 SetNameArg()

```
void rxdock::ExpCommand::SetNameArg (
            int i,
            std::string & n )  [inline], [virtual]
```

Implements rxdock::Command.

The documentation for this class was generated from the following file:

- Command.h

5.82 rxdock::PolarSF::f1prms Struct Reference

Public Member Functions

- **f1prms** (double R, double DMin, double DMax)

Public Attributes

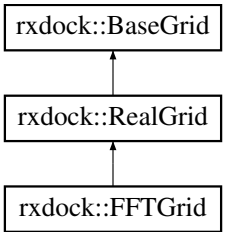
- double **R0**
- double **DRMin**
- double **DRMax**
- double **slope**

The documentation for this struct was generated from the following file:

- PolarSF.h

5.83 rxdock::FFTGrid Class Reference

Inheritance diagram for rxdock::FFTGrid:



Public Member Functions

- **FFTGrid** (const Coord &gridMin, const Coord &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **FFTGrid** (json j)
- **FFTGrid** (const FFTGrid &)
- **FFTGrid** (const RealGrid &)
- **FFTGrid** (const BaseGrid &)
- FFTGrid & **operator=** (const FFTGrid &)
- FFTGrid & **operator=** (const RealGrid &)
- FFTGrid & **operator=** (const BaseGrid &)
- virtual void Print (std::ostream &ostr) const
- FFTPeakMap **FindPeaks** (double threshold, unsigned int minVol=1)
- FFTPeak **FindMaxPeak** () const

Public Member Functions inherited from rxdock::RealGrid

- RBTDLL_EXPORT **RealGrid** (const Coord &gridMin, const Coord &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- RBTDLL_EXPORT **RealGrid** (json j)
- **RealGrid** (const RealGrid &)
- **RealGrid** (const BaseGrid &)
- RealGrid & **operator=** (const RealGrid &)
- RealGrid & **operator=** (const BaseGrid &)
- virtual void Print (std::ostream &ostr) const
- float * **GetGridData** ()
- double **GetTolerance** () const
- void **SetTolerance** (double tol)
- double **GetValue** (const Coord &c) const
- double **GetValue** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetValue** (unsigned int iXYZ) const
- RBTDLL_EXPORT double **GetSmoothedValue** (const Coord &c) const
- void **SetValue** (const Coord &c, double val)
- void **SetValue** (unsigned int iX, unsigned int iY, unsigned int iZ, double val)
- void **SetValue** (unsigned int iXYZ, double val)
- RBTDLL_EXPORT void **SetAllValues** (double val)
- void **ReplaceValue** (double oldVal, double newVal)
- void **ReplaceValueRange** (double oldValMin, double oldValMax, double newVal)
- void **SetSphere** (const Coord &c, double radius, double val, bool bOverwrite=true)
- void **SetSurface** (const Coord &c, double innerRad, double outerRad, double val, bool bOverwrite=true)

- void **CreateSurface** (double oldVal, double adjVal, double newVal)
- bool **isValueWithinSphere** (const Coord &c, double radius, double val)
- void **SetAccessible** (double radius, double oldVal, double adjVal, double newVal, bool bCenterOnly=true)
- unsigned int **Count** (double val) const
- unsigned int **CountRange** (double valMin, double valMax) const
- double **MinValue** () const
- double **MaxValue** () const
- unsigned int **FindMinValue** () const
- unsigned int **FindMaxValue** () const
- RBTDLL_EXPORT void **PrintInsightGrid** (std::ostream &s) const

Public Member Functions inherited from rxdock::BaseGrid

- **BaseGrid** (const Coord &gridMin, const Vector &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **BaseGrid** (json j)
- **BaseGrid** (const BaseGrid &)
- BaseGrid & **operator=** (const BaseGrid &)
- virtual void **Print** (std::ostream &ostr) const
- unsigned int **GetNX** () const
- unsigned int **GetNY** () const
- unsigned int **GetNZ** () const
- unsigned int **GetN** () const
- unsigned int **GetStrideX** () const
- unsigned int **GetStrideY** () const
- unsigned int **GetStrideZ** () const
- int **GetnXMin** () const
- int **GetnYMin** () const
- int **GetnZMin** () const
- int **GetnXMax** () const
- int **GetnYMax** () const
- int **GetnZMax** () const
- const Coord & **GetGridMin** () const
- const Coord & **GetGridMax** () const
- const Vector & **GetGridStep** () const
- Coord **GetGridCenter** () const
- Coord **GetGridSize** () const
- unsigned int **GetPad** () const
- const Coord & **GetPadMin** () const
- const Coord & **GetPadMax** () const
- void **SetGridMin** (const Coord &gridMin)
- void **TranslateGrid** (const Vector &vec)
- void **SetGridCenter** (const Coord &gridCenter)
- void **SetGridStep** (const Vector &gridStep)
- void **SetPad** (unsigned int NPad=0)
- bool **isValid** (const Coord &c) const
- bool **isValid** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- bool **isValid** (unsigned int iXYZ) const
- unsigned int **GetIX** (const Coord &c) const
- unsigned int **GetIY** (const Coord &c) const
- unsigned int **GetIZ** (const Coord &c) const
- unsigned int **GetIX** (double x) const
- unsigned int **GetIY** (double y) const
- unsigned int **GetIZ** (double z) const

- unsigned int **GetIX** (unsigned int iXYZ) const
- unsigned int **GetIY** (unsigned int iXYZ) const
- unsigned int **GetIZ** (unsigned int iXYZ) const
- unsigned int **GetIXYZ** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- unsigned int **GetIXYZ** (const Coord &c) const
- Coord **GetCoord** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetXCoord** (unsigned int iX) const
- double **GetYCoord** (unsigned int iY) const
- double **GetZCoord** (unsigned int iZ) const
- Coord **GetCoord** (unsigned int iXYZ) const
- CoordList **GetCoordList** (const std::set< unsigned int > &iXYZSet) const
- void **GetSphereIndices** (const Coord &c, double radius, std::vector< unsigned int > &sIndices) const

Static Public Attributes

- static const std::string **_CT** = "FFTGrid"

Static Public Attributes inherited from rxdock::RealGrid

- static const std::string **_CT** = "RealGrid"

Static Public Attributes inherited from rxdock::BaseGrid

- static const std::string **_CT** = "BaseGrid"

Protected Member Functions

- void **OwnPrint** (std::ostream &ostr) const

Protected Member Functions inherited from rxdock::RealGrid

- void **OwnPrint** (std::ostream &ostr) const

Protected Member Functions inherited from rxdock::BaseGrid

- void **OwnPrint** (std::ostream &ostr) const

5.83.1 Member Function Documentation

5.83.1.1 Print()

```
void FFTGrid::Print (
            std::ostream & ostr ) const  [virtual]
```

Reimplemented from rxdock::RealGrid.

The documentation for this class was generated from the following files:

- FFTGrid.h
- FFTGrid.cxx

5.84 rxdock::FFTPeak Class Reference

Public Attributes

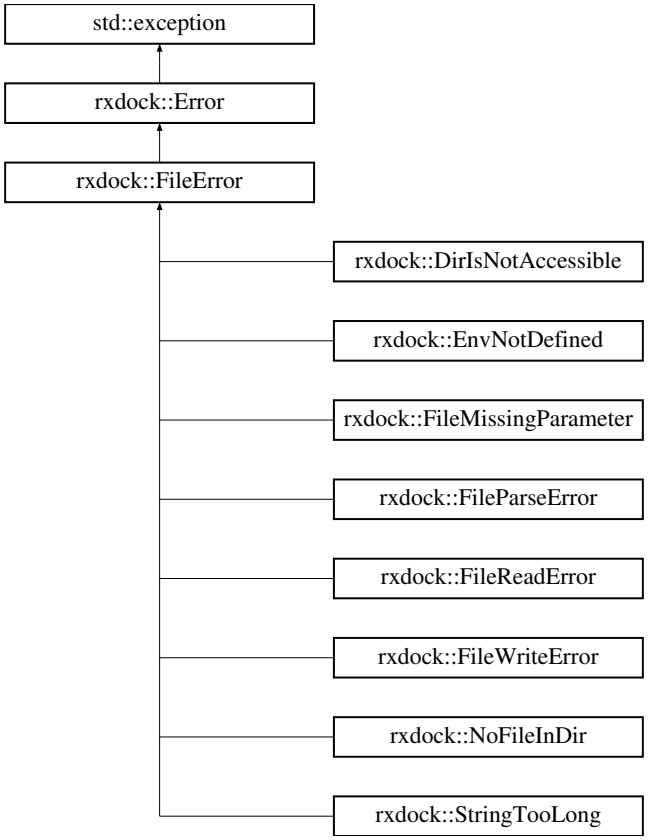
- unsigned int **index**
- Coord **coord**
- double **height**
- unsigned int **volume**
- std::set< unsigned int > **points**

The documentation for this class was generated from the following file:

- FFTGrid.h

5.85 rxdock::FileError Class Reference

Inheritance diagram for rxdock::FileError:



Public Member Functions

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Protected Member Functions

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

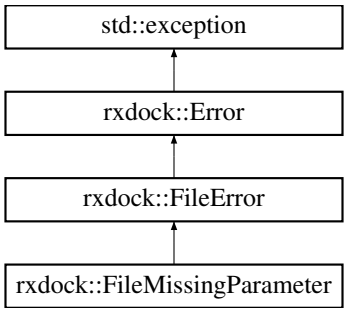
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.86 rxdock::FileMissingParameter Class Reference

Inheritance diagram for rxdock::FileMissingParameter:



Public Member Functions

- **FileMissingParameter** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

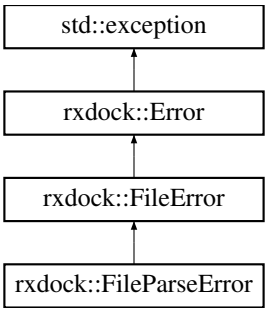
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.87 rxdock::FileParseError Class Reference

Inheritance diagram for rxdock::FileParseError:



Public Member Functions

- **FileParseError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

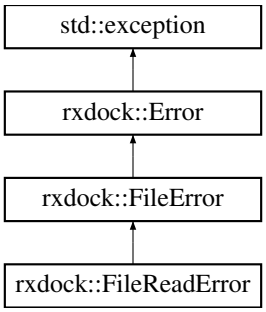
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.88 rxdock::FileReadError Class Reference

Inheritance diagram for rxdock::FileReadError:



Public Member Functions

- **FileReadError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

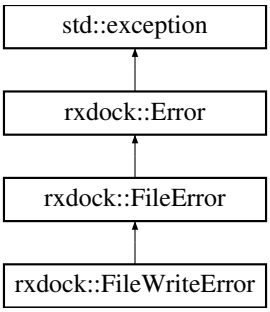
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.89 rxdock::FileWriteError Class Reference

Inheritance diagram for rxdock::FileWriteError:



Public Member Functions

- **FileWriteError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

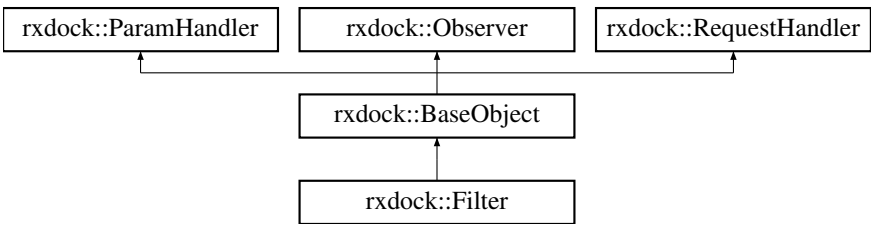
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.90 rxdock::Filter Class Reference

Inheritance diagram for rxdock::Filter:



Public Member Functions

- RBTDLL_EXPORT **Filter** (std::string strfilter, bool filter=false)
- virtual void Update (Subject *theChangedSubject)
- void **SetupReceptor** ()
- void **SetupLigand** ()
- void **SetupScore** ()
- RBTDLL_EXPORT bool **Write** ()
- RBTDLL_EXPORT bool **Terminate** ()
- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- void **SetMaxNRuns** (int n)

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨Filter¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Friends

- void **to_json** (json &j, const Filter &filter)
- void **from_json** (const json &j, Filter &filter)

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.90.1 Member Function Documentation

5.90.1.1 Update()

```
void Filter::Update (
            Subject * theChangedSubject )  [virtual]
```

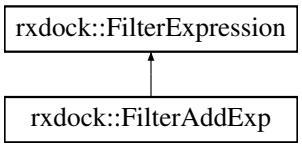
Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- Filter.h
- Filter.cxx

5.91 rxdock::FilterAddExp Class Reference

Inheritance diagram for rxdock::FilterAddExp:



Public Member Functions

- **FilterAddExp** (FilterExpressionPtr, FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = "FilterExpression"

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.91.1 Member Function Documentation

5.91.1.1 Accept()

```
void FilterAddExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.91.1.2 GetNOps()

```
virtual int rxdock::FilterAddExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.91.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterAddExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.91.1.4 Print()

```
void rxdock::FilterAddExp::Print ( )  [inline], [virtual]
```

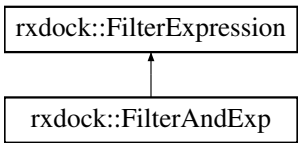
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.92 rxdock::FilterAndExp Class Reference

Inheritance diagram for rxdock::FilterAndExp:



Public Member Functions

- **FilterAndExp** (FilterExpressionPtr, FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.92.1 Member Function Documentation

5.92.1.1 Accept()

```
void FilterAndExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.92.1.2 GetNOps()

```
virtual int rxdock::FilterAndExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.92.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterAndExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.92.1.4 Print()

```
void rxdock::FilterAndExp::Print ( )  [inline], [virtual]
```

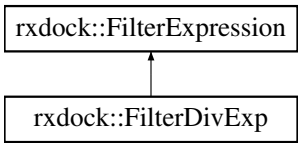
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.93 rxdock::FilterDivExp Class Reference

Inheritance diagram for rxdock::FilterDivExp:



Public Member Functions

- **FilterDivExp** (FilterExpressionPtr, FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.93.1 Member Function Documentation

5.93.1.1 Accept()

```
void FilterDivExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.93.1.2 GetNOps()

```
virtual int rxdock::FilterDivExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.93.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterDivExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.93.1.4 Print()

```
void rxdock::FilterDivExp::Print ( )  [inline], [virtual]
```

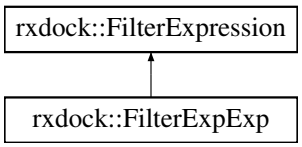
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.94 rxdock::FilterExpExp Class Reference

Inheritance diagram for rxdock::FilterExpExp:



Public Member Functions

- **FilterExpExp** (FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.94.1 Member Function Documentation

5.94.1.1 Accept()

```
void FilterExpExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.94.1.2 GetNOps()

```
virtual int rxdock::FilterExpExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.94.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterExpExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.94.1.4 Print()

```
void rxdock::FilterExpExp::Print ( )  [inline], [virtual]
```

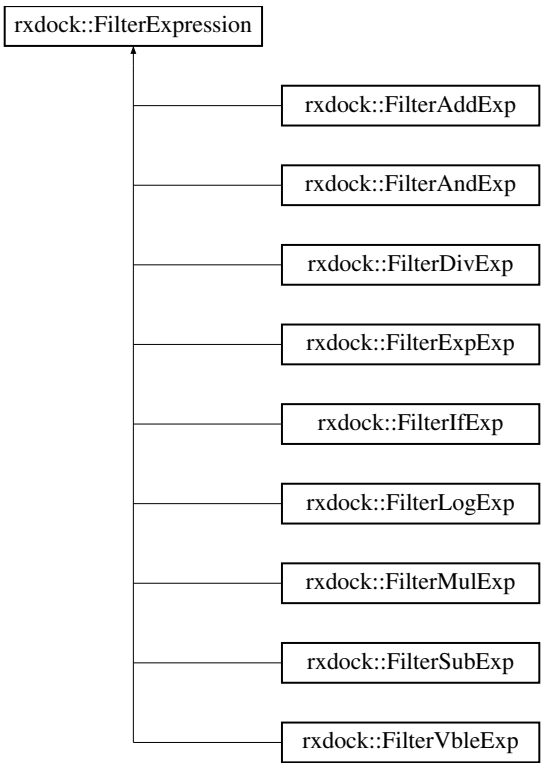
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.95 rxdock::FilterExpression Class Reference

Inheritance diagram for rxdock::FilterExpression:



Public Member Functions

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Static Public Attributes

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes

- ReturnType **value**

Friends

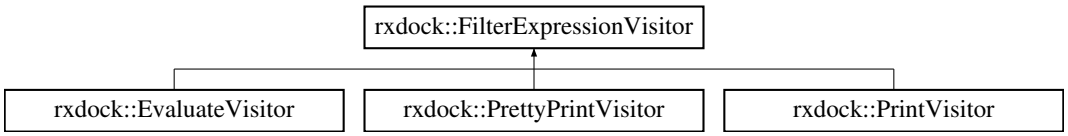
- void **to_json** (json &j, const FilterExpression &filterExpression)
- void **from_json** (const json &j, FilterExpression &filterExpression)

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.96 rxdock::FilterExpressionVisitor Class Reference

Inheritance diagram for rxdock::FilterExpressionVisitor:



Public Member Functions

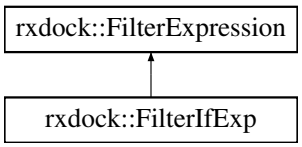
- virtual void **VisitVbleExp** (FilterVbleExp *)=0
- virtual void **VisitAddExp** (FilterAddExp *)=0
- virtual void **VisitSubExp** (FilterSubExp *)=0
- virtual void **VisitMulExp** (FilterMulExp *)=0
- virtual void **VisitDivExp** (FilterDivExp *)=0
- virtual void **VisitAndExp** (FilterAndExp *)=0
- virtual void **VisitLogExp** (FilterLogExp *)=0
- virtual void **VisitExpExp** (FilterExpExp *)=0
- virtual void **VisitIfExp** (FilterIfExp *)=0

The documentation for this class was generated from the following files:

- FilterExpressionVisitor.h
- FilterExpressionVisitor.cxx

5.97 rxdock::FilterIfExp Class Reference

Inheritance diagram for rxdock::FilterIfExp:



Public Member Functions

- **FilterIfExp** (FilterExpressionPtr, FilterExpressionPtr, FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.97.1 Member Function Documentation

5.97.1.1 Accept()

```
void FilterIfExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.97.1.2 GetNOps()

```
virtual int rxdock::FilterIfExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.97.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterIfExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.97.1.4 Print()

```
void rxdock::FilterIfExp::Print ( )  [inline], [virtual]
```

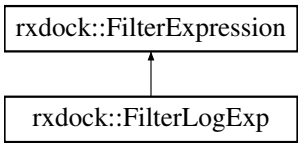
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.98 rxdock::FilterLogExp Class Reference

Inheritance diagram for rxdock::FilterLogExp:



Public Member Functions

- **FilterLogExp** (FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.98.1 Member Function Documentation

5.98.1.1 Accept()

```
void FilterLogExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.98.1.2 GetNOps()

```
virtual int rxdock::FilterLogExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.98.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterLogExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.98.1.4 Print()

```
void rxdock::FilterLogExp::Print ( )  [inline], [virtual]
```

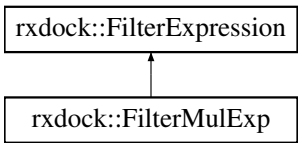
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.99 rxdock::FilterMulExp Class Reference

Inheritance diagram for rxdock::FilterMulExp:



Public Member Functions

- **FilterMulExp** (FilterExpressionPtr, FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.99.1 Member Function Documentation

5.99.1.1 Accept()

```
void FilterMulExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.99.1.2 GetNOps()

```
virtual int rxdock::FilterMulExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.99.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterMulExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.99.1.4 Print()

```
void rxdock::FilterMulExp::Print ( )  [inline], [virtual]
```

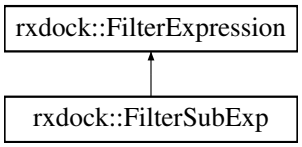
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.100 rxdock::FilterSubExp Class Reference

Inheritance diagram for rxdock::FilterSubExp:



Public Member Functions

- **FilterSubExp** (FilterExpressionPtr, FilterExpressionPtr)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- virtual int GetNOps ()
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.100.1 Member Function Documentation

5.100.1.1 Accept()

```
void FilterSubExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.100.1.2 GetNOps()

```
virtual int rxdock::FilterSubExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.100.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterSubExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.100.1.4 Print()

```
void rxdock::FilterSubExp::Print ( )  [inline], [virtual]
```

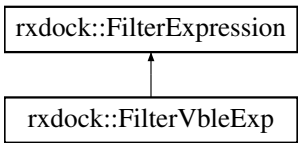
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.101 rxdock::FilterVbleExp Class Reference

Inheritance diagram for rxdock::FilterVbleExp:



Public Member Functions

- **FilterVbleExp** (const Vble &)
- void Accept (FilterExpressionVisitor &visitor)
- void Print ()
- int GetNOps ()
- const Vble & **GetVble** () const
- FilterExpressionPtr GetOp (int i)

Public Member Functions inherited from rxdock::FilterExpression

- virtual void **Print** ()=0
- virtual void **Accept** (FilterExpressionVisitor &)=0
- virtual int **GetNOps** ()=0
- virtual SmartPtr< FilterExpression > **GetOp** (int)=0
- ReturnType **GetValue** ()
- void **SetValue** (ReturnType v)

Additional Inherited Members

Static Public Attributes inherited from rxdock::FilterExpression

- static const std::string **_CT** = ¨FilterExpression¨

Protected Attributes inherited from rxdock::FilterExpression

- ReturnType **value**

5.101.1 Member Function Documentation

5.101.1.1 Accept()

```
void FilterVbleExp::Accept (
            FilterExpressionVisitor & visitor )  [virtual]
```

Implements rxdock::FilterExpression.

5.101.1.2 GetNOps()

```
int rxdock::FilterVbleExp::GetNOps ( )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.101.1.3 GetOp()

```
FilterExpressionPtr rxdock::FilterVbleExp::GetOp (
            int i )  [inline], [virtual]
```

Implements rxdock::FilterExpression.

5.101.1.4 Print()

```
void rxdock::FilterVbleExp::Print ( )  [inline], [virtual]
```

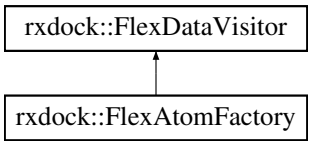
Implements rxdock::FilterExpression.

The documentation for this class was generated from the following files:

- FilterExpression.h
- FilterExpression.cxx

5.102 rxdock::FlexAtomFactory Class Reference

Inheritance diagram for rxdock::FlexAtomFactory:



Public Member Functions

- **FlexAtomFactory** (Model ∗)
- **FlexAtomFactory** (ModelList)
- void **Visit** (Model ∗)
- void **Clear** ()
- virtual void VisitReceptorFlexData (ReceptorFlexData ∗)
- virtual void VisitLigandFlexData (LigandFlexData ∗)
- virtual void VisitSolventFlexData (SolventFlexData ∗)
- AtomRList **GetFixedAtomList** () const
- AtomRList **GetTetheredAtomList** () const
- AtomRList **GetFreeAtomList** () const

- virtual void **VisitReceptorFlexData** (ReceptorFlexData ∗)=0
- virtual void **VisitLigandFlexData** (LigandFlexData ∗)=0
- virtual void **VisitSolventFlexData** (SolventFlexData ∗)=0

Friends

- void **to_json** (json &j, const FlexAtomFactory &flexAtomFactory)
- void **from_json** (const json &j, FlexAtomFactory &flexAtomFactory)

5.102.1 Member Function Documentation

5.102.1.1 VisitLigandFlexData()

```
void FlexAtomFactory::VisitLigandFlexData (
            LigandFlexData * pFlexData )  [virtual]
```

Implements rxdock::FlexDataVisitor.

5.102.1.2 VisitReceptorFlexData()

```
void FlexAtomFactory::VisitReceptorFlexData (
            ReceptorFlexData * pFlexData )  [virtual]
```

Implements rxdock::FlexDataVisitor.

5.102.1.3 VisitSolventFlexData()

```
void FlexAtomFactory::VisitSolventFlexData (
            SolventFlexData * pFlexData )  [virtual]
```

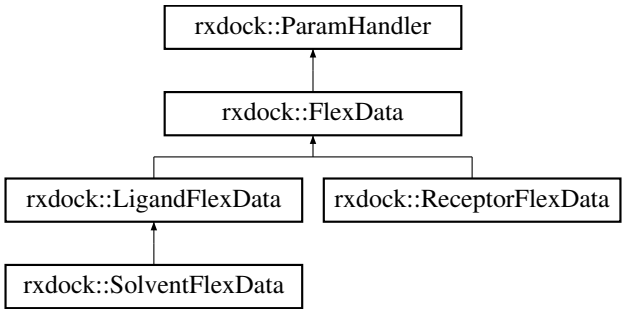
Implements rxdock::FlexDataVisitor.

The documentation for this class was generated from the following files:

- FlexAtomFactory.h
- FlexAtomFactory.cxx

5.103 rxdock::FlexData Class Reference

Inheritance diagram for rxdock::FlexData:



Public Member Functions

- Model * **GetModel** () const
- void **SetModel** (Model *pModel)
- DockingSite * **GetDockingSite** () const
- virtual void **Accept** (FlexDataVisitor &)=0

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Protected Member Functions

- **FlexData** (DockingSite *pDockSite)

Protected Member Functions inherited from rxdock::ParamHandler

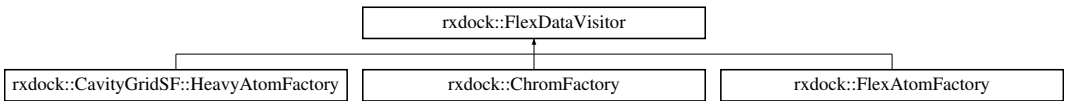
- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

The documentation for this class was generated from the following file:

- FlexData.h

5.104 rxdock::FlexDataVisitor Class Reference

Inheritance diagram for rxdock::FlexDataVisitor:



Public Member Functions

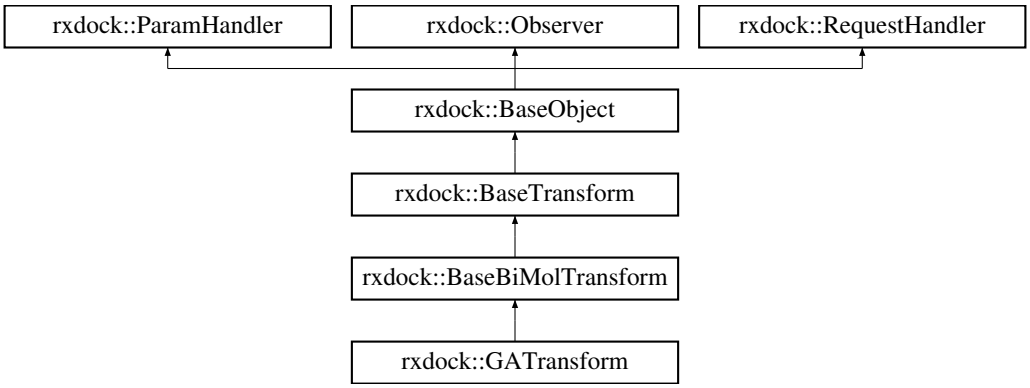
- virtual void **VisitReceptorFlexData** (ReceptorFlexData *)=0
- virtual void **VisitLigandFlexData** (LigandFlexData *)=0
- virtual void **VisitSolventFlexData** (SolventFlexData *)=0

The documentation for this class was generated from the following file:

- FlexDataVisitor.h

5.105 rxdock::GATransform Class Reference

Inheritance diagram for rxdock::GATransform:



Public Member Functions

- RBTDLL_EXPORT **GATransform** (const std::string &strName="GAGENRW")

Public Member Functions inherited from rxdock::BaseBiMolTransform

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform *)
- virtual void **Remove** (BaseTransform *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform * **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform * **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨GATransform¨
- static const std::string **_NEW_FRACTION** = ¨fraction-of-new-individuals¨
- static const std::string **_PCROSSOVER** = ¨crossover-probability¨
- static const std::string **_XOVERMUT** = ¨crossover-mutation¨
- static const std::string **_CMUTATE** = ¨cauchy-mutation¨
- static const std::string **_STEP_SIZE** = ¨step-size¨
- static const std::string **_EQUALITY_THRESHOLD** = ¨equality-threshold¨
- static const std::string **_NCYCLES** = ¨number-of-cycles¨
- static const std::string **_NCONVERGENCE** = ¨number-for-convergence¨
- static const std::string **_HISTORY_FREQ** = ¨history-frequency¨

Static Public Attributes inherited from rxdock::BaseBiMolTransform

- static const std::string **_CT** = ¨BaseBiMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupTransform ()
- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseBiMolTransform

- **BaseBiMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupReceptor** ()=0
- virtual void **SetupLigand** ()=0
- virtual void **SetupSolvent** ()
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.105.1 Member Function Documentation

5.105.1.1 Execute()

```
void GATransform::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.105.1.2 SetupLigand()

```
void GATransform::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.105.1.3 SetupReceptor()

```
void GATransform::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.105.1.4 SetupTransform()

```
void GATransform::SetupTransform ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

The documentation for this class was generated from the following files:

- GATransform.h
- GATransform.cxx

5.106 rxdock::Genome Class Reference

Public Member Functions

- **Genome** (ChromElement *pChr)
- **Genome** (const Genome &)
- Genome & **operator=** (const Genome &)
- Genome * **clone** () const
- ChromElement * **GetChrom** () const
- void **SetScore** (BaseSF *pSF)
- double **GetScore** () const
- double **SetRWFitness** (double sigmaOffset, double partialSum)
- void **NormaliseRWFitness** (double total)
- double **GetRWFitness** () const
- bool **Equals** (const Genome &g, double threshold) const
- void **Print** (std::ostream &) const

Static Public Attributes

- static const std::string **_CT** = ¨Genome¨

Friends

- void **to_json** (json &j, const Genome &genome)
- void **from_json** (const json &j, Genome &genome)
- bool **operator==** (const Genome &g1, const Genome &g2)
- std::ostream & **operator<<** (std::ostream &s, const Genome &g)

The documentation for this class was generated from the following files:

- Genome.h
- Genome.cxx

5.107 rxdock::GenomeCmp_Score Class Reference

Public Member Functions

- bool **operator()** (const Genome *pG1, const Genome *pG2) const

The documentation for this class was generated from the following file:

- Genome.h

5.108 rxdock::geneticprogram::GPChromosome Class Reference

Public Member Functions

- **GPChromosome** (int, int, int, int, int, int)
- **GPChromosome** (const GPChromosome &)
- void **Clear** ()
- std::ostream & **Print** (std::ostream &) const
- const int & **operator[ ]** (int idx) const
- int & **operator[ ]** (int idx)
- GPChromosome & **operator=** (const GPChromosome &c)
- int **size** () const
- CellPtr **Cells** (int idx) const
- void **SetConstant** (ReturnType, int)
- void **ResetConstant** (int)
- int **GetStartingCell** ()
- ReturnType **GetRCte** (int ncell)
- Commands **GetCommand** (int ncell)
- int **GetArgument** (int ncell, int narg)
- bool **IsProgramInput** (int ncell)

Static Public Attributes

- static const std::string **_CT** = ¨GPChromosome¨

Friends

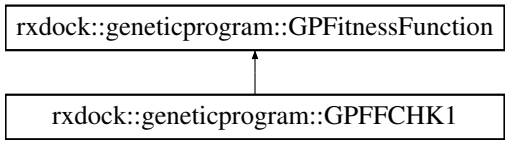
- std::ostream & **operator**<< (std::ostream &s, const GPChromosome &p)
- std::istream & **operator**>> (std::istream &s, GPChromosome &p)

The documentation for this class was generated from the following files:

- GPChromosome.h
- GPChromosome.cxx

5.109 rxdock::geneticprogram::GPFFCHK1 Class Reference

Inheritance diagram for rxdock::geneticprogram::GPFFCHK1:



Public Member Functions

- **GPFFCHK1** (ContextPtr c)
- void ReadTables (std::istream &, ReturnTypeArray &, ReturnTypeArray &)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)

Public Member Functions inherited from rxdock::geneticprogram::GPFitnessFunction

- **GPFitnessFunction** (const GPFitnessFunction &)
- GPFitnessFunction & **operator=** (const GPFitnessFunction &)
- virtual void **ReadTables** (std::istream &, ReturnTypeArray &, ReturnTypeArray &)=0
- double **GetObjective** () const
- double **GetFitness** () const
- void **SetFitness** (double)
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)=0
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)=0

Static Public Attributes

- static const std::string **_CT** = ¨GPFFCHK1¨

Static Public Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- static const std::string **_CT** = ¨GPFitnessFunction¨

Additional Inherited Members

Protected Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- Rand & **m_rand**
- double **objective**
- double **fitness**
- ReturnTypeArray **inputTable**
- ReturnTypeArray **SFTable**

5.109.1 Member Function Documentation

5.109.1.1 CalculateFitness() [1/2]

```
double GPFFCHK1::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.109.1.2 CalculateFitness() [2/2]

```
double GPFFCHK1::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            double hitlimit,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.109.1.3 ReadTables()

```
void GPFFCHK1::ReadTables (
            std::istream & in,
            ReturnTypeArray & it,
            ReturnTypeArray & sft )  [virtual]
```

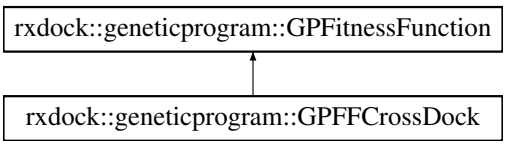
Implements rxdock::geneticprogram::GPFitnessFunction.

The documentation for this class was generated from the following files:

- GPFFCHK1.h
- GPFFCHK1.cxx

5.110 rxdock::geneticprogram::GPFFCrossDock Class Reference

Inheritance diagram for rxdock::geneticprogram::GPFFCrossDock:



Public Member Functions

- void ReadTables (std::istream &, ReturnTypeArray &, ReturnTypeArray &)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)

Public Member Functions inherited from rxdock::geneticprogram::GPFitnessFunction

- **GPFitnessFunction** (const GPFitnessFunction &)
- GPFitnessFunction & **operator=** (const GPFitnessFunction &)
- virtual void **ReadTables** (std::istream &, ReturnTypeArray &, ReturnTypeArray &)=0
- double **GetObjective** () const
- double **GetFitness** () const
- void **SetFitness** (double)
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)=0
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)=0

Static Public Attributes

- static const std::string **_CT** = ¨GPFFCrossDock¨

Static Public Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- static const std::string **_CT** = ¨GPFitnessFunction¨

Additional Inherited Members

Protected Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- Rand & **m_rand**
- double **objective**
- double **fitness**
- ReturnTypeArray **inputTable**
- ReturnTypeArray **SFTable**

5.110.1 Member Function Documentation

5.110.1.1 CalculateFitness() [1/2]

```
double GPFFCrossDock::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.110.1.2 CalculateFitness() [2/2]

```
double GPFFCrossDock::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            double hitlimit,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.110.1.3 ReadTables()

```
void GPFFCrossDock::ReadTables (
            std::istream & in,
            ReturnTypeArray & it,
            ReturnTypeArray & sft )  [virtual]
```

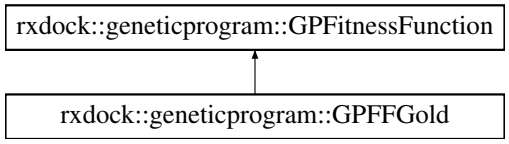
Implements rxdock::geneticprogram::GPFitnessFunction.

The documentation for this class was generated from the following files:

- GPFFCrossDock.h
- GPFFCrossDock.cxx

5.111 rxdock::geneticprogram::GPFFGold Class Reference

Inheritance diagram for rxdock::geneticprogram::GPFFGold:



Public Member Functions

- void ReadTables (std::istream &, ReturnTypeArray &, ReturnTypeArray &)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)

Public Member Functions inherited from rxdock::geneticprogram::GPFitnessFunction

- **GPFitnessFunction** (const GPFitnessFunction &)
- GPFitnessFunction & **operator=** (const GPFitnessFunction &)
- virtual void **ReadTables** (std::istream &, ReturnTypeArray &, ReturnTypeArray &)=0
- double **GetObjective** () const
- double **GetFitness** () const
- void **SetFitness** (double)
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)=0
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)=0

Static Public Attributes

- static const std::string **_CT** = ¨GPFFGold¨

Static Public Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- static const std::string **_CT** = ¨GPFitnessFunction¨

Additional Inherited Members

Protected Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- Rand & **m_rand**
- double **objective**
- double **fitness**
- ReturnTypeArray **inputTable**
- ReturnTypeArray **SFTable**

5.111.1 Member Function Documentation

5.111.1.1 CalculateFitness() [1/2]

```
double GPFFGold::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.111.1.2 CalculateFitness() [2/2]

```
double GPFFGold::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            double hitlimit,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.111.1.3 ReadTables()

```
void GPFFGold::ReadTables (
            std::istream & in,
            ReturnTypeArray & ,
            ReturnTypeArray &  )  [virtual]
```

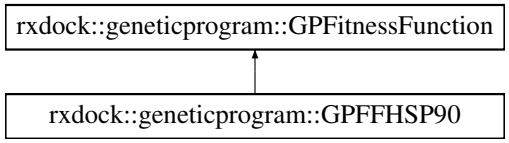
Implements rxdock::geneticprogram::GPFitnessFunction.

The documentation for this class was generated from the following files:

- GPFFGold.h
- GPFFGold.cxx

5.112 rxdock::geneticprogram::GPFFHSP90 Class Reference

Inheritance diagram for rxdock::geneticprogram::GPFFHSP90:



Public Member Functions

- **GPFFHSP90** (ContextPtr c)
- void ReadTables (std::istream &, ReturnTypeArray &, ReturnTypeArray &)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)

Public Member Functions inherited from rxdock::geneticprogram::GPFitnessFunction

- **GPFitnessFunction** (const GPFitnessFunction &)
- GPFitnessFunction & **operator=** (const GPFitnessFunction &)
- virtual void **ReadTables** (std::istream &, ReturnTypeArray &, ReturnTypeArray &)=0
- double **GetObjective** () const
- double **GetFitness** () const
- void **SetFitness** (double)
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)=0
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)=0

Static Public Attributes

- static const std::string **_CT** = ¨GPFFHSP90¨

Static Public Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- static const std::string **_CT** = ¨GPFitnessFunction¨

Additional Inherited Members

Protected Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- Rand & **m_rand**
- double **objective**
- double **fitness**
- ReturnTypeArray **inputTable**
- ReturnTypeArray **SFTable**

5.112.1 Member Function Documentation

5.112.1.1 CalculateFitness() [1/2]

```
double GPFFHSP90::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.112.1.2 CalculateFitness() [2/2]

```
double GPFFHSP90::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            double limit,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.112.1.3 ReadTables()

```
void GPFFHSP90::ReadTables (
            std::istream & in,
            ReturnTypeArray & it,
            ReturnTypeArray & sft )  [virtual]
```

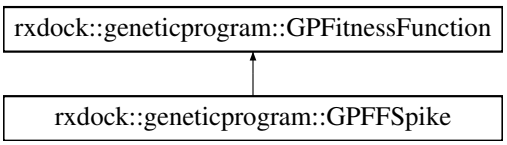
Implements rxdock::geneticprogram::GPFitnessFunction.

The documentation for this class was generated from the following files:

- GPFFHSP90.h
- GPFFHSP90.cxx

5.113 rxdock::geneticprogram::GPFFSpike Class Reference

Inheritance diagram for rxdock::geneticprogram::GPFFSpike:



Public Member Functions

- **GPFFSpike** (ContextPtr c)
- void ReadTables (std::istream &, ReturnTypeArray &, ReturnTypeArray &)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)
- double CalculateFitness (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)

Public Member Functions inherited from rxdock::geneticprogram::GPFitnessFunction

- **GPFitnessFunction** (const GPFitnessFunction &)
- GPFitnessFunction & **operator=** (const GPFitnessFunction &)
- virtual void **ReadTables** (std::istream &, ReturnTypeArray &, ReturnTypeArray &)=0
- double **GetObjective** () const
- double **GetFitness** () const
- void **SetFitness** (double)
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)=0
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)=0

Static Public Attributes

- static const std::string **_CT** = ¨GPFFSpike¨

Static Public Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- static const std::string **_CT** = ¨GPFitnessFunction¨

Additional Inherited Members

Protected Attributes inherited from rxdock::geneticprogram::GPFitnessFunction

- Rand & **m_rand**
- double **objective**
- double **fitness**
- ReturnTypeArray **inputTable**
- ReturnTypeArray **SFTable**

5.113.1 Member Function Documentation

5.113.1.1 CalculateFitness() [1/2]

```
double GPFFSpike::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.113.1.2 CalculateFitness() [2/2]

```
double GPFFSpike::CalculateFitness (
            GPGenomePtr g,
            ReturnTypeArray & it,
            ReturnTypeArray & sft,
            double hitlimit,
            bool function )  [virtual]
```

Implements rxdock::geneticprogram::GPFitnessFunction.

5.113.1.3 ReadTables()

```
void GPFFSpike::ReadTables (
            std::istream & in,
            ReturnTypeArray & it,
            ReturnTypeArray & sft )  [virtual]
```

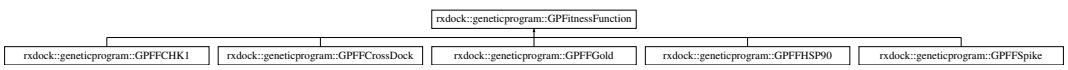
Implements rxdock::geneticprogram::GPFitnessFunction.

The documentation for this class was generated from the following files:

- GPFFSpike.h
- GPFFSpike.cxx

5.114 rxdock::geneticprogram::GPFitnessFunction Class Reference

Inheritance diagram for rxdock::geneticprogram::GPFitnessFunction:



Public Member Functions

- **GPFitnessFunction** (const GPFitnessFunction &)
- GPFitnessFunction & **operator=** (const GPFitnessFunction &)
- virtual void **ReadTables** (std::istream &, ReturnTypeArray &, ReturnTypeArray &)=0
- double **GetObjective** () const
- double **GetFitness** () const
- void **SetFitness** (double)
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, double, bool)=0
- virtual double **CalculateFitness** (GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &, bool)=0

Static Public Attributes

- static const std::string **_CT** = ¨GPFitnessFunction¨

Protected Attributes

- Rand & **m_rand**
- double **objective**
- double **fitness**
- ReturnTypeArray **inputTable**
- ReturnTypeArray **SFTable**

The documentation for this class was generated from the following files:

- GPFitnessFunction.h
- GPFitnessFunction.cxx

5.115 rxdock::geneticprogram::GPGenome Class Reference

Public Member Functions

- **GPGenome** (const GPGenome &)
- **GPGenome** (std::string)
- **GPGenome** (std::istream &)
- GPGenome & **operator=** (const GPGenome &)
- GPChromosomePtr **GetChrom** () const
- void **Initialise** ()
- void **Mutate** (double)
- void **SetFitness** (double f)
- double **GetFitness** () const
- void **UniformCrossover** (const GPGenome &, const GPGenome &)
- void **Crossover** (GPGenome &)
- std::ostream & **Print** (std::ostream &) const

Static Public Member Functions

- static void **SetStructure** (int, int, int, int, int, int, int, int)
- static int **GetNIP** ()
- static void **SetNIP** (int n)
- static int **GetNIF** ()
- static int **GetNN** ()
- static int **GetNO** ()
- static int **GetNSFI** ()
- static void **SetNSFI** (int n)

Static Public Attributes

- static const std::string **_CT** = ¨GPGenome¨

Friends

- std::ostream & **operator**<< (std::ostream &s, const GPGenome &p)

The documentation for this class was generated from the following files:

- GPGenome.h
- GPGenome.cxx

5.116 rxdock::geneticprogram::GPParser Class Reference

Public Member Functions

- **GPParser** (int, int, int, int)
- **GPParser** (const GPParser &)
- void **Clear** ()
- std::ostream & **Print** (std::ostream &) const
- ReturnTypeList **Parse** (GPChromosomePtr, ReturnTypeList &)
- std::string **PrintParse** (std::istream &, GPChromosomePtr, bool, bool)
- bool **IsConstantInt** (int i)
- bool **IsConstantFloat** (int i)

Static Public Attributes

- static const std::string **_CT** = ¨GPParser¨

Friends

- std::ostream & **operator**<< (std::ostream &s, const GPParser &p)

The documentation for this class was generated from the following files:

- GPParser.h
- GPParser.cxx

5.117 rxdock::geneticprogram::GPPopulation Class Reference

Public Member Functions

- **GPPopulation** (int, int, GPFitnessFunctionPtr, ReturnTypeArray &, ReturnTypeArray &)
- void **SetTrainingSets** (ReturnTypeArray &it, ReturnTypeArray sf)
- **GPPopulation** (const GPPopulation &)
- void **Initialise** (double, bool)
- void **ScaleFitness** ()
- void **Eval** (double, bool)
- GPGenomePtr **Select** (std::string) const
- void **SelectionUpdate** (std::string)
- void **GAstep** (std::string, double, double, double, double, double, bool)
- void **EPstep** (std::string, double, double, double, double, double, bool)
- GPGenomePtr **Best** () const
- std::ostream & **Print** (std::ostream &) const
- void **Swap** (SmartPtr< GPPopulation >, int)
- void **QSort** (GPGenomeList &)
- void **QSort** ()
- void **MergePops** ()
- int **GetSize** ()
- int **GetNrepl** ()

Static Public Member Functions

- static bool **Gen_eq** (GPGenome *, GPGenome *)

Static Public Attributes

- static const std::string **_CT** = ¨GPPopulation¨

Friends

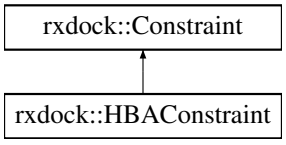
- std::ostream & **operator**<< (std::ostream &, const GPPopulation &)

The documentation for this class was generated from the following files:

- GPPopulation.h
- GPPopulation.cxx

5.118 rxdock::HBAConstraint Class Reference

Inheritance diagram for rxdock::HBAConstraint:



Public Member Functions

- **HBAConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.118.1 Member Function Documentation

5.118.1.1 AddAtomList()

```
void HBAConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

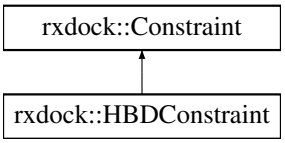
Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.119 rxdock::HBDConstraint Class Reference

Inheritance diagram for rxdock::HBDConstraint:



Public Member Functions

- **HBDConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.119.1 Member Function Documentation

5.119.1.1 AddAtomList()

```
void HBDConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

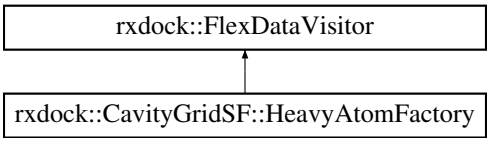
Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.120 rxdock::CavityGridSF::HeavyAtomFactory Class Reference

Inheritance diagram for rxdock::CavityGridSF::HeavyAtomFactory:



Public Member Functions

- **HeavyAtomFactory** (ModelList modelList)
- virtual void VisitReceptorFlexData (ReceptorFlexData *)
- virtual void VisitLigandFlexData (LigandFlexData *)
- virtual void VisitSolventFlexData (SolventFlexData *)
- AtomRList **GetAtomList** () const

- virtual void **VisitReceptorFlexData** (ReceptorFlexData *)=0
- virtual void **VisitLigandFlexData** (LigandFlexData *)=0
- virtual void **VisitSolventFlexData** (SolventFlexData *)=0

Friends

- void **to_json** (json &j, const HeavyAtomFactory &heavyAtomFactory)
- void **from_json** (const json &j, HeavyAtomFactory &heavyAtomFactory)

5.120.1 Member Function Documentation

5.120.1.1 VisitLigandFlexData()

```
void CavityGridSF::HeavyAtomFactory::VisitLigandFlexData (
            LigandFlexData * pFlexData )  [virtual]
```

Implements rxdock::FlexDataVisitor.

5.120.1.2 VisitReceptorFlexData()

```
void CavityGridSF::HeavyAtomFactory::VisitReceptorFlexData (
            ReceptorFlexData * pFlexData )  [virtual]
```

Implements rxdock::FlexDataVisitor.

5.120.1.3 VisitSolventFlexData()

```
void CavityGridSF::HeavyAtomFactory::VisitSolventFlexData (
            SolventFlexData * pFlexData )  [virtual]
```

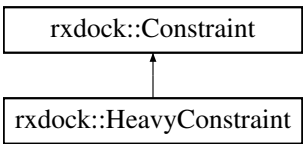
Implements rxdock::FlexDataVisitor.

The documentation for this class was generated from the following files:

- CavityGridSF.h
- CavityGridSF.cxx

5.121 rxdock::HeavyConstraint Class Reference

Inheritance diagram for rxdock::HeavyConstraint:



Public Member Functions

- **HeavyConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int counter

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.121.1 Member Function Documentation

5.121.1.1 AddAtomList()

```
void HeavyConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

Implements rxdock::Constraint.

5.121.2 Member Data Documentation

5.121.2.1 counter

```
unsigned int HeavyConstraint::counter  [static]
```

Initial value:

```
=
    0
```

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.122 rxdock::HHS_Solvation Class Reference

Public Member Functions

- **HHS_Solvation** (HHSType::eType t, Atom *a, double p=1.0, double r=1.0, double s=0.0)
- double **GetA_i** (void) const
- double **GetS_i** (void) const
- double **GetP_i** (void) const
- double **GetR_i** (void) const
- double **GetSigma** (void) const
- HHSType::eType **GetHHSType** () const
- Atom * **GetAtom** () const
- double **GetArea** () const
- double **GetEnergy** () const
- void **Init** ()
- void **Save** ()
- void **Restore** ()
- void **Overlap** (HHS_Solvation *h, double p_ij)
- const std::vector< HHS_Solvation * > & **GetVariable** () const
- int **GetNumVariable** () const
- void **AddVariable** (HHS_Solvation *anAtom)
- void **OverlapVariable** ()
- void **OverlapVariableEnabledOnly** ()
- void **Partition** (double d=0.0)

Static Public Attributes

- static const double **r_s** = 0.6
- static const double d_s
- static const double Pij_12
- static const double **Pij_13** = 0.3516
- static const double **Pij_14** = 0.3156

5.122.1 Member Data Documentation

5.122.1.1 d_s

```
const double HHS_Solvation::d_s  [static]
```

Initial value:

```
=
    2.0 * HHS_Solvation::r_s
```

5.122.1.2 Pij_12

```
const double HHS_Solvation::Pij_12  [static]
```

Initial value:

```
=
    0.8875
```

The documentation for this class was generated from the following files:

- SATypes.h
- SATypes.cxx

5.123 rxdock::HHSType Class Reference

Public Types

- enum **eType** {
  UNDEFINED = 0 , **C_sp3** , **CH_sp3** , **CH2_sp3** ,
  CH3_sp3 , **C_sp2** , **CH_sp2** , **CH2_sp2** ,
  C_sp2p , **C_ar** , **CH_ar** , **C_sp** ,
  C_sp3_P , **CH_sp3_P** , **CH2_sp3_P** , **CH3_sp3_P** ,
  C_sp2_P , **CH_sp2_P** , **CH2_sp2_P** , **C_ar_P** ,
  CH_ar_P , **H** , **HO** , **HN** ,
  HNp , **HS** , **O_sp3** , **OH_sp3** ,
  OW , **O_tri** , **OH_tri** , **O_sp2** ,
  ON , **Om** , **N_sp3** , **NH_sp3** ,
  NH2_sp3 , **N_sp3p** , **N_tri** , **NH_tri** ,
  NH2_tri , **N_sp2** , **N_sp2p** , **N_ar** ,
  N_sp , **S_sp3** , **S_sp2** , **P** ,
  F , **Cl** , **Br** , **I** ,
  Metal , **MAXTYPES** }

Public Member Functions

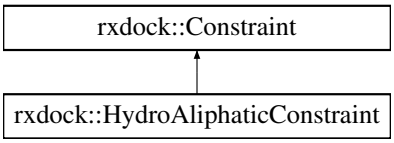
- eType **operator()** (Atom ∗anAtom) const
- std::string **Type2Str** (eType) const
- eType **Str2Type** (const std::string &) const

The documentation for this class was generated from the following files:

- SATypes.h
- SATypes.cxx

5.124 rxdock::HydroAliphaticConstraint Class Reference

Inheritance diagram for rxdock::HydroAliphaticConstraint:



Public Member Functions

- **HydroAliphaticConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.124.1 Member Function Documentation

5.124.1.1 AddAtomList()

```
void HydroAliphaticConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

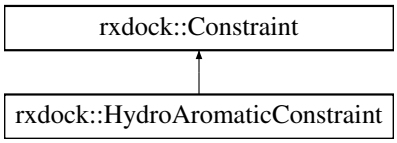
Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.125 rxdock::HydroAromaticConstraint Class Reference

Inheritance diagram for rxdock::HydroAromaticConstraint:



Public Member Functions

- **HydroAromaticConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.125.1 Member Function Documentation

5.125.1.1 AddAtomList()

```
void HydroAromaticConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

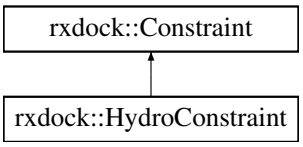
Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.126 rxdock::HydroConstraint Class Reference

Inheritance diagram for rxdock::HydroConstraint:



Public Member Functions

- **HydroConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.126.1 Member Function Documentation

5.126.1.1 AddAtomList()

```
void HydroConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

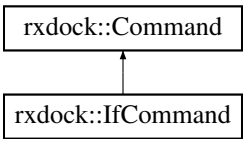
Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.127 rxdock::IfCommand Class Reference

Inheritance diagram for rxdock::IfCommand:



Public Member Functions

- int GetNArgs ()
- std::string GetName ()
- void SetArg (int i, ReturnType f)
- void SetNameArg (int i, std::string &n)
- ReturnType Execute ()

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Additional Inherited Members

Static Public Member Functions inherited from rxdock::Command

- static void **Clear** ()

Static Public Attributes inherited from rxdock::Command

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions inherited from rxdock::Command

- std::string **tabs** (int n)

5.127.1 Member Function Documentation

5.127.1.1 Execute()

```
ReturnType rxdock::IfCommand::Execute ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.127.1.2 GetName()

```
std::string rxdock::IfCommand::GetName ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.127.1.3 GetNArgs()

```
int rxdock::IfCommand::GetNArgs ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.127.1.4 SetArg()

```
void rxdock::IfCommand::SetArg (
            int i,
            ReturnType f )  [inline], [virtual]
```

Implements rxdock::Command.

5.127.1.5 SetNameArg()

```
void rxdock::IfCommand::SetNameArg (
            int i,
            std::string & n )  [inline], [virtual]
```

Implements rxdock::Command.

The documentation for this class was generated from the following file:

- Command.h

5.128 rxdock::TriposAtomType::info Struct Reference

Public Member Functions

- **info** (const std::string &n, int a, int h)

Public Attributes

- std::string **name**
- int **atomicNo**
- int **hybrid**

The documentation for this struct was generated from the following file:

- TriposAtomType.h

5.129 rxdock::InitHHS Class Reference

Public Member Functions

- void **operator()** (HHS_Solvation *pHHS)

The documentation for this class was generated from the following file:

- SATypes.h

5.130 rxdock::InteractionCenter Class Reference

Public Types

- enum **eLP** { **NONE** , **PLANE** , **LONEPAIR** }

Public Member Functions

- **InteractionCenter** (Atom *pAtom1=nullptr, Atom *pAtom2=nullptr, Atom *pAtom3=nullptr, eLP LP=NONE)
- Atom * **GetAtom1Ptr** () const
- Atom * **GetAtom2Ptr** () const
- Atom * **GetAtom3Ptr** () const
- eLP **LP** () const
- AtomRList **GetAtomList** () const
- bool **isSelected** () const

Friends

- void **to_json** (json &j, const InteractionCenter &interactionCenter)
- void **from_json** (const json &j, InteractionCenter &interactionCenter)

The documentation for this class was generated from the following files:

- InteractionGrid.h
- InteractionGrid.cxx

5.131 rxdock::InteractionCenterCmp Class Reference

Public Member Functions

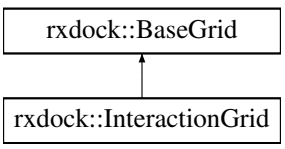
- bool **operator()** (const InteractionCenter *pIC1, const InteractionCenter *pIC2) const

The documentation for this class was generated from the following file:

- InteractionGrid.h

5.132 rxdock::InteractionGrid Class Reference

Inheritance diagram for rxdock::InteractionGrid:



Public Member Functions

- **InteractionGrid** (const Coord &gridMin, const Coord &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **InteractionGrid** (json j)
- **InteractionGrid** (const InteractionGrid &)
- **InteractionGrid** (const BaseGrid &)
- InteractionGrid & **operator=** (const InteractionGrid &)
- InteractionGrid & **operator=** (const BaseGrid &)
- virtual void Print (std::ostream &ostr) const
- const InteractionCenterList & **GetInteractionList** (unsigned int iXYZ) const
- const InteractionCenterList & **GetInteractionList** (const Coord &c) const
- void **SetInteractionLists** (InteractionCenter *pIntn, double radius)
- void **ClearInteractionLists** ()
- void **UniqueInteractionLists** ()

Public Member Functions inherited from rxdock::BaseGrid

- **BaseGrid** (const Coord &gridMin, const Vector &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **BaseGrid** (json j)
- **BaseGrid** (const BaseGrid &)
- BaseGrid & **operator=** (const BaseGrid &)
- virtual void **Print** (std::ostream &ostr) const
- unsigned int **GetNX** () const
- unsigned int **GetNY** () const
- unsigned int **GetNZ** () const
- unsigned int **GetN** () const
- unsigned int **GetStrideX** () const
- unsigned int **GetStrideY** () const
- unsigned int **GetStrideZ** () const
- int **GetnXMin** () const
- int **GetnYMin** () const
- int **GetnZMin** () const
- int **GetnXMax** () const
- int **GetnYMax** () const
- int **GetnZMax** () const
- const Coord & **GetGridMin** () const
- const Coord & **GetGridMax** () const
- const Vector & **GetGridStep** () const
- Coord **GetGridCenter** () const
- Coord **GetGridSize** () const
- unsigned int **GetPad** () const
- const Coord & **GetPadMin** () const
- const Coord & **GetPadMax** () const
- void **SetGridMin** (const Coord &gridMin)
- void **TranslateGrid** (const Vector &vec)
- void **SetGridCenter** (const Coord &gridCenter)
- void **SetGridStep** (const Vector &gridStep)
- void **SetPad** (unsigned int NPad=0)
- bool **isValid** (const Coord &c) const
- bool **isValid** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- bool **isValid** (unsigned int iXYZ) const
- unsigned int **GetIX** (const Coord &c) const

- unsigned int **GetIY** (const Coord &c) const
- unsigned int **GetIZ** (const Coord &c) const
- unsigned int **GetIX** (double x) const
- unsigned int **GetIY** (double y) const
- unsigned int **GetIZ** (double z) const
- unsigned int **GetIX** (unsigned int iXYZ) const
- unsigned int **GetIY** (unsigned int iXYZ) const
- unsigned int **GetIZ** (unsigned int iXYZ) const
- unsigned int **GetIXYZ** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- unsigned int **GetIXYZ** (const Coord &c) const
- Coord **GetCoord** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetXCoord** (unsigned int iX) const
- double **GetYCoord** (unsigned int iY) const
- double **GetZCoord** (unsigned int iZ) const
- Coord **GetCoord** (unsigned int iXYZ) const
- CoordList **GetCoordList** (const std::set< unsigned int > &iXYZSet) const
- void **GetSphereIndices** (const Coord &c, double radius, std::vector< unsigned int > &sIndices) const

Static Public Attributes

- static const std::string **_CT** = ¨InteractionGrid¨

Static Public Attributes inherited from rxdock::BaseGrid

- static const std::string **_CT** = ¨BaseGrid¨

Protected Member Functions

- void **OwnPrint** (std::ostream &ostr) const

Protected Member Functions inherited from rxdock::BaseGrid

- void **OwnPrint** (std::ostream &ostr) const

5.132.1 Member Function Documentation

5.132.1.1 Print()

```
void InteractionGrid::Print (
            std::ostream & ostr ) const  [virtual]
```

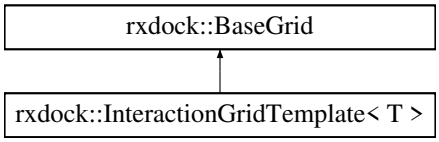
Reimplemented from rxdock::BaseGrid.

The documentation for this class was generated from the following files:

- InteractionGrid.h
- InteractionGrid.cxx

5.133 rxdock::InteractionGridTemplate< T > Class Template Reference

Inheritance diagram for rxdock::InteractionGridTemplate< T >:



Public Member Functions

- **InteractionGridTemplate** (const Coord &gridMin, const Coord &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **InteractionGridTemplate** (std::istream &istr)
- **InteractionGridTemplate** (const InteractionGridTemplate &aGrid)
- **InteractionGridTemplate** (const BaseGrid &)
- InteractionGridTemplate & **operator=** (const InteractionGridTemplate &aGrid)
- InteractionGridTemplate & **operator=** (const BaseGrid &aGrid)
- virtual void Print (std::ostream &ostr) const
- const InteractionNodeList< T > & **GetInteractionList** (unsigned int iXYZ) const
- const InteractionNodeList< T > & **GetInteractionList** (const Coord &c) const
- void **SetInteractionLists** (T *pIntn, double radius)
- void **ClearInteractionLists** ()

Public Member Functions inherited from rxdock::BaseGrid

- **BaseGrid** (const Coord &gridMin, const Vector &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **BaseGrid** (json j)
- **BaseGrid** (const BaseGrid &)
- BaseGrid & **operator=** (const BaseGrid &)
- virtual void **Print** (std::ostream &ostr) const
- unsigned int **GetNX** () const
- unsigned int **GetNY** () const
- unsigned int **GetNZ** () const
- unsigned int **GetN** () const
- unsigned int **GetStrideX** () const
- unsigned int **GetStrideY** () const
- unsigned int **GetStrideZ** () const
- int **GetnXMin** () const
- int **GetnYMin** () const
- int **GetnZMin** () const
- int **GetnXMax** () const
- int **GetnYMax** () const
- int **GetnZMax** () const
- const Coord & **GetGridMin** () const
- const Coord & **GetGridMax** () const
- const Vector & **GetGridStep** () const
- Coord **GetGridCenter** () const
- Coord **GetGridSize** () const
- unsigned int **GetPad** () const

- const Coord & **GetPadMin** () const
- const Coord & **GetPadMax** () const
- void **SetGridMin** (const Coord &gridMin)
- void **TranslateGrid** (const Vector &vec)
- void **SetGridCenter** (const Coord &gridCenter)
- void **SetGridStep** (const Vector &gridStep)
- void **SetPad** (unsigned int NPad=0)
- bool **isValid** (const Coord &c) const
- bool **isValid** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- bool **isValid** (unsigned int iXYZ) const
- unsigned int **GetIX** (const Coord &c) const
- unsigned int **GetIY** (const Coord &c) const
- unsigned int **GetIZ** (const Coord &c) const
- unsigned int **GetIX** (double x) const
- unsigned int **GetIY** (double y) const
- unsigned int **GetIZ** (double z) const
- unsigned int **GetIX** (unsigned int iXYZ) const
- unsigned int **GetIY** (unsigned int iXYZ) const
- unsigned int **GetIZ** (unsigned int iXYZ) const
- unsigned int **GetIXYZ** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- unsigned int **GetIXYZ** (const Coord &c) const
- Coord **GetCoord** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetXCoord** (unsigned int iX) const
- double **GetYCoord** (unsigned int iY) const
- double **GetZCoord** (unsigned int iZ) const
- Coord **GetCoord** (unsigned int iXYZ) const
- CoordList **GetCoordList** (const std::set< unsigned int > &iXYZSet) const
- void **GetSphereIndices** (const Coord &c, double radius, std::vector< unsigned int > &sIndices) const

Static Public Attributes

- static std::string **_CT**

Static Public Attributes inherited from rxdock::BaseGrid

- static const std::string **_CT** = ¨BaseGrid¨

Protected Member Functions

- void **OwnPrint** (std::ostream &ostr) const

Protected Member Functions inherited from rxdock::BaseGrid

- void **OwnPrint** (std::ostream &ostr) const

5.133.1 Member Function Documentation

5.133.1.1 Print()

```
template<class T >
virtual void rxdock::InteractionGridTemplate< T >::Print (
            std::ostream & ostr ) const  [inline], [virtual]
```

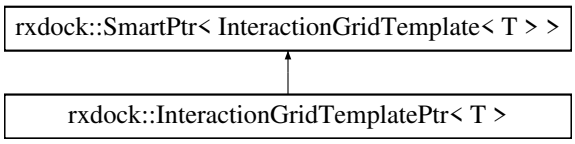
Reimplemented from rxdock::BaseGrid.

The documentation for this class was generated from the following file:

- InteractionTemplate.h

5.134 rxdock::InteractionGridTemplatePtr< T > Class Template Reference

Inheritance diagram for rxdock::InteractionGridTemplatePtr< T >:



Additional Inherited Members

Public Member Functions inherited from rxdock::SmartPtr< InteractionGridTemplate< T > >

- **SmartPtr** (InteractionGridTemplate< T > *pT)
- **SmartPtr** (const SmartPtr< InteractionGridTemplate< T > > &sp)
- **SmartPtr** (const SmartPtr< T2 > &sp)
- const SmartPtr< InteractionGridTemplate< T > > & **operator=** (const SmartPtr< InteractionGridTemplate< T > > &sp)
- const SmartPtr< InteractionGridTemplate< T > > & **operator=** (const SmartPtr< T2 > &sp)
- bool **Null** () const
- unsigned * **GetCountPtr** () const
- InteractionGridTemplate< T > * **Ptr** ()
- const InteractionGridTemplate< T > * **Ptr** () const
- void **SetNull** ()
- InteractionGridTemplate< T > * **operator->** ()
- const InteractionGridTemplate< T > * **operator->** () const
- InteractionGridTemplate< T > & **operator*** ()
- const InteractionGridTemplate< T > & **operator*** () const
- **operator InteractionGridTemplate< T >** * () const

The documentation for this class was generated from the following file:

- InteractionTemplate.h

5.135 rxdock::InteractionNode< T > Class Template Reference

Public Member Functions

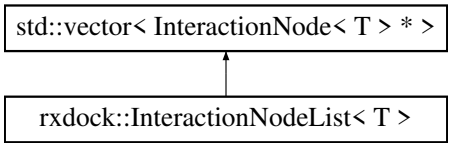
- **InteractionNode** (T aProperty)
- T **GetProperty** () const
- void **SetProperty** (T aProperty)

The documentation for this class was generated from the following file:

- InteractionTemplate.h

5.136 rxdock::InteractionNodeList< T > Class Template Reference

Inheritance diagram for rxdock::InteractionNodeList< T >:

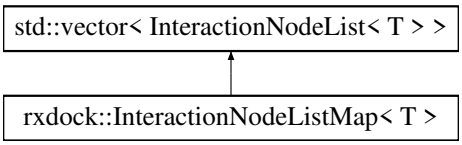


The documentation for this class was generated from the following file:

- InteractionTemplate.h

5.137 rxdock::InteractionNodeListMap< T > Class Template Reference

Inheritance diagram for rxdock::InteractionNodeListMap< T >:

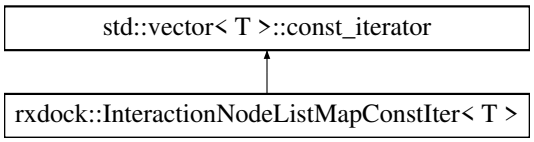


The documentation for this class was generated from the following file:

- InteractionTemplate.h

5.138 rxdock::InteractionNodeListMapConstIter< T > Class Template Reference

Inheritance diagram for rxdock::InteractionNodeListMapConstIter< T >:

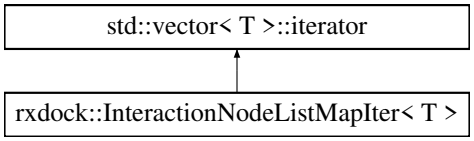


The documentation for this class was generated from the following file:

- InteractionTemplate.h

5.139 rxdock::InteractionNodeListMapIter< T > Class Template Reference

Inheritance diagram for rxdock::InteractionNodeListMapIter< T >:

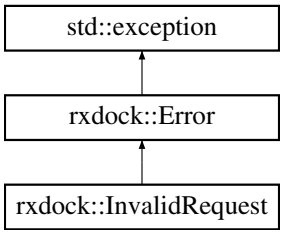


The documentation for this class was generated from the following file:

- InteractionTemplate.h

5.140 rxdock::InvalidRequest Class Reference

Inheritance diagram for rxdock::InvalidRequest:



Public Member Functions

- **InvalidRequest** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::Error

- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- Error.h

5.141 rxdock::InvertSelectAtom Class Reference

Public Member Functions

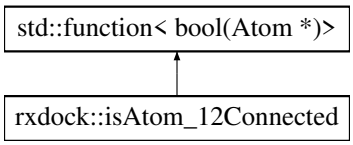
- void **operator()** (Atom *pAtom)

The documentation for this class was generated from the following file:

- Atom.h

5.142 rxdock::isAtom_12Connected Class Reference

Inheritance diagram for rxdock::isAtom_12Connected:



Public Member Functions

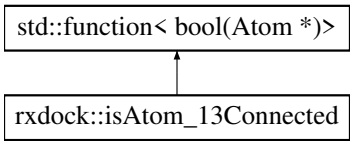
- **isAtom_12Connected** (Atom ∗spAtom)
- bool **operator()** (Atom ∗spAtom2) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.143 rxdock::isAtom_13Connected Class Reference

Inheritance diagram for rxdock::isAtom_13Connected:



Public Member Functions

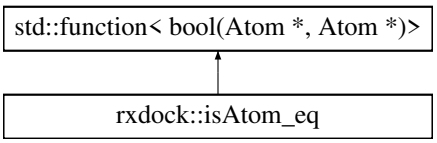
- **isAtom_13Connected** (Atom ∗pAtom)
- bool **operator()** (Atom ∗pAtom2) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.144 rxdock::isAtom_eq Class Reference

Inheritance diagram for rxdock::isAtom_eq:



Public Member Functions

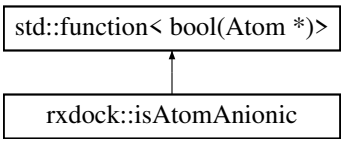
- bool **operator()** (const Atom ∗pAtom1, const Atom ∗pAtom2) const

The documentation for this class was generated from the following file:

- Atom.h

5.145 rxdock::isAtomAnionic Class Reference

Inheritance diagram for rxdock::isAtomAnionic:



Public Member Functions

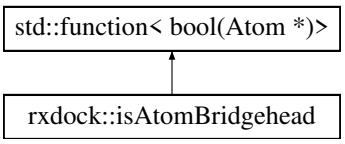
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.146 rxdock::isAtomBridgehead Class Reference

Inheritance diagram for rxdock::isAtomBridgehead:



Public Member Functions

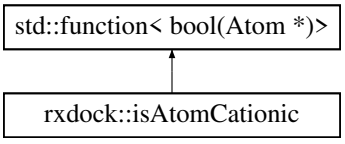
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.147 rxdock::isAtomCationic Class Reference

Inheritance diagram for rxdock::isAtomCationic:



Public Member Functions

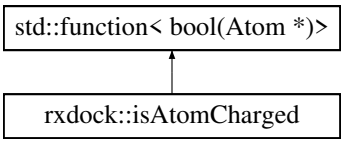
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.148 rxdock::isAtomCharged Class Reference

Inheritance diagram for rxdock::isAtomCharged:



Public Member Functions

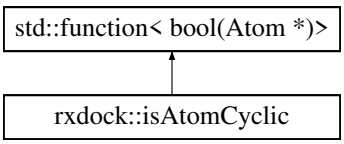
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.149 rxdock::isAtomCyclic Class Reference

Inheritance diagram for rxdock::isAtomCyclic:



Public Member Functions

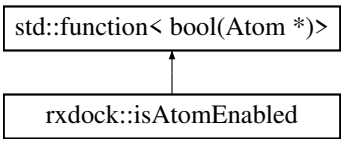
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.150 rxdock::isAtomEnabled Class Reference

Inheritance diagram for rxdock::isAtomEnabled:



Public Member Functions

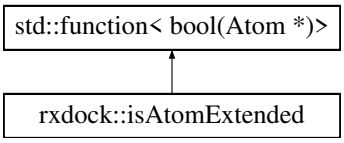
- bool **operator()** (Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.151 rxdock::isAtomExtended Class Reference

Inheritance diagram for rxdock::isAtomExtended:



Public Member Functions

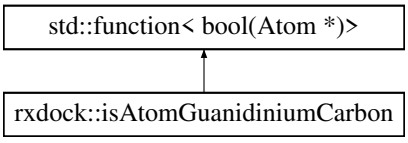
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.152 rxdock::isAtomGuanidiniumCarbon Class Reference

Inheritance diagram for rxdock::isAtomGuanidiniumCarbon:



Public Member Functions

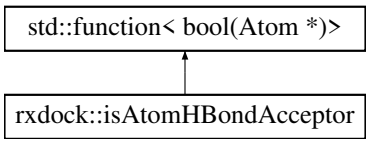
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.153 rxdock::isAtomHBondAcceptor Class Reference

Inheritance diagram for rxdock::isAtomHBondAcceptor:



Public Member Functions

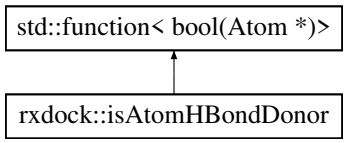
- RBTDLL_EXPORT bool **operator()** (const Atom ∗) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.154 rxdock::isAtomHBondDonor Class Reference

Inheritance diagram for rxdock::isAtomHBondDonor:



Public Member Functions

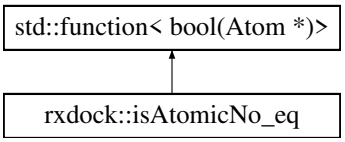
- RBTDLL_EXPORT bool **operator()** (const Atom ∗) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.155 rxdock::isAtomicNo_eq Class Reference

Inheritance diagram for rxdock::isAtomicNo_eq:



Public Member Functions

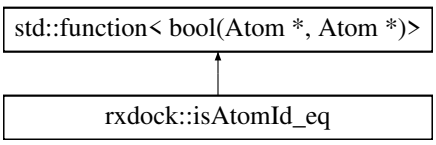
- **isAtomicNo_eq** (int nn)
- bool **operator()** (const Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.156 rxdock::isAtomId_eq Class Reference

Inheritance diagram for rxdock::isAtomId_eq:



Public Member Functions

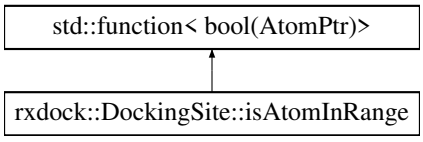
- bool **operator()** (const Atom *pAtom1, const Atom *pAtom2) const

The documentation for this class was generated from the following file:

- Atom.h

5.157 rxdock::DockingSite::isAtomInRange Class Reference

Inheritance diagram for rxdock::DockingSite::isAtomInRange:



Public Member Functions

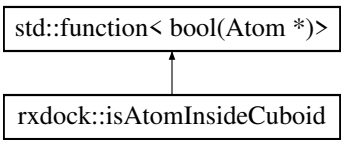
- **isAtomInRange** (RealGrid *pGrid, double minDist, double maxDist)
- bool **operator()** (Atom *pAtom) const

The documentation for this class was generated from the following files:

- DockingSite.h
- DockingSite.cxx

5.158 rxdock::isAtomInsideCuboid Class Reference

Inheritance diagram for rxdock::isAtomInsideCuboid:



Public Member Functions

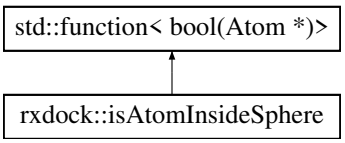
- **isAtomInsideCuboid** (const Coord &ccmin, const Coord &ccmax)
- bool **operator()** (const Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.159 rxdock::isAtomInsideSphere Class Reference

Inheritance diagram for rxdock::isAtomInsideSphere:



Public Member Functions

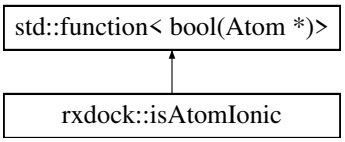
- **isAtomInsideSphere** (const Coord &cc, double rr)
- bool **operator()** (const Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.160 rxdock::isAtomIonic Class Reference

Inheritance diagram for rxdock::isAtomIonic:



Public Member Functions

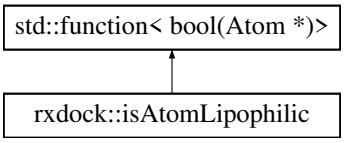
- bool **operator()** (const Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.161 rxdock::isAtomLipophilic Class Reference

Inheritance diagram for rxdock::isAtomLipophilic:



Public Member Functions

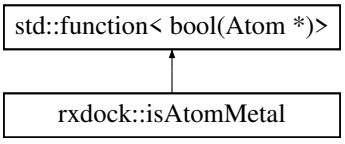
- RBTDLL_EXPORT bool **operator()** (Atom *pAtom) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.162 rxdock::isAtomMetal Class Reference

Inheritance diagram for rxdock::isAtomMetal:



Public Member Functions

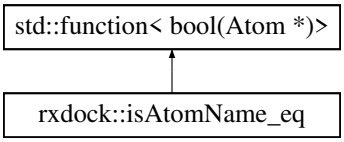
- RBTDLL_EXPORT bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.163 rxdock::isAtomName_eq Class Reference

Inheritance diagram for rxdock::isAtomName_eq:



Public Member Functions

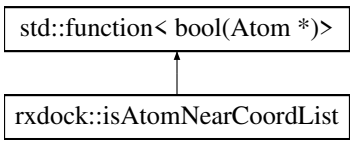
- **isAtomName_eq** (std::string ss)
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.164 rxdock::isAtomNearCoordList Class Reference

Inheritance diagram for rxdock::isAtomNearCoordList:



Public Member Functions

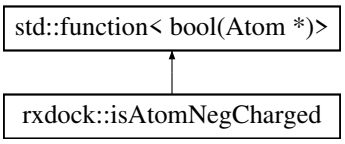
- **isAtomNearCoordList** (const CoordList &ccl, double rr)
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.165 rxdock::isAtomNegCharged Class Reference

Inheritance diagram for rxdock::isAtomNegCharged:



Public Member Functions

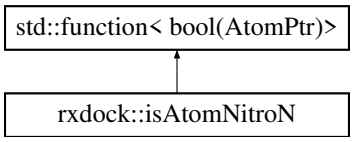
- bool **operator()** (const Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.166 rxdock::isAtomNitroN Class Reference

Inheritance diagram for rxdock::isAtomNitroN:



Public Member Functions

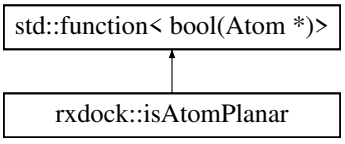
- bool **operator()** (AtomPtr spAtom) const

The documentation for this class was generated from the following file:

- rblist.cxx

5.167 rxdock::isAtomPlanar Class Reference

Inheritance diagram for rxdock::isAtomPlanar:



Public Member Functions

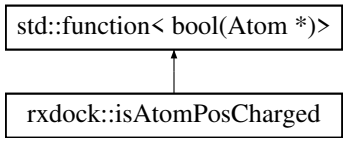
- bool **operator()** (const Atom ∗) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.168 rxdock::isAtomPosCharged Class Reference

Inheritance diagram for rxdock::isAtomPosCharged:



Public Member Functions

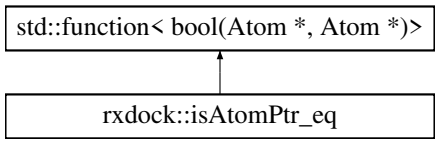
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.169 rxdock::isAtomPtr_eq Class Reference

Inheritance diagram for rxdock::isAtomPtr_eq:



Public Member Functions

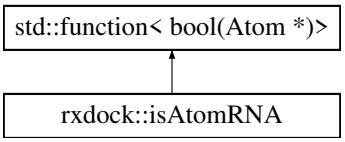
- bool **operator()** (const Atom ∗pAtom1, const Atom ∗pAtom2) const

The documentation for this class was generated from the following file:

- Atom.h

5.170 rxdock::isAtomRNA Class Reference

Inheritance diagram for rxdock::isAtomRNA:



Public Member Functions

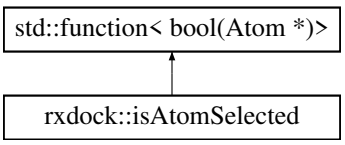
- bool **operator()** (const Atom ∗) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.171 rxdock::isAtomSelected Class Reference

Inheritance diagram for rxdock::isAtomSelected:



Public Member Functions

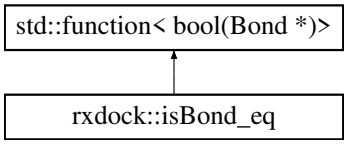
- bool **operator()** (Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.172 rxdock::isBond_eq Class Reference

Inheritance diagram for rxdock::isBond_eq:



Public Member Functions

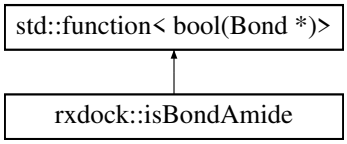
- **isBond_eq** (Bond ∗pp)
- bool **operator()** (Bond ∗pBond) const

The documentation for this class was generated from the following file:

- Bond.h

5.173 rxdock::isBondAmide Class Reference

Inheritance diagram for rxdock::isBondAmide:



Public Member Functions

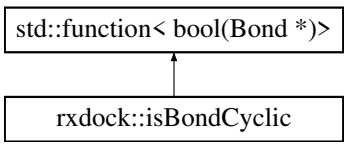
- RBTDLL_EXPORT bool **operator()** (Bond ∗) const

The documentation for this class was generated from the following files:

- Bond.h
- Bond.cxx

5.174 rxdock::isBondCyclic Class Reference

Inheritance diagram for rxdock::isBondCyclic:



Public Member Functions

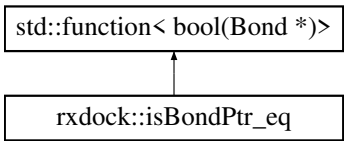
- bool **operator()** (Bond ∗pBond) const

The documentation for this class was generated from the following file:

- Bond.h

5.175 rxdock::isBondPtr_eq Class Reference

Inheritance diagram for rxdock::isBondPtr_eq:



Public Member Functions

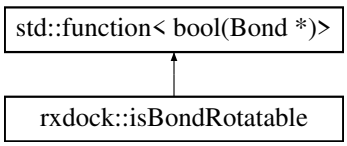
- **isBondPtr_eq** (Bond ∗pp)
- bool **operator()** (Bond ∗pBond) const

The documentation for this class was generated from the following file:

- Bond.h

5.176 rxdock::isBondRotatable Class Reference

Inheritance diagram for rxdock::isBondRotatable:



Public Member Functions

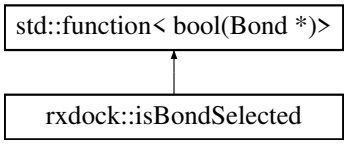
- RBTDLL_EXPORT bool **operator()** (Bond ∗) const

The documentation for this class was generated from the following files:

- Bond.h
- Bond.cxx

5.177 rxdock::isBondSelected Class Reference

Inheritance diagram for rxdock::isBondSelected:



Public Member Functions

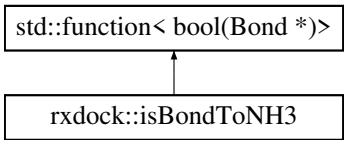
- bool **operator()** (Bond ∗pBond) const

The documentation for this class was generated from the following file:

- Bond.h

5.178 rxdock::isBondToNH3 Class Reference

Inheritance diagram for rxdock::isBondToNH3:



Public Member Functions

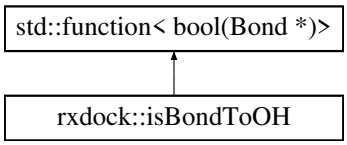
- RBTDLL_EXPORT bool **operator()** (Bond ∗) const

The documentation for this class was generated from the following files:

- Bond.h
- Bond.cxx

5.179 rxdock::isBondToOH Class Reference

Inheritance diagram for rxdock::isBondToOH:



Public Member Functions

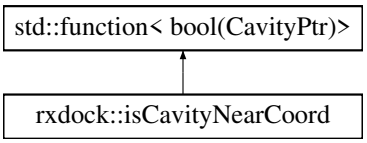
- RBTDLL_EXPORT bool **operator()** (Bond ∗pBond) const

The documentation for this class was generated from the following files:

- Bond.h
- Bond.cxx

5.180 rxdock::isCavityNearCoord Class Reference

Inheritance diagram for rxdock::isCavityNearCoord:



Public Member Functions

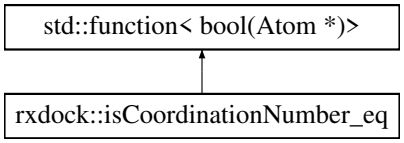
- **isCavityNearCoord** (const Coord &cc, double rr)
- bool **operator()** (CavityPtr spCavity) const

The documentation for this class was generated from the following file:

- Cavity.h

5.181 rxdock::isCoordinationNumber_eq Class Reference

Inheritance diagram for rxdock::isCoordinationNumber_eq:



Public Member Functions

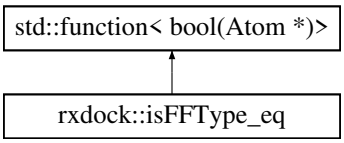
- **isCoordinationNumber_eq** (unsigned int nn)
- **isCoordinationNumber_eq** (unsigned int nn, int nAt)
- **isCoordinationNumber_eq** (unsigned int nn, std::string strType)
- **isCoordinationNumber_eq** (unsigned int nn, Atom::eHybridState eState)
- bool **operator()** (Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.182 rxdock::isFFType_eq Class Reference

Inheritance diagram for rxdock::isFFType_eq:



Public Member Functions

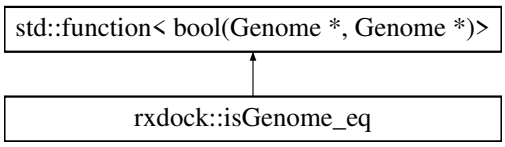
- **isFFType_eq** (std::string ss)
- bool **operator()** (const Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.183 rxdock::isGenome_eq Class Reference

Inheritance diagram for rxdock::isGenome_eq:



Public Member Functions

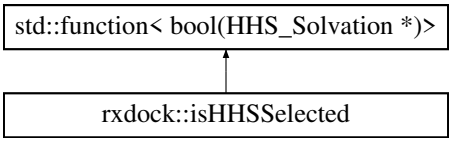
- **isGenome_eq** (double threshold)
- bool **operator()** (const Genome *pG1, const Genome *pG2) const

The documentation for this class was generated from the following file:

- Genome.h

5.184 rxdock::isHHSSelected Class Reference

Inheritance diagram for rxdock::isHHSSelected:



Public Member Functions

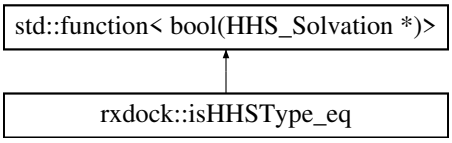
- bool **operator()** (const HHS_Solvation ∗pHHS) const

The documentation for this class was generated from the following files:

- SATypes.h
- SATypes.cxx

5.185 rxdock::isHHSType_eq Class Reference

Inheritance diagram for rxdock::isHHSType_eq:



Public Member Functions

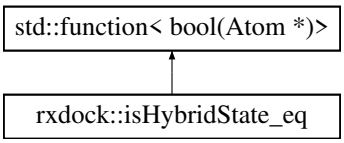
- **isHHSType_eq** (HHSType::eType tt)
- bool **operator()** (const HHS_Solvation ∗pHHS) const

The documentation for this class was generated from the following file:

- SATypes.h

5.186 rxdock::isHybridState_eq Class Reference

Inheritance diagram for rxdock::isHybridState_eq:



Public Member Functions

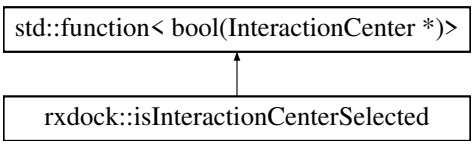
- **isHybridState_eq** (Atom::eHybridState ee)
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.187 rxdock::isInteractionCenterSelected Class Reference

Inheritance diagram for rxdock::isInteractionCenterSelected:



Public Member Functions

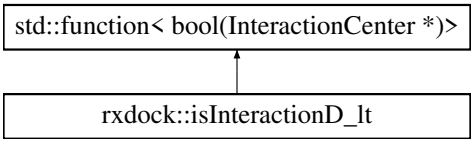
- bool **operator()** (const InteractionCenter *pIC) const

The documentation for this class was generated from the following file:

- InteractionGrid.h

5.188 rxdock::isInteractionD_lt Class Reference

Inheritance diagram for rxdock::isInteractionD_lt:



Public Member Functions

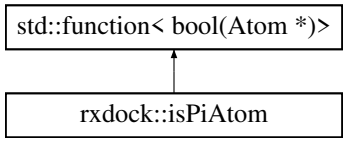
- **isInteractionD_lt** (const InteractionCenter *pIC1, double dd)
- bool **operator()** (const InteractionCenter *pIC2) const

The documentation for this class was generated from the following file:

- InteractionGrid.h

5.189 rxdock::isPiAtom Class Reference

Inheritance diagram for rxdock::isPiAtom:



Public Member Functions

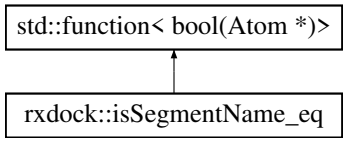
- RBTDLL_EXPORT bool **operator()** (const Atom ∗) const

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.190 rxdock::isSegmentName_eq Class Reference

Inheritance diagram for rxdock::isSegmentName_eq:



Public Member Functions

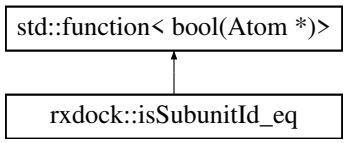
- **isSegmentName_eq** (std::string ss)
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.191 rxdock::isSubunitId_eq Class Reference

Inheritance diagram for rxdock::isSubunitId_eq:



Public Member Functions

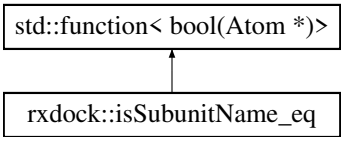
- **isSubunitId_eq** (std::string ss)
- bool **operator()** (const Atom ∗pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.192 rxdock::isSubunitName_eq Class Reference

Inheritance diagram for rxdock::isSubunitName_eq:



Public Member Functions

- **isSubunitName_eq** (std::string ss)
- bool **operator()** (const Atom *pAtom) const

The documentation for this class was generated from the following file:

- Atom.h

5.193 rxdock::neldermead::IterationCriterion Class Reference

Public Member Functions

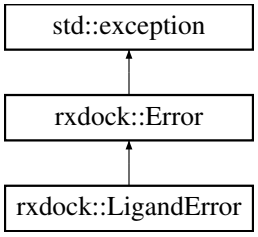
- **IterationCriterion** (long max_iterations)
- template<class State >
  bool **operator()** (const State &state)

The documentation for this class was generated from the following file:

- NMCriteria.h

5.194 rxdock::LigandError Class Reference

Inheritance diagram for rxdock::LigandError:



Public Member Functions

- **LigandError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Protected Member Functions

- **LigandError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &str↩ Message="")

Protected Member Functions inherited from rxdock::Error

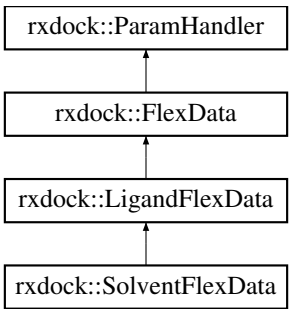
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- LigandError.h

5.195 rxdock::LigandFlexData Class Reference

Inheritance diagram for rxdock::LigandFlexData:



Public Member Functions

- RBTDLL_EXPORT **LigandFlexData** (DockingSite *pDockSite)
- virtual void Accept (FlexDataVisitor &v)

Public Member Functions inherited from rxdock::FlexData

- Model * **GetModel** () const
- void **SetModel** (Model *pModel)
- DockingSite * **GetDockingSite** () const
- virtual void **Accept** (FlexDataVisitor &)=0

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Static Public Member Functions

- static RBTDLL_EXPORT const std::string & **GetTransMode** ()
- static RBTDLL_EXPORT const std::string & **GetRotMode** ()

Static Public Attributes

- static const std::string & **_TRANS_STEP** = "translational-step"
- static const std::string & **_ROT_STEP** = "rotational-step"
- static const std::string & **_DIHEDRAL_STEP** = "dihedral-step"
- static const std::string & **_TRANS_MODE** = "translational-mode"
- static const std::string & **_ROT_MODE** = "rotational-mode"
- static const std::string & **_DIHEDRAL_MODE** = "dihedral-mode"
- static const std::string & **_MAX_TRANS** = "maximum-translation"
- static const std::string & **_MAX_ROT** = "maximum-rotation"
- static const std::string & **_MAX_DIHEDRAL** = "maximum-dihedral"

Additional Inherited Members

Protected Member Functions inherited from rxdock::FlexData

- **FlexData** (DockingSite *pDockSite)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.195.1 Member Function Documentation

5.195.1.1 Accept()

```
virtual void rxdock::LigandFlexData::Accept (
            FlexDataVisitor & v )  [inline], [virtual]
```

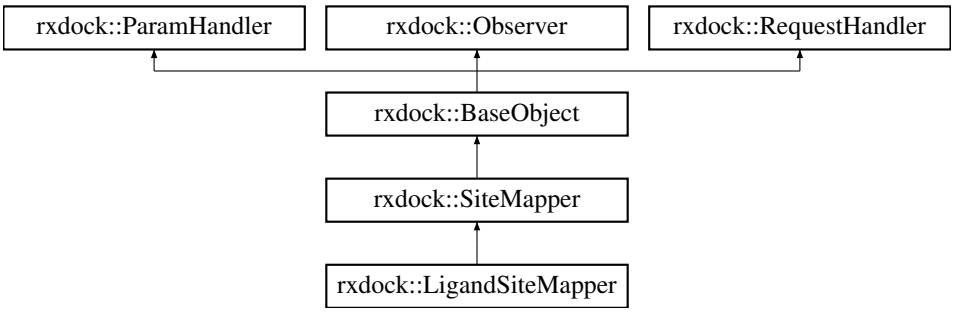
Implements rxdock::FlexData.

The documentation for this class was generated from the following files:

- LigandFlexData.h
- LigandFlexData.cxx

5.196 rxdock::LigandSiteMapper Class Reference

Inheritance diagram for rxdock::LigandSiteMapper:



Public Member Functions

- **LigandSiteMapper** (const std::string &strName=¨LIGAND_MAPPER¨)
- virtual CavityList operator() ()

Public Member Functions inherited from rxdock::SiteMapper

- ModelPtr **GetReceptor** () const
- virtual CavityList **operator()** ()=0
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨LigandSiteMapper¨
- static const std::string **_REF_MOL** = ¨reference-molecule¨
- static const std::string _VOL_INCR
- static const std::string **_SMALL_SPHERE** = ¨small-sphere-radius¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_RADIUS** = ¨region-radius¨
- static const std::string **_MIN_VOLUME** = ¨minimum-cavity-volume¨
- static const std::string **_MAX_CAVITIES** = ¨maximum-cavities¨

Static Public Attributes inherited from rxdock::SiteMapper

- static const std::string **_CT** = ¨SiteMapper¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Additional Inherited Members

Protected Member Functions inherited from rxdock::SiteMapper

- **SiteMapper** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.196.1 Member Function Documentation

5.196.1.1 operator()()

```
CavityList LigandSiteMapper::operator() ( )  [virtual]
```

Implements rxdock::SiteMapper.

5.196.2 Member Data Documentation

5.196.2.1 _VOL_INCR

```
const std::string LigandSiteMapper::_VOL_INCR  [static]
```

Initial value:

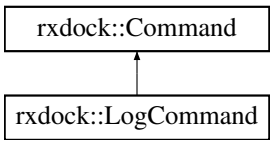
```
=
    "excluded-volume-radius-increment"
```

The documentation for this class was generated from the following files:

- LigandSiteMapper.h
- LigandSiteMapper.cxx

5.197 rxdock::LogCommand Class Reference

Inheritance diagram for rxdock::LogCommand:



Public Member Functions

- int GetNArgs ()
- std::string GetName ()
- void SetArg (int i, ReturnType f)
- void SetNameArg (int i, std::string &n)
- ReturnType Execute ()

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Additional Inherited Members

Static Public Member Functions inherited from rxdock::Command

- static void **Clear** ()

Static Public Attributes inherited from rxdock::Command

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions inherited from rxdock::Command

- std::string **tabs** (int n)

5.197.1 Member Function Documentation

5.197.1.1 Execute()

```
ReturnType rxdock::LogCommand::Execute ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.197.1.2 GetName()

```
std::string rxdock::LogCommand::GetName ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.197.1.3 GetNArgs()

```
int rxdock::LogCommand::GetNArgs ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.197.1.4 SetArg()

```
void rxdock::LogCommand::SetArg (
            int i,
            ReturnType f )  [inline], [virtual]
```

Implements rxdock::Command.

5.197.1.5 SetNameArg()

```
void rxdock::LogCommand::SetNameArg (
            int i,
            std::string & n )  [inline], [virtual]
```

Implements rxdock::Command.

The documentation for this class was generated from the following file:

- Command.h

5.198 rxdock::MCStats Class Reference

Public Member Functions

- void **Init** (double score)
- void **InitBlock** (double score)
- void **Accumulate** (double score, bool bAccepted)
- double **Mean** () const
- double **Variance** () const
- double **AccRate** () const

Public Attributes

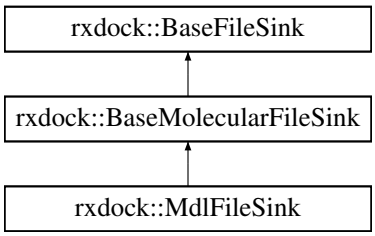
- double **_total**
- double **_total2**
- double **_blockInitial**
- double **_blockFinal**
- double **_blockMin**
- double **_blockMax**
- double **_initial**
- double **_final**
- double **_min**
- double **_max**
- int **_steps**
- int **_accepted**

The documentation for this class was generated from the following files:

- SimAnnTransform.h
- SimAnnTransform.cxx

5.199 rxdock::MdlFileSink Class Reference

Inheritance diagram for rxdock::MdlFileSink:



Public Member Functions

- RBTDLL_EXPORT **MdlFileSink** (const std::string &fileName, ModelPtr spModel)
- virtual void Render ()

Public Member Functions inherited from rxdock::BaseMolecularFileSink

- **BaseMolecularFileSink** (const std::string &fileName, ModelPtr spModel, bool bUseModelSegmentNames=true)
- virtual bool **isMultiConfSupported** ()
- ModelPtr **GetModel** () const
- RBTDLL_EXPORT void **SetModel** (ModelPtr spModel, bool bUseModelSegmentNames=false)
- void **SetNextSegmentId** (int nSegmentId)
- bool **GetMultiConf** () const
- void **SetMultiConf** (bool bMultiConf)
- void **WriteMultiConf** ()
- ModelList **GetSolvent** () const
- void **SetSolvent** (ModelList solventList)

Public Member Functions inherited from rxdock::BaseFileSink

- **BaseFileSink** (const std::string &fileName)
- std::string **GetFileName** () const
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- Error **Status** ()
- virtual void **Render** ()=0

Friends

- void **to_json** (json &j, const MdlFileSink &mdlFileSink)
- void **from_json** (const json &j, MdlFileSink &mdlFileSink)

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseMolecularFileSink

- void **Reset** ()

Protected Member Functions inherited from rxdock::BaseFileSink

- void **Write** (bool bClearCache=true)
- void **AddLine** (const std::string &fileRec)
- void **ReplaceLine** (const std::string &fileRec, unsigned int nRec)
- bool **isCacheEmpty** () const
- bool **GetAppend** () const
- void **SetAppend** (bool bAppend)

Protected Attributes inherited from rxdock::BaseMolecularFileSink

- int **m_nAtomId**
- int **m_nSubunitId**
- int **m_nSegmentId**
- bool **m_bUseModelSegmentNames**

5.199.1 Member Function Documentation

5.199.1.1 Render()

```
void MdlFileSink::Render ( )  [virtual]
```

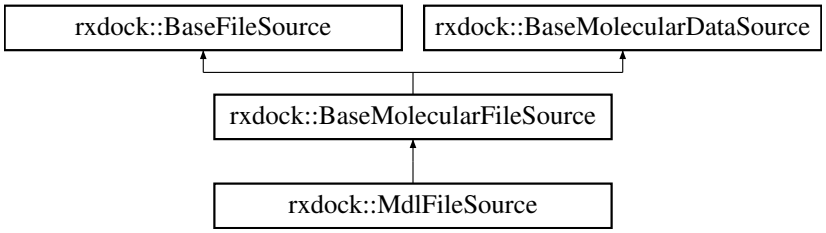
Implements rxdock::BaseFileSink.

The documentation for this class was generated from the following files:

- MdlFileSink.h
- MdlFileSink.cxx

5.200 rxdock::MdlFileSource Class Reference

Inheritance diagram for rxdock::MdlFileSource:



Public Member Functions

- RBTDLL_EXPORT **MdlFileSource** (const std::string &fileName, bool bPosIonisable=true, bool bNeg↩ Ionisable=true, bool bImplHydrogens=true)
- virtual bool isTitleListSupported ()
- virtual bool isAtomListSupported ()
- virtual bool isCoordinatesSupported ()
- virtual bool isBondListSupported ()
- virtual bool isDataSupported ()

Public Member Functions inherited from rxdock::BaseMolecularFileSource

- RBTDLL_EXPORT **BaseMolecularFileSource** (const std::string &fileName, const std::string &sourceName)
- **BaseMolecularFileSource** (const std::string &fileName, const std::string &strRecDelim, const std::string &sourceName)
- void Reset ()
- int GetNumTitles ()
- int GetNumAtoms ()
- int GetNumBonds ()
- int GetNumSegments ()
- std::vector< std::string > GetTitleList ()
- AtomList GetAtomList ()
- BondList GetBondList ()
- SegmentMap GetSegmentMap ()

- virtual int GetNumData ()
- virtual std::vector< std::string > GetDataFieldList ()
- virtual StringVariantMap GetDataMap ()
- virtual bool isDataFieldPresent (const std::string &strDataField)
- virtual Variant GetDataValue (const std::string &strDataField)
- SegmentMap **GetSegmentFilterMap** ()
- RBTDLL_EXPORT void **SetSegmentFilterMap** (const SegmentMap &segmentFilterMap)
- void **ClearSegmentFilterMap** ()
- bool **isSegmentFilterMapDefined** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Public Member Functions inherited from rxdock::BaseMolecularDataSource

- **BaseMolecularDataSource** (std::string strName)
- std::string **GetName** ()
- virtual bool **isTitleListSupported** ()
- virtual bool **isAtomListSupported** ()
- virtual bool **isCoordinatesSupported** ()
- virtual bool **isBondListSupported** ()
- virtual void **Reset** ()=0
- virtual int **GetNumTitles** ()=0
- virtual int **GetNumAtoms** ()=0
- virtual int **GetNumBonds** ()=0
- virtual int **GetNumSegments** ()=0
- virtual std::vector< std::string > **GetTitleList** ()=0
- virtual AtomList **GetAtomList** ()=0
- virtual BondList **GetBondList** ()=0
- virtual SegmentMap **GetSegmentMap** ()=0
- virtual bool **isDataSupported** ()
- virtual int **GetNumData** ()=0
- virtual std::vector< std::string > **GetDataFieldList** ()=0
- virtual StringVariantMap **GetDataMap** ()=0
- virtual bool **isDataFieldPresent** (const std::string &strDataField)=0
- virtual Variant **GetDataValue** (const std::string &strDataField)=0

Protected Member Functions

- virtual void Parse ()

Protected Member Functions inherited from rxdock::BaseMolecularFileSource

- void **ClearMolCache** ()
- void **RemoveAtom** (AtomPtr spAtom)
- void **RenumberAtomsAndBonds** ()
- void **SetupPartialIonicGroups** (AtomList &atoms, ParameterFileSourcePtr spParamSource)

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Friends

- void **to_json** (json &j, const MdlFileSource &mdlFileSrc)
- void **from_json** (const json &j, MdlFileSource &mdlFileSrc)

Additional Inherited Members

Protected Attributes inherited from rxdock::BaseMolecularFileSource

- std::vector< std::string > **m_titleList**
- AtomList **m_atomList**
- BondList **m_bondList**
- SegmentMap **m_segmentMap**
- StringVariantMap **m_dataMap**

Protected Attributes inherited from rxdock::BaseFileSource

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

5.200.1 Member Function Documentation

5.200.1.1 isAtomListSupported()

```
virtual bool rxdock::MdlFileSource::isAtomListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.200.1.2 isBondListSupported()

```
virtual bool rxdock::MdlFileSource::isBondListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.200.1.3 isCoordinatesSupported()

```
virtual bool rxdock::MdlFileSource::isCoordinatesSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.200.1.4 isDataSupported()

```
virtual bool rxdock::MdlFileSource::isDataSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.200.1.5 isTitleListSupported()

```
virtual bool rxdock::MdlFileSource::isTitleListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.200.1.6 Parse()

```
void MdlFileSource::Parse (
            void  )  [protected], [virtual]
```

Implements rxdock::BaseFileSource.

The documentation for this class was generated from the following files:

- MdlFileSource.h
- MdlFileSource.cxx

5.201 rxdock::Model Class Reference

Public Member Functions

- RBTDLL_EXPORT **Model** (BaseMolecularFileSource *pMolSource)
- RBTDLL_EXPORT **Model** (AtomList &atomList, BondList &bondList)
- **Model** (json j)
- std::string **GetName** () const
- int **GetNumTitles** () const
- std::vector< std::string > **GetTitleList** () const
- void **SetTitle** (std::size_t index, const std::string &strTitle)
- int **GetNumAtoms** () const
- AtomList **GetAtomList** () const
- int **GetNumBonds** () const
- BondList **GetBondList** () const
- int **GetNumSegments** () const
- SegmentMap **GetSegmentMap** () const
- int **GetNumRings** () const

- AtomListList **GetRingAtomLists** () const
- int **GetNumData** () const
- std::vector< std::string > **GetDataFieldList** () const
- StringVariantMap **GetDataMap** () const
- bool **isDataFieldPresent** (const std::string &strDataField) const
- RBTDLL_EXPORT Variant **GetDataValue** (const std::string &strDataField) const
- RBTDLL_EXPORT void **SetDataValue** (const std::string &strDataField, const Variant &dataValue)
- void **ClearDataField** (const std::string &strDataField)
- RBTDLL_EXPORT void **ClearAllDataFields** (const std::string &strDataFieldPrefix)
- void **ClearAllDataFields** ()
- PseudoAtomPtr **AddPseudoAtom** (const AtomList &atomList)
- void **ClearPseudoAtoms** ()
- void **UpdatePseudoAtoms** ()
- unsigned int **GetNumPseudoAtoms** () const
- PseudoAtomList **GetPseudoAtomList** () const
- unsigned int **GetNumTetheredAtoms** () const
- AtomList **GetTetheredAtomList** () const
- double **GetOccupancy** () const
- bool **GetEnabled** () const
- RBTDLL_EXPORT void **SetOccupancy** (double occupancy, double threshold=0.5)
- void **UpdateCoords** (BaseMolecularFileSource *pMolSource)
- void **Translate** (const Vector &vector)
- void **Rotate** (const Vector &axis, double thetaDeg)
- void **Rotate** (const Vector &axis, double thetaDeg, const Coord &center)
- RBTDLL_EXPORT void **RotateBond** (BondPtr spBond, double thetaDeg)
- void **RotateBond** (BondPtr spBond, double thetaDeg, AtomPtr spFixedAtom)
- RBTDLL_EXPORT void **RotateBond** (BondPtr spBond, double thetaDeg, bool bSwap)
- void **SaveCoords** (const std::string &coordName="")
- RBTDLL_EXPORT void **RevertCoords** (const std::string &coordName="")
- std::map< std::string, int > **GetSavedCoordNames** () const
- int **GetNumSavedCoords** () const
- int **GetCurrentCoords** () const
- RBTDLL_EXPORT void **RevertCoords** (int)
- Coord **GetCenterOfMass** () const
- void **SetCenterOfMass** (const Coord &c)
- RBTDLL_EXPORT double **GetTotalMass** () const
- PrincipalAxes **GetPrincipalAxes** () const
- void **AlignPrincipalAxes** (const PrincipalAxes &alignAxes=PrincipalAxes(), bool bAlignCOM=true)
- RBTDLL_EXPORT void **SetFlexData** (FlexData *pFlexData)
- RBTDLL_EXPORT FlexData * **GetFlexData** () const
- RBTDLL_EXPORT ChromElement * **GetChrom** () const
- bool **isFlexible** () const
- const AtomRList & **GetFlexIntns** (Atom *pAtom) const
- BondList **GetFlexBonds** () const
- void **SelectFlexAtoms** ()
- void **SelectFlexAtoms** (Atom *pAtom)
- void **SetAtomSelectionFlags** (bool bSelected=true)
- unsigned int **GetNumSelectedAtoms** ()
- AtomList **GetSelectedAtomList** ()
- void **SetAtomCyclicFlags** (bool bCyclic=true)
- unsigned int **GetNumCyclicAtoms** ()
- AtomList **GetCyclicAtomList** ()
- void **SetAtomUser1Flags** (bool bUser1=true)
- void **SetAtomUser1Values** (double dUser1=0.0)
- void **SetAtomUser2Values** (double dUser2=0.0)

- unsigned int **GetNumHBondAcceptorAtoms** ()
- AtomList **GetHBondAcceptorAtomList** ()
- unsigned int **GetNumHBondDonorAtoms** ()
- AtomList **GetHBondDonorAtomList** ()
- unsigned int **GetNumChargedAtoms** ()
- AtomList **GetChargedAtomList** ()
- unsigned int **GetNumPlanarAtoms** ()
- AtomList **GetPlanarAtomList** ()
- unsigned int **GetNumAtomsWithAtomicNo** (int nAtomicNo)
- AtomList **GetAtomListWithAtomicNo** (int nAtomicNo)
- unsigned int **GetNumAtomsWithFFType** (std::string strFFType)
- AtomList **GetAtomListWithFFType** (std::string strFFType)
- template<class Predicate >
  unsigned int **GetNumBonds** (const Predicate &pred)
- template<class Predicate >
  BondList **GetBondList** (const Predicate &pred)
- void **SetBondSelectionFlags** (bool bSelected=true)
- unsigned int **GetNumSelectedBonds** ()
- BondList **GetSelectedBondList** ()
- void **SetBondCyclicFlags** (bool bCyclic=true)
- unsigned int **GetNumCyclicBonds** ()
- BondList **GetCyclicBondList** ()
- void **GetMinMaxCoords** (Coord &minCoord, Coord &maxCoord, bool bInit=true)
- std::map< std::string, int > **GetAtomTypeMap** ()
- std::map< std::string, int > **GetBondTypeMap** ()

Friends

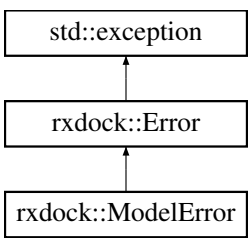
- class **ModelMutator**
- void **to_json** (json &j, const Model &model)
- void **from_json** (const json &j, Model &model)

The documentation for this class was generated from the following files:

- Model.h
- Model.cxx

5.202 rxdock::ModelError Class Reference

Inheritance diagram for rxdock::ModelError:



Public Member Functions

- **ModelError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Protected Member Functions

- **ModelError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- ModelError.h

5.203 rxdock::ModelMutator Class Reference

Public Member Functions

- **ModelMutator** (Model *pModel, const BondList &rotBonds, const AtomList &tetheredAtoms)
- const AtomRListList & **GetFlexIntns** () const
- const AtomRListList & **GetFlexAtoms** () const
- BondList **GetFlexBonds** () const

Friends

- void **to_json** (json &j, const ModelMutator &modelMutator)
- void **from_json** (const json &j, ModelMutator &modelMutator)

The documentation for this class was generated from the following files:

- ModelMutator.h
- ModelMutator.cxx

5.204 rxdock::MOEGrid Class Reference

Public Member Functions

- **MOEGrid** (MOEGridShape &, MOEGridData &, std::string)
- void **SetShape** (MOEGridShape &a_shape)
- void **SetData** (MOEGridData &a_data)
- void **SetOutputFileName** (std::string &a_filename)
- RBTDLL_EXPORT void **CalculateCommonExtents** (std::vector< std::string > strPrmFiles)
- void **GetDockingSiteExtents** (std::string &a_strPrmFile)
- Coord **GetMinExtents** ()
- Coord **GetMaxExtents** ()
- RBTDLL_EXPORT long **WriteGrid** (std::ios_base::openmode mode)
- long **ReadGrid** ()

The documentation for this class was generated from the following files:

- MOEGrid.h
- MOEGrid.cxx

5.205 rxdock::MOEGridPoint Class Reference

Public Member Functions

- **MOEGridPoint** (std::vector< double >, double)
- **MOEGridPoint** (Coord &, double)
- double **GetValue** ()
- const std::vector< double > & **GetCoords** ()
- void **SetCoords** (double a_val)
- void **SetValue** (double a_val)

The documentation for this class was generated from the following files:

- MOEGrid.h
- MOEGrid.cxx

5.206 rxdock::MOEGridShape Class Reference

Public Member Functions

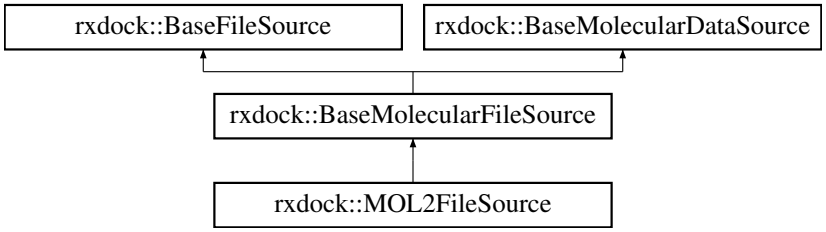
- RBTDLL_EXPORT **MOEGridShape** (std::vector< double >, std::vector< double >, double)
- const std::vector< double > & **GetOrigin** ()
- const std::vector< double > & **GetExtents** ()
- const std::vector< int > & **GetIExtents** ()
- double **GetStepSize** ()
- long **GetDataSize** ()

The documentation for this class was generated from the following files:

- MOEGrid.h
- MOEGrid.cxx

5.207 rxdock::MOL2FileSource Class Reference

Inheritance diagram for rxdock::MOL2FileSource:



Public Member Functions

- **MOL2FileSource** (const std::string &fileName, bool bImplHydrogens=true)
- bool isTitleListSupported ()
- bool isAtomListSupported ()
- bool isCoordinatesSupported ()
- bool isBondListSupported ()
- bool isDataSupported ()

Public Member Functions inherited from rxdock::BaseMolecularFileSource

- RBTDLL_EXPORT **BaseMolecularFileSource** (const std::string &fileName, const std::string &sourceName)
- **BaseMolecularFileSource** (const std::string &fileName, const std::string &strRecDelim, const std::string &sourceName)
- void Reset ()
- int GetNumTitles ()
- int GetNumAtoms ()
- int GetNumBonds ()
- int GetNumSegments ()
- std::vector< std::string > GetTitleList ()
- AtomList GetAtomList ()
- BondList GetBondList ()
- SegmentMap GetSegmentMap ()
- virtual int GetNumData ()
- virtual std::vector< std::string > GetDataFieldList ()
- virtual StringVariantMap GetDataMap ()
- virtual bool isDataFieldPresent (const std::string &strDataField)
- virtual Variant GetDataValue (const std::string &strDataField)
- SegmentMap **GetSegmentFilterMap** ()
- RBTDLL_EXPORT void **SetSegmentFilterMap** (const SegmentMap &segmentFilterMap)
- void **ClearSegmentFilterMap** ()
- bool **isSegmentFilterMapDefined** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Public Member Functions inherited from rxdock::BaseMolecularDataSource

- **BaseMolecularDataSource** (std::string strName)
- std::string **GetName** ()
- virtual bool **isTitleListSupported** ()
- virtual bool **isAtomListSupported** ()
- virtual bool **isCoordinatesSupported** ()
- virtual bool **isBondListSupported** ()
- virtual void **Reset** ()=0
- virtual int **GetNumTitles** ()=0
- virtual int **GetNumAtoms** ()=0
- virtual int **GetNumBonds** ()=0
- virtual int **GetNumSegments** ()=0
- virtual std::vector< std::string > **GetTitleList** ()=0
- virtual AtomList **GetAtomList** ()=0
- virtual BondList **GetBondList** ()=0
- virtual SegmentMap **GetSegmentMap** ()=0
- virtual bool **isDataSupported** ()
- virtual int **GetNumData** ()=0
- virtual std::vector< std::string > **GetDataFieldList** ()=0
- virtual StringVariantMap **GetDataMap** ()=0
- virtual bool **isDataFieldPresent** (const std::string &strDataField)=0
- virtual Variant **GetDataValue** (const std::string &strDataField)=0

Static Public Attributes

- static const std::string **_CT** = ¨MOL2FileSource¨
- static const std::string **_TRIPOS_DELIM** = ¨@<TRIPOS>¨
- static const unsigned int _TRIPOS_DELIM_SIZE
- static const std::string **_IDS_MOL2_RECDELIM** = ¨@<TRIPOS>MOLECULE¨

Protected Member Functions

- virtual void Parse ()

Protected Member Functions inherited from rxdock::BaseMolecularFileSource

- void **ClearMolCache** ()
- void **RemoveAtom** (AtomPtr spAtom)
- void **RenumberAtomsAndBonds** ()
- void **SetupPartialIonicGroups** (AtomList &atoms, ParameterFileSourcePtr spParamSource)

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Friends

- void **to_json** (json &j, const MOL2FileSource &mol2FileSrc)
- void **from_json** (const json &j, MOL2FileSource &mol2FileSrc)

Additional Inherited Members

Protected Attributes inherited from rxdock::BaseMolecularFileSource

- std::vector< std::string > **m_titleList**
- AtomList **m_atomList**
- BondList **m_bondList**
- SegmentMap **m_segmentMap**
- StringVariantMap **m_dataMap**

Protected Attributes inherited from rxdock::BaseFileSource

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

5.207.1 Member Function Documentation

5.207.1.1 isAtomListSupported()

```
bool rxdock::MOL2FileSource::isAtomListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.207.1.2 isBondListSupported()

```
bool rxdock::MOL2FileSource::isBondListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.207.1.3 isCoordinatesSupported()

```
bool rxdock::MOL2FileSource::isCoordinatesSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.207.1.4 isDataSupported()

```
bool rxdock::MOL2FileSource::isDataSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.207.1.5 isTitleListSupported()

```
bool rxdock::MOL2FileSource::isTitleListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.207.1.6 Parse()

```
void MOL2FileSource::Parse (
            void  )  [protected], [virtual]
```

Implements rxdock::BaseFileSource.

5.207.2 Member Data Documentation

5.207.2.1 _TRIPOS_DELIM_SIZE

```
const unsigned int MOL2FileSource::_TRIPOS_DELIM_SIZE  [static]
```

Initial value:

```
=
    MOL2FileSource::_TRIPOS_DELIM.size()
```

The documentation for this class was generated from the following files:

- MOL2FileSource.h
- MOL2FileSource.cxx

5.208 rxdock::MOL2Substructure Class Reference

Public Member Functions

- **MOL2Substructure** (const std::string &name, int rr, const std::string &cc, const std::string &ss)
- std::string **GetName** () const
- int **GetRoot** () const
- std::string **GetChain** () const
- std::string **GetType** () const

Friends

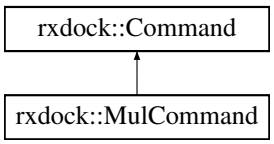
- void **to_json** (json &j, const MOL2Substructure &mol2Structure)
- void **from_json** (const json &j, MOL2Substructure &mol2Structure)

The documentation for this class was generated from the following file:

- MOL2FileSource.h

5.209 rxdock::MulCommand Class Reference

Inheritance diagram for rxdock::MulCommand:



Public Member Functions

- int GetNArgs ()
- std::string GetName ()
- void SetArg (int i, ReturnType f)
- void SetNameArg (int i, std::string &n)
- ReturnType Execute ()

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Additional Inherited Members

Static Public Member Functions inherited from rxdock::Command

- static void **Clear** ()

Static Public Attributes inherited from rxdock::Command

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions inherited from rxdock::Command

- std::string **tabs** (int n)

5.209.1 Member Function Documentation

5.209.1.1 Execute()

```
ReturnType rxdock::MulCommand::Execute ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.209.1.2 GetName()

```
std::string rxdock::MulCommand::GetName ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.209.1.3 GetNArgs()

```
int rxdock::MulCommand::GetNArgs ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.209.1.4 SetArg()

```
void rxdock::MulCommand::SetArg (
            int i,
            ReturnType f )  [inline], [virtual]
```

Implements rxdock::Command.

5.209.1.5 SetNameArg()

```
void rxdock::MulCommand::SetNameArg (
            int i,
            std::string & n )  [inline], [virtual]
```

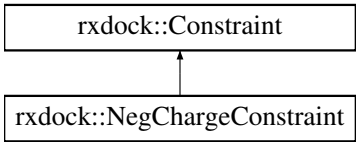
Implements rxdock::Command.

The documentation for this class was generated from the following file:

- Command.h

5.210 rxdock::NegChargeConstraint Class Reference

Inheritance diagram for rxdock::NegChargeConstraint:



Public Member Functions

- **NegChargeConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.210.1 Member Function Documentation

5.210.1.1 AddAtomList()

```
void NegChargeConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

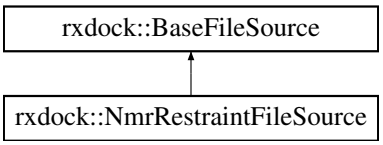
Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.211 rxdock::NmrRestraintFileSource Class Reference

Inheritance diagram for rxdock::NmrRestraintFileSource:



Public Member Functions

- **NmrRestraintFileSource** (const std::string &fileName)
- unsigned int **GetNumNoeRestraints** ()
- NoeRestraintNamesList **GetNoeRestraintList** ()
- unsigned int **GetNumStdRestraints** ()
- StdRestraintNamesList **GetStdRestraintList** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Friends

- void **to_json** (json &j, const NmrRestraintFileSource &nmrRestrainfFileSrc)
- void **from_json** (const json &j, NmrRestraintFileSource &nmrRestrainfFileSrc)

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Protected Attributes inherited from rxdock::BaseFileSource

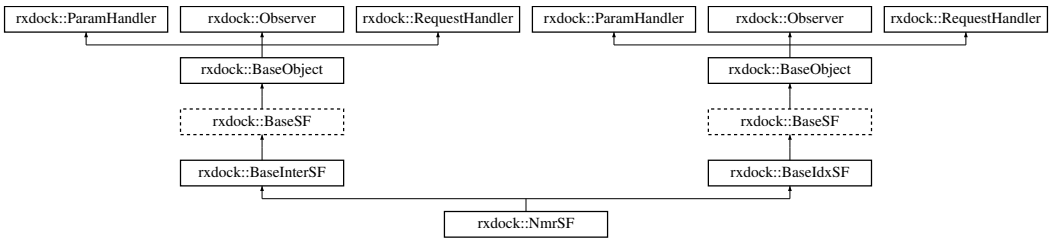
- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

The documentation for this class was generated from the following files:

- NmrRestraintFileSource.h
- NmrRestraintFileSource.cxx

5.212 rxdock::NmrSF Class Reference

Inheritance diagram for rxdock::NmrSF:



Public Member Functions

- **NmrSF** (const std::string &strName=¨nmr¨)

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseIdxSF

- double **GetGridStep** () const
- void **SetGridStep** (double step)
- double **GetBorder** () const
- void **SetBorder** (double border)

Static Public Attributes

- static const std::string **_CT** = ¨NmrSF¨
- static const std::string **_FILENAME** = ¨file-name¨
- static const std::string **_QUADRATIC** = ¨quadratic¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::BaseIdxSF

- static const std::string **_CT** = ¨BaseIdxSF¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_BORDER** = ¨border¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseIdxSF

- InteractionGridPtr **CreateInteractionGrid** () const
- NonBondedGridPtr **CreateNonBondedGrid** () const
- NonBondedHHSGridPtr **CreateNonBondedHHSGrid** () const
- double **GetMaxError** () const
- double **GetCorrectedRange** () const
- void **OwnParameterUpdated** (const std::string &strName)

5.212.1 Member Function Documentation

5.212.1.1 ParameterUpdated()

```
void NmrSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.212.1.2 RawScore()

```
double NmrSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.212.1.3 SetupLigand()

```
void NmrSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.212.1.4 SetupReceptor()

```
void NmrSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.212.1.5 SetupScore()

```
void NmrSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- NmrSF.h
- NmrSF.cxx

5.213 rxdock::NoeEndAtoms Class Reference

Public Member Functions

- **NoeEndAtoms** (const NoeEndNames &n, const AtomList &atomList)
- bool **isOK** () const

Public Attributes

- AtomList **atoms**
- eNoeType **type**

Friends

- void **to_json** (json &j, const NoeEndAtoms &noeEndAtoms)
- void **from_json** (const json &j, NoeEndAtoms &noeEndAtoms)

The documentation for this class was generated from the following files:

- NoeRestraint.h
- NoeRestraint.cxx

5.214 rxdock::NoeEndNames Class Reference

Public Member Functions

- bool **isOK** () const

Public Attributes

- std::vector< std::string > **names**
- eNoeType **type**

Friends

- void **to_json** (json &j, const NoeEndNames &noeEndNames)
- void **from_json** (const json &j, NoeEndNames &noeEndNames)

The documentation for this class was generated from the following files:

- NoeRestraint.h
- NoeRestraint.cxx

5.215 rxdock::NoeRestraintAtoms Class Reference

Public Member Functions

- **NoeRestraintAtoms** (const NoeRestraintNames &n, const AtomList &atomList)
- bool **isOK** () const
- bool **isSimple** () const

Public Attributes

- NoeEndAtoms **from**
- NoeEndAtoms **to**
- double **maxDist**

Friends

- void **to_json** (json &j, const NoeRestraintAtoms &noeRestraintAtoms)
- void **from_json** (const json &j, NoeRestraintAtoms &noeRestraintAtoms)

The documentation for this class was generated from the following files:

- NoeRestraint.h
- NoeRestraint.cxx

5.216 rxdock::NoeRestraintNames Class Reference

Public Member Functions

- bool **isOK** () const

Public Attributes

- NoeEndNames **from**
- NoeEndNames **to**
- double **maxDist**

Friends

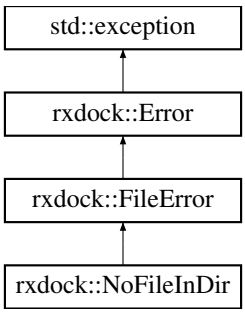
- void **to_json** (json &j, const NoeRestraintNames &noeRestraintNames)
- void **from_json** (const json &j, NoeRestraintNames &noeRestraintNames)

The documentation for this class was generated from the following files:

- NoeRestraint.h
- NoeRestraint.cxx

5.217 rxdock::NoFileInDir Class Reference

Inheritance diagram for rxdock::NoFileInDir:



Public Member Functions

- **NoFileInDir** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

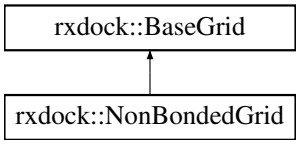
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.218 rxdock::NonBondedGrid Class Reference

Inheritance diagram for rxdock::NonBondedGrid:



Public Member Functions

- **NonBondedGrid** (const Coord &gridMin, const Coord &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **NonBondedGrid** (json j)
- **NonBondedGrid** (const NonBondedGrid &)
- **NonBondedGrid** (const BaseGrid &)
- NonBondedGrid & **operator=** (const NonBondedGrid &)
- NonBondedGrid & **operator=** (const BaseGrid &)
- virtual void Print (std::ostream &ostr) const
- const AtomRList & **GetAtomList** (unsigned int iXYZ) const
- const AtomRList & **GetAtomList** (const Coord &c) const
- void **SetAtomLists** (Atom *pAtom, double radius)
- void **ClearAtomLists** ()
- void **UniqueAtomLists** ()

Public Member Functions inherited from rxdock::BaseGrid

- **BaseGrid** (const Coord &gridMin, const Vector &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **BaseGrid** (json j)
- **BaseGrid** (const BaseGrid &)
- BaseGrid & **operator=** (const BaseGrid &)
- virtual void **Print** (std::ostream &ostr) const
- unsigned int **GetNX** () const
- unsigned int **GetNY** () const
- unsigned int **GetNZ** () const
- unsigned int **GetN** () const
- unsigned int **GetStrideX** () const
- unsigned int **GetStrideY** () const
- unsigned int **GetStrideZ** () const

- int **GetnXMin** () const
- int **GetnYMin** () const
- int **GetnZMin** () const
- int **GetnXMax** () const
- int **GetnYMax** () const
- int **GetnZMax** () const
- const Coord & **GetGridMin** () const
- const Coord & **GetGridMax** () const
- const Vector & **GetGridStep** () const
- Coord **GetGridCenter** () const
- Coord **GetGridSize** () const
- unsigned int **GetPad** () const
- const Coord & **GetPadMin** () const
- const Coord & **GetPadMax** () const
- void **SetGridMin** (const Coord &gridMin)
- void **TranslateGrid** (const Vector &vec)
- void **SetGridCenter** (const Coord &gridCenter)
- void **SetGridStep** (const Vector &gridStep)
- void **SetPad** (unsigned int NPad=0)
- bool **isValid** (const Coord &c) const
- bool **isValid** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- bool **isValid** (unsigned int iXYZ) const
- unsigned int **GetIX** (const Coord &c) const
- unsigned int **GetIY** (const Coord &c) const
- unsigned int **GetIZ** (const Coord &c) const
- unsigned int **GetIX** (double x) const
- unsigned int **GetIY** (double y) const
- unsigned int **GetIZ** (double z) const
- unsigned int **GetIX** (unsigned int iXYZ) const
- unsigned int **GetIY** (unsigned int iXYZ) const
- unsigned int **GetIZ** (unsigned int iXYZ) const
- unsigned int **GetIXYZ** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- unsigned int **GetIXYZ** (const Coord &c) const
- Coord **GetCoord** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetXCoord** (unsigned int iX) const
- double **GetYCoord** (unsigned int iY) const
- double **GetZCoord** (unsigned int iZ) const
- Coord **GetCoord** (unsigned int iXYZ) const
- CoordList **GetCoordList** (const std::set< unsigned int > &iXYZSet) const
- void **GetSphereIndices** (const Coord &c, double radius, std::vector< unsigned int > &sIndices) const

Static Public Attributes

- static const std::string **_CT** = ¨NonBondedGrid¨

Static Public Attributes inherited from rxdock::BaseGrid

- static const std::string **_CT** = ¨BaseGrid¨

Protected Member Functions

- void **OwnPrint** (std::ostream &ostr) const

Protected Member Functions inherited from rxdock::BaseGrid

- void **OwnPrint** (std::ostream &ostr) const

5.218.1 Member Function Documentation

5.218.1.1 Print()

```
void NonBondedGrid::Print (
            std::ostream & ostr ) const  [virtual]
```

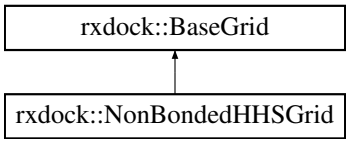
Reimplemented from rxdock::BaseGrid.

The documentation for this class was generated from the following files:

- NonBondedGrid.h
- NonBondedGrid.cxx

5.219 rxdock::NonBondedHHSGrid Class Reference

Inheritance diagram for rxdock::NonBondedHHSGrid:



Public Member Functions

- **NonBondedHHSGrid** (const Coord &gridMin, const Coord &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **NonBondedHHSGrid** (json j)
- **NonBondedHHSGrid** (const NonBondedHHSGrid &)
- **NonBondedHHSGrid** (const BaseGrid &)
- NonBondedHHSGrid & **operator=** (const NonBondedHHSGrid &)
- NonBondedHHSGrid & **operator=** (const BaseGrid &)
- virtual void Print (std::ostream &ostr) const
- const HHS_SolvationRList & **GetHHSList** (unsigned int iXYZ) const
- const HHS_SolvationRList & **GetHHSList** (const Coord &c) const
- void **SetHHSLists** (HHS_Solvation ∗pHHS, double radius)
- void **ClearHHSLists** (void)

Public Member Functions inherited from rxdock::BaseGrid

- **BaseGrid** (const Coord &gridMin, const Vector &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **BaseGrid** (json j)
- **BaseGrid** (const BaseGrid &)
- BaseGrid & **operator=** (const BaseGrid &)
- virtual void **Print** (std::ostream &ostr) const
- unsigned int **GetNX** () const
- unsigned int **GetNY** () const
- unsigned int **GetNZ** () const
- unsigned int **GetN** () const
- unsigned int **GetStrideX** () const
- unsigned int **GetStrideY** () const
- unsigned int **GetStrideZ** () const
- int **GetnXMin** () const
- int **GetnYMin** () const
- int **GetnZMin** () const
- int **GetnXMax** () const
- int **GetnYMax** () const
- int **GetnZMax** () const
- const Coord & **GetGridMin** () const
- const Coord & **GetGridMax** () const
- const Vector & **GetGridStep** () const
- Coord **GetGridCenter** () const
- Coord **GetGridSize** () const
- unsigned int **GetPad** () const
- const Coord & **GetPadMin** () const
- const Coord & **GetPadMax** () const
- void **SetGridMin** (const Coord &gridMin)
- void **TranslateGrid** (const Vector &vec)
- void **SetGridCenter** (const Coord &gridCenter)
- void **SetGridStep** (const Vector &gridStep)
- void **SetPad** (unsigned int NPad=0)
- bool **isValid** (const Coord &c) const
- bool **isValid** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- bool **isValid** (unsigned int iXYZ) const
- unsigned int **GetIX** (const Coord &c) const
- unsigned int **GetIY** (const Coord &c) const
- unsigned int **GetIZ** (const Coord &c) const
- unsigned int **GetIX** (double x) const
- unsigned int **GetIY** (double y) const
- unsigned int **GetIZ** (double z) const
- unsigned int **GetIX** (unsigned int iXYZ) const
- unsigned int **GetIY** (unsigned int iXYZ) const
- unsigned int **GetIZ** (unsigned int iXYZ) const
- unsigned int **GetIXYZ** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- unsigned int **GetIXYZ** (const Coord &c) const
- Coord **GetCoord** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetXCoord** (unsigned int iX) const
- double **GetYCoord** (unsigned int iY) const
- double **GetZCoord** (unsigned int iZ) const
- Coord **GetCoord** (unsigned int iXYZ) const
- CoordList **GetCoordList** (const std::set< unsigned int > &iXYZSet) const
- void **GetSphereIndices** (const Coord &c, double radius, std::vector< unsigned int > &sIndices) const

Static Public Attributes

- static const std::string **_CT** = ¨NonBondedHHSGrid¨

Static Public Attributes inherited from rxdock::BaseGrid

- static const std::string **_CT** = ¨BaseGrid¨

Protected Member Functions

- void **OwnPrint** (std::ostream &ostr) const

Protected Member Functions inherited from rxdock::BaseGrid

- void **OwnPrint** (std::ostream &ostr) const

5.219.1 Member Function Documentation

5.219.1.1 Print()

```
void NonBondedHHSGrid::Print (
            std::ostream & ostr ) const  [virtual]
```

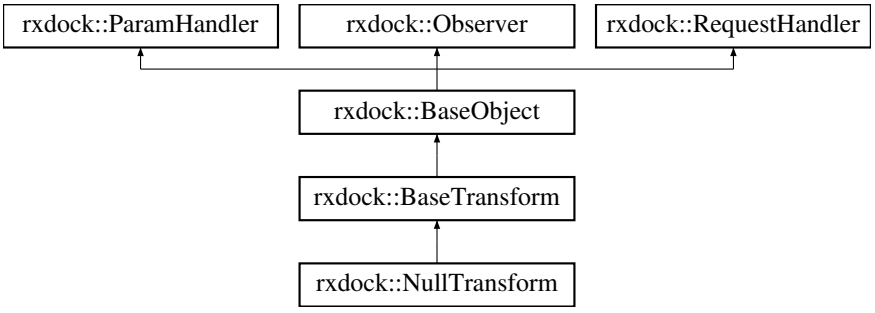
Reimplemented from rxdock::BaseGrid.

The documentation for this class was generated from the following files:

- NonBondedHHSGrid.h
- NonBondedHHSGrid.cxx

5.220 rxdock::NullTransform Class Reference

Inheritance diagram for rxdock::NullTransform:



Public Member Functions

- **NullTransform** (const std::string &strName="NULL")
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform *)
- virtual void **Remove** (BaseTransform *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform * **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform * **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨NullTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.220.1 Member Function Documentation

5.220.1.1 Execute()

```
void NullTransform::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.220.1.2 Update()

```
void NullTransform::Update (
            Subject * theChangedSubject )  [virtual]
```

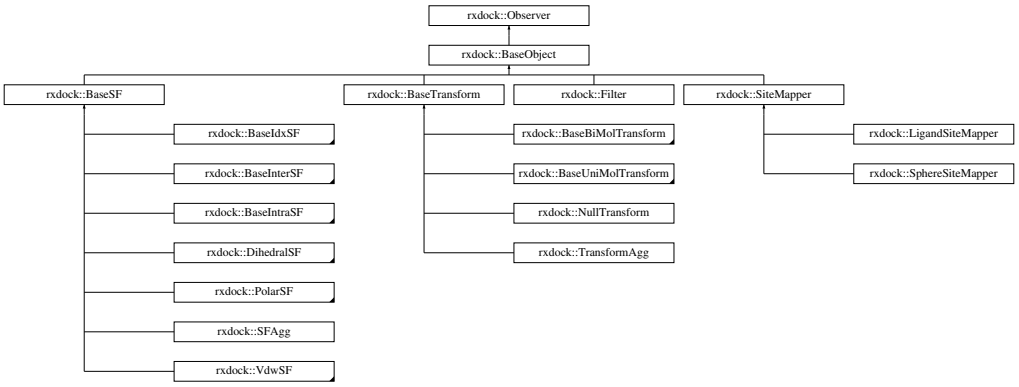
Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- NullTransform.h
- NullTransform.cxx

5.221 rxdock::Observer Class Reference

Inheritance diagram for rxdock::Observer:



Public Member Functions

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

5.221.1 Member Function Documentation

5.221.1.1 Update()

```
virtual void rxdock::Observer::Update (
            Subject * theChangedSubject )  [pure virtual]
```

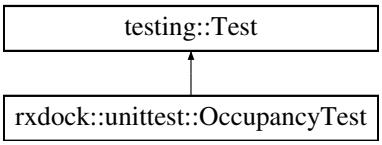
Implemented in rxdock::PMFIdxSF.

The documentation for this class was generated from the following files:

- Observer.h
- Observer.cxx

5.222 rxdock::unittest::OccupancyTest Class Reference

Inheritance diagram for rxdock::unittest::OccupancyTest:



Protected Member Functions

- void **SetUp** () override
- void **TearDown** () override
- double **CompareScoresForDisabledAndNoSolvent** ()
- double **CompareScoresForSolventModes** ()
- void **CheckFlexAtomFactory** (Model ∗pModel, int expected[9][3])

Protected Attributes

- BiMolWorkSpacePtr **m_workSpace**
- ModelList **m_solventList**

Static Protected Attributes

- static double **TINY** = 1E-4

The documentation for this class was generated from the following files:

- OccupancyTest.h
- OccupancyTest.cxx

5.223 rxdock::OverlapHHS Class Reference

Public Member Functions

- **OverlapHHS** (HHS_Solvation ∗pHHS, double p)
- void **operator()** (HHS_Solvation ∗pHHSj)

The documentation for this class was generated from the following file:

- SATypes.h

5.224 rxdock::OverlapVariableHHS Class Reference

Public Member Functions

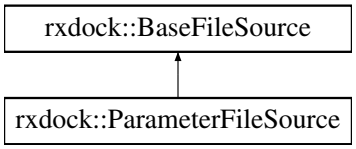
- void **operator()** (HHS_Solvation *pHHS)

The documentation for this class was generated from the following file:

- SATypes.h

5.225 rxdock::ParameterFileSource Class Reference

Inheritance diagram for rxdock::ParameterFileSource:



Public Member Functions

- **ParameterFileSource** (const char *fileName)
- RBTDLL_EXPORT **ParameterFileSource** (const std::string &fileName)
- RBTDLL_EXPORT std::string **GetTitle** ()
- std::string **GetVersion** ()
- unsigned int **GetNumParameters** ()
- RBTDLL_EXPORT std::vector< std::string > **GetParameterList** ()
- double **GetParameterValue** (const std::string &strParamName)
- RBTDLL_EXPORT std::string **GetParameterValueAsString** (const std::string &strParamName)
- bool **isParameterPresent** (const std::string &strParamName)
- int **GetNumSections** ()
- std::vector< std::string > **GetSectionList** ()
- std::string **GetSection** () const
- RBTDLL_EXPORT void **SetSection** (const std::string &strSection="")

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Friends

- void **to_json** (json &j, const ParameterFileSource &parameterFileSrc)
- void **from_json** (const json &j, ParameterFileSource &parameterFileSrc)

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Protected Attributes inherited from rxdock::BaseFileSource

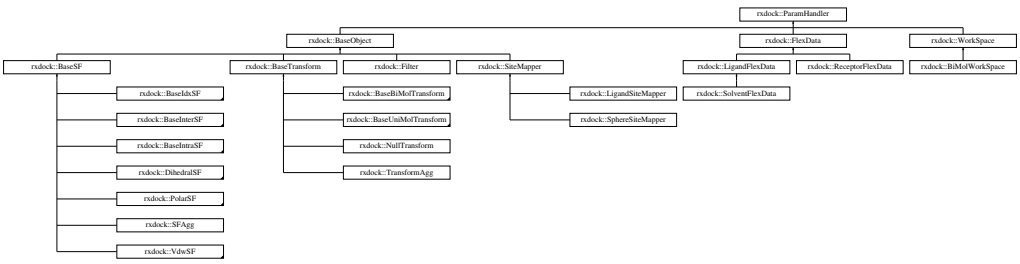
- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

The documentation for this class was generated from the following files:

- ParameterFileSource.h
- ParameterFileSource.cxx

5.226 rxdock::ParamHandler Class Reference

Inheritance diagram for rxdock::ParamHandler:



Public Member Functions

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Protected Member Functions

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const ParamHandler &paramHandler)
- void **from_json** (const json &j, ParamHandler &paramHandler)

The documentation for this class was generated from the following files:

- ParamHandler.h
- ParamHandler.cxx

5.227 rxdock::Parser Class Reference

Public Member Functions

- **Parser** (const Parser &)
- FilterExpressionPtr **Parse** (TokenIterPtr, ContextPtr)

Static Public Attributes

- static const std::string **_CT** = ¨Parser¨

The documentation for this class was generated from the following files:

- Parser.h
- Parser.cxx

5.228 rxdock::PartitionHHS Class Reference

Public Member Functions

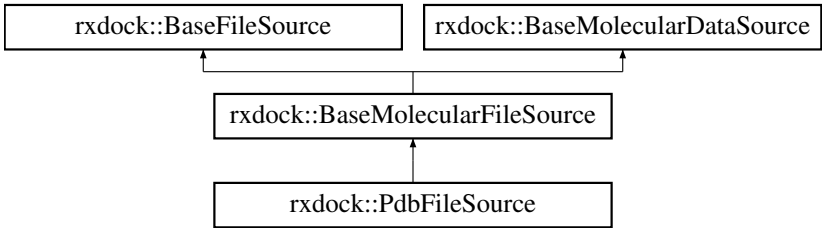
- **PartitionHHS** (double dd)
- void **operator()** (HHS_Solvation ∗pHHS)

The documentation for this class was generated from the following file:

- SATypes.h

5.229 rxdock::PdbFileSource Class Reference

Inheritance diagram for rxdock::PdbFileSource:



Public Member Functions

- **PdbFileSource** (const char *fileName)
- **PdbFileSource** (const std::string fileName)
- virtual bool isTitleListSupported ()
- virtual bool isAtomListSupported ()
- virtual bool isCoordinatesSupported ()
- virtual bool isBondListSupported ()

Public Member Functions inherited from rxdock::BaseMolecularFileSource

- RBTDLL_EXPORT **BaseMolecularFileSource** (const std::string &fileName, const std::string &sourceName)
- **BaseMolecularFileSource** (const std::string &fileName, const std::string &strRecDelim, const std::string &sourceName)
- void Reset ()
- int GetNumTitles ()
- int GetNumAtoms ()
- int GetNumBonds ()
- int GetNumSegments ()
- std::vector< std::string > GetTitleList ()
- AtomList GetAtomList ()
- BondList GetBondList ()
- SegmentMap GetSegmentMap ()
- virtual int GetNumData ()
- virtual std::vector< std::string > GetDataFieldList ()
- virtual StringVariantMap GetDataMap ()
- virtual bool isDataFieldPresent (const std::string &strDataField)
- virtual Variant GetDataValue (const std::string &strDataField)
- SegmentMap **GetSegmentFilterMap** ()
- RBTDLL_EXPORT void **SetSegmentFilterMap** (const SegmentMap &segmentFilterMap)
- void **ClearSegmentFilterMap** ()
- bool **isSegmentFilterMapDefined** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Public Member Functions inherited from rxdock::BaseMolecularDataSource

- **BaseMolecularDataSource** (std::string strName)
- std::string **GetName** ()
- virtual bool **isTitleListSupported** ()
- virtual bool **isAtomListSupported** ()
- virtual bool **isCoordinatesSupported** ()
- virtual bool **isBondListSupported** ()
- virtual void **Reset** ()=0
- virtual int **GetNumTitles** ()=0
- virtual int **GetNumAtoms** ()=0
- virtual int **GetNumBonds** ()=0
- virtual int **GetNumSegments** ()=0
- virtual std::vector< std::string > **GetTitleList** ()=0
- virtual AtomList **GetAtomList** ()=0
- virtual BondList **GetBondList** ()=0
- virtual SegmentMap **GetSegmentMap** ()=0
- virtual bool **isDataSupported** ()
- virtual int **GetNumData** ()=0
- virtual std::vector< std::string > **GetDataFieldList** ()=0
- virtual StringVariantMap **GetDataMap** ()=0
- virtual bool **isDataFieldPresent** (const std::string &strDataField)=0
- virtual Variant **GetDataValue** (const std::string &strDataField)=0

Protected Member Functions

- virtual void Parse ()

Protected Member Functions inherited from rxdock::BaseMolecularFileSource

- void **ClearMolCache** ()
- void **RemoveAtom** (AtomPtr spAtom)
- void **RenumberAtomsAndBonds** ()
- void **SetupPartiallonicGroups** (AtomList &atoms, ParameterFileSourcePtr spParamSource)

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Additional Inherited Members

Protected Attributes inherited from rxdock::BaseMolecularFileSource

- std::vector< std::string > **m_titleList**
- AtomList **m_atomList**
- BondList **m_bondList**
- SegmentMap **m_segmentMap**
- StringVariantMap **m_dataMap**

Protected Attributes inherited from rxdock::BaseFileSource

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

5.229.1 Member Function Documentation

5.229.1.1 isAtomListSupported()

```
virtual bool rxdock::PdbFileSource::isAtomListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.229.1.2 isBondListSupported()

```
virtual bool rxdock::PdbFileSource::isBondListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.229.1.3 isCoordinatesSupported()

```
virtual bool rxdock::PdbFileSource::isCoordinatesSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.229.1.4 isTitleListSupported()

```
virtual bool rxdock::PdbFileSource::isTitleListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.229.1.5 Parse()

```
void PdbFileSource::Parse (
            void  )  [protected], [virtual]
```

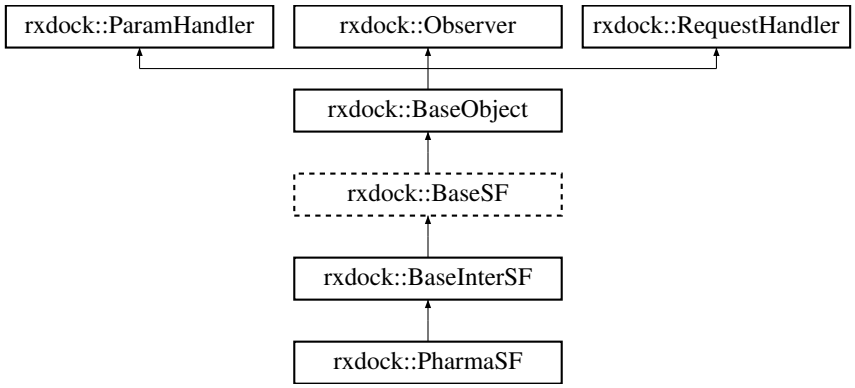
Implements rxdock::BaseFileSource.

The documentation for this class was generated from the following files:

- PdbFileSource.h
- PdbFileSource.cxx

5.230 rxdock::PharmaSF Class Reference

Inheritance diagram for rxdock::PharmaSF:



Public Member Functions

- **PharmaSF** (const std::string &strName="pharma")
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨PharmaSF¨
- static const std::string **_CONSTRAINTS_FILE** = ¨constraints-file¨
- static const std::string **_OPTIONAL_FILE** = ¨optional-file¨
- static const std::string **_NOPT** = ¨number-of-optional-parameters¨
- static const std::string **_WRITE_ERRORS** = ¨write-errors¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const PharmaSF &pharmaSF)
- void **from_json** (const json &j, PharmaSF &pharmaSF)

5.230.1 Member Function Documentation

5.230.1.1 ParameterUpdated()

```
void PharmaSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.230.1.2 RawScore()

```
double PharmaSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.230.1.3 ScoreMap()

```
void PharmaSF::ScoreMap (
            StringVariantMap & scoreMap ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.230.1.4 SetupLigand()

```
void PharmaSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.230.1.5 SetupReceptor()

```
void PharmaSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.230.1.6 SetupScore()

```
void PharmaSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- PharmaSF.h
- PharmaSF.cxx

5.231 rxdock::Plane Class Reference

Public Member Functions

- **Plane** (const Vector &v, double d)
- **Plane** (const Coord &c0, const Coord &c1, const Coord &c2)
- double **D** () const
- Vector **VNorm** () const
- void **Normalise** ()

Friends

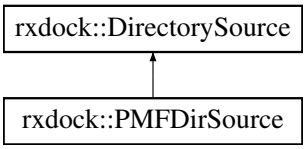
- void **to_json** (json &j, const Plane &plane)
- void **from_json** (const json &j, Plane &plane)
- std::ostream & **operator**<< (std::ostream &s, const Plane &plane)

The documentation for this class was generated from the following file:

- Plane.h

5.232 rxdock::PMFDirSource Class Reference

Inheritance diagram for rxdock::PMFDirSource:



Public Member Functions

- **PMFDirSource** (const std::string &)
- virtual void **ReadFiles** (std::vector< std::vector< PMFValue > > *aVect, std::vector< std::string > *a↩ NameVect, std::vector< PMFValue > *aSlopeVect)
- virtual void **ParseLines** (std::vector< std::string > anStrVect, std::vector< PMFValue > *aValueVect)

Public Member Functions inherited from rxdock::DirectorySource

- DirectorySource (const std::string &)

Additional Inherited Members

Static Public Attributes inherited from rxdock::DirectorySource

- static const std::string **_CT** = ¨DirectorySource¨

Protected Member Functions inherited from rxdock::DirectorySource

- void **CheckDirectory** (std::string &)

Protected Attributes inherited from rxdock::DirectorySource

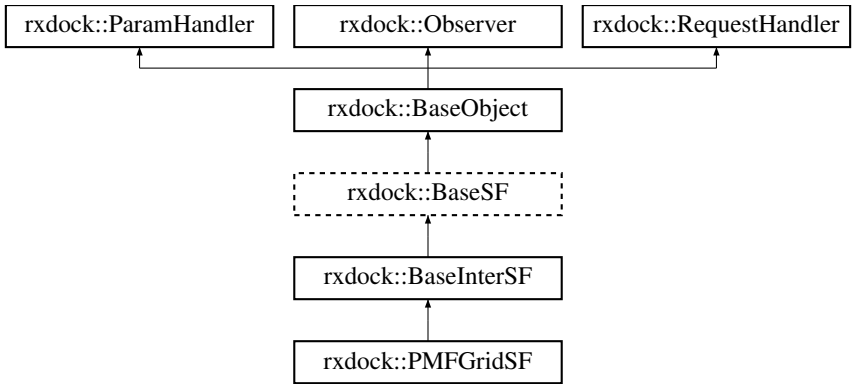
- struct stat **fStat**
- int **fNum**
- struct dirent ∗∗ **fNameList**
- std::ifstream **inFile**
- std::string **thePath**

The documentation for this class was generated from the following files:

- PMFDirSource.h
- PMFDirSource.cxx

5.233 rxdock::PMFGridSF Class Reference

Inheritance diagram for rxdock::PMFGridSF:



Public Member Functions

- **PMFGridSF** (const std::string &strName=¨pmf-grid¨)

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨PMFGridSF¨
- static const std::string **_GRID** = ¨grid¨
- static const std::string **_SMOOTHED** = ¨smoothed¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- unsigned int **GetCorrectedType** (PMFType aType) const

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const PMFGridSF &sf)
- void **from_json** (const json &j, PMFGridSF &sf)

5.233.1 Member Function Documentation

5.233.1.1 RawScore()

```
double PMFGridSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.233.1.2 SetupLigand()

```
void PMFGridSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.233.1.3 SetupReceptor()

```
void PMFGridSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.233.1.4 SetupScore()

```
virtual void rxdock::PMFGridSF::SetupScore ( )  [inline], [protected], [virtual]
```

Implements rxdock::BaseInterSF.

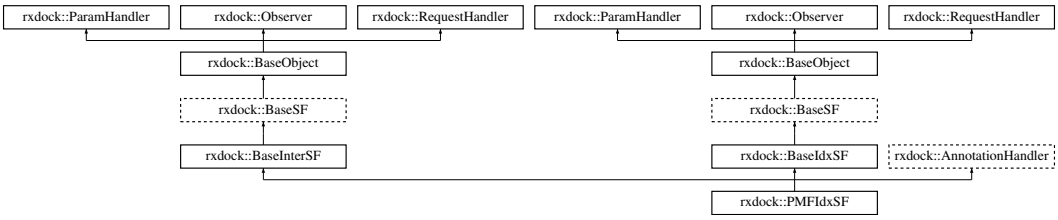
The documentation for this class was generated from the following files:

- PMFGridSF.h
- PMFGridSF.cxx

5.234 rxdock::PMFIdxSF Class Reference

`#include <PMFIdxSF.h>`

Inheritance diagram for rxdock::PMFIdxSF:



Public Member Functions

- PMFIdxSF (const std::string &strName="pmf")
- virtual ~PMFIdxSF ()
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseIdxSF

- double **GetGridStep** () const
- void **SetGridStep** (double step)
- double **GetBorder** () const
- void **SetBorder** (double border)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Attributes

- static const std::string _CT = ¨PMFIdxSF¨
- static const std::string _PMFDIR = ¨pmf-directory¨
- static const std::string **_CC_CUTOFF** = ¨C-C-cutoff¨
- static const std::string **_SLOPE** = ¨slope¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::BaseIdxSF

- static const std::string **_CT** = ¨BaseIdxSF¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_BORDER** = ¨border¨

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore (void) const
- double GetLinearCloseRangeValue (double aDist, PMFType aRecType, PMFType aLigType) const

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseIdxSF

- InteractionGridPtr **CreateInteractionGrid** () const
- NonBondedGridPtr **CreateNonBondedGrid** () const
- NonBondedHHSGridPtr **CreateNonBondedHHSGrid** () const
- double **GetMaxError** () const
- double **GetCorrectedRange** () const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

5.234.1 Detailed Description

PMFIdxSF class for PMF scoring.

5.234.2 Constructor & Destructor Documentation

5.234.2.1 PMFIdxSF()

```
PMFIdxSF::PMFIdxSF (
            const std::string & strName = ¨pmf¨ )
```

The only one constructor

5.234.2.2 ~PMFIdxSF()

```
PMFIdxSF::~PMFIdxSF ( )  [virtual]
```

The virtual destructor

5.234.3 Member Function Documentation

5.234.3.1 GetLinearCloseRangeValue()

```
double PMFIdxSF::GetLinearCloseRangeValue (
            double aDist,
            PMFType aRecType,
            PMFType aLigType ) const  [protected]
```

Estimate value for short distances instead of using plateau in PMFs

5.234.3.2 RawScore()

```
double PMFIdxSF::RawScore (
            void  ) const  [protected], [virtual]
```

RawScore(void) returning with the actual PMF score

Implements rxdock::BaseSF.

5.234.3.3 SetupLigand()

```
void PMFIdxSF::SetupLigand ( )  [protected], [virtual]
```

determine PMF types for all ligand atoms

Implements rxdock::BaseInterSF.

5.234.3.4 SetupReceptor()

```
void PMFIdxSF::SetupReceptor ( )  [protected], [virtual]
```

Setting up receptor means atom typing for all the receptor atoms and create NonBondedGrid for NN

SetupReceptor: determine PMF atom types for all receptor atoms create NonBondedGrid of nearest neighbours

Implements rxdock::BaseInterSF.

5.234.3.5 SetupScore()

```
void PMFIdxSF::SetupScore ( )  [protected], [virtual]
```

Setting up scores

Implements rxdock::BaseInterSF.

5.234.3.6 Update()

```
void PMFIdxSF::Update (
            Subject * theChangedSubject )  [virtual]
```

Observer functionalities pure virtual, so it must be overloaded.

Reimplemented from rxdock::BaseInterSF.

5.234.4 Member Data Documentation

5.234.4.1 _CT

```
const std::string PMFIdxSF::_CT = ¨PMFIdxSF¨  [static]
```

This string serves as a name for the weight factor of the PMF score. Should be read from the .prm file.

5.234.4.2 _PMFDIR

```
const std::string PMFIdxSF::_PMFDIR = ¨pmf-directory¨  [static]
```

String pointing to the dir with PMF tables

The documentation for this class was generated from the following files:

- PMFIdxSF.h
- PMFIdxSF.cxx

5.235 rxdock::PMFValue Struct Reference

```
#include <PMF.h>
```

Public Attributes

- float **distance**
- float **density**

5.235.1 Detailed Description

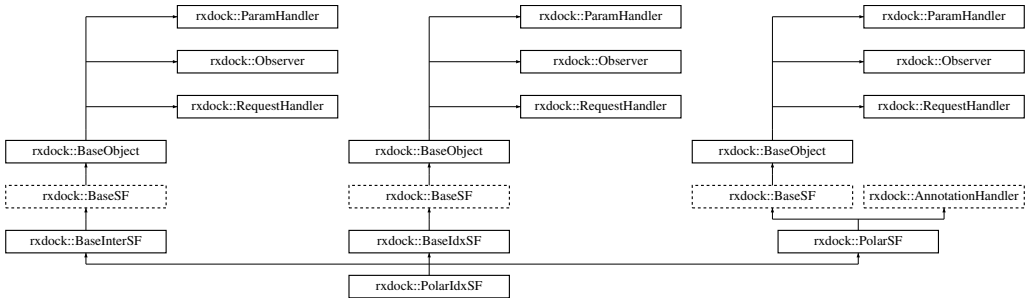
Basic data type to represent distance-value pairs of the PMF function

The documentation for this struct was generated from the following file:

- PMF.h

5.236 rxdock::PolarIdxSF Class Reference

Inheritance diagram for rxdock::PolarIdxSF:



Public Member Functions

- RBTDLL_EXPORT **PolarIdxSF** (const std::string &strName="polar")
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseIdxSF

- double **GetGridStep** () const
- void **SetGridStep** (double step)
- double **GetBorder** () const
- void **SetBorder** (double border)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Member Functions

- static RBTDLL_EXPORT const std::string & **GetIncr** ()

Static Public Attributes

- static const std::string **_CT** = ¨PolarIdxSF¨
- static const std::string **_INCR** = ¨increment¨
- static const std::string **_ATTR** = ¨attractive¨
- static const std::string **_THRESHOLD_POS** = ¨threshold-positive¨
- static const std::string **_THRESHOLD_NEG** = ¨threshold-negative¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::BaseIdxSF

- static const std::string **_CT** = ¨BaseIdxSF¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_BORDER** = ¨border¨

Static Public Attributes inherited from rxdock::PolarSF

- static const std::string **_CT** = ¨PolarSF¨
- static const std::string **_INCR**
- static const std::string **_R12FACTOR** = ¨r12-factor¨
- static const std::string **_R12INCR** = ¨r12-increment¨
- static const std::string **_DR12MIN** = ¨dr12-minimum¨
- static const std::string **_DR12MAX** = ¨dr12-maximum¨
- static const std::string **_A1** = ¨a1¨
- static const std::string **_DA1MIN** = ¨da1-minimum¨
- static const std::string **_DA1MAX** = ¨da1-maximum¨
- static const std::string **_A2** = ¨a2¨
- static const std::string **_DA2MIN** = ¨da2-minimum¨
- static const std::string **_DA2MAX** = ¨da2-maximum¨
- static const std::string **_INCMETAL** = ¨inc-metal¨
- static const std::string **_INCHBD** = ¨inc-HBD¨
- static const std::string **_INCHBA** = ¨inc-HBA¨

- static const std::string **_INCGUAN** = ¨inc-guan¨
- static const std::string **_GUAN_PLANE** = ¨guan-plane¨
- static const std::string **_ABS_DR12** = ¨abs-dr12¨
- static const std::string **_LP_OSP2** = ¨lp-osp2¨
- static const std::string **_LP_PHI** = ¨lp-phi¨
- static const std::string **_LP_DPHIMIN** = ¨lp-dphi-minimum¨
- static const std::string **_LP_DPHIMAX** = ¨lp-dphi-maximum¨
- static const std::string **_LP_DTHETAMIN** = ¨lp-dtheta-minimum¨
- static const std::string **_LP_DTHETAMAX** = ¨lp-dtheta-maximum¨

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void **ClearReceptor** ()
- void **ClearLigand** ()
- void **ClearSolvent** ()
- void **DeleteList** (InteractionCenterList &icList)
- void ParameterUpdated (const std::string &strName)

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseIdxSF

- InteractionGridPtr **CreateInteractionGrid** () const
- NonBondedGridPtr **CreateNonBondedGrid** () const
- NonBondedHHSGridPtr **CreateNonBondedHHSGrid** () const
- double **GetMaxError** () const
- double **GetCorrectedRange** () const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::PolarSF

- InteractionCenterList **CreateAcceptorInteractionCenters** (const AtomList &atomList) const
- InteractionCenterList **CreateDonorInteractionCenters** (const AtomList &atomList) const
- void **BuildIntraMap** (const InteractionCenterList &ICList1, const InteractionCenterList &ICList2, InteractionListMap &intns) const
- void **BuildIntraMap** (const InteractionCenterList &ICList, InteractionListMap &intns) const
- double **IntraScore** (const InteractionCenterList &posList, const InteractionCenterList &negList, const InteractionListMap &prtIntns, bool attr) const
- void **Partition** (const InteractionCenterList &posList, const InteractionCenterList &negList, const InteractionListMap &intns, InteractionListMap &prtIntns, double dist=0.0) const
- f1prms **GetRprms** () const
- f1prms **GetA1prms** () const
- f1prms **GetA2prms** () const
- double **PolarScore** (const InteractionCenter *intn, const InteractionCenterList &intnList, const f1prms &Rprms, const f1prms &A1prms, const f1prms &A2prms) const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

5.236.1 Member Function Documentation

5.236.1.1 ParameterUpdated()

```
void PolarIdxSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.236.1.2 RawScore()

```
double PolarIdxSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.236.1.3 ScoreMap()

```
void PolarIdxSF::ScoreMap (
            StringVariantMap & scoreMap ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.236.1.4 SetupLigand()

```
void PolarIdxSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.236.1.5 SetupReceptor()

```
void PolarIdxSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.236.1.6 SetupScore()

```
void PolarIdxSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.236.1.7 SetupSolvent()

```
void PolarIdxSF::SetupSolvent ( )  [protected], [virtual]
```

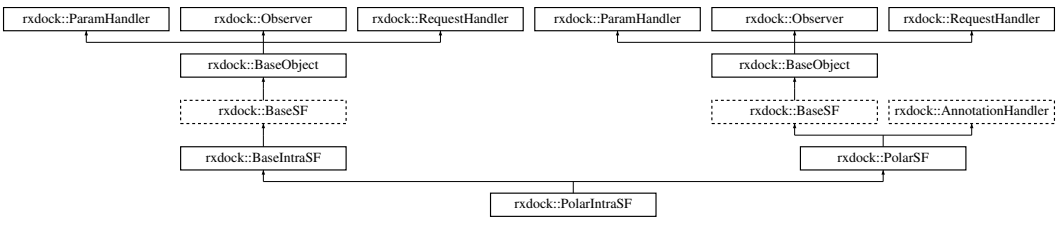
Reimplemented from rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- PolarIdxSF.h
- PolarIdxSF.cxx

5.237 rxdock::PolarIntraSF Class Reference

Inheritance diagram for rxdock::PolarIntraSF:



Public Member Functions

- **PolarIntraSF** (const std::string &strName="polar")

Public Member Functions inherited from rxdock::BaseIntraSF

- ModelPtr **GetLigand** () const
- virtual void Update (Subject *theChangedSubject)
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Attributes

- static const std::string **_CT** = "PolarIntraSF"
- static const std::string **_ATTR** = "attractive"

Static Public Attributes inherited from rxdock::BaseIntraSF

- static const std::string **_CT** = "BaseIntraSF"

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = "BaseSF"
- static const std::string **_WEIGHT** = "weight"
- static const std::string **_RANGE** = "range"
- static const std::string **_SYSTEM_SF** = "rxdock.score.system"
- static const std::string **_INTRA_SF** = "rxdock.score.intra"

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = "BaseObject"
- static const std::string **_CLASS** = "CLASS"
- static const std::string **_NAME** = "NAME"
- static const std::string **_ENABLED** = "ENABLED"

Static Public Attributes inherited from rxdock::PolarSF

- static const std::string **_CT** = "PolarSF"
- static const std::string **_INCR**
- static const std::string **_R12FACTOR** = "r12-factor"
- static const std::string **_R12INCR** = "r12-increment"
- static const std::string **_DR12MIN** = "dr12-minimum"
- static const std::string **_DR12MAX** = "dr12-maximum"
- static const std::string **_A1** = "a1"
- static const std::string **_DA1MIN** = "da1-minimum"
- static const std::string **_DA1MAX** = "da1-maximum"
- static const std::string **_A2** = "a2"
- static const std::string **_DA2MIN** = "da2-minimum"
- static const std::string **_DA2MAX** = "da2-maximum"
- static const std::string **_INCMETAL** = "inc-metal"
- static const std::string **_INCHBD** = "inc-HBD"
- static const std::string **_INCHBA** = "inc-HBA"
- static const std::string **_INCGUAN** = "inc-guan"
- static const std::string **_GUAN_PLANE** = "guan-plane"
- static const std::string **_ABS_DR12** = "abs-dr12"
- static const std::string **_LP_OSP2** = "lp-osp2"
- static const std::string **_LP_PHI** = "lp-phi"
- static const std::string **_LP_DPHIMIN** = "lp-dphi-minimum"
- static const std::string **_LP_DPHIMAX** = "lp-dphi-maximum"
- static const std::string **_LP_DTHETAMIN** = "lp-dtheta-minimum"
- static const std::string **_LP_DTHETAMAX** = "lp-dtheta-maximum"

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- virtual void SetupScore ()
- virtual double RawScore () const
- virtual void HandleRequest (RequestPtr spRequest)
- void **ClearModel** ()
- void ParameterUpdated (const std::string &strName)

- virtual void **SetupScore** ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::PolarSF

- InteractionCenterList **CreateAcceptorInteractionCenters** (const AtomList &atomList) const
- InteractionCenterList **CreateDonorInteractionCenters** (const AtomList &atomList) const
- void **BuildIntraMap** (const InteractionCenterList &ICList1, const InteractionCenterList &ICList2, InteractionListMap &intns) const
- void **BuildIntraMap** (const InteractionCenterList &ICList, InteractionListMap &intns) const
- double **IntraScore** (const InteractionCenterList &posList, const InteractionCenterList &negList, const InteractionListMap &prtIntns, bool attr) const
- void **Partition** (const InteractionCenterList &posList, const InteractionCenterList &negList, const InteractionListMap &intns, InteractionListMap &prtIntns, double dist=0.0) const
- f1prms **GetRprms** () const
- f1prms **GetA1prms** () const
- f1prms **GetA2prms** () const
- double **PolarScore** (const InteractionCenter *intn, const InteractionCenterList &intnList, const f1prms &Rprms, const f1prms &A1prms, const f1prms &A2prms) const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

Friends

- void **to_json** (json &j, const PolarIntraSF &polarIntraSF)
- void **from_json** (const json &j, PolarIntraSF &polarIntraSF)

5.237.1 Member Function Documentation

5.237.1.1 HandleRequest()

```
void PolarIntraSF::HandleRequest (
            RequestPtr spRequest )  [protected], [virtual]
```

Reimplemented from rxdock::BaseObject.

5.237.1.2 ParameterUpdated()

```
void PolarIntraSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.237.1.3 RawScore()

```
double PolarIntraSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.237.1.4 SetupScore()

```
void PolarIntraSF::SetupScore ( )  [protected], [virtual]
```

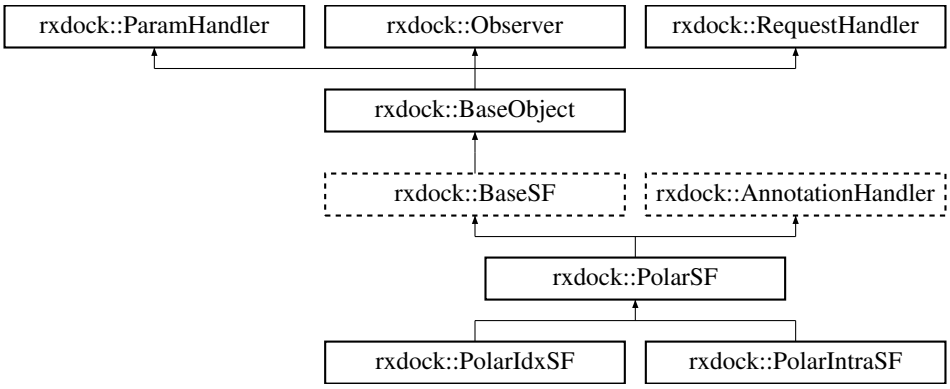
Implements rxdock::BaseIntraSF.

The documentation for this class was generated from the following files:

- PolarIntraSF.h
- PolarIntraSF.cxx

5.238 rxdock::PolarSF Class Reference

Inheritance diagram for rxdock::PolarSF:



Classes

- struct f1prms

Static Public Attributes

- static const std::string **_CT** = "PolarSF"
- static const std::string **_INCR**
- static const std::string **_R12FACTOR** = "r12-factor"
- static const std::string **_R12INCR** = "r12-increment"
- static const std::string **_DR12MIN** = "dr12-minimum"
- static const std::string **_DR12MAX** = "dr12-maximum"
- static const std::string **_A1** = "a1"
- static const std::string **_DA1MIN** = "da1-minimum"
- static const std::string **_DA1MAX** = "da1-maximum"
- static const std::string **_A2** = "a2"
- static const std::string **_DA2MIN** = "da2-minimum"
- static const std::string **_DA2MAX** = "da2-maximum"
- static const std::string **_INCMETAL** = "inc-metal"
- static const std::string **_INCHBD** = "inc-HBD"
- static const std::string **_INCHBA** = "inc-HBA"
- static const std::string **_INCGUAN** = "inc-guan"
- static const std::string **_GUAN_PLANE** = "guan-plane"
- static const std::string **_ABS_DR12** = "abs-dr12"
- static const std::string **_LP_OSP2** = "lp-osp2"
- static const std::string **_LP_PHI** = "lp-phi"
- static const std::string **_LP_DPHIMIN** = "lp-dphi-minimum"
- static const std::string **_LP_DPHIMAX** = "lp-dphi-maximum"
- static const std::string **_LP_DTHETAMIN** = "lp-dtheta-minimum"
- static const std::string **_LP_DTHETAMAX** = "lp-dtheta-maximum"

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- InteractionCenterList **CreateAcceptorInteractionCenters** (const AtomList &atomList) const
- InteractionCenterList **CreateDonorInteractionCenters** (const AtomList &atomList) const
- void **BuildIntraMap** (const InteractionCenterList &ICList1, const InteractionCenterList &ICList2, InteractionListMap &intns) const
- void **BuildIntraMap** (const InteractionCenterList &ICList, InteractionListMap &intns) const
- double **IntraScore** (const InteractionCenterList &posList, const InteractionCenterList &negList, const InteractionListMap &prtIntns, bool attr) const
- void **Partition** (const InteractionCenterList &posList, const InteractionCenterList &negList, const InteractionListMap &intns, InteractionListMap &prtIntns, double dist=0.0) const
- f1prms **GetRprms** () const
- f1prms **GetA1prms** () const
- f1prms **GetA2prms** () const
- double **PolarScore** (const InteractionCenter *intn, const InteractionCenterList &intnList, const f1prms &Rprms, const f1prms &A1prms, const f1prms &A2prms) const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

Friends

- void **to_json** (json &j, const PolarSF &polarSF)
- void **from_json** (const json &j, PolarSF &polarSF)

Additional Inherited Members

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Public Member Functions inherited from rxdock::Observer

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

The documentation for this class was generated from the following files:

- PolarSF.h
- PolarSF.cxx

5.239 rxdock::geneticprogram::PopCmp Class Reference

Public Member Functions

- bool **operator()** (const GPPopulationPtr &x, const GPPopulationPtr &y) const

The documentation for this class was generated from the following file:

- GPPopulation.h

5.240 rxdock::Population Class Reference

Public Member Functions

- RBTDLL_EXPORT **Population** (ChromElement *pChr, int size, BaseSF *pSF)
- int **GetMaxSize** () const
- int **GetActualSize** () const
- RBTDLL_EXPORT GenomePtr **Best** () const
- double **GetScoreMean** () const
- double **GetScoreVariance** () const
- const GenomeList & **GetGenomeList** () const
- BaseSF * **GetSF** () const
- void **SetSF** (BaseSF *pSF)
- RBTDLL_EXPORT void **GAstep** (int nReplicates, double relStepSize, double equalityThreshold, double pcross, bool xovermut, bool cmutate)
- RBTDLL_EXPORT GenomePtr **RouletteWheelSelect** () const
- void **Print** (std::ostream &) const

Static Public Attributes

- static const std::string **_CT** = ¨Population¨

Friends

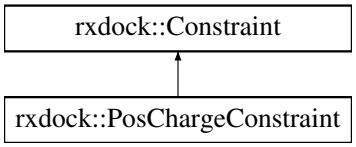
- void **to_json** (json &j, const Population &population)
- void **from_json** (const json &j, Population &population)
- std::ostream & **operator<<** (std::ostream &, const Population &)

The documentation for this class was generated from the following files:

- Population.h
- Population.cxx

5.241 rxdock::PosChargeConstraint Class Reference

Inheritance diagram for rxdock::PosChargeConstraint:



Public Member Functions

- **PosChargeConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.241.1 Member Function Documentation

5.241.1.1 AddAtomList()

```
void PosChargeConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

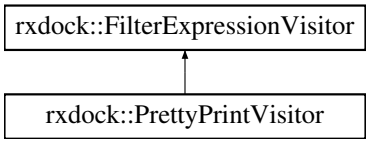
Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.242 rxdock::PrettyPrintVisitor Class Reference

Inheritance diagram for rxdock::PrettyPrintVisitor:



Public Member Functions

- **PrettyPrintVisitor** (ContextPtr)
- virtual void VisitVbleExp (FilterVbleExp ∗)
- virtual void VisitAddExp (FilterAddExp ∗)
- virtual void VisitSubExp (FilterSubExp ∗)
- virtual void VisitMulExp (FilterMulExp ∗)
- virtual void VisitDivExp (FilterDivExp ∗)
- virtual void VisitAndExp (FilterAndExp ∗)
- virtual void VisitLogExp (FilterLogExp ∗)
- virtual void VisitExpExp (FilterExpExp ∗)
- virtual void VisitIfExp (FilterIfExp ∗)

- virtual void **VisitVbleExp** (FilterVbleExp ∗)=0
- virtual void **VisitAddExp** (FilterAddExp ∗)=0
- virtual void **VisitSubExp** (FilterSubExp ∗)=0
- virtual void **VisitMulExp** (FilterMulExp ∗)=0
- virtual void **VisitDivExp** (FilterDivExp ∗)=0
- virtual void **VisitAndExp** (FilterAndExp ∗)=0
- virtual void **VisitLogExp** (FilterLogExp ∗)=0
- virtual void **VisitExpExp** (FilterExpExp ∗)=0
- virtual void **VisitIfExp** (FilterIfExp ∗)=0

5.242.1 Member Function Documentation

5.242.1.1 VisitAddExp()

```
void PrettyPrintVisitor::VisitAddExp (
            FilterAddExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.2 VisitAndExp()

```
void PrettyPrintVisitor::VisitAndExp (
            FilterAndExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.3 VisitDivExp()

```
void PrettyPrintVisitor::VisitDivExp (
            FilterDivExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.4 VisitExpExp()

```
void PrettyPrintVisitor::VisitExpExp (
            FilterExpExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.5 VisitIfExp()

```
void PrettyPrintVisitor::VisitIfExp (
            FilterIfExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.6 VisitLogExp()

```
void PrettyPrintVisitor::VisitLogExp (
            FilterLogExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.7 VisitMulExp()

```
void PrettyPrintVisitor::VisitMulExp (
            FilterMulExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.8 VisitSubExp()

```
void PrettyPrintVisitor::VisitSubExp (
            FilterSubExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.242.1.9 VisitVbleExp()

```
void PrettyPrintVisitor::VisitVbleExp (
            FilterVbleExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

The documentation for this class was generated from the following files:

- FilterExpressionVisitor.h
- FilterExpressionVisitor.cxx

5.243 rxdock::PrincipalAxes Class Reference

Public Attributes

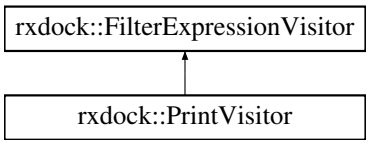
- Coord **com**
- Vector **axis1**
- Vector **axis2**
- Vector **axis3**
- double **moment1**
- double **moment2**
- double **moment3**

The documentation for this class was generated from the following file:

- PrincipalAxes.h

5.244 rxdock::PrintVisitor Class Reference

Inheritance diagram for rxdock::PrintVisitor:



Public Member Functions

- **PrintVisitor** (ContextPtr)
- virtual void VisitVbleExp (FilterVbleExp *)
- virtual void VisitAddExp (FilterAddExp *)
- virtual void VisitSubExp (FilterSubExp *)
- virtual void VisitMulExp (FilterMulExp *)
- virtual void VisitDivExp (FilterDivExp *)
- virtual void VisitAndExp (FilterAndExp *)
- virtual void VisitLogExp (FilterLogExp *)
- virtual void VisitExpExp (FilterExpExp *)
- virtual void VisitIfExp (FilterIfExp *)

- virtual void **VisitVbleExp** (FilterVbleExp *)=0
- virtual void **VisitAddExp** (FilterAddExp *)=0
- virtual void **VisitSubExp** (FilterSubExp *)=0
- virtual void **VisitMulExp** (FilterMulExp *)=0
- virtual void **VisitDivExp** (FilterDivExp *)=0
- virtual void **VisitAndExp** (FilterAndExp *)=0
- virtual void **VisitLogExp** (FilterLogExp *)=0
- virtual void **VisitExpExp** (FilterExpExp *)=0
- virtual void **VisitIfExp** (FilterIfExp *)=0

5.244.1 Member Function Documentation

5.244.1.1 VisitAddExp()

```
void PrintVisitor::VisitAddExp (
            FilterAddExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.2 VisitAndExp()

```
void PrintVisitor::VisitAndExp (
            FilterAndExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.3 VisitDivExp()

```
void PrintVisitor::VisitDivExp (
            FilterDivExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.4 VisitExpExp()

```
void PrintVisitor::VisitExpExp (
            FilterExpExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.5 VisitIfExp()

```
void PrintVisitor::VisitIfExp (
            FilterIfExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.6 VisitLogExp()

```
void PrintVisitor::VisitLogExp (
            FilterLogExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.7 VisitMulExp()

```
void PrintVisitor::VisitMulExp (
            FilterMulExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.8 VisitSubExp()

```
void PrintVisitor::VisitSubExp (
            FilterSubExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

5.244.1.9 VisitVbleExp()

```
void PrintVisitor::VisitVbleExp (
            FilterVbleExp * fe )  [virtual]
```

Implements rxdock::FilterExpressionVisitor.

The documentation for this class was generated from the following files:

- FilterExpressionVisitor.h
- FilterExpressionVisitor.cxx

5.245 rxdock::PRMFactory Class Reference

Public Member Functions

- RBTDLL_EXPORT **PRMFactory** (ParameterFileSource ∗pParamSource)
- RBTDLL_EXPORT **PRMFactory** (ParameterFileSource ∗pParamSource, DockingSite ∗pDS)
- DockingSite ∗ **GetDockingSite** () const
- void **SetDockingSite** (DockingSite ∗pDS)
- RBTDLL_EXPORT ModelPtr **CreateReceptor** ()
- RBTDLL_EXPORT ModelPtr **CreateLigand** (BaseMolecularFileSource ∗pSource)
- RBTDLL_EXPORT ModelList **CreateSolvent** ()

Static Public Attributes

- static const std::string & **_CT** = ¨PRMFactory¨
- static const std::string & **_REC_SECTION** = ¨receptor¨
- static const std::string & **_REC_FILE** = ¨file¨
- static const std::string & **_REC_TOPOL_FILE** = ¨topology-file¨
- static const std::string & **_REC_COORD_FILE** = ¨coordinate-file¨
- static const std::string & _REC_NUM_COORD_FILES
- static const std::string & **_REC_FLEX_DISTANCE** = ¨flexibility-distance¨
- static const std::string & _REC_DIHEDRAL_STEP
- static const std::string & **_REC_ALL_H** = ¨keep-all-hydrogens¨
- static const std::string & **_REC_MASSES_FILE** = ¨masses-file¨
- static const std::string & **_REC_SEGMENT_NAME** = ¨segment-name¨
- static const std::string & **_LIG_SECTION** = ¨ligand¨
- static const std::string & **_SOLV_SECTION** = ¨solvent¨
- static const std::string & **_SOLV_FILE** = ¨file¨

5.245.1 Member Data Documentation

5.245.1.1 _REC_DIHEDRAL_STEP

```
const std::string & PRMFactory::_REC_DIHEDRAL_STEP  [static]
```

Initial value:

```
=
    "dihedral-maximum-mutation-step"
```

5.245.1.2 _REC_NUM_COORD_FILES

```
const std::string & PRMFactory::_REC_NUM_COORD_FILES  [static]
```

Initial value:

```
=
    "number-of-coordinate-files"
```

The documentation for this class was generated from the following files:

- PRMFactory.h
- PRMFactory.cxx

5.246 rxdock::DihedralElement::prms Class Reference

Public Member Functions

- **prms** (double ss=3.0, double kk=0.2, double o=0.0)

Public Attributes

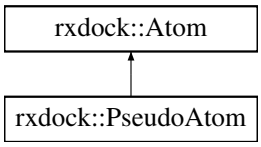
- double **s**
- double **k**
- double **sign**
- double **offset**

The documentation for this class was generated from the following file:

- DihedralSF.h

5.247 rxdock::PseudoAtom Class Reference

Inheritance diagram for rxdock::PseudoAtom:



Public Member Functions

- **PseudoAtom** (const AtomList &atomList, int nAtomId=-1, std::string strAtomName=¨PSEUDO¨)
- virtual std::ostream & Print (std::ostream &s) const
- unsigned int **GetNumAtoms** () const
- AtomList **GetAtomList** () const
- void **UpdateCoords** ()

Public Member Functions inherited from rxdock::Atom

- RBTDLL_EXPORT **Atom** (int nAtomId, int nAtomicNo=6, std::string strAtomName=¨C¨, std::string str↩ SubunitId=¨1¨, std::string strSubunitName=¨RES¨, std::string strSegmentName=¨SEG1¨, eHybridState e↩ State=UNDEFINED, unsigned int nHydrogens=0, int nFormalCharge=0)
- **Atom** (json j)
- **Atom** (const Atom &atom)
- Atom & **operator=** (const Atom &atom)
- virtual std::ostream & **Print** (std::ostream &s) const
- RBTDLL_EXPORT bool **GetEnabled** () const
- int **GetAtomId** () const
- void **SetAtomId** (const int nAtomId)
- int **GetAtomicNo** () const

- void **SetAtomicNo** (const int nAtomicNo)
- std::string **GetName** () const
- void **SetAtomName** (const std::string &strAtomName)
- RBTDLL_EXPORT std::string **GetFullAtomName** () const
- std::string **GetSubunitId** () const
- void **SetSubunitId** (const std::string &strSubunitId)
- std::string **GetSubunitName** () const
- void **SetSubunitName** (const std::string &strSubunitName)
- std::string **GetSegmentName** () const
- void **SetSegmentName** (const std::string &strSegmentName)
- eHybridState **GetHybridState** () const
- void **SetHybridState** (const eHybridState eState)
- unsigned int **GetNumImplicitHydrogens** () const
- void **SetNumImplicitHydrogens** (const unsigned int nHydrogens)
- int **GetFormalCharge** () const
- void **SetFormalCharge** (const int nFormalCharge)
- bool **GetCyclicFlag** () const
- void **SetCyclicFlag** (bool bCyclic=true)
- bool **GetSelectionFlag** () const
- void **SetSelectionFlag** (bool bSelected=true)
- void **InvertSelectionFlag** ()
- bool **GetUser1Flag** () const
- void **SetUser1Flag** (bool bUser1=true)
- double **GetUser1Value** () const
- void **SetUser1Value** (double dUser1=0.0)
- double **GetUser2Value** () const
- void **SetUser2Value** (double dUser2=0.0)
- Model * **GetModelPtr** () const
- void **SetModelPtr** (Model *pModel=nullptr)
- unsigned int **GetNumBonds** () const
- const BondMap & **GetBondMap** () const
- bool **AddBond** (Bond *pBond)
- bool **RemoveBond** (Bond *pBond)
- unsigned int **GetNumCyclicBonds** () const
- BondMap **GetCyclicBondMap** () const
- int **GetTotalFormalBondOrder** () const
- int **GetMaxFormalBondOrder** () const
- const Coord & **GetCoords** () const
- double **GetX** () const
- double **GetY** () const
- double **GetZ** () const
- void **SetCoords** (const Coord &coord)
- void **SetCoords** (const double x, const double y, const double z)
- void **SetX** (const double x)
- void **SetY** (const double y)
- void **SetZ** (const double z)
- double **GetPartialCharge** () const
- void **SetPartialCharge** (const double dPartialCharge)
- double **GetGroupCharge** () const
- void **SetGroupCharge** (const double dGroupCharge)
- double **GetAtomicMass** () const
- void **SetAtomicMass** (const double dAtomicMass)
- double **GetVdwRadius** () const
- void **SetVdwRadius** (const double dVdwRadius)
- std::string **GetFFType** () const

- void **SetFFType** (const std::string &strFFType)
- PMFType **GetPMFType** () const
- void **SetPMFType** (PMFType aType)
- TriposAtomType::eType **GetTriposType** () const
- void **SetTriposType** (TriposAtomType::eType aType)
- void **SaveCoords** (unsigned int coordNum=0)
- void **RevertCoords** (unsigned int coordNum=0)
- void **Translate** (const Vector &vector)
- void **Translate** (const double vx, const double vy, const double vz)
- void **RotateUsingQuat** (const Quat &q)
- unsigned int **GetCoordinationNumber** () const
- unsigned int **GetCoordinationNumber** (int nAtomicNo) const
- unsigned int **GetCoordinationNumber** (const std::string &strFFType) const
- unsigned int **GetCoordinationNumber** (eHybridState e) const

Friends

- void **to_json** (json &j, const PseudoAtom &pseudoAtom)
- void **from_json** (const json &j, PseudoAtom &pseudoAtom)

Additional Inherited Members

Public Types inherited from rxdock::Atom

- enum **eHybridState** {
  UNDEFINED = 0 , **SP** = 1 , **SP2** = 2 , **SP3** = 3 ,
  AROM = 4 , **TRI** = 5 }

5.247.1 Member Function Documentation

5.247.1.1 Print()

```
std::ostream & PseudoAtom::Print (
            std::ostream & s ) const  [virtual]
```

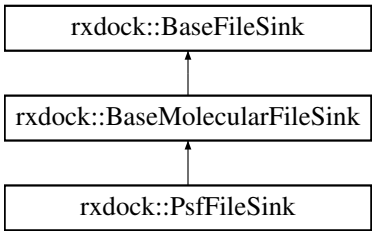
Reimplemented from rxdock::Atom.

The documentation for this class was generated from the following files:

- PseudoAtom.h
- PseudoAtom.cxx

5.248 rxdock::PsfFileSink Class Reference

Inheritance diagram for rxdock::PsfFileSink:



Public Member Functions

- RBTDLL_EXPORT **PsfFileSink** (const std::string &fileName, ModelPtr spModel)
- virtual void Render ()

Public Member Functions inherited from rxdock::BaseMolecularFileSink

- **BaseMolecularFileSink** (const std::string &fileName, ModelPtr spModel, bool bUseModelSegment↩ Names=true)
- virtual bool **isMultiConfSupported** ()
- ModelPtr **GetModel** () const
- RBTDLL_EXPORT void **SetModel** (ModelPtr spModel, bool bUseModelSegmentNames=false)
- void **SetNextSegmentId** (int nSegmentId)
- bool **GetMultiConf** () const
- void **SetMultiConf** (bool bMultiConf)
- void **WriteMultiConf** ()
- ModelList **GetSolvent** () const
- void **SetSolvent** (ModelList solventList)

Public Member Functions inherited from rxdock::BaseFileSink

- **BaseFileSink** (const std::string &fileName)
- std::string **GetFileName** () const
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- Error **Status** ()
- virtual void **Render** ()=0

Additional Inherited Members

Protected Member Functions inherited from rxdock::BaseMolecularFileSink

- void **Reset** ()

Protected Member Functions inherited from rxdock::BaseFileSink

- void **Write** (bool bClearCache=true)
- void **AddLine** (const std::string &fileRec)
- void **ReplaceLine** (const std::string &fileRec, unsigned int nRec)
- bool **isCacheEmpty** () const
- bool **GetAppend** () const
- void **SetAppend** (bool bAppend)

Protected Attributes inherited from rxdock::BaseMolecularFileSink

- int **m_nAtomId**
- int **m_nSubunitId**
- int **m_nSegmentId**
- bool **m_bUseModelSegmentNames**

5.248.1 Member Function Documentation

5.248.1.1 Render()

```
void PsfFileSink::Render ( )  [virtual]
```

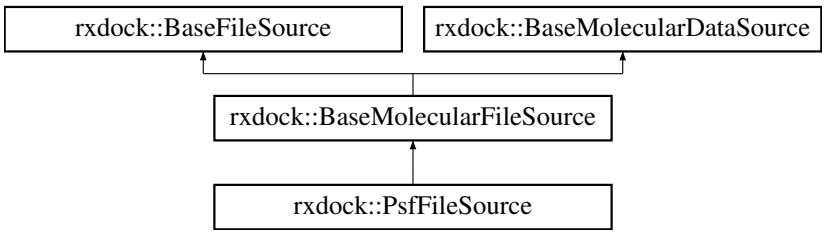
Implements rxdock::BaseFileSink.

The documentation for this class was generated from the following files:

- PsfFileSink.h
- PsfFileSink.cxx

5.249 rxdock::PsfFileSource Class Reference

Inheritance diagram for rxdock::PsfFileSource:



Public Member Functions

- **PsfFileSource** (const std::string &fileName, const std::string &strMassesFile=GetDataFileName(¨data¨, ¨masses.rtf¨), bool bImplHydrogens=true)
- virtual bool isTitleListSupported ()
- virtual bool isAtomListSupported ()
- virtual bool isCoordinatesSupported ()
- virtual bool isBondListSupported ()

Public Member Functions inherited from rxdock::BaseMolecularFileSource

- RBTDLL_EXPORT **BaseMolecularFileSource** (const std::string &fileName, const std::string &sourceName)
- **BaseMolecularFileSource** (const std::string &fileName, const std::string &strRecDelim, const std::string &sourceName)
- void Reset ()
- int GetNumTitles ()
- int GetNumAtoms ()
- int GetNumBonds ()
- int GetNumSegments ()
- std::vector< std::string > GetTitleList ()
- AtomList GetAtomList ()
- BondList GetBondList ()
- SegmentMap GetSegmentMap ()
- virtual int GetNumData ()
- virtual std::vector< std::string > GetDataFieldList ()
- virtual StringVariantMap GetDataMap ()
- virtual bool isDataFieldPresent (const std::string &strDataField)
- virtual Variant GetDataValue (const std::string &strDataField)
- SegmentMap **GetSegmentFilterMap** ()
- RBTDLL_EXPORT void **SetSegmentFilterMap** (const SegmentMap &segmentFilterMap)
- void **ClearSegmentFilterMap** ()
- bool **isSegmentFilterMapDefined** ()

Public Member Functions inherited from rxdock::BaseFileSource

- **BaseFileSource** (const std::string &fileName)
- **BaseFileSource** (const std::string &fileName, const std::string &strRecDelim)
- RBTDLL_EXPORT std::string **GetFileName** ()
- void **SetFileName** (const std::string &fileName)
- bool **StatusOK** ()
- RBTDLL_EXPORT Error **Status** ()
- RBTDLL_EXPORT bool **FileStatusOK** ()
- Error **FileStatus** ()
- bool **isMultiRecordSupported** ()
- RBTDLL_EXPORT void **NextRecord** ()
- void **Rewind** ()
- RBTDLL_EXPORT std::size_t **GetEstimatedNumRecords** ()

Public Member Functions inherited from rxdock::BaseMolecularDataSource

- **BaseMolecularDataSource** (std::string strName)
- std::string **GetName** ()
- virtual bool **isTitleListSupported** ()
- virtual bool **isAtomListSupported** ()
- virtual bool **isCoordinatesSupported** ()
- virtual bool **isBondListSupported** ()
- virtual void **Reset** ()=0
- virtual int **GetNumTitles** ()=0
- virtual int **GetNumAtoms** ()=0
- virtual int **GetNumBonds** ()=0
- virtual int **GetNumSegments** ()=0
- virtual std::vector< std::string > **GetTitleList** ()=0
- virtual AtomList **GetAtomList** ()=0
- virtual BondList **GetBondList** ()=0
- virtual SegmentMap **GetSegmentMap** ()=0
- virtual bool **isDataSupported** ()
- virtual int **GetNumData** ()=0
- virtual std::vector< std::string > **GetDataFieldList** ()=0
- virtual StringVariantMap **GetDataMap** ()=0
- virtual bool **isDataFieldPresent** (const std::string &strDataField)=0
- virtual Variant **GetDataValue** (const std::string &strDataField)=0

Protected Member Functions

- virtual void Parse ()

Protected Member Functions inherited from rxdock::BaseMolecularFileSource

- void **ClearMolCache** ()
- void **RemoveAtom** (AtomPtr spAtom)
- void **RenumberAtomsAndBonds** ()
- void **SetupPartiallonicGroups** (AtomList &atoms, ParameterFileSourcePtr spParamSource)

Protected Member Functions inherited from rxdock::BaseFileSource

- virtual void **Parse** ()=0
- void **Read** (bool aDelimiterAtEnd=true)

Additional Inherited Members

Protected Attributes inherited from rxdock::BaseMolecularFileSource

- std::vector< std::string > **m_titleList**
- AtomList **m_atomList**
- BondList **m_bondList**
- SegmentMap **m_segmentMap**
- StringVariantMap **m_dataMap**

Protected Attributes inherited from rxdock::BaseFileSource

- bool **m_bParsedOK**
- FileRecList **m_lineRecs**

5.249.1 Member Function Documentation

5.249.1.1 isAtomListSupported()

```
virtual bool rxdock::PsfFileSource::isAtomListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.249.1.2 isBondListSupported()

```
virtual bool rxdock::PsfFileSource::isBondListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.249.1.3 isCoordinatesSupported()

```
virtual bool rxdock::PsfFileSource::isCoordinatesSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.249.1.4 isTitleListSupported()

```
virtual bool rxdock::PsfFileSource::isTitleListSupported ( )  [inline], [virtual]
```

Reimplemented from rxdock::BaseMolecularDataSource.

5.249.1.5 Parse()

```
void PsfFileSource::Parse (
            void  )  [protected], [virtual]
```

Implements rxdock::BaseFileSource.

The documentation for this class was generated from the following files:

- PsfFileSource.h
- PsfFileSource.cxx

5.250 rxdock::Quat Class Reference

Public Member Functions

- **Quat** (double s1, double vx, double vy, double vz)
- **Quat** (double s1, const Vector &v1)
- **Quat** (const Vector &axis, double phi)
- **Quat** (const Quat &quat)
- Quat & **operator=** (const Quat &quat)
- void **operator+=** (const Quat &quat)
- void **operator-=** (const Quat &quat)
- void **operator*=** (const double &d)
- void **operator/=** (const double &d)
- double **Length** () const
- Quat **Unit** () const
- double **Dot** (const Quat &quat) const
- Quat **Conj** () const
- Coord **Rotate** (const Coord &w) const

Public Attributes

- double **s**
- Vector **v**

Friends

- void **to_json** (json &j, const Quat &quat)
- void **from_json** (const json &j, Quat &quat)
- std::ostream & **operator<<** (std::ostream &s, const Quat &quat)
- bool **operator==** (const Quat &quat1, const Quat &quat2)
- bool **operator!=** (const Quat &quat1, const Quat &quat2)
- Quat **operator+** (const Quat &quat1, const Quat &quat2)
- Quat **operator-** (const Quat &quat1, const Quat &quat2)
- Quat **operator-** (const Quat &quat)
- Quat **operator*** (const Quat &quat1, const Quat &quat2)
- Quat **operator*** (const Quat &quat, const double &d)
- Quat **operator*** (const double &d, const Quat &quat)
- Quat **operator/** (const Quat &quat, const double &d)

The documentation for this class was generated from the following file:

- Quat.h

5.251 rxdock::Rand Class Reference

Public Member Functions

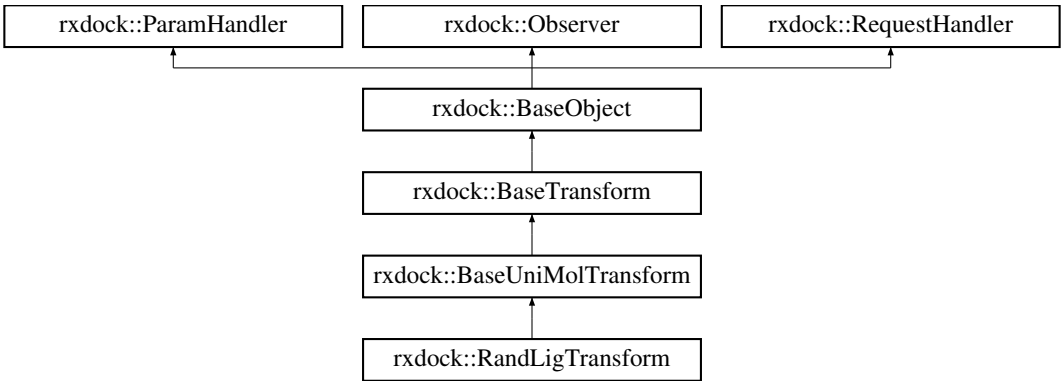
- RBTDLL_EXPORT void **Seed** (int seed=0)
- void **SeedFromRandomDevice** ()
- RBTDLL_EXPORT int **GetSeed** ()
- RBTDLL_EXPORT double **GetRandom01** ()
- int **GetRandomInt** (int nMax)
- Vector **GetRandomUnitVector** ()
- double **GetGaussianRandom** (double, double)
- double **GetCauchyRandom** (double, double)

The documentation for this class was generated from the following files:

- Rand.h
- Rand.cxx

5.252 rxdock::RandLigTransform Class Reference

Inheritance diagram for rxdock::RandLigTransform:



Public Member Functions

- **RandLigTransform** (const std::string &strName="RANDLIG")

Public Member Functions inherited from rxdock::BaseUniMolTransform

- ModelPtr **GetLigand** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform ∗)
- virtual void **Remove** (BaseTransform ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform ∗ **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform ∗ **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨RandLigTransform¨
- static const std::string **_TORS_STEP** = ¨torsion-step¨

Static Public Attributes inherited from rxdock::BaseUniMolTransform

- static const std::string **_CT** = ¨BaseUniMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupTransform ()
- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseUniMolTransform

- **BaseUniMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const RandLigTransform &randLigTrans)
- void **from_json** (const json &j, RandLigTransform &randLigTrans)

5.252.1 Member Function Documentation

5.252.1.1 Execute()

```
void RandLigTransform::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.252.1.2 SetupTransform()

```
void RandLigTransform::SetupTransform ( )  [protected], [virtual]
```

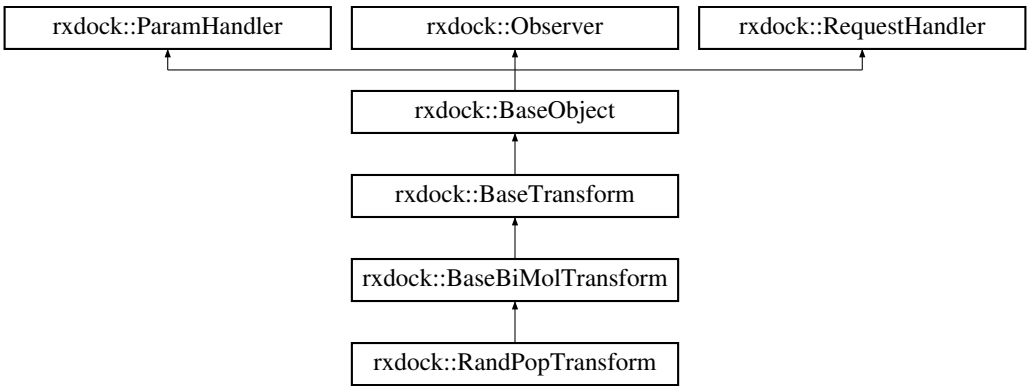
Implements rxdock::BaseUniMolTransform.

The documentation for this class was generated from the following files:

- RandLigTransform.h
- RandLigTransform.cxx

5.253 rxdock::RandPopTransform Class Reference

Inheritance diagram for rxdock::RandPopTransform:



Public Member Functions

- RBTDLL_EXPORT **RandPopTransform** (const std::string &strName=¨RANDPOP¨)

Public Member Functions inherited from rxdock::BaseBiMolTransform

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform ∗)
- virtual void **Remove** (BaseTransform ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform ∗ **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform ∗ **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨RandPopTransform¨
- static const std::string **_POP_SIZE** = ¨population-size¨
- static const std::string _SCALE_CHROM_LENGTH

Static Public Attributes inherited from rxdock::BaseBiMolTransform

- static const std::string **_CT** = ¨BaseBiMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupTransform ()
- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseBiMolTransform

- **BaseBiMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupReceptor** ()=0
- virtual void **SetupLigand** ()=0
- virtual void **SetupSolvent** ()
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.253.1 Member Function Documentation

5.253.1.1 Execute()

```
void RandPopTransform::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.253.1.2 SetupLigand()

```
void RandPopTransform::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.253.1.3 SetupReceptor()

```
void RandPopTransform::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.253.1.4 SetupSolvent()

```
void RandPopTransform::SetupSolvent ( )  [protected], [virtual]
```

Reimplemented from rxdock::BaseBiMolTransform.

5.253.1.5 SetupTransform()

```
void RandPopTransform::SetupTransform ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.253.2 Member Data Documentation

5.253.2.1 _SCALE_CHROM_LENGTH

```
const std::string RandPopTransform::_SCALE_CHROM_LENGTH  [static]
```

Initial value:

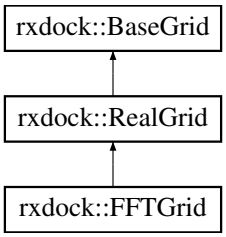
```
=
    "scale-chromosome-length"
```

The documentation for this class was generated from the following files:

- RandPopTransform.h
- RandPopTransform.cxx

5.254 rxdock::RealGrid Class Reference

Inheritance diagram for rxdock::RealGrid:



Public Member Functions

- RBTDLL_EXPORT **RealGrid** (const Coord &gridMin, const Coord &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- RBTDLL_EXPORT **RealGrid** (json j)
- **RealGrid** (const RealGrid &)
- **RealGrid** (const BaseGrid &)
- RealGrid & **operator=** (const RealGrid &)
- RealGrid & **operator=** (const BaseGrid &)
- virtual void Print (std::ostream &ostr) const
- float * **GetGridData** ()
- double **GetTolerance** () const
- void **SetTolerance** (double tol)
- double **GetValue** (const Coord &c) const
- double **GetValue** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetValue** (unsigned int iXYZ) const
- RBTDLL_EXPORT double **GetSmoothedValue** (const Coord &c) const
- void **SetValue** (const Coord &c, double val)
- void **SetValue** (unsigned int iX, unsigned int iY, unsigned int iZ, double val)
- void **SetValue** (unsigned int iXYZ, double val)
- RBTDLL_EXPORT void **SetAllValues** (double val)
- void **ReplaceValue** (double oldVal, double newVal)
- void **ReplaceValueRange** (double oldValMin, double oldValMax, double newVal)
- void **SetSphere** (const Coord &c, double radius, double val, bool bOverwrite=true)
- void **SetSurface** (const Coord &c, double innerRad, double outerRad, double val, bool bOverwrite=true)
- void **CreateSurface** (double oldVal, double adjVal, double newVal)
- bool **isValueWithinSphere** (const Coord &c, double radius, double val)
- void **SetAccessible** (double radius, double oldVal, double adjVal, double newVal, bool bCenterOnly=true)
- unsigned int **Count** (double val) const
- unsigned int **CountRange** (double valMin, double valMax) const
- double **MinValue** () const
- double **MaxValue** () const
- unsigned int **FindMinValue** () const
- unsigned int **FindMaxValue** () const
- RBTDLL_EXPORT void **PrintInsightGrid** (std::ostream &s) const

Public Member Functions inherited from rxdock::BaseGrid

- **BaseGrid** (const Coord &gridMin, const Vector &gridStep, unsigned int NX, unsigned int NY, unsigned int NZ, unsigned int NPad=0)
- **BaseGrid** (json j)
- **BaseGrid** (const BaseGrid &)
- BaseGrid & **operator=** (const BaseGrid &)
- virtual void **Print** (std::ostream &ostr) const
- unsigned int **GetNX** () const
- unsigned int **GetNY** () const
- unsigned int **GetNZ** () const
- unsigned int **GetN** () const
- unsigned int **GetStrideX** () const
- unsigned int **GetStrideY** () const
- unsigned int **GetStrideZ** () const
- int **GetnXMin** () const
- int **GetnYMin** () const
- int **GetnZMin** () const
- int **GetnXMax** () const
- int **GetnYMax** () const
- int **GetnZMax** () const
- const Coord & **GetGridMin** () const
- const Coord & **GetGridMax** () const
- const Vector & **GetGridStep** () const
- Coord **GetGridCenter** () const
- Coord **GetGridSize** () const
- unsigned int **GetPad** () const
- const Coord & **GetPadMin** () const
- const Coord & **GetPadMax** () const
- void **SetGridMin** (const Coord &gridMin)
- void **TranslateGrid** (const Vector &vec)
- void **SetGridCenter** (const Coord &gridCenter)
- void **SetGridStep** (const Vector &gridStep)
- void **SetPad** (unsigned int NPad=0)
- bool **isValid** (const Coord &c) const
- bool **isValid** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- bool **isValid** (unsigned int iXYZ) const
- unsigned int **GetIX** (const Coord &c) const
- unsigned int **GetIY** (const Coord &c) const
- unsigned int **GetIZ** (const Coord &c) const
- unsigned int **GetIX** (double x) const
- unsigned int **GetIY** (double y) const
- unsigned int **GetIZ** (double z) const
- unsigned int **GetIX** (unsigned int iXYZ) const
- unsigned int **GetIY** (unsigned int iXYZ) const
- unsigned int **GetIZ** (unsigned int iXYZ) const
- unsigned int **GetIXYZ** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- unsigned int **GetIXYZ** (const Coord &c) const
- Coord **GetCoord** (unsigned int iX, unsigned int iY, unsigned int iZ) const
- double **GetXCoord** (unsigned int iX) const
- double **GetYCoord** (unsigned int iY) const
- double **GetZCoord** (unsigned int iZ) const
- Coord **GetCoord** (unsigned int iXYZ) const
- CoordList **GetCoordList** (const std::set< unsigned int > &iXYZSet) const
- void **GetSphereIndices** (const Coord &c, double radius, std::vector< unsigned int > &sIndices) const

Static Public Attributes

- static const std::string **_CT** = ¨RealGrid¨

Static Public Attributes inherited from rxdock::BaseGrid

- static const std::string **_CT** = ¨BaseGrid¨

Protected Member Functions

- void **OwnPrint** (std::ostream &ostr) const

Protected Member Functions inherited from rxdock::BaseGrid

- void **OwnPrint** (std::ostream &ostr) const

Friends

- RBTDLL_EXPORT friend void **to_json** (json &j, const RealGrid &grid)
- void **from_json** (const json &j, RealGrid &grid)

5.254.1 Member Function Documentation

5.254.1.1 Print()

```
void RealGrid::Print (
            std::ostream & ostr ) const  [virtual]
```

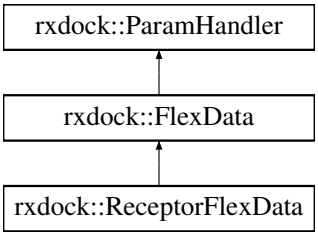
Reimplemented from rxdock::BaseGrid.

The documentation for this class was generated from the following files:

- RealGrid.h
- RealGrid.cxx

5.255 rxdock::ReceptorFlexData Class Reference

Inheritance diagram for rxdock::ReceptorFlexData:



Public Member Functions

- RBTDLL_EXPORT **ReceptorFlexData** (DockingSite *pDockSite)
- virtual void Accept (FlexDataVisitor &v)

Public Member Functions inherited from rxdock::FlexData

- Model * **GetModel** () const
- void **SetModel** (Model *pModel)
- DockingSite * **GetDockingSite** () const
- virtual void **Accept** (FlexDataVisitor &)=0

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Static Public Attributes

- static const std::string & **_FLEX_DISTANCE** = ¨flexibility-distance¨
- static const std::string & **_DIHEDRAL_STEP** = ¨dihedral-step¨

Additional Inherited Members

Protected Member Functions inherited from rxdock::FlexData

- **FlexData** (DockingSite *pDockSite)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.255.1 Member Function Documentation

5.255.1.1 Accept()

```
virtual void rxdock::ReceptorFlexData::Accept (
            FlexDataVisitor & v )  [inline], [virtual]
```

Implements rxdock::FlexData.

The documentation for this class was generated from the following files:

- ReceptorFlexData.h
- ReceptorFlexData.cxx

5.256 rxdock::neldermead::RelativeValueCriterion< DataType > Class Template Reference

Public Member Functions

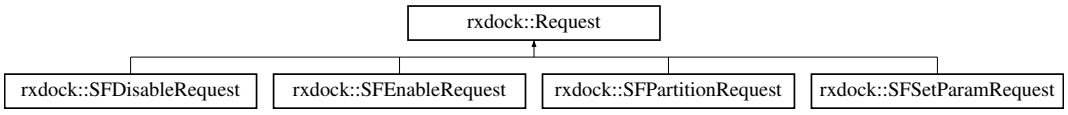
- **RelativeValueCriterion** (DataType ftol)
- template<class State >
  bool **operator()** (const State &state)

The documentation for this class was generated from the following file:

- NMCriteria.h

5.257 rxdock::Request Class Reference

Inheritance diagram for rxdock::Request:



Public Member Functions

- **Request** (json j)
- RequestID **GetID** () const
- unsigned int **GetNumParameters** () const
- VariantList **GetParameters** () const
- **Request** (RequestID id)
- void **AddParameter** (const Variant &v)

Friends

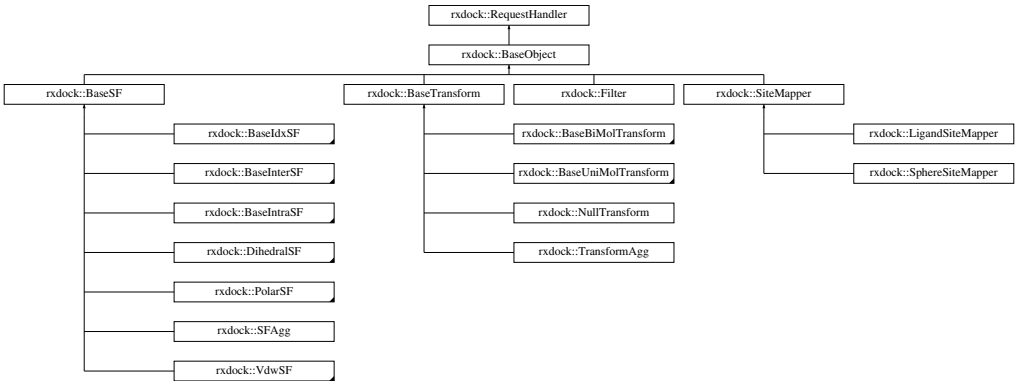
- void **to_json** (json &j, const Request &request)
- void **from_json** (const json &j, Request &request)

The documentation for this class was generated from the following file:

- Request.h

5.258 rxdock::RequestHandler Class Reference

Inheritance diagram for rxdock::RequestHandler:



Public Member Functions

- virtual void **HandleRequest** (RequestPtr spRequest)

The documentation for this class was generated from the following file:

- RequestHandler.h

5.259 rxdock::RestoreHHS Class Reference

Public Member Functions

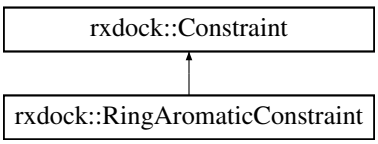
- void **operator()** (HHS_Solvation ∗pHHS)

The documentation for this class was generated from the following file:

- SATypes.h

5.260 rxdock::RingAromaticConstraint Class Reference

Inheritance diagram for rxdock::RingAromaticConstraint:



Public Member Functions

- **RingAromaticConstraint** (Coord c, double t)
- void AddAtomList (ModelPtr, bool bCheck=true)

Public Member Functions inherited from rxdock::Constraint

- **Constraint** (Coord, double)
- **Constraint** (const Constraint &)
- **Constraint** (json j)
- void **copy** (const Constraint &)
- std::ostream & **Print** (std::ostream &) const
- Coord **GetCoords** () const
- double **GetTolerance** () const
- virtual void **AddAtomList** (ModelPtr, bool bCheck=true)=0
- double **Score** () const

Static Public Attributes

- static unsigned int **counter** = 0

Static Public Attributes inherited from rxdock::Constraint

- static const std::string **_CT** = ¨Constraint¨

Additional Inherited Members

Protected Attributes inherited from rxdock::Constraint

- Coord **coord**
- double **tolerance**
- AtomList **m_atomList**

5.260.1 Member Function Documentation

5.260.1.1 AddAtomList()

```
void RingAromaticConstraint::AddAtomList (
            ModelPtr lig,
            bool bCheck = true )  [virtual]
```

Implements rxdock::Constraint.

The documentation for this class was generated from the following files:

- Constraint.h
- Constraint.cxx

5.261 rxdock::RotateAtomUsingQuat Class Reference

Public Member Functions

- **RotateAtomUsingQuat** (const Quat &qq)
- void **operator()** (Atom *pAtom)

The documentation for this class was generated from the following file:

- Atom.h

5.262 rxdock::RotateAtomUsingQuatIfSelected Class Reference

Public Member Functions

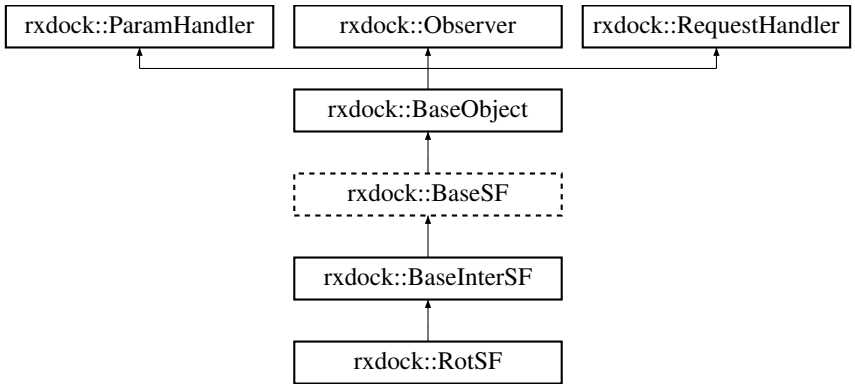
- **RotateAtomUsingQuatIfSelected** (const Quat &qq)
- void **operator()** (Atom *pAtom)

The documentation for this class was generated from the following file:

- Atom.h

5.263 rxdock::RotSF Class Reference

Inheritance diagram for rxdock::RotSF:



Public Member Functions

- **RotSF** (const std::string &strName="rot")

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨RotSF¨
- static const std::string **_INCNH3** = ¨inc-NH3¨
- static const std::string **_INCOH** = ¨inc-OH¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const RotSF &rotsf)
- void **from_json** (const json &j, RotSF &rotsf)

5.263.1 Member Function Documentation

5.263.1.1 ParameterUpdated()

```
void RotSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.263.1.2 RawScore()

```
double RotSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.263.1.3 SetupLigand()

```
void RotSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.263.1.4 SetupReceptor()

```
void RotSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.263.1.5 SetupScore()

```
void RotSF::SetupScore ( )  [protected], [virtual]
```

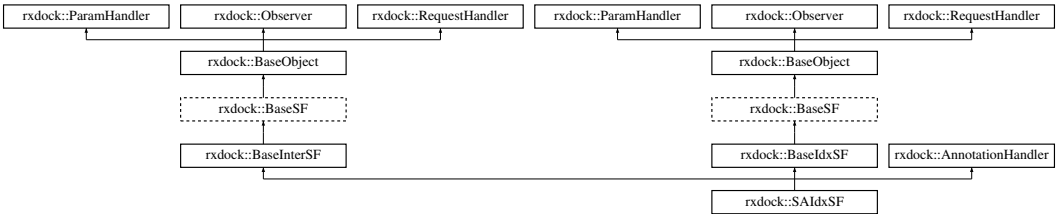
Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- RotSF.h
- RotSF.cxx

5.264 rxdock::SAIdxSF Class Reference

Inheritance diagram for rxdock::SAIdxSF:



Public Member Functions

- RBTDLL_EXPORT **SAIdxSF** (const std::string &strName="solv")
- virtual void ScoreMap (StringVariantMap &scoreMap) const
- virtual void HandleRequest (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseIdxSF

- double **GetGridStep** () const
- void **SetGridStep** (double step)
- double **GetBorder** () const
- void **SetBorder** (double border)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Attributes

- static const std::string **_CT** = ¨SAIdxSF¨
- static const std::string **_INCR** = ¨incr¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::BaseIdxSF

- static const std::string **_CT** = ¨BaseIdxSF¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_BORDER** = ¨border¨

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void SetupScore ()
- virtual double RawScore (void) const
- void **ClearReceptor** (void)
- void **ClearLigand** (void)
- void **ClearSolvent** (void)
- double **GetASP** (HHSType::eType, double) const
- double **GetP_i** (HHSType::eType) const
- double **GetR_i** (HHSType::eType) const
- void **PrintWeightMatrix** () const

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseIdxSF

- InteractionGridPtr **CreateInteractionGrid** () const
- NonBondedGridPtr **CreateNonBondedGrid** () const
- NonBondedHHSGridPtr **CreateNonBondedHHSGrid** () const
- double **GetMaxError** () const
- double **GetCorrectedRange** () const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

5.264.1 Member Function Documentation

5.264.1.1 HandleRequest()

```
void SAIdxSF::HandleRequest (
            RequestPtr spRequest )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.264.1.2 RawScore()

```
double SAIdxSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.264.1.3 ScoreMap()

```
void SAIdxSF::ScoreMap (
            StringVariantMap & scoreMap ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.264.1.4 SetupLigand()

```
void SAIdxSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.264.1.5 SetupReceptor()

```
void SAIdxSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.264.1.6 SetupScore()

```
void SAIdxSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.264.1.7 SetupSolvent()

```
void SAIdxSF::SetupSolvent ( )  [protected], [virtual]
```

Reimplemented from rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- SAIdxSF.h
- SAIdxSF.cxx

5.265 rxdock::SaveHHS Class Reference

Public Member Functions

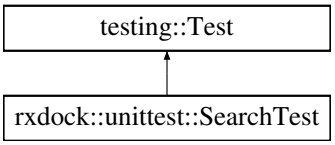
- void **operator()** (HHS_Solvation *pHHS)

The documentation for this class was generated from the following file:

- SATypes.h

5.266 rxdock::unittest::SearchTest Class Reference

Inheritance diagram for rxdock::unittest::SearchTest:



Protected Member Functions

- void **SetUp** () override
- void **TearDown** () override
- double **rmsd** (const CoordList &rc, const CoordList &c)

Protected Attributes

- AtomList **m_atomList**
- BiMolWorkSpacePtr **m_workSpace**
- SFAggPtr **m_SF**

The documentation for this class was generated from the following files:

- SearchTest.h
- SearchTest.cxx

5.267 rxdock::SelectAtom Class Reference

Public Member Functions

- **SelectAtom** (bool bb)
- void **operator()** (Atom ∗pAtom)

The documentation for this class was generated from the following file:

- Atom.h

5.268 rxdock::SelectFlexAtoms Class Reference

Public Member Functions

- void **operator()** (Atom ∗pAtom)

The documentation for this class was generated from the following files:

- Atom.h
- Atom.cxx

5.269 rxdock::SelectInteractionCenter Class Reference

Public Member Functions

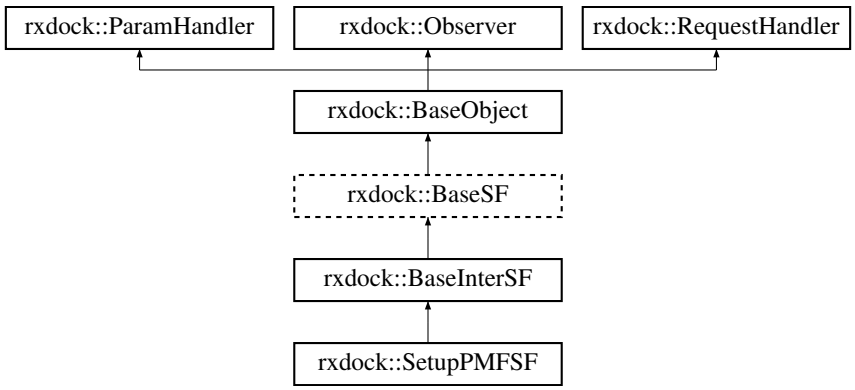
- **SelectInteractionCenter** (bool bb)
- void **operator()** (InteractionCenter *pIC)

The documentation for this class was generated from the following files:

- InteractionGrid.h
- InteractionGrid.cxx

5.270 rxdock::SetupPMFSF Class Reference

Inheritance diagram for rxdock::SetupPMFSF:



Public Member Functions

- **SetupPMFSF** (const std::string &strName="setup-pmf")

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨SetupPMFSF¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void **SetupReceptorPMFTypes** (void)
- void **SetupLigandPMFTypes** (void)
- PMFType GetPMFfor_IC (AtomPtr)
- PMFType **GetPMFfor_IN** (AtomPtr)
- PMFType **GetPMFfor_IO** (AtomPtr)
- PMFType **GetPMFfor_IS** (AtomPtr)
- PMFType **GetPMFfor_rC** (AtomPtr)
- PMFType **GetPMFfor_rN** (AtomPtr)
- PMFType **GetPMFfor_rO** (AtomPtr)
- PMFType **GetPMFfor_rS** (AtomPtr)
- bool **IsChargedNitrogen** (AtomPtr)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const SetupPMFSF &pmfsf)
- void **from_json** (const json &j, SetupPMFSF &pmfsf)

5.270.1 Member Function Documentation

5.270.1.1 GetPMFfor_lC()

```
PMFType SetupPMFSF::GetPMFfor_lC (
            AtomPtr anAtom )  [protected]
```

Routines to figure out complex PMF types

5.270.1.2 RawScore()

```
double SetupPMFSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.270.1.3 SetupLigand()

```
void SetupPMFSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.270.1.4 SetupReceptor()

```
void SetupPMFSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.270.1.5 SetupScore()

```
void SetupPMFSF::SetupScore ( )  [protected], [virtual]
```

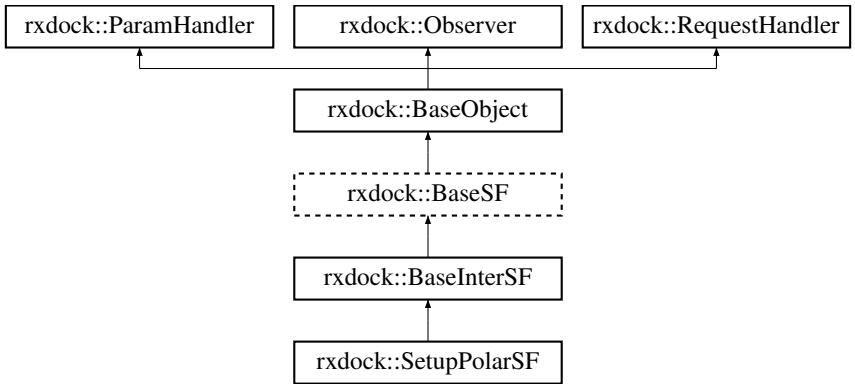
Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- SetupPMFSF.h
- SetupPMFSF.cxx

5.271 rxdock::SetupPolarSF Class Reference

Inheritance diagram for rxdock::SetupPolarSF:



Public Member Functions

- RBTDLL_EXPORT **SetupPolarSF** (const std::string &strName=¨setup-polar¨)

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨SetupPolarSF¨
- static const std::string **_RADIUS** = ¨radius¨
- static const std::string **_NORM** = ¨norm¨
- static const std::string **_POWER** = ¨power¨
- static const std::string **_CHGFACTOR** = ¨charge-factor¨
- static const std::string **_GUANFACTOR** = ¨guanidine-factor¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void SetupScore ()
- virtual double RawScore () const

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.271.1 Member Function Documentation

5.271.1.1 RawScore()

```
double SetupPolarSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.271.1.2 SetupLigand()

```
void SetupPolarSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.271.1.3 SetupReceptor()

```
void SetupPolarSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.271.1.4 SetupScore()

```
void SetupPolarSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.271.1.5 SetupSolvent()

```
void SetupPolarSF::SetupSolvent ( )  [protected], [virtual]
```

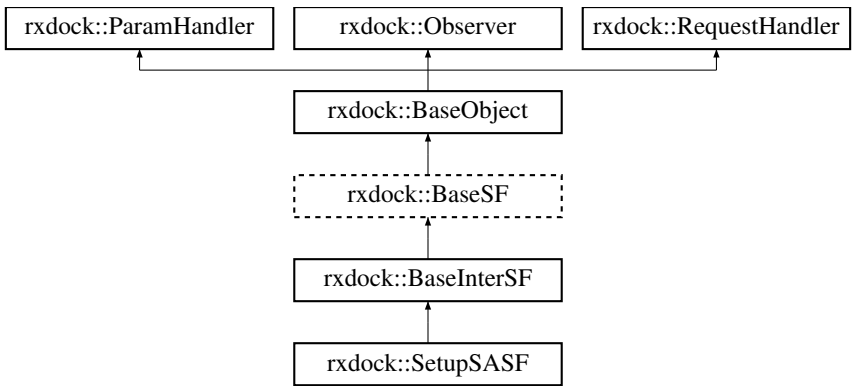
Reimplemented from rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- SetupPolarSF.h
- SetupPolarSF.cxx

5.272 rxdock::SetupSASF Class Reference

Inheritance diagram for rxdock::SetupSASF:



Public Member Functions

- **SetupSASF** (const std::string &strName="setup-sa")

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = "SetupSASF"

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = "BaseInterSF"

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = "BaseSF"
- static const std::string **_WEIGHT** = "weight"
- static const std::string **_RANGE** = "range"
- static const std::string **_SYSTEM_SF** = "rxdock.score.system"
- static const std::string **_INTRA_SF** = "rxdock.score.intra"

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = "BaseObject"
- static const std::string **_CLASS** = "CLASS"
- static const std::string **_NAME** = "NAME"
- static const std::string **_ENABLED** = "ENABLED"

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void **SetupReceptorSATypes** (void)
- void **SetupLigandSATypes** (void)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const SetupSASF &sasf)
- void **from_json** (const json &j, SetupSASF &sasf)

5.272.1 Member Function Documentation

5.272.1.1 RawScore()

```
double SetupSASF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.272.1.2 SetupLigand()

```
void SetupSASF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.272.1.3 SetupReceptor()

```
void SetupSASF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.272.1.4 SetupScore()

```
void SetupSASF::SetupScore ( )  [protected], [virtual]
```

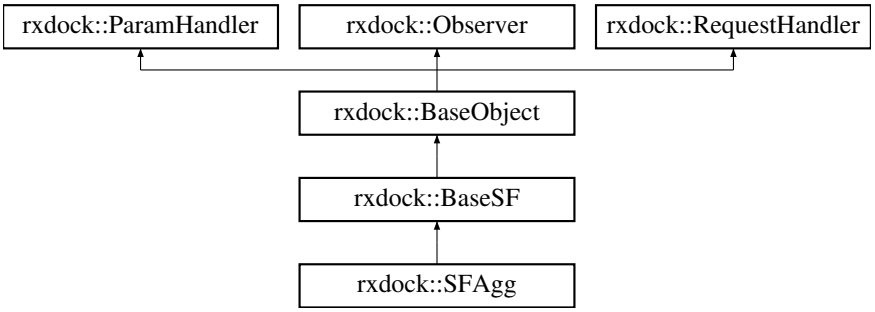
Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- SetupSASF.h
- SetupSASF.cxx

5.273 rxdock::SFAgg Class Reference

Inheritance diagram for rxdock::SFAgg:



Public Member Functions

- RBTDLL_EXPORT **SFAgg** (const std::string &strName=GetMetaDataPrefix()+"score")
- virtual void ScoreMap (StringVariantMap &scoreMap) const
- virtual void Add (BaseSF *)
- virtual void Remove (BaseSF *)
- virtual bool isAgg () const
- virtual unsigned int GetNumSF () const
- virtual BaseSF * GetSF (unsigned int iSF) const
- virtual void Register (WorkSpace *)
- virtual void Unregister ()
- virtual void Update (Subject *theChangedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨SFAgg¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual double RawScore () const

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const SFAgg &sfAgg)
- void **from_json** (const json &j, SFAgg &sfAgg)

5.273.1 Member Function Documentation

5.273.1.1 Add()

```
void SFAgg::Add (
            BaseSF * pSF )  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.273.1.2 GetNumSF()

```
unsigned int SFAgg::GetNumSF ( ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.273.1.3 GetSF()

```
BaseSF * SFAgg::GetSF (
            unsigned int iSF ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.273.1.4 HandleRequest()

```
void SFAgg::HandleRequest (
            RequestPtr spRequest )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.273.1.5 isAgg()

```
bool SFAgg::isAgg ( ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.273.1.6 Print()

```
void SFAgg::Print (
            std::ostream & s ) const  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.273.1.7 RawScore()

```
double SFAgg::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.273.1.8 Register()

```
void SFAgg::Register (
            WorkSpace * pWorkSpace )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.273.1.9 Remove()

```
void SFAgg::Remove (
            BaseSF * pSF )  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.273.1.10 ScoreMap()

```
void SFAgg::ScoreMap (
            StringVariantMap & scoreMap ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.273.1.11 Unregister()

```
void SFAgg::Unregister ( )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.273.1.12 Update()

```
void SFAgg::Update (
            Subject * theChangedSubject )  [virtual]
```

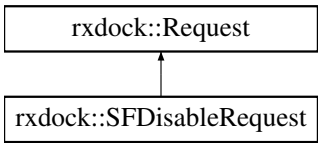
Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- SFAgg.h
- SFAgg.cxx

5.274 rxdock::SFDisableRequest Class Reference

Inheritance diagram for rxdock::SFDisableRequest:



Public Member Functions

- **SFDisableRequest** (const std::string &sfName)

Public Member Functions inherited from rxdock::Request

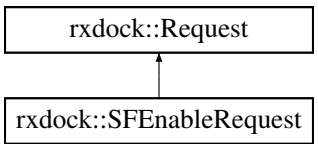
- **Request** (json j)
- RequestID **GetID** () const
- unsigned int **GetNumParameters** () const
- VariantList **GetParameters** () const
- **Request** (RequestID id)
- void **AddParameter** (const Variant &v)

The documentation for this class was generated from the following file:

- SFRequest.h

5.275 rxdock::SFEnableRequest Class Reference

Inheritance diagram for rxdock::SFEnableRequest:



Public Member Functions

- **SFEnableRequest** (const std::string &sfName)

Public Member Functions inherited from rxdock::Request

- **Request** (json j)
- RequestID **GetID** () const
- unsigned int **GetNumParameters** () const
- VariantList **GetParameters** () const
- **Request** (RequestID id)
- void **AddParameter** (const Variant &v)

The documentation for this class was generated from the following file:

- SFRequest.h

5.276 rxdock::SFFactory Class Reference

Public Member Functions

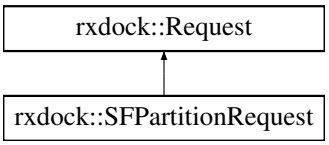
- virtual BaseSF * **Create** (const std::string &_strSFClass, const std::string &strName)
- virtual SFAgg * **CreateAggFromFile** (ParameterFileSourcePtr spPrmSource, const std::string &strName, const std::string &strSFClasses=std::string())

The documentation for this class was generated from the following files:

- SFFactory.h
- SFFactory.cxx

5.277 rxdock::SFPartitionRequest Class Reference

Inheritance diagram for rxdock::SFPartitionRequest:



Public Member Functions

- **SFPartitionRequest** (double dist)
- **SFPartitionRequest** (const std::string &sfName, double dist)

Public Member Functions inherited from rxdock::Request

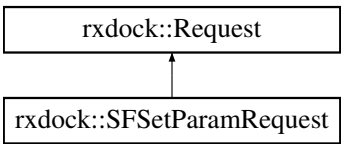
- **Request** (json j)
- RequestID **GetID** () const
- unsigned int **GetNumParameters** () const
- VariantList **GetParameters** () const
- **Request** (RequestID id)
- void **AddParameter** (const Variant &v)

The documentation for this class was generated from the following file:

- SFRequest.h

5.278 rxdock::SFSetParamRequest Class Reference

Inheritance diagram for rxdock::SFSetParamRequest:



Public Member Functions

- **SFSetParamRequest** (const std::string &paramName, const Variant &paramValue)
- **SFSetParamRequest** (const std::string &sfName, const std::string &paramName, const Variant &param↩ Value)

Public Member Functions inherited from rxdock::Request

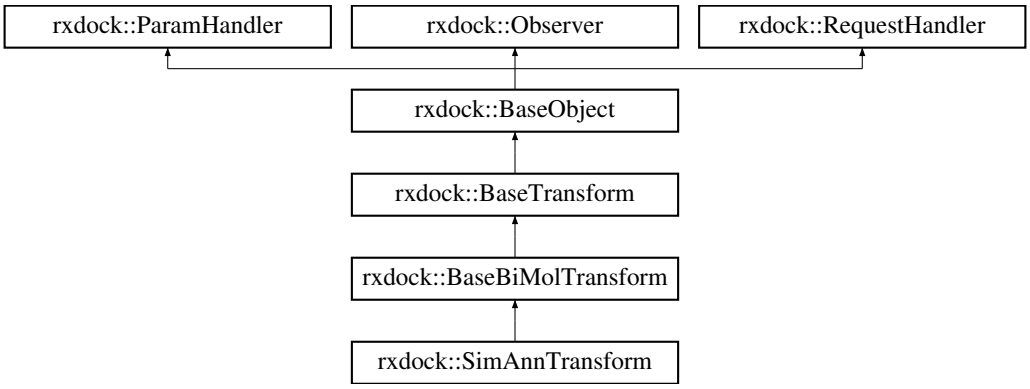
- **Request** (json j)
- RequestID **GetID** () const
- unsigned int **GetNumParameters** () const
- VariantList **GetParameters** () const
- **Request** (RequestID id)
- void **AddParameter** (const Variant &v)

The documentation for this class was generated from the following file:

- SFRequest.h

5.279 rxdock::SimAnnTransform Class Reference

Inheritance diagram for rxdock::SimAnnTransform:



Public Member Functions

- RBTDLL_EXPORT **SimAnnTransform** (const std::string &strName="SIMANN")

Public Member Functions inherited from rxdock::BaseBiMolTransform

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform *)
- virtual void **Remove** (BaseTransform *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform * **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform * **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Member Functions

- static RBTDLL_EXPORT const std::string & **GetStartT** ()
- static RBTDLL_EXPORT const std::string & **GetFinalT** ()
- static RBTDLL_EXPORT const std::string & **GetBlockLength** ()
- static RBTDLL_EXPORT const std::string & **GetNumBlocks** ()
- static RBTDLL_EXPORT const std::string & **GetStepSize** ()
- static RBTDLL_EXPORT const std::string & **GetPartitionDist** ()
- static RBTDLL_EXPORT const std::string & **GetPartitionFreq** ()

Static Public Attributes

- static const std::string **_CT** = ¨SimAnnTransform¨
- static const std::string **_START_T** = ¨start-temperature¨
- static const std::string **_FINAL_T** = ¨final-temperature¨
- static const std::string **_BLOCK_LENGTH** = ¨block-length¨
- static const std::string _SCALE_CHROM_LENGTH
- static const std::string **_NUM_BLOCKS** = ¨number-of-blocks¨
- static const std::string **_STEP_SIZE** = ¨step-size¨
- static const std::string _MIN_ACC_RATE
- static const std::string **_PARTITION_DIST** = ¨partition-distance¨
- static const std::string **_PARTITION_FREQ** = ¨partition-frequency¨
- static const std::string **_HISTORY_FREQ** = ¨history-frequency¨

Static Public Attributes inherited from rxdock::BaseBiMolTransform

- static const std::string **_CT** = ¨BaseBiMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupTransform ()
- void **MC** (double t, int blockLen, double stepSize)
- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseBiMolTransform

- **BaseBiMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupReceptor** ()=0
- virtual void **SetupLigand** ()=0
- virtual void **SetupSolvent** ()
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const SimAnnTransform &simAnnTransform)
- void **from_json** (const json &j, SimAnnTransform &simAnnTransform)

5.279.1 Member Function Documentation

5.279.1.1 Execute()

```
void SimAnnTransform::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.279.1.2 SetupLigand()

```
void SimAnnTransform::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.279.1.3 SetupReceptor()

```
void SimAnnTransform::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.279.1.4 SetupTransform()

```
void SimAnnTransform::SetupTransform ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.279.2 Member Data Documentation

5.279.2.1 _MIN_ACC_RATE

```
const std::string SimAnnTransform::_MIN_ACC_RATE  [static]
```

Initial value:

```
=
    "minimum-metropolis-acceptance-rate"
```

5.279.2.2 _SCALE_CHROM_LENGTH

```
const std::string SimAnnTransform::_SCALE_CHROM_LENGTH  [static]
```

Initial value:

```
=
    "scale-chromosome-length"
```

The documentation for this class was generated from the following files:

- SimAnnTransform.h
- SimAnnTransform.cxx

5.280 rxdock::SimpleSolvationParameters Struct Reference

Public Attributes

- double **p**
- double **r**

The documentation for this struct was generated from the following file:

- SetupSASF.h

5.281 rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion > Class Template Reference

```
#include <NMSimplex.h>
```

Public Member Functions

- **Simplex** (const Criterion &criterion)
- void **Optimize** (Function &fun)
- const ParameterType & GetBestParameters () const
- DataType GetBestValue () const
- void **SetStartPoint** (const ParameterType &point)
- void **SetDelta** (DataType delta)
- void **SetDelta** (ParameterType deltas)

5.281.1 Detailed Description

template<class DataType, class ParameterType, class Function, class Criterion>
class rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion >

The Nelder-Mead Simplex algorithm Complies to a simple and standard interface

Parameters

DataType	is the type of inner values to consider
ParameterType	is the type of ParameterType (Eigen if possible)
Function	is the type of Function to optimize
Criterion	is the type of the stopping Criterion

5.281.2 Member Function Documentation

5.281.2.1 GetBestParameters()

```
template<class DataType , class ParameterType , class Function , class Criterion >
const ParameterType & rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion
>::GetBestParameters ( ) const  [inline]
```

Retrieves the best parameters

5.281.2.2 GetBestValue()

```
template<class DataType , class ParameterType , class Function , class Criterion >
DataType rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion >::Get↩
BestValue ( ) const  [inline]
```

Retrieves the best final value

The documentation for this class was generated from the following file:

- NMSimplex.h

5.282 rxdock::SimplexCostFunction Class Reference

Public Types

- typedef double **DataType**
- typedef Eigen::VectorXd **ParameterType**

Public Member Functions

- **SimplexCostFunction** (BaseSF ∗pSF, ChromElementPtr chrom)
- double **operator()** (const ParameterType &parameters)

Public Attributes

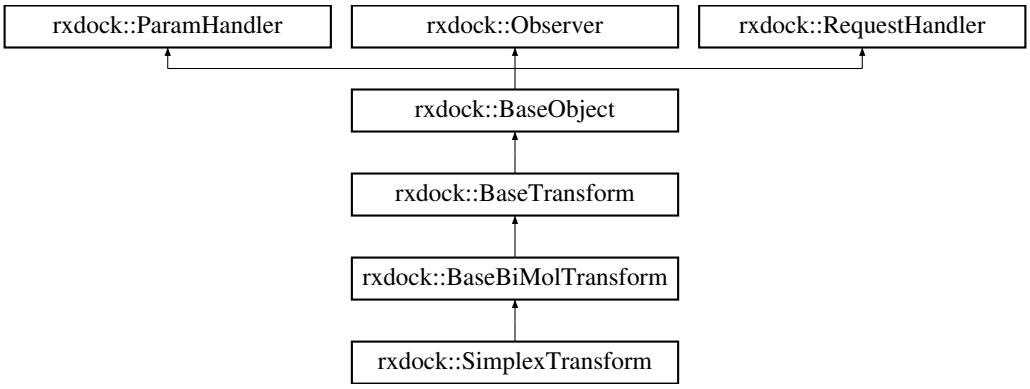
- unsigned long int **nCalls** = 0

The documentation for this class was generated from the following file:

- SimplexCostFunction.h

5.283 rxdock::SimplexTransform Class Reference

Inheritance diagram for rxdock::SimplexTransform:



Public Member Functions

- RBTDLL_EXPORT **SimplexTransform** (const std::string &strName="SIMPLEX")

Public Member Functions inherited from rxdock::BaseBiMolTransform

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform *)
- virtual void **Remove** (BaseTransform *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform * **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform * **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Member Functions

- static RBTDLL_EXPORT const std::string & **GetMaxCalls** ()
- static RBTDLL_EXPORT const std::string & **GetNCycles** ()
- static RBTDLL_EXPORT const std::string & **GetStepSize** ()

Static Public Attributes

- static const std::string **_CT** = ¨SimplexTransform¨
- static const std::string **_MAX_CALLS** = ¨maximum-number-of-calls¨
- static const std::string **_NCYCLES** = ¨number-of-cycles¨
- static const std::string _STOPPING_STEP_LENGTH
- static const std::string **_PARTITION_DIST** = ¨partition-distance¨
- static const std::string **_STEP_SIZE** = ¨step-size¨
- static const std::string **_CONVERGENCE** = ¨convergence¨

Static Public Attributes inherited from rxdock::BaseBiMolTransform

- static const std::string **_CT** = ¨BaseBiMolTransform¨

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = ¨BaseTransform¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupTransform ()
- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseBiMolTransform

- **BaseBiMolTransform** (const std::string &strClass, const std::string &strName)
- virtual void **SetupReceptor** ()=0
- virtual void **SetupLigand** ()=0
- virtual void **SetupSolvent** ()
- virtual void **SetupTransform** ()=0

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.283.1 Member Function Documentation

5.283.1.1 Execute()

```
void SimplexTransform::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.283.1.2 SetupLigand()

```
void SimplexTransform::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.283.1.3 SetupReceptor()

```
void SimplexTransform::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.283.1.4 SetupSolvent()

```
void SimplexTransform::SetupSolvent ( )  [protected], [virtual]
```

Reimplemented from rxdock::BaseBiMolTransform.

5.283.1.5 SetupTransform()

```
void SimplexTransform::SetupTransform ( )  [protected], [virtual]
```

Implements rxdock::BaseBiMolTransform.

5.283.2 Member Data Documentation

5.283.2.1 _STOPPING_STEP_LENGTH

```
const std::string SimplexTransform::_STOPPING_STEP_LENGTH  [static]
```

Initial value:

```
=
    "stopping-step-length"
```

The documentation for this class was generated from the following files:

- SimplexTransform.h
- SimplexTransform.cxx

5.284 rxdock::Singleton< singleton_type > Class Template Reference

Static Public Member Functions

- static singleton_type & **instance** ()

Protected Types

- enum **action** { **DESTROY** , **CREATE** , **GET** }

Static Protected Member Functions

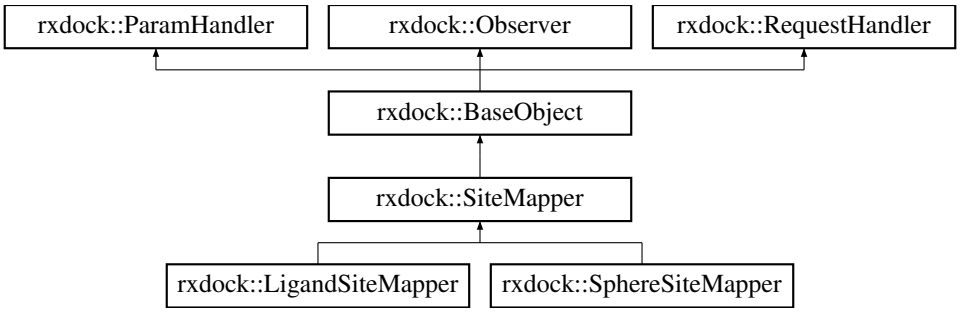
- static singleton_type * **object** (const action &a)

The documentation for this class was generated from the following file:

- Singleton.h

5.285 rxdock::SiteMapper Class Reference

Inheritance diagram for rxdock::SiteMapper:



Public Member Functions

- ModelPtr **GetReceptor** () const
- virtual CavityList **operator()** ()=0
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = "SiteMapper"

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = "BaseObject"
- static const std::string **_CLASS** = "CLASS"
- static const std::string **_NAME** = "NAME"
- static const std::string **_ENABLED** = "ENABLED"

Protected Member Functions

- **SiteMapper** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const SiteMapper &siteMapper)
- void **from_json** (const json &j, SiteMapper &siteMapper)

5.285.1 Member Function Documentation

5.285.1.1 Update()

```
void SiteMapper::Update (
            Subject * theChangedSubject )  [virtual]
```

Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- SiteMapper.h
- SiteMapper.cxx

5.286 rxdock::SiteMapperFactory Class Reference

Public Member Functions

- virtual SiteMapper * **Create** (const std::string &_strMapperClass, const std::string &strName)
- virtual SiteMapper * **CreateFromFile** (ParameterFileSourcePtr spPrmSource, const std::string &strName)

The documentation for this class was generated from the following files:

- SiteMapperFactory.h
- SiteMapperFactory.cxx

5.287 rxdock::SmartPtr< T > Class Template Reference

Public Member Functions

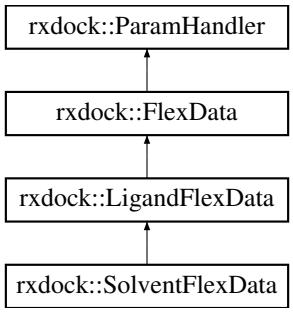
- **SmartPtr** (T *pT)
- **SmartPtr** (const SmartPtr< T > &sp)
- template<class T2 >
  SmartPtr (const SmartPtr< T2 > &sp)
- const SmartPtr< T > & **operator=** (const SmartPtr< T > &sp)
- template<class T2 >
  const SmartPtr< T > & **operator=** (const SmartPtr< T2 > &sp)
- bool **Null** () const
- unsigned * **GetCountPtr** () const
- T * **Ptr** ()
- const T * **Ptr** () const
- void **SetNull** ()
- T * **operator->** ()
- const T * **operator->** () const
- T & **operator*** ()
- const T & **operator*** () const
- **operator T*** () const

The documentation for this class was generated from the following file:

- SmartPointer.h

5.288 rxdock::SolventFlexData Class Reference

Inheritance diagram for rxdock::SolventFlexData:



Public Member Functions

- **SolventFlexData** (DockingSite ∗pDockSite)
- virtual void Accept (FlexDataVisitor &v)

Public Member Functions inherited from rxdock::LigandFlexData

- RBTDLL_EXPORT **LigandFlexData** (DockingSite ∗pDockSite)
- virtual void Accept (FlexDataVisitor &v)

Public Member Functions inherited from rxdock::FlexData

- Model ∗ **GetModel** () const
- void **SetModel** (Model ∗pModel)
- DockingSite ∗ **GetDockingSite** () const
- virtual void **Accept** (FlexDataVisitor &)=0

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Static Public Attributes

- static const std::string & **_OCCUPANCY** = ¨occupancy¨
- static const std::string & **_OCCUPANCY_STEP** = ¨occupancy-step¨

Static Public Attributes inherited from rxdock::LigandFlexData

- static const std::string & **_TRANS_STEP** = ¨translational-step¨
- static const std::string & **_ROT_STEP** = ¨rotational-step¨
- static const std::string & **_DIHEDRAL_STEP** = ¨dihedral-step¨
- static const std::string & **_TRANS_MODE** = ¨translational-mode¨
- static const std::string & **_ROT_MODE** = ¨rotational-mode¨
- static const std::string & **_DIHEDRAL_MODE** = ¨dihedral-mode¨
- static const std::string & **_MAX_TRANS** = ¨maximum-translation¨
- static const std::string & **_MAX_ROT** = ¨maximum-rotation¨
- static const std::string & **_MAX_DIHEDRAL** = ¨maximum-dihedral¨

Additional Inherited Members

Static Public Member Functions inherited from rxdock::LigandFlexData

- static RBTDLL_EXPORT const std::string & **GetTransMode** ()
- static RBTDLL_EXPORT const std::string & **GetRotMode** ()

Protected Member Functions inherited from rxdock::FlexData

- **FlexData** (DockingSite *pDockSite)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.288.1 Member Function Documentation

5.288.1.1 Accept()

```
virtual void rxdock::SolventFlexData::Accept (
            FlexDataVisitor & v )  [inline], [virtual]
```

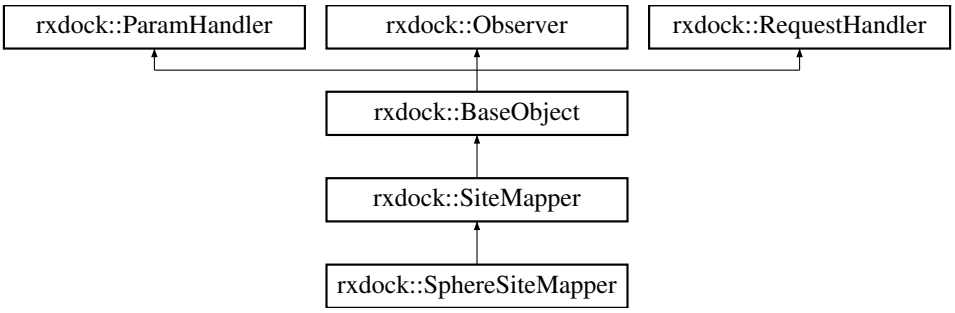
Reimplemented from rxdock::LigandFlexData.

The documentation for this class was generated from the following files:

- SolventFlexData.h
- SolventFlexData.cxx

5.289 rxdock::SphereSiteMapper Class Reference

Inheritance diagram for rxdock::SphereSiteMapper:



Public Member Functions

- **SphereSiteMapper** (const std::string &strName=¨SPHERE_MAPPER¨)
- virtual CavityList operator() ()

Public Member Functions inherited from rxdock::SiteMapper

- ModelPtr **GetReceptor** () const
- virtual CavityList **operator()** ()=0
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨SphereSiteMapper¨
- static const std::string _VOL_INCR
- static const std::string **_SMALL_SPHERE** = ¨small-sphere-radius¨
- static const std::string **_LARGE_SPHERE** = ¨large-sphere-radius¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_CENTER** = ¨center¨
- static const std::string **_RADIUS** = ¨radius¨
- static const std::string **_MIN_VOLUME** = ¨minimum-cavity-volume¨
- static const std::string **_MAX_CAVITIES** = ¨maximum-cavities¨

Static Public Attributes inherited from rxdock::SiteMapper

- static const std::string **_CT** = ¨SiteMapper¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Additional Inherited Members

Protected Member Functions inherited from rxdock::SiteMapper

- **SiteMapper** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.289.1 Member Function Documentation

5.289.1.1 operator()()

```
CavityList SphereSiteMapper::operator() ( )  [virtual]
```

Implements rxdock::SiteMapper.

5.289.2 Member Data Documentation

5.289.2.1 _VOL_INCR

```
const std::string SphereSiteMapper::_VOL_INCR  [static]
```

Initial value:

```
=
    "excluded-volume-radius-increment"
```

The documentation for this class was generated from the following files:

- SphereSiteMapper.h
- SphereSiteMapper.cxx

5.290 rxdock::neldermead::State< DataType, ParameterType > Struct Template Reference

```
#include <NMState.h>
```

Public Attributes

- ParameterType **bestParameters**
- DataType **bestValue**
- ParameterType **currentParameters**
- DataType **currentValue**
- ParameterType **formerParameters**
- DataType **formerValue**
- long **iteration**

Friends

- void **to_json** (json &j, const State &state)
- void **from_json** (const json &j, State &state)

5.290.1 Detailed Description

template<class DataType, class ParameterType>
struct rxdock::neldermead::State< DataType, ParameterType >

Optimization state for an optimizer

The documentation for this struct was generated from the following file:

- NMState.h

5.291 rxdock::StdRestraintAtoms Class Reference

Public Member Functions

- **StdRestraintAtoms** (const StdRestraintNames &n, const AtomList &atomList)
- bool **isOK** () const
- bool **isSimple** () const

Public Attributes

- NoeEndAtoms **from**
- double **maxDist**

Friends

- void **to_json** (json &j, const StdRestraintAtoms &stdRestraintAtoms)
- void **from_json** (const json &j, StdRestraintAtoms &stdRestraintAtoms)

The documentation for this class was generated from the following files:

- NoeRestraint.h
- NoeRestraint.cxx

5.292 rxdock::StdRestraintNames Class Reference

Public Member Functions

- bool **isOK** () const

Public Attributes

- NoeEndNames **from**
- double **maxDist**

Friends

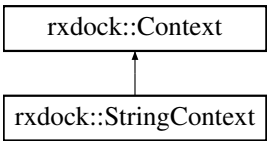
- void **to_json** (json &j, const StdRestraintNames &stdRestraintNames)
- void **from_json** (const json &j, StdRestraintNames &stdRestraintNames)

The documentation for this class was generated from the following files:

- NoeRestraint.h
- NoeRestraint.cxx

5.293 rxdock::StringContext Class Reference

Inheritance diagram for rxdock::StringContext:



Public Member Functions

- **StringContext** (SmartPtr< std::ifstream > ifile)
- **StringContext** (const StringContext &c)
- void Assign (std::string key, ReturnType val)
- void Assign (int i, ReturnType val)
- double **Get** (ModelPtr lig, std::string name)
- double **Get** (ModelPtr rec, DockingSitePtr site, std::string name)
- double **Get** (BaseSF *spSF, std::string name, ModelPtr lig)
- const Vble & GetVble (std::string key)
- const Vble & GetVble (int key)
- void SetVble (int key, const Vble &v)
- void **UpdateLigs** (ModelPtr lig)
- void **UpdateSite** (ModelPtr rec, DockingSitePtr site)
- void **UpdateScores** (BaseSF *spSF, ModelPtr lig)

Public Member Functions inherited from rxdock::Context

- **Context** (const Context &c)
- virtual void **Assign** (std::string, ReturnType)=0
- virtual void **Assign** (int, ReturnType)=0
- virtual const Vble & **GetVble** (int)=0
- virtual const Vble & **GetVble** (std::string)=0
- virtual void **SetVble** (int key, const Vble &v)=0

Additional Inherited Members

Static Public Attributes inherited from rxdock::Context

- static const std::string **_CT** = ¨Context¨

5.293.1 Member Function Documentation

5.293.1.1 Assign() [1/2]

```
void rxdock::StringContext::Assign (
            int i,
            ReturnType val )  [inline], [virtual]
```

Implements rxdock::Context.

5.293.1.2 Assign() [2/2]

```
void rxdock::StringContext::Assign (
            std::string key,
            ReturnType val )  [inline], [virtual]
```

Implements rxdock::Context.

5.293.1.3 GetVble() [1/2]

```
const Vble & rxdock::StringContext::GetVble (
            int key )  [inline], [virtual]
```

Implements rxdock::Context.

5.293.1.4 GetVble() [2/2]

```
const Vble & rxdock::StringContext::GetVble (
            std::string key )  [inline], [virtual]
```

Implements rxdock::Context.

5.293.1.5 SetVble()

```
void rxdock::StringContext::SetVble (
            int key,
            const Vble & v )  [inline], [virtual]
```

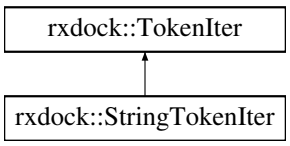
Implements rxdock::Context.

The documentation for this class was generated from the following files:

- Context.h
- Context.cxx

5.294 rxdock::StringTokenIter Class Reference

Inheritance diagram for rxdock::StringTokenIter:



Public Member Functions

- **StringTokenIter** (const StringTokenIter &)
- **StringTokenIter** (const istreamPtr, ContextPtr)
- void **copy** (const StringTokenIter &)
- void Next (ContextPtr)
- TokenPtr Current ()

- virtual void **Next** (ContextPtr)=0
- virtual TokenPtr **Current** ()=0

Static Public Attributes

- static const std::string **_CT** = ¨StringTokenIter¨

5.294.1 Member Function Documentation

5.294.1.1 Current()

```
TokenPtr StringTokenIter::Current ( )  [virtual]
```

Implements rxdock::TokenIter.

5.294.1.2 Next()

```
void StringTokenIter::Next (
            ContextPtr  )  [virtual]
```

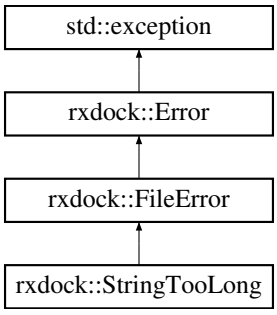
Implements rxdock::TokenIter.

The documentation for this class was generated from the following files:

- StringTokenIter.h
- StringTokenIter.cxx

5.295 rxdock::StringTooLong Class Reference

Inheritance diagram for rxdock::StringTooLong:



Public Member Functions

- **StringTooLong** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strFile, int nLine, const std::string &strMessage="")

Public Member Functions inherited from rxdock::Error

- **Error** (const std::string &strFile, int nLine, const std::string &strMessage="")
- virtual const char * **what** () const noexcept
- std::string **File** () const
- int **Line** () const
- std::string **Message** () const
- std::string **Name** () const
- bool **isOK** () const
- void **AddMessage** (const std::string &strMessage)

Additional Inherited Members

Protected Member Functions inherited from rxdock::FileError

- **FileError** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

Protected Member Functions inherited from rxdock::Error

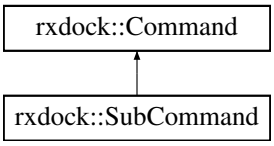
- **Error** (const std::string &strName, const std::string &strFile, int nLine, const std::string &strMessage="")

The documentation for this class was generated from the following file:

- FileError.h

5.296 rxdock::SubCommand Class Reference

Inheritance diagram for rxdock::SubCommand:



Public Member Functions

- int GetNArgs ()
- std::string GetName ()
- void SetArg (int i, ReturnType f)
- void SetNameArg (int i, std::string &n)
- ReturnType Execute ()

- virtual ReturnType **Execute** ()=0
- virtual int **GetNArgs** ()=0
- virtual std::string **GetName** ()=0
- virtual void **SetArg** (int i, ReturnType f)=0
- virtual void **SetNameArg** (int i, std::string &n)=0

Additional Inherited Members

Static Public Member Functions inherited from rxdock::Command

- static void **Clear** ()

Static Public Attributes inherited from rxdock::Command

- static int **ntabs** = 0
- static bool **inside** = false

Protected Member Functions inherited from rxdock::Command

- std::string **tabs** (int n)

5.296.1 Member Function Documentation

5.296.1.1 Execute()

```
ReturnType rxdock::SubCommand::Execute ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.296.1.2 GetName()

```
std::string rxdock::SubCommand::GetName ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.296.1.3 GetNArgs()

```
int rxdock::SubCommand::GetNArgs ( )  [inline], [virtual]
```

Implements rxdock::Command.

5.296.1.4 SetArg()

```
void rxdock::SubCommand::SetArg (
            int i,
            ReturnType f )  [inline], [virtual]
```

Implements rxdock::Command.

5.296.1.5 SetNameArg()

```
void rxdock::SubCommand::SetNameArg (
            int i,
            std::string & n )  [inline], [virtual]
```

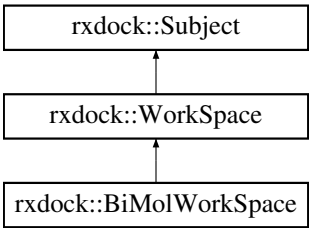
Implements rxdock::Command.

The documentation for this class was generated from the following file:

- Command.h

5.297 rxdock::Subject Class Reference

Inheritance diagram for rxdock::Subject:



Public Member Functions

- virtual void **Attach** (Observer ∗)
- virtual void **Detach** (Observer ∗)
- virtual void **Notify** ()

The documentation for this class was generated from the following files:

- Subject.h
- Subject.cxx

5.298 rxdock::SymBond Class Reference

Public Member Functions

- **SymBond** (BondPtr bond, int n, bool swap)

Public Attributes

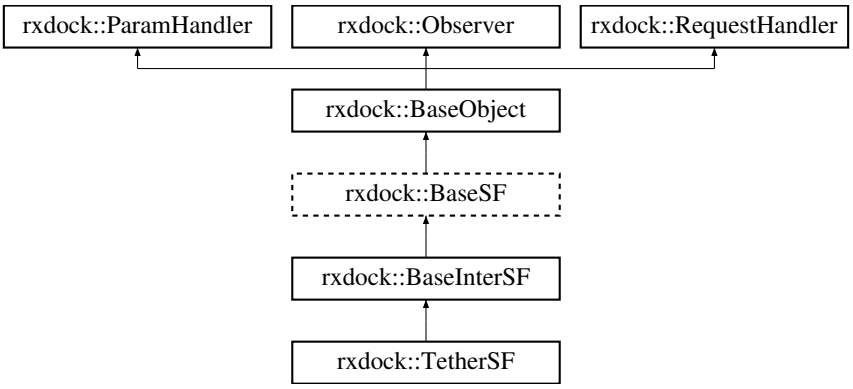
- BondPtr **m_bond**
- int **m_n**
- bool **m_swap**
- double **m_dih**

The documentation for this class was generated from the following file:

- rbrms.cxx

5.299 rxdock::TetherSF Class Reference

Inheritance diagram for rxdock::TetherSF:



Public Member Functions

- **TetherSF** (const std::string &strName="tether")

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject *theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = ¨TetherSF¨
- static const std::string **_REFERENCE_FILE** = ¨reference-file¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseInterSF

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const TetherSF &tetsf)
- void **from_json** (const json &j, TetherSF &tetsf)

5.299.1 Member Function Documentation

5.299.1.1 ParameterUpdated()

```
void TetherSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.299.1.2 RawScore()

```
double TetherSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.299.1.3 SetupLigand()

```
void TetherSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.299.1.4 SetupReceptor()

```
void TetherSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.299.1.5 SetupScore()

```
void TetherSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- TetherSF.h
- TetherSF.cxx

5.300 rxdock::Token Class Reference

Public Member Functions

- **Token** (const Vble &)
- **Token** (Commands)
- **Token** (const Token &)
- const Vble & **GetVble** () const
- bool **IsVble** ()
- bool **IsLog** ()
- bool **IsExp** ()
- bool **IsAdd** ()
- bool **IsSub** ()
- bool **IsMul** ()
- bool **IsDiv** ()
- bool **IsAnd** ()
- bool **IsIf** ()

Static Public Attributes

- static const std::string **_CT** = ¨Token¨

Friends

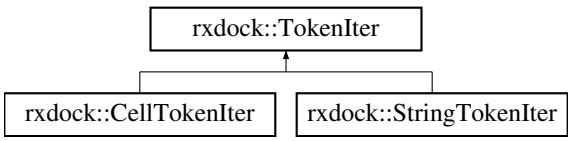
- void **to_json** (json &j, const Token &token)
- void **from_json** (const json &j, Token &token)

The documentation for this class was generated from the following files:

- Token.h
- Token.cxx

5.301 rxdock::TokenIter Class Reference

Inheritance diagram for rxdock::TokenIter:



Public Member Functions

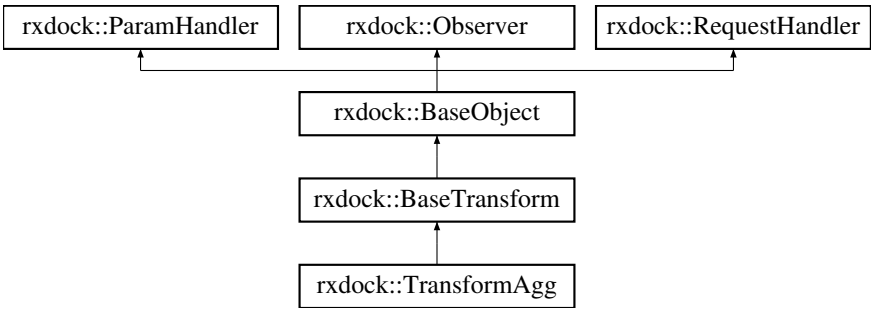
- virtual void **Next** (ContextPtr)=0
- virtual TokenPtr **Current** ()=0

The documentation for this class was generated from the following file:

- TokenIter.h

5.302 rxdock::TransformAgg Class Reference

Inheritance diagram for rxdock::TransformAgg:



Public Member Functions

- RBTDLL_EXPORT **TransformAgg** (const std::string &strName=¨dock¨)
- virtual void Add (BaseTransform ∗)
- virtual void Remove (BaseTransform ∗)
- virtual bool isAgg () const
- virtual unsigned int GetNumTransforms () const
- virtual BaseTransform ∗ GetTransform (unsigned int iTransform) const
- virtual void Register (WorkSpace ∗)
- virtual void Unregister ()
- virtual void Update (Subject ∗theChangedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::BaseTransform

- std::string GetFullName () const
- void **Go** ()
- virtual void **Add** (BaseTransform ∗)
- virtual void **Remove** (BaseTransform ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumTransforms** () const
- virtual BaseTransform ∗ **GetTransform** (unsigned int) const
- void **Orphan** ()
- BaseTransform ∗ **GetParentTransform** () const
- void **AddSFRequest** (RequestPtr)
- void **ClearSFRequests** ()
- void **SendSFRequests** ()

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Attributes

- static const std::string **_CT** = "TransformAgg"

Static Public Attributes inherited from rxdock::BaseTransform

- static const std::string **_CT** = "BaseTransform"

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = "BaseObject"
- static const std::string **_CLASS** = "CLASS"
- static const std::string **_NAME** = "NAME"
- static const std::string **_ENABLED** = "ENABLED"

Protected Member Functions

- virtual void Execute ()

Protected Member Functions inherited from rxdock::BaseTransform

- **BaseTransform** (const std::string &strClass, const std::string &strName)

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

5.302.1 Member Function Documentation

5.302.1.1 Add()

```
void TransformAgg::Add (
            BaseTransform * pTransform )  [virtual]
```

Reimplemented from rxdock::BaseTransform.

5.302.1.2 Execute()

```
void TransformAgg::Execute ( )  [protected], [virtual]
```

Implements rxdock::BaseTransform.

5.302.1.3 GetNumTransforms()

```
unsigned int TransformAgg::GetNumTransforms ( ) const  [virtual]
```

Reimplemented from rxdock::BaseTransform.

5.302.1.4 GetTransform()

```
BaseTransform * TransformAgg::GetTransform (
            unsigned int iTransform ) const  [virtual]
```

Reimplemented from rxdock::BaseTransform.

5.302.1.5 HandleRequest()

```
void TransformAgg::HandleRequest (
            RequestPtr spRequest )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.302.1.6 isAgg()

```
bool TransformAgg::isAgg ( ) const  [virtual]
```

Reimplemented from rxdock::BaseTransform.

5.302.1.7 Print()

```
void TransformAgg::Print (
            std::ostream & s ) const  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.302.1.8 Register()

```
void TransformAgg::Register (
            WorkSpace * pWorkSpace )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.302.1.9 Remove()

```
void TransformAgg::Remove (
            BaseTransform * pTransform )  [virtual]
```

Reimplemented from rxdock::BaseTransform.

5.302.1.10 Unregister()

```
void TransformAgg::Unregister ( )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.302.1.11 Update()

```
void TransformAgg::Update (
            Subject * theChangedSubject )  [virtual]
```

Implements rxdock::Observer.

The documentation for this class was generated from the following files:

- TransformAgg.h
- TransformAgg.cxx

5.303 rxdock::TransformFactory Class Reference

Public Member Functions

- virtual BaseTransform * **Create** (const std::string &strTransformClass, const std::string &strName)
- virtual TransformAgg * **CreateAggFromFile** (ParameterFileSourcePtr spPrmSource, const std::string &str↩ Name, const std::string &strTransformClasses=std::string())

The documentation for this class was generated from the following files:

- TransformFactory.h
- TransformFactory.cxx

5.304 rxdock::TranslateAtom Class Reference

Public Member Functions

- **TranslateAtom** (const Vector &vv)
- void **operator()** (Atom *pAtom)

The documentation for this class was generated from the following file:

- Atom.h

5.305 rxdock::TranslateAtomIfSelected Class Reference

Public Member Functions

- **TranslateAtomIfSelected** (const Vector &vv)
- void **operator()** (Atom *pAtom)

The documentation for this class was generated from the following file:

- Atom.h

5.306 rxdock::TriposAtomType Class Reference

Classes

- struct info

Public Types

- enum **eType** {
  UNDEFINED = 0 , **Al** , **Br** , **C_cat** ,
  C_1 , **C_1_H1** , **C_2** , **C_2_H1** ,
  C_2_H2 , **C_3** , **C_3_H1** , **C_3_H2** ,
  C_3_H3 , **C_ar** , **C_ar_H1** , **Ca** ,
  Cl , **Du** , **F** , **H** ,
  H_P , **I** , **K** , **Li** ,
  LP , **N_1** , **N_2** , **N_3** ,
  N_4 , **N_am** , **N_ar** , **N_pl3** ,
  Na , **O_2** , **O_3** , **O_co2** ,
  P_3 , **S_2** , **S_3** , **S_o** ,
  S_o2 , **Si** , **MAXTYPES** }

Public Member Functions

- eType **operator()** (Atom ∗pAtom, bool useExtendedTypes=false) const
- RBTDLL_EXPORT std::string **Type2Str** (eType) const
- int **Type2Hybrid** (eType) const
- int **Type2AtomicNo** (eType) const
- RBTDLL_EXPORT eType **Str2Type** (const std::string &) const

Friends

- void **to_json** (json &j, const TriposAtomType &triposAtomType)
- void **from_json** (const json &j, TriposAtomType &triposAtomType)

The documentation for this class was generated from the following files:

- TriposAtomType.h
- TriposAtomType.cxx

5.307 rxdock::Variant Class Reference

Public Member Functions

- **Variant** (int i)
- **Variant** (double d)
- **Variant** (const std::string &s)
- **Variant** (const char ∗c)
- **Variant** (const std::vector< std::string > &sl)
- **Variant** (bool b)
- **Variant** (const Coord &c)
- **Variant** (const std::vector< double > &dl, int maxCols, int precision)
- Variant & **operator=** (int i)
- Variant & **operator=** (double d)
- Variant & **operator=** (const std::string &s)
- Variant & **operator=** (const char ∗c)
- Variant & **operator=** (const std::vector< std::string > &sl)
- Variant & **operator=** (bool b)

- Variant & **operator=** (const Coord &c)
- void **operator+=** (const Variant &v)
- **operator int** () const
- **operator unsigned int** () const
- **operator double** () const
- **operator std::string** () const
- **operator std::vector< std::string >** () const
- **operator bool** () const
- **operator Coord** () const
- double **GetDouble** () const
- std::string **GetString** () const
- std::vector< std::string > **GetStringList** () const
- bool **GetBool** () const
- Coord **GetCoord** () const
- unsigned int **Size** () const
- bool **isEmpty** () const

Friends

- std::ostream & **operator<<** (std::ostream &s, const Variant &v)
- void **to_json** (json &j, const Variant &v)
- void **from_json** (const json &j, Variant &v)

The documentation for this class was generated from the following file:

- Variant.h

5.308 rxdock::Vble Class Reference

Public Types

- enum **VbleType** { **CTE** , **LIG** , **SCORE** , **SITE** }

Public Member Functions

- **Vble** (std::string s, ReturnType val=0.0)
- **Vble** (const Vble &v)
- void **SetValue** (ReturnType val)
- ReturnType **GetValue** () const
- Vble & **operator=** (const Vble &v)
- void **SetName** (std::string nm)
- std::string **GetName** () const
- bool **IsLig** ()
- bool **IsScore** ()
- bool **IsSite** ()

Static Public Attributes

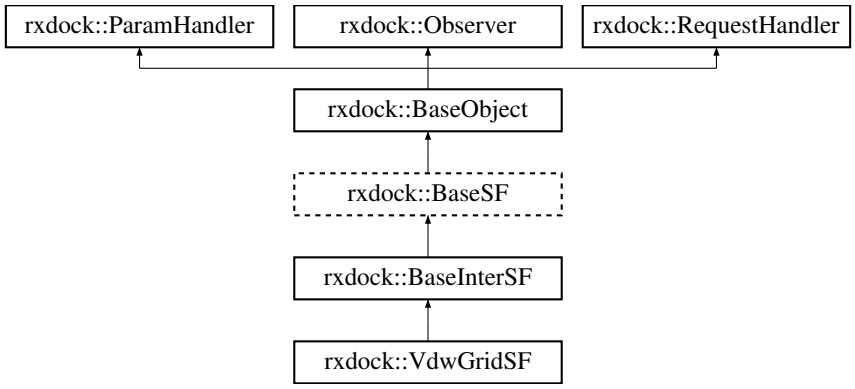
- static const std::string **_CT** = ¨Vble¨

The documentation for this class was generated from the following files:

- Vble.h
- Context.cxx

5.309 rxdock::VdwGridSF Class Reference

Inheritance diagram for rxdock::VdwGridSF:



Public Member Functions

- **VdwGridSF** (const std::string &strName=¨vdw¨)

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Static Public Member Functions

- static RBTDLL_EXPORT const std::string & **GetCt** ()

Static Public Attributes

- static const std::string **_CT** = ¨VdwGridSF¨
- static const std::string **_GRID** = ¨grid¨
- static const std::string **_SMOOTHED** = ¨smoothed¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Friends

- void **to_json** (json &j, const VdwGridSF &vdwGridSF)
- void **from_json** (const json &j, VdwGridSF &vdwGridSF)

5.309.1 Member Function Documentation

5.309.1.1 ParameterUpdated()

```
void VdwGridSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.309.1.2 RawScore()

```
double VdwGridSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.309.1.3 SetupLigand()

```
void VdwGridSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.309.1.4 SetupReceptor()

```
void VdwGridSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.309.1.5 SetupScore()

```
void VdwGridSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.309.1.6 SetupSolvent()

```
void VdwGridSF::SetupSolvent ( )  [protected], [virtual]
```

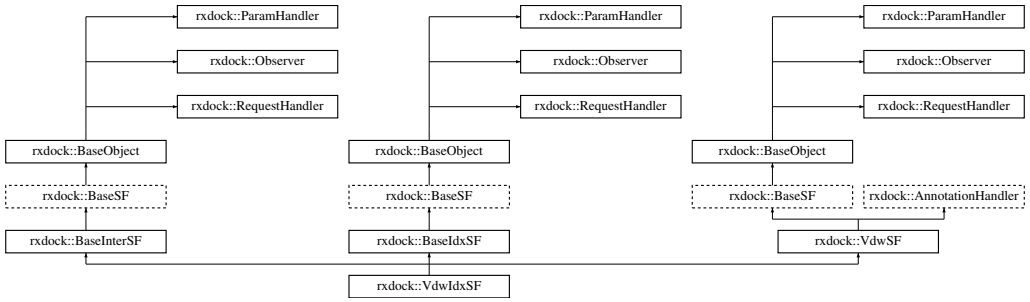
Reimplemented from rxdock::BaseInterSF.

The documentation for this class was generated from the following files:

- VdwGridSF.h
- VdwGridSF.cxx

5.310 rxdock::VdwIdxSF Class Reference

Inheritance diagram for rxdock::VdwIdxSF:



Public Member Functions

- RBTDLL_EXPORT **VdwIdxSF** (const std::string &strName=¨vdw¨)
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseInterSF

- ModelPtr **GetReceptor** () const
- ModelPtr **GetLigand** () const
- ModelList **GetSolvent** () const
- virtual void Update (Subject ∗theChangedSubject)

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseIdxSF

- double **GetGridStep** () const
- void **SetGridStep** (double step)
- double **GetBorder** () const
- void **SetBorder** (double border)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Attributes

- static const std::string **_CT** = ¨VdwIdxSF¨
- static const std::string **_THRESHOLD_ATTR** = ¨attribute-threshold¨
- static const std::string **_THRESHOLD_REP** = ¨representation-threshold¨
- static const std::string _ANNOTATION_LIPO
- static const std::string **_ANNOTATE** = ¨enable-annotations¨
- static const std::string **_FAST_SOLVENT** = ¨fast-solvent¨

Static Public Attributes inherited from rxdock::BaseInterSF

- static const std::string **_CT** = ¨BaseInterSF¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::BaseIdxSF

- static const std::string **_CT** = ¨BaseIdxSF¨
- static const std::string **_GRIDSTEP** = ¨grid-step¨
- static const std::string **_BORDER** = ¨border¨

Static Public Attributes inherited from rxdock::VdwSF

- static const std::string **_CT** = ¨VdwSF¨
- static const std::string **_USE_4_8** = ¨use-4-8¨
- static const std::string **_USE_TRIPOS** = ¨use-tripos¨
- static const std::string **_RMAX** = ¨rmax¨
- static const std::string **_ECUT** = ¨ecut¨
- static const std::string **_E0** = ¨e0¨

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- virtual void SetupReceptor ()
- virtual void SetupLigand ()
- virtual void SetupSolvent ()
- virtual void SetupScore ()
- virtual double RawScore () const
- double **InterScore** () const
- double **ReceptorScore** () const
- double **SolventScore** () const
- double **ReceptorSolventScore** () const
- double **LigandSolventScore** () const
- void ParameterUpdated (const std::string &strName)

- virtual void SetupReceptor ()=0
- virtual void SetupLigand ()=0
- virtual void **SetupSolvent** ()
- virtual void SetupScore ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::BaseIdxSF

- InteractionGridPtr **CreateInteractionGrid** () const
- NonBondedGridPtr **CreateNonBondedGrid** () const
- NonBondedHHSGridPtr **CreateNonBondedHHSGrid** () const
- double **GetMaxError** () const
- double **GetCorrectedRange** () const
- void **OwnParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::VdwSF

- void **OwnParameterUpdated** (const std::string &strName)
- double **VdwScore** (const Atom *pAtom, const AtomRList &atomList) const
- double **VdwScoreEnabledOnly** (const Atom *pAtom, const AtomRList &atomList) const
- double **MaxVdwRange** (const Atom *pAtom) const
- double **MaxVdwRange** (TriposAtomType::eType t) const
- void **BuildIntraMap** (const AtomRList &atomList, AtomRListList &intns) const
- void **BuildIntraMap** (const AtomRList &atomList1, const AtomRList &atomList2, AtomRListList &intns) const
- void **Partition** (const AtomRList &atomList, const AtomRListList &intns, AtomRListList &prtIntns, double dist=0.0) const

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

Friends

- void **to_json** (json &j, const VdwIdxSF &vdwIdxSF)
- void **from_json** (const json &j, VdwIdxSF &vdwIdxSF)

Additional Inherited Members

Static Public Member Functions inherited from rxdock::VdwSF

- static RBTDLL_EXPORT const std::string & **GetEcut** ()

5.310.1 Member Function Documentation

5.310.1.1 ParameterUpdated()

```
void VdwIdxSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.310.1.2 RawScore()

```
double VdwIdxSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.310.1.3 ScoreMap()

```
void VdwIdxSF::ScoreMap (
            StringVariantMap & scoreMap ) const  [virtual]
```

Reimplemented from rxdock::BaseSF.

5.310.1.4 SetupLigand()

```
void VdwIdxSF::SetupLigand ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.310.1.5 SetupReceptor()

```
void VdwIdxSF::SetupReceptor ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.310.1.6 SetupScore()

```
void VdwIdxSF::SetupScore ( )  [protected], [virtual]
```

Implements rxdock::BaseInterSF.

5.310.1.7 SetupSolvent()

```
void VdwIdxSF::SetupSolvent ( )  [protected], [virtual]
```

Reimplemented from rxdock::BaseInterSF.

5.310.2 Member Data Documentation

5.310.2.1 _ANNOTATION_LIPO

```
const std::string VdwIdxSF::_ANNOTATION_LIPO  [static]
```

Initial value:

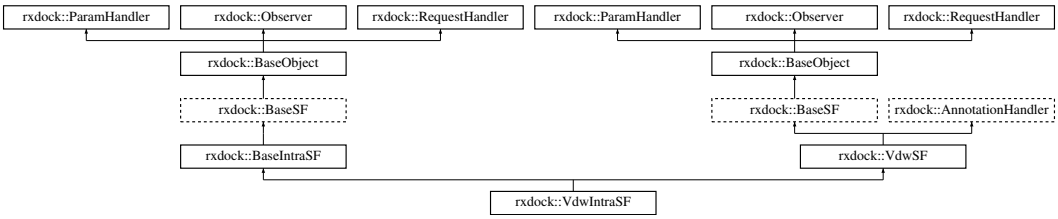
```
=
    "lipophilic-annotation-threshold"
```

The documentation for this class was generated from the following files:

- VdwIdxSF.h
- VdwIdxSF.cxx

5.311 rxdock::VdwIntraSF Class Reference

Inheritance diagram for rxdock::VdwIntraSF:



Public Member Functions

- RBTDLL_EXPORT **VdwIntraSF** (const std::string &strName=¨vdw¨)
- virtual void HandleRequest (RequestPtr spRequest)

Public Member Functions inherited from rxdock::BaseIntraSF

- ModelPtr **GetLigand** () const
- virtual void Update (Subject ∗theChangedSubject)
- virtual void ScoreMap (StringVariantMap &scoreMap) const

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF ∗)
- virtual void **Remove** (BaseSF ∗)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF ∗ **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF ∗ **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace ∗)
- virtual void **Unregister** ()
- WorkSpace ∗ **GetWorkSpace** () const
- virtual void Deleted (Subject ∗theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

- virtual void Update (Subject ∗theChangedSubject)=0
- virtual void **Deleted** (Subject ∗theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::AnnotationHandler

- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

Static Public Attributes

- static const std::string **_CT** = "VdwIntraSF"

Static Public Attributes inherited from rxdock::BaseIntraSF

- static const std::string **_CT** = "BaseIntraSF"

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = "BaseSF"
- static const std::string **_WEIGHT** = "weight"
- static const std::string **_RANGE** = "range"
- static const std::string **_SYSTEM_SF** = "rxdock.score.system"
- static const std::string **_INTRA_SF** = "rxdock.score.intra"

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = "BaseObject"
- static const std::string **_CLASS** = "CLASS"
- static const std::string **_NAME** = "NAME"
- static const std::string **_ENABLED** = "ENABLED"

Static Public Attributes inherited from rxdock::VdwSF

- static const std::string **_CT** = "VdwSF"
- static const std::string **_USE_4_8** = "use-4-8"
- static const std::string **_USE_TRIPOS** = "use-tripos"
- static const std::string **_RMAX** = "rmax"
- static const std::string **_ECUT** = "ecut"
- static const std::string **_E0** = "e0"

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = "annotations"

Protected Member Functions

- virtual void SetupScore ()
- virtual double RawScore () const
- void ParameterUpdated (const std::string &strName)

- virtual void **SetupScore** ()=0

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::VdwSF

- void **OwnParameterUpdated** (const std::string &strName)
- double **VdwScore** (const Atom *pAtom, const AtomRList &atomList) const
- double **VdwScoreEnabledOnly** (const Atom *pAtom, const AtomRList &atomList) const
- double **MaxVdwRange** (const Atom *pAtom) const
- double **MaxVdwRange** (TriposAtomType::eType t) const
- void **BuildIntraMap** (const AtomRList &atomList, AtomRListList &intns) const
- void **BuildIntraMap** (const AtomRList &atomList1, const AtomRList &atomList2, AtomRListList &intns) const
- void **Partition** (const AtomRList &atomList, const AtomRListList &intns, AtomRListList &prtIntns, double dist=0.0) const

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

Friends

- void **to_json** (json &j, const VdwIntraSF &vdwIntraSF)
- void **from_json** (const json &j, VdwIntraSF &vdwIntraSF)

Additional Inherited Members

Static Public Member Functions inherited from rxdock::VdwSF

- static RBTDLL_EXPORT const std::string & **GetEcut** ()

5.311.1 Member Function Documentation

5.311.1.1 HandleRequest()

```
void VdwIntraSF::HandleRequest (
            RequestPtr spRequest )  [virtual]
```

Reimplemented from rxdock::BaseObject.

5.311.1.2 ParameterUpdated()

```
void VdwIntraSF::ParameterUpdated (
            const std::string & strName )  [protected], [virtual]
```

Reimplemented from rxdock::BaseSF.

5.311.1.3 RawScore()

```
double VdwIntraSF::RawScore (
            void  ) const  [protected], [virtual]
```

Implements rxdock::BaseSF.

5.311.1.4 SetupScore()

```
void VdwIntraSF::SetupScore ( )  [protected], [virtual]
```

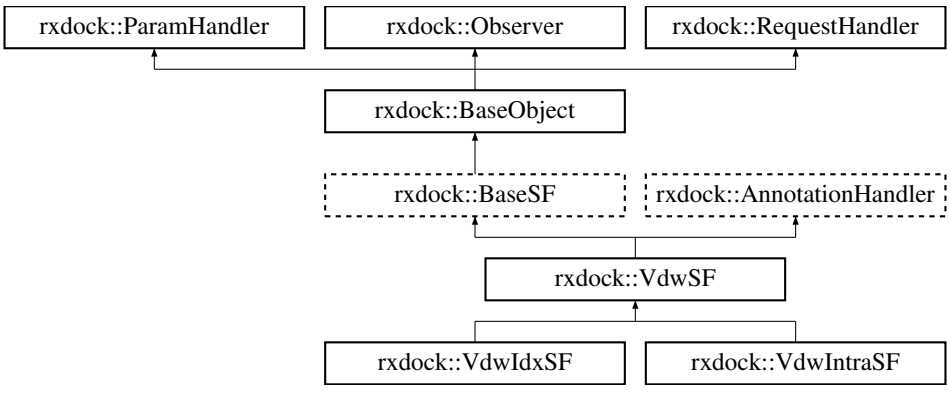
Implements rxdock::BaseIntraSF.

The documentation for this class was generated from the following files:

- VdwIntraSF.h
- VdwIntraSF.cxx

5.312 rxdock::VdwSF Class Reference

Inheritance diagram for rxdock::VdwSF:



Static Public Member Functions

- static RBTDLL_EXPORT const std::string & **GetEcut** ()

Static Public Attributes

- static const std::string **_CT** = ¨VdwSF¨
- static const std::string **_USE_4_8** = ¨use-4-8¨
- static const std::string **_USE_TRIPOS** = ¨use-tripos¨
- static const std::string **_RMAX** = ¨rmax¨
- static const std::string **_ECUT** = ¨ecut¨
- static const std::string **_E0** = ¨e0¨

Static Public Attributes inherited from rxdock::BaseSF

- static const std::string **_CT** = ¨BaseSF¨
- static const std::string **_WEIGHT** = ¨weight¨
- static const std::string **_RANGE** = ¨range¨
- static const std::string **_SYSTEM_SF** = ¨rxdock.score.system¨
- static const std::string **_INTRA_SF** = ¨rxdock.score.intra¨

Static Public Attributes inherited from rxdock::BaseObject

- static const std::string **_CT** = ¨BaseObject¨
- static const std::string **_CLASS** = ¨CLASS¨
- static const std::string **_NAME** = ¨NAME¨
- static const std::string **_ENABLED** = ¨ENABLED¨

Static Public Attributes inherited from rxdock::AnnotationHandler

- static const std::string **_ANNOTATION_FIELD** = ¨annotations¨

Protected Member Functions

- void **OwnParameterUpdated** (const std::string &strName)
- double **VdwScore** (const Atom ∗pAtom, const AtomRList &atomList) const
- double **VdwScoreEnabledOnly** (const Atom ∗pAtom, const AtomRList &atomList) const
- double **MaxVdwRange** (const Atom ∗pAtom) const
- double **MaxVdwRange** (TriposAtomType::eType t) const
- void **BuildIntraMap** (const AtomRList &atomList, AtomRListList &intns) const
- void **BuildIntraMap** (const AtomRList &atomList1, const AtomRList &atomList2, AtomRListList &intns) const
- void **Partition** (const AtomRList &atomList, const AtomRListList &intns, AtomRListList &prtIntns, double dist=0.0) const

Protected Member Functions inherited from rxdock::BaseSF

- **BaseSF** (const std::string &strClass, const std::string &strName)
- virtual double RawScore () const =0
- void ParameterUpdated (const std::string &strName)
- void **AddToParentMapEntry** (StringVariantMap &scoreMap, double rs) const

Protected Member Functions inherited from rxdock::BaseObject

- **BaseObject** (const std::string &strClass, const std::string &strName)
- void ParameterUpdated (const std::string &strName)

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

Protected Member Functions inherited from rxdock::AnnotationHandler

- void **AddAnnotation** (AnnotationPtr spAnnotation) const
- void **ClearAnnotationList** () const
- void **EnableAnnotations** (bool bEnabled) const

Friends

- void **to_json** (json &j, const VdwSF &vswSF)
- void **from_json** (const json &j, VdwSF &vswSF)

Additional Inherited Members

Public Member Functions inherited from rxdock::BaseSF

- std::string GetFullName () const
- double **GetWeight** () const
- void **SetWeight** (double)
- double **GetRange** () const
- RBTDLL_EXPORT void **SetRange** (double)
- RBTDLL_EXPORT double **Score** () const
- virtual void **ScoreMap** (StringVariantMap &scoreMap) const
- virtual void **Add** (BaseSF *)
- virtual void **Remove** (BaseSF *)
- virtual bool **isAgg** () const
- virtual unsigned int **GetNumSF** () const
- virtual BaseSF * **GetSF** (unsigned int iSF) const
- void **Orphan** ()
- BaseSF * **GetParentSF** () const

Public Member Functions inherited from rxdock::BaseObject

- std::string **GetClass** () const
- std::string **GetName** () const
- void **SetName** (const std::string &)
- virtual std::string **GetFullName** () const
- void **Enable** ()
- void **Disable** ()
- bool **isEnabled** () const
- virtual void **Register** (WorkSpace *)
- virtual void **Unregister** ()
- WorkSpace * **GetWorkSpace** () const
- virtual void Deleted (Subject *theDeletedSubject)
- virtual void HandleRequest (RequestPtr spRequest)
- virtual void Print (std::ostream &s) const

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Public Member Functions inherited from rxdock::Observer

- virtual void Update (Subject *theChangedSubject)=0
- virtual void **Deleted** (Subject *theDeletedSubject)=0

- virtual void **HandleRequest** (RequestPtr spRequest)

Public Member Functions inherited from rxdock::AnnotationHandler

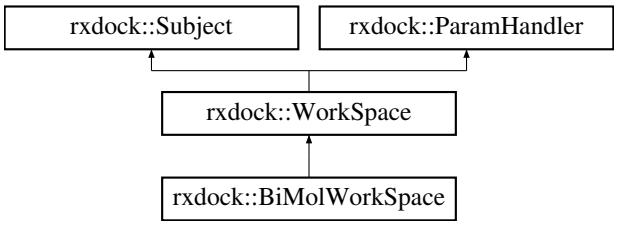
- bool **isAnnotationEnabled** () const
- const AnnotationList & **GetAnnotationList** () const
- int **GetNumAnnotations** () const
- void **RenderAnnotationList** (const std::string &strName, std::vector< std::string > &retVal) const

The documentation for this class was generated from the following files:

- VdwSF.h
- VdwSF.cxx

5.313 rxdock::WorkSpace Class Reference

Inheritance diagram for rxdock::WorkSpace:



Public Member Functions

- **WorkSpace** (unsigned int nModels=2)
- RBTDLL_EXPORT std::string **GetName** () const
- RBTDLL_EXPORT void **SetName** (const std::string &)
- RBTDLL_EXPORT unsigned int **GetNumModels** () const
- RBTDLL_EXPORT ModelList **GetModels** () const
- RBTDLL_EXPORT ModelPtr **GetModel** (unsigned int iModel) const
- void **SetModel** (unsigned int iModel, ModelPtr spModel)
- ModelList **GetModels** (unsigned int iModel) const
- void **AddModels** (ModelList modelList)
- void **SetModels** (unsigned int iModel, ModelList modelList)
- void **RemoveModels** (unsigned int iModel)
- MolecularFileSinkPtr **GetSink** () const
- RBTDLL_EXPORT void **SetSink** (MolecularFileSinkPtr)
- virtual void **Save** (bool bSaveScores=true)
- MolecularFileSinkPtr **GetHistorySink** () const
- RBTDLL_EXPORT void **SetHistorySink** (MolecularFileSinkPtr)
- virtual void **SaveHistory** (bool bSaveScores=true)
- RBTDLL_EXPORT BaseSF * **GetSF** () const
- RBTDLL_EXPORT void **SetSF** (BaseSF *)
- BaseTransform * **GetTransform** () const
- RBTDLL_EXPORT void **SetTransform** (BaseTransform *)
- virtual void **Run** ()
- void **SetPopulation** (PopulationPtr population)
- PopulationPtr **GetPopulation** () const
- void **ClearPopulation** ()
- RBTDLL_EXPORT DockingSitePtr **GetDockingSite** () const
- RBTDLL_EXPORT void **SetDockingSite** (DockingSitePtr spDockSite)
- FilterPtr **GetFilter** () const
- RBTDLL_EXPORT void **SetFilter** (FilterPtr spFilter)

Public Member Functions inherited from rxdock::Subject

- virtual void **Attach** (Observer *)
- virtual void **Detach** (Observer *)
- virtual void **Notify** ()

Public Member Functions inherited from rxdock::ParamHandler

- unsigned int **GetNumParameters** () const
- Variant **GetParameter** (const std::string &strName) const
- bool **isParameterValid** (const std::string &strName) const
- std::vector< std::string > **GetParameterNames** () const
- StringVariantMap **GetParameters** () const
- RBTDLL_EXPORT void **SetParameter** (const std::string &strName, const Variant &vValue)
- virtual void **Print** (std::ostream &s) const

Static Public Attributes

- static const std::string **_CT** = ¨WorkSpace¨
- static const std::string **_NAME** = ¨NAME¨

Additional Inherited Members

Protected Member Functions inherited from rxdock::ParamHandler

- void **AddParameter** (const std::string &strName, const Variant &vValue)
- void **DeleteParameter** (const std::string &strName)
- void **ClearParameters** ()
- virtual void **ParameterUpdated** (const std::string &strName)

The documentation for this class was generated from the following files:

- WorkSpace.h
- WorkSpace.cxx

Chapter 6

File Documentation

6.1 AlignTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Aligns ligand with the principal axes of one of the active site cavities
00014
00015 #ifndef _RBTALIGNTRANSFORM_H_
00016 #define _RBTALIGNTRANSFORM_H_
00017
00018 #include "rxdock/BaseBiMolTransform.h"
00019 #include "rxdock/Cavity.h"
00020 #include "rxdock/Rand.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class AlignTransform : public BaseBiMolTransform {
00029 public:
00030   // Static data member for class type
00031   static const std::string _CT;
00032   // Parameter names
00033   static const std::string _COM;
00034   static const std::string _AXES;
00035
00037   // Constructors/destructors
00038   AlignTransform(const std::string &strName = "ALIGN");
00039   virtual ~AlignTransform();
00040
00041   friend void to_json(json &j, const AlignTransform &alignTransform);
00042   friend void from_json(const json &j, AlignTransform &alignTransform);
00043
00045   // Public methods
00047
00048 protected:
00050   // Protected methods
00052   virtual void SetupReceptor(); // Called by Update when receptor is changed
00053   virtual void SetupLigand();   // Called by Update when ligand is changed
00054   virtual void
00055   SetupTransform(); // Called by Update when either model has changed
00056   virtual void Execute();
00057
00058 private:
00060   // Private methods
00062   AlignTransform(
00063       const AlignTransform &); // Copy constructor disabled by default
00064   AlignTransform &
00065   operator=(const AlignTransform &); // Copy assignment disabled by default
```

```
00066
00067 protected:
00069   // Protected data
00071
00072 private:
00074   // Private data
00076   Rand &m_rand; // keep a reference to the singleton random number generator
00077   CavityList m_cavities;        // List of active site cavities to choose from
00078   std::vector<int> m_cumulSize; // Cumulative sizes, for weighted probabilities
00079   int m_totalSize;              // Total size of all cavities
00080 };
00081
00082 void to_json(json &j, const AlignTransform &alignTransform);
00083 void from_json(const json &j, AlignTransform &alignTransform);
00084
00085 // Useful typedefs
00086 typedef SmartPtr<AlignTransform> AlignTransformPtr; // Smart pointer
00087
00088 } // namespace rxdock
00089
00090 #endif //_RBTALIGNTRANSFORM_H_
```

6.2 Annotation.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple struct for holding annotation information associated with
00014 // intermolecular scores
00015
00016 #ifndef _RBTANNOTATION_H_
00017 #define _RBTANNOTATION_H_
00018
00019 #include "rxdock/Atom.h"
00020 #include "rxdock/Config.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class Annotation {
00029 public:
00031   // Constructors/destructors
00032   Annotation(const Atom *pAtom1, const Atom *pAtom2, double dist, double score);
00033   Annotation(json j);
00034   virtual ~Annotation();
00035
00036   friend void to_json(json &j, const Annotation &annotation);
00037   friend void from_json(const json &j, Annotation &annotation);
00038
00040   // Public methods
00042   // Get
00043   const Atom *GetAtom1Ptr() const;
00044   const Atom *GetAtom2Ptr() const;
00045   double GetDistance() const;
00046   double GetScore() const;
00047   // Get the fully qualified (FQ) residue name for atom 2 (target atom)
00048   std::string GetFQResName() const;
00049
00050   // Set
00051   void SetAtom1Ptr(const Atom *pAt1);
00052   void SetAtom2Ptr(const Atom *pAt2);
00053   void SetDistance(double d);
00054   void SetScore(double s);
00055
00056   // Render annotation into string in rDock Viewer format
00057   std::string Render() const;
00058
00059   // Operators
00060   // Special meaning of operator+ for accumulating annotations by residue
00061   void operator+=(const Annotation &ann);
00062
```

```
00063 protected:
00065   // Protected methods
00067
00068 private:
00070   // Private methods
00072   Annotation(); // Disable default constructor
00073   // Annotation(const Annotation&);//Copy constructor disabled by default
00074   // Annotation& operator=(const Annotation&);//Copy assignment disabled
00075   // by default
00076
00077 protected:
00079   // Protected data
00081
00082 private:
00084   // Private data
00086   const Atom *m_pAtom1;
00087   const Atom *m_pAtom2;
00088   double m_dist;
00089   double m_score;
00090 };
00091
00092 // Useful typedefs
00093 typedef SmartPtr<Annotation> AnnotationPtr;        // Smart pointer
00094 typedef std::vector<AnnotationPtr> AnnotationList; // Vector of smart pointers
00095 typedef AnnotationList::iterator AnnotationListIter;
00096 typedef AnnotationList::const_iterator AnnotationListConstIter;
00097
00099 // Non-member functions (in rxdock namespace)
00101 void to_json(json &j, const Annotation &annotation);
00102 void from_json(const json &j, Annotation &annotation);
00103
00105 // Comparison functions for sorting Annotation* containers
00106 // For use by STL sort algorithms
00108
00109 // Less than operator for sorting Annotation*'s by the atom ID of atom 2
00110 // (target)
00111 class Ann_Cmp_AtomId2 {
00112 public:
00113   bool operator()(Annotation *pAnn1, Annotation *pAnn2) const {
00114     return pAnn1->GetAtom2Ptr()->GetAtomId() <
00115            pAnn2->GetAtom2Ptr()->GetAtomId();
00116   }
00117 };
00118
00119 } // namespace rxdock
00120
00121 #endif //_RBTANNOTATION_H_
```

6.3 AnnotationHandler.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base implementation class for managing an annotation list. Scoring function
00014 // classes wishing to store annotations for later retrieval should derive from
00015 // this class.
00016
00017 #ifndef _RBTANNOTATIONHANDLER_H_
00018 #define _RBTANNOTATIONHANDLER_H_
00019
00020 #include "rxdock/Annotation.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class AnnotationHandler {
00029 public:
00030   static const std::string _ANNOTATION_FIELD;
00031
00033   // Constructors/destructors
00034   virtual ~AnnotationHandler(); // Default destructor
```

```
00035
00036   friend void to_json(json &j, const AnnotationHandler &annotationHandler);
00037   friend void from_json(const json &j, AnnotationHandler &annotationHandler);
00038
00040   // Public methods
00042   bool isAnnotationEnabled() const;
00043   // Get a const ref to the annotation list (read-only)
00044   const AnnotationList &GetAnnotationList() const;
00045   int GetNumAnnotations() const;
00046   void RenderAnnotationList(const std::string &strName,
00047                             std::vector<std::string> &retVal) const;
00048
00049 protected:
00051   // Protected methods
00053   RBTDLL_EXPORT AnnotationHandler();
00054   void AddAnnotation(AnnotationPtr spAnnotation) const;
00055   void ClearAnnotationList() const;
00056   void EnableAnnotations(bool bEnabled) const;
00057
00058 private:
00060   // Private methods
00062   AnnotationHandler(
00063       const AnnotationHandler &); // Copy constructor disabled by default
00064   AnnotationHandler &
00065   operator=(const AnnotationHandler &); // Copy assignment disabled by default
00066
00067 protected:
00069   // Protected data
00071
00072 private:
00074   // Private data
00076   mutable bool m_bEnabled;
00077   mutable AnnotationList m_annotationList;
00078 };
00079
00080 void to_json(json &j, const AnnotationHandler &annotationHandler);
00081 void from_json(const json &j, AnnotationHandler &annotationHandler);
00082
00083 } // namespace rxdock
00084
00085 #endif //_RBTANNOTATIONHANDLER_H_
```

6.4 AromIdxSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Indexed-grid-based intermolecular aromatic scoring function
00014
00015 #ifndef _RBTAROMIDXSF_H_
00016 #define _RBTAROMIDXSF_H_
00017
00018 #include "rxdock/AnnotationHandler.h"
00019 #include "rxdock/BaseIdxSF.h"
00020 #include "rxdock/BaseInterSF.h"
00021 #include "rxdock/Plane.h"
00022
00023 #include <nlohmann/json.hpp>
00024
00025 using json = nlohmann::json;
00026
00027 namespace rxdock {
00028
00029 class AromIdxSF : public BaseInterSF,
00030                   public BaseIdxSF,
00031                   public AnnotationHandler {
00032 public:
00033   // Class type string
00034   static const std::string _CT;
00035   // Parameter names
00036   static const std::string _INCR;
00037   static const std::string _R12;
00038   static const std::string _DR12MIN;
00039   static const std::string _DR12MAX;
```

```
00040   static const std::string _DAMIN;
00041   static const std::string _DAMAX;
00042   // DM 12 Jun 2002 - score threshold used for counting aromatic interactions
00043   static const std::string _THRESHOLD;
00044
00045   AromIdxSF(const std::string &strName = "arom");
00046   virtual ~AromIdxSF();
00047
00048   friend void to_json(json &j, const AromIdxSF &aromIdxSF);
00049   friend void from_json(const json &j, AromIdxSF &aromIdxSF);
00050
00051   // Override BaseSF::ScoreMap to provide additional raw descriptors
00052   // virtual void ScoreMap(StringVariantMap& scoreMap) const;
00053
00054 protected:
00055   virtual void SetupReceptor();
00056   virtual void SetupLigand();
00057   virtual void SetupScore();
00058   virtual double RawScore() const;
00059
00060   // Clear the receptor and ligand grids and lists respectively
00061   // As we are not using smart pointers, there is some memory management to do
00062   void ClearReceptor();
00063   void ClearLigand();
00064
00065   // DM 25 Oct 2000 - track changes to parameter values in local data members
00066   // ParameterUpdated is invoked by ParamHandler::SetParameter
00067   void ParameterUpdated(const std::string &strName);
00068
00069 private:
00071   // DM 6 Feb 2003
00072   // To be consistent with the Polar score, we should really define a base
00073   // AromSF class which will provide the scoring function primitives, then
00074   // have an AromIdxSF subclass for intermolecular aromatic interactions, and
00075   // a AromIntraSF subclass for ligand intramolecular interactions. Currently
00076   // there is no AromIntraSF class so everything is contained within
00077   // AromIdxSF
00078
00079   // Generic scoring function params
00080   struct f1prms {
00081     double R0, DRMin, DRMax, slope;
00082     f1prms(double R, double DMin, double DMax)
00083         : R0(R), DRMin(DMin), DRMax(DMax), slope(1.0 / (DMax - DMin)) {}
00084   };
00085
00086   inline f1prms GetRprms() const { return f1prms(m_R12, m_DR12Min, m_DR12Max); }
00087   inline f1prms GetAprms() const { return f1prms(0.0, m_DAMin, m_DAMax); }
00088
00089   // Generic scoring function primitive (deviation from ideal geometry)
00090   inline double f1(double DR, const f1prms &prms) const {
00091     return (DR > prms.DRMax)   ? 0.0
00092          : (DR > prms.DRMin) ? 1.0 - prms.slope * (DR - prms.DRMin)
00093                              : 1.0;
00094   }
00095
00096   // The actual aromatic score, between a given interaction center and a list of
00097   // near neighbour centers
00098   double AromScore(const InteractionCenter *pIC1,
00099                    const InteractionCenterList &IC2List, const f1prms &Rprms,
00100                    const f1prms &Aprms) const;
00101   double PiScore(const InteractionCenter *pIC1,
00102                  const InteractionCenterList &IC2List) const;
00103   // End of section that should ultimately be moved to AromSF base class
00105
00106   InteractionGridPtr m_spAromGrid;
00107   InteractionGridPtr m_spGuanGrid;
00108   InteractionCenterList m_recepAromList;
00109   InteractionCenterList m_recepGuanList;
00110   InteractionCenterList m_ligAromList;
00111   InteractionCenterList m_ligGuanList;
00112
00113   // DM 25 Oct 2000 - local copies of params
00114   double m_R12;
00115   double m_DR12Min;
00116   double m_DR12Max;
00117   double m_DAMin;
00118   double m_DAMax;
00119
00120   // DM 12 Jun 2002 - keep track of number of ligand rings and guan carbons
00121   // involved in non-zero arom interactions
00122   mutable int m_nArom;
00123   mutable int m_nGuan;
00124   mutable double m_ss; // Sigma-sigma score
00125   mutable double m_sp; // Sigma-pi score
00126   mutable double m_pp; // pi-pi score
00127   double m_threshold;
00128 };
```

```
00129
00130 void to_json(json &j, const Bond &bond);
00131 void from_json(const json &j, Bond &bond);
00132
00133 } // namespace rxdock
00134
00135 #endif //_RBTAROMIDXSF_H_
```

6.5 Atom.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Handles all the properties for a single atom
00014
00015 #ifndef _RBTATOM_H_
00016 #define _RBTATOM_H_
00017
00018 #include "rxdock/Config.h"
00019 #include "rxdock/Coord.h"
00020 #include "rxdock/PMF.h"
00021 #include "rxdock/Quat.h"
00022 #include "rxdock/TriposAtomType.h"
00023
00024 #include <list>
00025
00026 #include <nlohmann/json.hpp>
00027
00028 using json = nlohmann::json;
00029
00030 namespace rxdock {
00031
00032 class Model; // Forward declaration
00033 class Bond;  // Forward declaration
00034
00035 // DM 23 Apr 1999 - provide custom comparison function
00036 // to bond map, so that map is sorted by bond ID, not by pointer
00037 class BondPCmp_BondId {
00038 public:
00039   bool operator()(Bond *pBond1, Bond *pBond2) const;
00040 };
00041
00042 // Map of bonds the atom is bonded to
00043 // Key is regular Bond*, value is Bool
00044 // Bool is true if this atom is atom1 in the bond, false if atom2
00045 typedef std::map<Bond *, bool, BondPCmp_BondId> BondMap;
00046 typedef BondMap::iterator BondMapIter;
00047 typedef BondMap::const_iterator BondMapConstIter;
00048
00049 class Atom {
00050 public:
00052   // Enums
00054
00055   // Hybridisation state
00056   enum eHybridState {
00057     UNDEFINED = 0,
00058     SP = 1,   // SP hybridised (linear geom)
00059     SP2 = 2,  // SP2 hybridised (planar geom)
00060     SP3 = 3,  // SP3 hybridised (tetrahedral geom)
00061     AROM = 4, // Aromatic SP2 (planar geom)
00062     TRI = 5,  // Trigonal nitrogen (planar geom)
00063   };
00064
00066   // Constructors / destructors
00068
00069   // Default constructor
00070   RBTDLL_EXPORT Atom();
00071
00072   // Constructor supplying all 2-D parameters
00073   RBTDLL_EXPORT
00074   explicit Atom(int nAtomId, int nAtomicNo = 6, std::string strAtomName = "C",
00075                 std::string strSubunitId = "1",
00076                 std::string strSubunitName = "RES",
00077                 std::string strSegmentName = "SEG1",
```

```
00078                 eHybridState eState =
00079                     UNDEFINED, // DM 8 Dec 1998 Changed from SP3 to UNDEFINED
00080                 unsigned int nHydrogens = 0, int nFormalCharge = 0);
00081   Atom(json j);
00082
00083   // Default destructor
00084   virtual ~Atom();
00085
00086   // Copy constructor
00087   Atom(const Atom &atom);
00088
00089   // Copy assignment
00090   Atom &operator=(const Atom &atom);
00091
00093   // Stream functions
00095
00096   // Insertion operator (primarily for debugging)
00097   // Note: needs to be friend so can access atom private data
00098   // without using the accessor functions
00099   RBTDLL_EXPORT friend std::ostream &operator«(std::ostream &s,
00100                                                const Atom &atom);
00101   friend void to_json(json &j, const Atom &atom);
00102   friend void from_json(const json &j, Atom &atom);
00103
00104   // Virtual function for dumping atom details to an output stream
00105   // Derived classes (e.g. pseudoatom) can override if required
00106   virtual std::ostream &Print(std::ostream &s) const;
00107
00109   // Public accessor functions
00110   // 2-D attributes
00112
00113   // DM 7 June 2006
00114   // atom enabled state is controlled by the parent Model
00115   // enabled state
00116   RBTDLL_EXPORT bool GetEnabled() const;
00117
00118   // AtomId
00119   int GetAtomId() const { return m_nAtomId; }
00120   void SetAtomId(const int nAtomId) { m_nAtomId = nAtomId; }
00121
00122   // AtomicNo
00123   int GetAtomicNo() const { return m_nAtomicNo; }
00124   void SetAtomicNo(const int nAtomicNo) { m_nAtomicNo = nAtomicNo; }
00125
00126   // AtomName
00127   std::string GetName() const { return m_strAtomName; }
00128   void SetAtomName(const std::string &strAtomName) {
00129     m_strAtomName = strAtomName;
00130   }
00131   RBTDLL_EXPORT std::string
00132   GetFullAtomName() const; // Returns composite of segment name,
00133                            // subunit id and name, and atom name
00134
00135   // SubunitId
00136   std::string GetSubunitId() const { return m_strSubunitId; }
00137   void SetSubunitId(const std::string &strSubunitId) {
00138     m_strSubunitId = strSubunitId;
00139   }
00140
00141   // SubunitName
00142   std::string GetSubunitName() const { return m_strSubunitName; }
00143   void SetSubunitName(const std::string &strSubunitName) {
00144     m_strSubunitName = strSubunitName;
00145   }
00146
00147   // SegmentName
00148   std::string GetSegmentName() const { return m_strSegmentName; }
00149   void SetSegmentName(const std::string &strSegmentName) {
00150     m_strSegmentName = strSegmentName;
00151   }
00152
00153   // Hybridisation state
00154   eHybridState GetHybridState() const { return m_eState; }
00155   void SetHybridState(const eHybridState eState) { m_eState = eState; }
00156
00157   // Attached hydrogens
00158   unsigned int GetNumImplicitHydrogens() const { return m_nHydrogens; }
00159   void SetNumImplicitHydrogens(const unsigned int nHydrogens) {
00160     m_nHydrogens = nHydrogens;
00161   }
00162
00163   // FormalCharge (DM 24 Mar 1999 - changed from double to int)
00164   int GetFormalCharge() const { return m_nFormalCharge; }
00165   void SetFormalCharge(const int nFormalCharge) {
00166     m_nFormalCharge = nFormalCharge;
00167   }
00168
```

```
00169   // CyclicFlag - flag to indicate atom is in a ring (set by Model::FindRing)
00170   bool GetCyclicFlag() const { return m_bCyclic; }
00171   void SetCyclicFlag(bool bCyclic = true) { m_bCyclic = bCyclic; }
00172
00173   // SelectionFlag - general purpose flag can be set/cleared by various search
00174   // algorithms (e.g. FindRings)
00175   bool GetSelectionFlag() const { return m_bSelected; }
00176   void SetSelectionFlag(bool bSelected = true) { m_bSelected = bSelected; }
00177   void InvertSelectionFlag() { m_bSelected = !m_bSelected; }
00178
00179   // DM 21 Jul 1999 - user-defined flag
00180   bool GetUser1Flag() const { return m_bUser1; }
00181   void SetUser1Flag(bool bUser1 = true) { m_bUser1 = bUser1; }
00182
00183   // DM 29 Jan 2000 - user-defined double
00184   double GetUser1Value() const { return m_dUser1; }
00185   void SetUser1Value(double dUser1 = 0.0) { m_dUser1 = dUser1; }
00186
00187   // DM 27 Jul 2000 - user-defined double#2
00188   double GetUser2Value() const { return m_dUser2; }
00189   void SetUser2Value(double dUser2 = 0.0) { m_dUser2 = dUser2; }
00190
00191   // ModelPtr - pointer to the parent model
00192   Model *GetModelPtr() const { return m_pModel; }
00193   void SetModelPtr(Model *pModel = nullptr) { m_pModel = pModel; }
00194
00195   // DM 04 Dec 1998  Add functions to handle bond map
00196   // Returns number of bonds in map
00197   unsigned int GetNumBonds() const { return m_bondMap.size(); }
00198   // Returns the bond map
00199   // DM 26 Oct 2000 - return by reference
00200   // BondMap GetBondMap() const {return m_bondMap;}
00201   const BondMap &GetBondMap() const { return m_bondMap; }
00202   // Add a bond to the bond map - returns true if OK, false if this atom is not
00203   // a member of the bond or if bond already added
00204   bool AddBond(Bond *pBond);
00205   // Remove a bond from the bond map - returns true if OK, false if bond not
00206   // present in map
00207   bool RemoveBond(Bond *pBond);
00208   // Returns number of cyclic bonds in map
00209   unsigned int GetNumCyclicBonds() const;
00210   // Returns bond map with only cyclic bonds included
00211   BondMap GetCyclicBondMap() const;
00212   // Returns total formal bond order around atom
00213   int GetTotalFormalBondOrder() const;
00214   // Returns max formal bond order for all bonds to atom
00215   int GetMaxFormalBondOrder() const;
00216
00218   // Public accessor functions
00219   // 3-D attributes
00221
00222   // Coords
00223   // DM 8 Dec 1998 - make virtual so we can override in the pseudoatom class
00224   // DM 11 Jul 2000 - remove overhead of virtual methods
00225   // Pseudo atoms must now refresh their coords each time the constituent atoms
00226   // move DM 27 Oct 2000 - return by reference
00227   const Coord &GetCoords() const { return m_coord; }
00228   double GetX() const { return m_coord.xyz(0); }
00229   double GetY() const { return m_coord.xyz(1); }
00230   double GetZ() const { return m_coord.xyz(2); }
00231
00232   void SetCoords(const Coord &coord) { m_coord = coord; }
00233   void SetCoords(const double x, const double y, const double z) {
00234     m_coord.xyz(0) = x;
00235     m_coord.xyz(1) = y;
00236     m_coord.xyz(2) = z;
00237   }
00238   void SetX(const double x) { m_coord.xyz(0) = x; }
00239   void SetY(const double y) { m_coord.xyz(1) = y; }
00240   void SetZ(const double z) { m_coord.xyz(2) = z; }
00241
00242   // PartialCharge
00243   double GetPartialCharge() const { return m_dPartialCharge; }
00244   void SetPartialCharge(const double dPartialCharge) {
00245     m_dPartialCharge = dPartialCharge;
00246   }
00247
00248   // GroupCharge (added DM 24 Mar 1999, for ionic interaction group charges)
00249   double GetGroupCharge() const { return m_dGroupCharge; }
00250   void SetGroupCharge(const double dGroupCharge) {
00251     m_dGroupCharge = dGroupCharge;
00252   }
00253
00254   // AtomicMass
00255   double GetAtomicMass() const { return m_dAtomicMass; }
00256   void SetAtomicMass(const double dAtomicMass) { m_dAtomicMass = dAtomicMass; }
00257
```

```
00258   // VdwRadius
00259   double GetVdwRadius() const { return m_dVdwRadius; }
00260   void SetVdwRadius(const double dVdwRadius) { m_dVdwRadius = dVdwRadius; }
00261
00262   // AtomType
00263   std::string GetFFType() const { return m_strFFType; }
00264   void SetFFType(const std::string &strFFType) { m_strFFType = strFFType; }
00265   PMFType GetPMFType() const { return m_nPMFType; }
00266   void SetPMFType(PMFType aType) { m_nPMFType = aType; }
00267   TriposAtomType::eType GetTriposType() const { return m_triposType; }
00268   void SetTriposType(TriposAtomType::eType aType) { m_triposType = aType; }
00269
00270   // XB
00271   // reweighting factor
00272   // double GetReweight() const { return m_dReweight; }
00273   // void SetReweight(const double dReweight) { m_dReweight = dReweight; }
00274   // XB END MODIFICATIONS
00275
00277   // Other public methods
00279   // DM 11 Jan 1999 - give atoms the ability to save their current coords
00280   // and revert to the saved coords. Useful for Monte Carlo simulations.
00281   // DM 08 Feb 1999 - all saved coords are now saved in a
00282   // std::map<UInt,Coord> map key=0 is reserved for the default SaveCoords
00283   // and RevertCoords
00284   void SaveCoords(unsigned int coordNum = 0);
00285   void RevertCoords(unsigned int coordNum = 0);
00286
00287   // Translate - translate coordinates by the supplied vector
00288   void Translate(const Vector &vector) { m_coord += vector; }
00289
00290   void Translate(const double vx, const double vy, const double vz) {
00291     m_coord += Coord(vx, vy, vz);
00292   }
00293
00294   // DM 07 Jan 1999 - rotate coordinates using the supplied quaternion
00295   void RotateUsingQuat(const Quat &q) { m_coord = q.Rotate(m_coord); }
00296
00297   // DM 04 Dec 1998  Now we have the bond map, we can easily provide
00298   // coordination numbers This version returns the total number of coordinated
00299   // atoms (includes implicit hydrogens) Equivalent to
00300   // GetNumBonds()+GetNumImplicitHydrogens()
00301   unsigned int GetCoordinationNumber() const;
00302   // This version returns the number of coordinated atoms of a given element
00303   // Note: for hydrogens, the implicit atoms are excluded
00304   // so GetCoordinationNumber(1) will return the number of explicit hydrogens
00305   unsigned int GetCoordinationNumber(int nAtomicNo) const;
00306   // This version returns the number of coordinated atoms of a given force field
00307   // type
00308   unsigned int GetCoordinationNumber(const std::string &strFFType) const;
00309   // This version returns the number of coordinated atoms of a given
00310   // hybridisation state
00311   unsigned int GetCoordinationNumber(eHybridState e) const;
00312
00313 protected:
00314 private:
00315   // Private methods
00316   // Clears the bond map - should only need to be called by the copy
00317   // constructors, hence private
00318   void ClearBondMap();
00319
00320 private:
00321   // Private data
00322
00323   // These can be considered as 2-D params (i.e. define the chemistry and
00324   // topology)
00325   int m_nAtomicNo;              // Atomic number
00326   int m_nAtomId;                // Atom ID
00327   std::string m_strAtomName;    // Atom name
00328   std::string m_strSubunitId;   // Subunit(residue) ID (note: string not int)
00329   std::string m_strSubunitName; // Subunit(residue) name
00330   std::string m_strSegmentName; // Segment(molecule) name
00331   eHybridState m_eState;        // Hybridisation state
00332   unsigned int m_nHydrogens;    // Number of attached (implicit) hydrogens
00333   int m_nFormalCharge; // Formal charge (DM 24 Mar 1999 - changed from double
00334                        // to int)
00335   Model *m_pModel;     // Regular pointer to parent model
00336   BondMap m_bondMap;   // Map of bonds this atom is bonded to
00337   bool m_bCyclic;      // Is the atom in a ring ?
00338   bool m_bSelected;    // Can be set/cleared by various search algorithms (e.g.
00339                        // FindRings)
00340   bool m_bUser1;       // User flag 1
00341   double m_dUser1;     // User value 1
00342   double m_dUser2;     // User value 2
00343   PMFType m_nPMFType;  // PMF atom type: default
00344   TriposAtomType::eType m_triposType; // DM 14 May 2002: Tripos 5.2 atom type
00345
00346   // These can be considered as 3-D params (i.e. the extra info required for 3-D
```

```
00347   // calculations)
00348   Coord m_coord;           // cartesian coords
00349   double m_dPartialCharge; // partial charge
00350   double m_dGroupCharge;   // interaction group charge (added DM 24 Mar 1999)
00351   double m_dAtomicMass;    // atomic mass
00352   double m_dVdwRadius;     // atomic mass
00353   std::string m_strFFType; // force field atom type
00354   // double m_dReweight;      // XB reweighting factor
00355
00356   UIntCoordMap m_savedCoords; // DM 08 Feb 1999 - now store all saved coords
00357                               // in a std::map<UInt,Coord>
00358 };
00359
00360 // Useful typedefs
00361 typedef SmartPtr<Atom> AtomPtr;        // Smart pointer
00362 typedef std::vector<AtomPtr> AtomList; // Vector of smart pointers
00363 typedef AtomList::iterator AtomListIter;
00364 typedef AtomList::const_iterator AtomListConstIter;
00365
00366 typedef std::vector<Atom *> AtomRList; // Vector of regular pointers
00367 typedef AtomRList::iterator AtomRListIter;
00368 typedef AtomRList::const_iterator AtomRListConstIter;
00369
00370 typedef std::vector<AtomList>
00371     AtomListList; // A vector of atom vectors (e.g. for storing ring systems)
00372 typedef AtomListList::iterator AtomListListIter;
00373 typedef AtomListList::const_iterator AtomListListConstIter;
00374
00375 typedef std::vector<AtomRList>
00376     AtomRListList; // A vector of atom vectors (regular pointer version)
00377 typedef AtomRListList::iterator AtomRListListIter;
00378 typedef AtomRListList::const_iterator AtomRListListConstIter;
00379
00380 typedef std::list<AtomPtr> AtomTrueList; // A real list of smart pointers
00381 typedef AtomTrueList::iterator AtomTrueListIter;
00382 typedef AtomTrueList::const_iterator AtomTrueListConstIter;
00383
00385 // Non-member functions (in rxdock namespace)
00387
00388 std::ostream &operator«(std::ostream &s, const Atom &atom);
00389
00390 void to_json(json &j, const Atom &atom);
00391 void from_json(const json &j, Atom &site);
00392
00394 // Calculation functions
00396
00397 // DM 07 Jul 2000 - converted to regular Atom* versions
00398 // Returns the "bond" length in Angstroms between 2 atoms
00399 // Note: atoms do not have to be bonded
00400 inline double BondLength(Atom *pAtom1, Atom *pAtom2) {
00401   return Length(pAtom1->GetCoords(), pAtom2->GetCoords());
00402 }
00403
00404 // Returns the "bond" angle in degrees between 3 atoms
00405 // Note: atoms do not have to be bonded
00406 inline double BondAngle(Atom *pAtom1, Atom *pAtom2, Atom *pAtom3) {
00407   return Angle(pAtom1->GetCoords(), pAtom2->GetCoords(), pAtom3->GetCoords());
00408 }
00409
00410 // Returns the "bond" dihedral in degrees between 4 atoms
00411 // Note: atoms do not have to be bonded
00412 inline double BondDihedral(Atom *pAtom1, Atom *pAtom2, Atom *pAtom3,
00413                            Atom *pAtom4) {
00414   return Dihedral(pAtom1->GetCoords(), pAtom2->GetCoords(), pAtom3->GetCoords(),
00415                   pAtom4->GetCoords());
00416 }
00417
00419 // Conversion functions
00421
00422 // Converts hybridisation state enum to a string
00423 std::string ConvertHybridStateToString(Atom::eHybridState eState);
00424 // Convert formal charge to a string (e.g. +, -, +2, -2 etc)
00425 std::string ConvertFormalChargeToString(int nCharge);
00426
00428 // Predicates (for use by STL algorithms)
00430 // DM 30 Apr 2002 - convert to regular Atom* functions
00431 // More universal as AtomPtr parameters will be automatically degraded to
00432 // regular pointers
00433
00434 // Is atom enabled ?
00435 class isAtomEnabled : public std::function<bool(Atom *)> {
00436 public:
00437   explicit isAtomEnabled() = default;
00438   bool operator()(Atom *pAtom) const { return pAtom->GetEnabled(); }
00439 };
00440
00441 // Is atom selected ?
```

```
00442 class isAtomSelected : public std::function<bool(Atom *)> {
00443 public:
00444   explicit isAtomSelected() = default;
00445   bool operator()(Atom *pAtom) const { return pAtom->GetSelectionFlag(); }
00446 };
00447
00448 // Is atom cyclic ?
00449 class isAtomCyclic : public std::function<bool(Atom *)> {
00450 public:
00451   explicit isAtomCyclic() = default;
00452   bool operator()(const Atom *pAtom) const { return pAtom->GetCyclicFlag(); }
00453 };
00454
00455 // DM 9 Feb 2000 - is atom bridgehead? Checks for >2 cyclic bonds
00456 class isAtomBridgehead : public std::function<bool(Atom *)> {
00457 public:
00458   explicit isAtomBridgehead() = default;
00459   bool operator()(const Atom *pAtom) const {
00460     return (pAtom->GetNumCyclicBonds() > 2);
00461   }
00462 };
00463
00464 // Is atom a H-Bond Acceptor ?
00465 // Checks atomic number for O(8), N(7) or S(16)
00466 class isAtomHBondAcceptor : public std::function<bool(Atom *)> {
00467 public:
00468   explicit isAtomHBondAcceptor() = default;
00469   RBTDLL_EXPORT bool operator()(const Atom *) const;
00470 };
00471
00472 // Is atom a H-Bond Donor ?
00473 // Checks 1) is it a hydrogen, 2) does it make exactly one bond, 3) is the bond
00474 // to O, N or S ?
00475 class isAtomHBondDonor : public std::function<bool(Atom *)> {
00476 public:
00477   explicit isAtomHBondDonor() = default;
00478   RBTDLL_EXPORT bool operator()(const Atom *) const;
00479 };
00480
00481 // Is atom formally charged ?
00482 // Checks if formal charge is != zero
00483 class isAtomCharged : public std::function<bool(Atom *)> {
00484 public:
00485   explicit isAtomCharged() = default;
00486   bool operator()(const Atom *pAtom) const {
00487     return (pAtom->GetFormalCharge() != 0);
00488   }
00489 };
00490
00491 // Is atom formally positively charged ?
00492 // Checks if formal charge is > zero
00493 class isAtomPosCharged : public std::function<bool(Atom *)> {
00494 public:
00495   explicit isAtomPosCharged() = default;
00496   bool operator()(const Atom *pAtom) const {
00497     return (pAtom->GetFormalCharge() > 0);
00498   }
00499 };
00500
00501 // Is atom formally negatively charged ?
00502 // Checks if formal charge is < zero
00503 class isAtomNegCharged : public std::function<bool(Atom *)> {
00504 public:
00505   explicit isAtomNegCharged() = default;
00506   bool operator()(const Atom *pAtom) const {
00507     return (pAtom->GetFormalCharge() < 0);
00508   }
00509 };
00510
00511 // Does atom have implicit hydrogens
00512 class isAtomExtended : public std::function<bool(Atom *)> {
00513 public:
00514   explicit isAtomExtended() = default;
00515   bool operator()(const Atom *pAtom) const {
00516     return (pAtom->GetNumImplicitHydrogens() > 0);
00517   }
00518 };
00519
00520 // Is atom planar ?
00521 // Checks if 1) atom makes 2 bonds (in which case must be planar) or
00522 // 2) hybridisation state is SP2, AROM or TRI
00523 class isAtomPlanar : public std::function<bool(Atom *)> {
00524 public:
00525   explicit isAtomPlanar() = default;
00526   bool operator()(const Atom *) const;
00527 };
00528
```

```
00529 // Is atom a pi-atom ?
00530 // SP2,TRI or AROM, or special case of OSP3
00531 class isPiAtom : public std::function<bool(Atom *)> {
00532 public:
00533   explicit isPiAtom() = default;
00534   RBTDLL_EXPORT bool operator()(const Atom *) const;
00535 };
00536
00537 // Is atomic number equal to n ?
00538 class isAtomicNo_eq : public std::function<bool(Atom *)> {
00539   int n;
00540
00541 public:
00542   explicit isAtomicNo_eq(int nn) : n(nn) {}
00543   bool operator()(const Atom *pAtom) const { return pAtom->GetAtomicNo() == n; }
00544 };
00545
00546 // Is Force field type equal to s ?
00547 class isFFType_eq : public std::function<bool(Atom *)> {
00548   std::string s;
00549
00550 public:
00551   explicit isFFType_eq(std::string ss) : s(std::move(ss)) {}
00552   bool operator()(const Atom *pAtom) const { return pAtom->GetFFType() == s; }
00553 };
00554
00555 // Is Atom name equal to s ?
00556 class isAtomName_eq : public std::function<bool(Atom *)> {
00557   std::string s;
00558
00559 public:
00560   explicit isAtomName_eq(std::string ss) : s(std::move(ss)) {}
00561   bool operator()(const Atom *pAtom) const { return pAtom->GetName() == s; }
00562 };
00563
00564 // Is Subunit name equal to s ?
00565 class isSubunitName_eq : public std::function<bool(Atom *)> {
00566   std::string s;
00567
00568 public:
00569   explicit isSubunitName_eq(std::string ss) : s(std::move(ss)) {}
00570   bool operator()(const Atom *pAtom) const {
00571     return pAtom->GetSubunitName() == s;
00572   }
00573 };
00574
00575 // Is Subunit ID equal to s ?
00576 class isSubunitId_eq : public std::function<bool(Atom *)> {
00577   std::string s;
00578
00579 public:
00580   explicit isSubunitId_eq(std::string ss) : s(std::move(ss)) {}
00581   bool operator()(const Atom *pAtom) const {
00582     return pAtom->GetSubunitId() == s;
00583   }
00584 };
00585
00586 // Is Segment name equal to s ?
00587 class isSegmentName_eq : public std::function<bool(Atom *)> {
00588   std::string s;
00589
00590 public:
00591   explicit isSegmentName_eq(std::string ss) : s(std::move(ss)) {}
00592   bool operator()(const Atom *pAtom) const {
00593     return pAtom->GetSegmentName() == s;
00594   }
00595 };
00596
00597 // Is hybridisation state equal to e ? (DM 26 Mar 1999)
00598 class isHybridState_eq : public std::function<bool(Atom *)> {
00599   Atom::eHybridState e;
00600
00601 public:
00602   explicit isHybridState_eq(Atom::eHybridState ee) : e(ee) {}
00603   bool operator()(const Atom *pAtom) const {
00604     return pAtom->GetHybridState() == e;
00605   }
00606 };
00607
00608 // DM 6 Jan 1999
00609 // Is atom2 equal to atom1 (checks if underlying regular pointers match)
00610 // Note: this is a binary rather than unary predicate
00611 // DM 1 Feb 1999 - renamed from isAtom_eq to isAtomPtr_eq
00612 class isAtomPtr_eq : public std::function<bool(Atom *, Atom *)> {
00613 public:
00614   explicit isAtomPtr_eq() = default;
00615   bool operator()(const Atom *pAtom1, const Atom *pAtom2) const {
```

```
00616     return pAtom1 == pAtom2;
00617   }
00618 };
00619
00620 // DM 1 Feb 1999
00621 // Is atom2 equal to atom1 (checks if subunit name, subunit ID and atom name
00622 // match) Note: this is a binary rather than unary predicate
00623 class isAtom_eq : public std::function<bool(Atom *, Atom *)> {
00624 public:
00625   explicit isAtom_eq() = default;
00626   bool operator()(const Atom *pAtom1, const Atom *pAtom2) const {
00627     return ((pAtom1->GetSubunitId() == pAtom2->GetSubunitId()) &&
00628             (pAtom1->GetSubunitName() == pAtom2->GetSubunitName()) &&
00629             (pAtom1->GetName() == pAtom2->GetName()));
00630   }
00631 };
00632
00633 // DM 22 Apr 2002
00634 // Is atom2 ID equal to atom1 ID
00635 // Note: this is a binary rather than unary predicate
00636 class isAtomId_eq : public std::function<bool(Atom *, Atom *)> {
00637 public:
00638   explicit isAtomId_eq() = default;
00639   bool operator()(const Atom *pAtom1, const Atom *pAtom2) const {
00640     return (pAtom1->GetAtomId() == pAtom2->GetAtomId());
00641   }
00642 };
00643
00644 // DM 12 Apr 1999
00645 // Is atom inside a sphere defined by radius=r, center=c
00646 class isAtomInsideSphere : public std::function<bool(Atom *)> {
00647   const Coord &c; // center of sphere
00648   double r2;      // radius squared (to avoid taking square roots)
00649 public:
00650   explicit isAtomInsideSphere(const Coord &cc, double rr)
00651       : c(cc), r2(rr * rr) {}
00652   bool operator()(const Atom *pAtom) const {
00653     return Length2(pAtom->GetCoords() - c) <= r2;
00654   }
00655 };
00656
00657 // DM 12 Apr 1999
00658 // Is atom within a cuboid defined by cmin,cmax coords?
00659 class isAtomInsideCuboid : public std::function<bool(Atom *)> {
00660   const Coord &cmin;
00661   const Coord &cmax;
00662
00663 public:
00664   explicit isAtomInsideCuboid(const Coord &ccmin, const Coord &ccmax)
00665       : cmin(ccmin), cmax(ccmax) {}
00666   bool operator()(const Atom *pAtom) const {
00667     // DM 27 Oct 2000 - GetCoords now returns by reference
00668     const Coord &c = pAtom->GetCoords();
00669     return (c >= cmin) && (c <= cmax);
00670   }
00671 };
00672
00673 // DM 29 Jul 1999
00674 // Is atom within a given distance of any coord in the coord list
00675 class isAtomNearCoordList : public std::function<bool(Atom *)> {
00676   const CoordList &cl;
00677   double r2; // radius squared (to avoid taking square roots)
00678 public:
00679   explicit isAtomNearCoordList(const CoordList &ccl, double rr)
00680       : cl(ccl), r2(rr * rr) {}
00681   bool operator()(const Atom *pAtom) const;
00682 };
00683
00684 // Is atom2 1-2 connected (i.e. bonded) to atom1 ?
00685 class isAtom_12Connected : public std::function<bool(Atom *)> {
00686   Atom *pAtom1;
00687   AtomList bondedAtomList1;
00688
00689 public:
00690   explicit isAtom_12Connected(Atom *spAtom);
00691   bool operator()(Atom *spAtom2) const;
00692 };
00693
00694 // Is atom2 1-3 connected (i.e. via a bond angle) to atom1 ?
00695 class isAtom_13Connected : public std::function<bool(Atom *)> {
00696   Atom *pAtom1;
00697   AtomList bondedAtomList1;
00698
00699 public:
00700   explicit isAtom_13Connected(Atom *pAtom);
00701   bool operator()(Atom *pAtom2) const;
00702 };
```

```
00703
00704 // Is atom's subunit an RNA-type (i.e. A,G,C or U)
00705 class isAtomRNA : public std::function<bool(Atom *)> {
00706 public:
00707   explicit isAtomRNA() = default;
00708   bool operator()(const Atom *) const;
00709 };
00710
00711 // Is atom classified as an RNA receptor "anionic" atom
00712 // For our purposes, these are N7,O2,O4,O6,O1P and O2P
00713 // DM 22 Dec 1998 - center phosphate interactions on the P atom, not on O1P and
00714 // O2P to avoid double counting of favourable interactions class
00715 // isAtomRNAAnionic : public std::function<bool(Atom *)>  {
00716 //  isAtomRNA bIsAtomRNA;
00717 // public:
00718 //  explicit isAtomRNAAnionic() {}
00719 //  Bool operator() (AtomPtr) const;
00720 //};
00721
00722 // DM 22 Dec 1998 - need ligand anionic atoms so we can include repulsive
00723 // interactions with the RNA
00724 //
00725 // Is atom classified as a ligand "anionic" atom
00726 // Includes COO-, PO3-
00727 // class isAtomLigandAnionic : public std::function<bool(Atom *)>  {
00728 // public:
00729 //  explicit isAtomLigandAnionic() {}
00730 //  Bool operator() (AtomPtr) const;
00731 //};
00732
00733 // DM 22 Dec 1998 - renamed from isAtomRNACationic
00734 //
00735 // Is atom classified as a ligand "cationic" atom
00736 // The most important category is charged Nitrogens
00737 // class isAtomLigandCationic : public std::function<bool(Atom *)>  {
00738 // public:
00739 //  explicit isAtomLigandCationic() {}
00740 //  Bool operator() (AtomPtr) const;
00741 //};
00742
00744 // DM 30 Mar 1999
00745 // Now that the MolecularFileSources define the GroupCharge attribute in a
00746 // file-format independent manner, we don't need separate predicates for
00747 // isRNAAnionic, isLigandAnionic etc We just need isIonic, isCationic, isAnionic
00748
00749 // Is atom defined as an ionic interaction center ?
00750 // Checks if group charge is != zero
00751 class isAtomIonic : public std::function<bool(Atom *)> {
00752 public:
00753   explicit isAtomIonic() = default;
00754   bool operator()(const Atom *pAtom) const {
00755     return (std::fabs(pAtom->GetGroupCharge()) > 0.001);
00756   }
00757 };
00758
00759 // Is atom defined as a cationic interaction center ?
00760 // Checks if group charge is > zero
00761 class isAtomCationic : public std::function<bool(Atom *)> {
00762 public:
00763   explicit isAtomCationic() = default;
00764   bool operator()(const Atom *pAtom) const {
00765     return (pAtom->GetGroupCharge() > 0.001);
00766   }
00767 };
00768
00769 // Is atom defined as an anionic interaction center ?
00770 // Checks if group charge is < zero
00771 class isAtomAnionic : public std::function<bool(Atom *)> {
00772 public:
00773   explicit isAtomAnionic() = default;
00774   bool operator()(const Atom *pAtom) const {
00775     return (pAtom->GetGroupCharge() < -0.001);
00776   }
00777 };
00778
00779 // Is atom the central carbon in a guanidinium group ?
00780 // Checks for cationic sp2/arom carbon
00781 // DM 27 Jan 2000 - also check is acyclic
00782 class isAtomGuanidiniumCarbon : public std::function<bool(Atom *)> {
00783   isAtomicNo_eq bIsCarbon;
00784   isAtomCationic bIsCationic;
00785   isPiAtom bIsPi;
00786   isAtomCyclic bIsCyclic;
00787
00788 public:
00789   explicit isAtomGuanidiniumCarbon() : bIsCarbon(6) {}
00790   bool operator()(const Atom *pAtom) const {
```

```
00791     return bIsCationic(pAtom) && bIsCarbon(pAtom) && bIsPi(pAtom) &&
00792            !bIsCyclic(pAtom);
00793   }
00794 };
00795
00796 // DM 24 Jan 2001
00797 // Checks for common metal ions by atomic number
00798 // DM 19 Oct 2001 - extend to all common metals (Na,Mg,K->Zn)
00799 class isAtomMetal : public std::function<bool(Atom *)> {
00800 public:
00801   RBTDLL_EXPORT explicit isAtomMetal();
00802   RBTDLL_EXPORT bool operator()(const Atom *pAtom) const;
00803 };
00804
00805 // DM 21 Jul 1999 Is atom lipophilic ?
00806 // DM 16 May 2003 Total rewrite to be much more comprehensive
00807 class isAtomLipophilic : public std::function<bool(Atom *)> {
00808   isAtomIonic isIonic;
00809   isAtomHBondDonor isHBD;
00810   isAtomHBondAcceptor isHBA;
00811   isAtomMetal isMetal;
00812   isHybridState_eq isSP3;
00813   isHybridState_eq isSP2;
00814   isAtomicNo_eq isO;
00815   isAtomicNo_eq isN;
00816
00817 public:
00818   explicit isAtomLipophilic()
00819       : isSP3(Atom::SP3), isSP2(Atom::SP2), isO(8), isN(7) {}
00820   RBTDLL_EXPORT bool operator()(Atom *pAtom) const;
00821 };
00822
00824
00825 // DM 6 April 1999
00826 // Is coordination number of atom equal to n
00827 // First constructor checks total coordination number
00828 // Other constructors check atomic number, force field type and hybridisation
00829 // state coordination numbers
00830 class isCoordinationNumber_eq : public std::function<bool(Atom *)> {
00831   enum {
00832     TOTAL,
00833     ATNO,
00834     FFTYPE,
00835     HYBRID
00836   } eCNType;       // Type of coordination number to check
00837   unsigned int n; // Coordination number value to check
00838   int atNo;
00839   std::string ffType;
00840   Atom::eHybridState hybrid;
00841
00842 public:
00843   // Total coordination number
00844   explicit isCoordinationNumber_eq(unsigned int nn)
00845       : eCNType(TOTAL), n(nn), atNo(0), ffType(""), hybrid(Atom::UNDEFINED) {}
00846   // Atomic number coordination number
00847   explicit isCoordinationNumber_eq(unsigned int nn, int nAt)
00848       : eCNType(ATNO), n(nn), atNo(nAt), ffType(""), hybrid(Atom::UNDEFINED) {}
00849   // Force field type coordination number
00850   explicit isCoordinationNumber_eq(unsigned int nn, std::string strType)
00851       : eCNType(FFTYPE), n(nn), atNo(0), ffType(std::move(strType)),
00852         hybrid(Atom::UNDEFINED) {}
00853   // Hybridisation state coordination number
00854   explicit isCoordinationNumber_eq(unsigned int nn, Atom::eHybridState eState)
00855       : eCNType(HYBRID), n(nn), atNo(0), ffType(""), hybrid(eState) {}
00856   bool operator()(Atom *pAtom) const {
00857     switch (eCNType) {
00858     case TOTAL:
00859       return pAtom->GetCoordinationNumber() == n;
00860     case ATNO:
00861       return pAtom->GetCoordinationNumber(atNo) == n;
00862     case FFTYPE:
00863       return pAtom->GetCoordinationNumber(ffType) == n;
00864     case HYBRID:
00865       return pAtom->GetCoordinationNumber(hybrid) == n;
00866     default:
00867       return false;
00868     }
00869   }
00870 };
00871
00873 // Comparison functions for sorting Atom* containers
00874 // For use by STL sort algorithms
00876
00877 // Less than operator for sorting Atom*s by atom ID
00878 class AtomPtrCmp_AtomId {
00879 public:
00880   bool operator()(Atom *pAtom1, Atom *pAtom2) const {
```

```
00881     return pAtom1->GetAtomId() < pAtom2->GetAtomId();
00882   }
00883 };
00884
00885 // Less than operator for sorting Atom*s by atomic number
00886 class AtomPtrCmp_AtomicNo {
00887 public:
00888   bool operator()(Atom *pAtom1, Atom *pAtom2) const {
00889     return pAtom1->GetAtomicNo() < pAtom2->GetAtomicNo();
00890   }
00891 };
00892
00893 // Less than operator for sorting Atom*s by atom name
00894 class AtomPtrCmp_AtomName {
00895 public:
00896   bool operator()(Atom *pAtom1, Atom *pAtom2) const {
00897     return pAtom1->GetName() < pAtom2->GetName();
00898   }
00899 };
00900
00901 // Less than operator for sorting AtomPtrs by pointer value
00902 class AtomPtrCmp_Ptr {
00903 public:
00904   bool operator()(Atom *pAtom1, Atom *pAtom2) const { return pAtom1 < pAtom2; }
00905 };
00906
00908 // Functions objects for performing actions on atoms
00909 // For use by STL algorithms
00911
00912 // DM 28 Jul 1999 - extract atom coord (for use by std::transform)
00913 // DM 27 Oct 2000 - return by reference
00914 inline const Coord &ExtractAtomCoord(Atom *pAtom) { return pAtom->GetCoords(); }
00915
00916 // DM 09 Nov 1999 - Accumulate atomic mass (for use by std::accumulate)
00917 inline double AccumAtomicMass(double val, Atom *pAtom) {
00918   return val + pAtom->GetAtomicMass();
00919 }
00920
00921 // DM 11 Nov 1999 - Accumulate mass weighted coords (for use by std::accumulate)
00922 inline Coord AccumMassWeightedCoords(const Coord &val, Atom *pAtom) {
00923   return val + pAtom->GetAtomicMass() * pAtom->GetCoords();
00924 }
00925
00926 // Translate an atom by the supplied vector
00927 class TranslateAtom {
00928   const Vector v;
00929
00930 public:
00931   explicit TranslateAtom(const Vector &vv) : v(vv) {}
00932   void operator()(Atom *pAtom) { pAtom->Translate(v); }
00933 };
00934
00935 // Translate an atom by the supplied vector, but only if the selection flag is
00936 // true
00937 class TranslateAtomIfSelected {
00938   const Vector v;
00939
00940 public:
00941   explicit TranslateAtomIfSelected(const Vector &vv) : v(vv) {}
00942   void operator()(Atom *pAtom) {
00943     if (pAtom->GetSelectionFlag())
00944       pAtom->Translate(v);
00945   }
00946 };
00947
00948 // Rotate an atom by the supplied quaternion
00949 class RotateAtomUsingQuat {
00950   const Quat q;
00951
00952 public:
00953   explicit RotateAtomUsingQuat(const Quat &qq) : q(qq) {}
00954   void operator()(Atom *pAtom) { pAtom->RotateUsingQuat(q); }
00955 };
00956
00957 // Rotate an atom by the supplied quaternion, but only if the selection flag is
00958 // true
00959 class RotateAtomUsingQuatIfSelected {
00960   const Quat q;
00961
00962 public:
00963   explicit RotateAtomUsingQuatIfSelected(const Quat &qq) : q(qq) {}
00964   void operator()(Atom *pAtom) {
00965     if (pAtom->GetSelectionFlag())
00966       pAtom->RotateUsingQuat(q);
00967   }
00968 };
00969
```

```
00970 // Select/deselect the atom
00971 class SelectAtom {
00972   bool b;
00973
00974 public:
00975   explicit SelectAtom(bool bb) : b(bb) {}
00976   void operator()(Atom *pAtom) { pAtom->SetSelectionFlag(b); }
00977 };
00978
00979 // Invert the atom selection flag
00980 class InvertSelectAtom {
00981 public:
00982   explicit InvertSelectAtom() = default;
00983   void operator()(Atom *pAtom) { pAtom->InvertSelectionFlag(); }
00984 };
00985
00986 // Select the intramolecular flexible interactions to this atom (sets
00987 // SelectionFlag = true)
00988 class SelectFlexAtoms {
00989 public:
00990   explicit SelectFlexAtoms() = default;
00991   void operator()(Atom *pAtom);
00992 };
00993
00994 // Set/unset the cyclic flag
00995 class CyclicAtom {
00996   bool b;
00997
00998 public:
00999   explicit CyclicAtom(bool bb) : b(bb) {}
01000   void operator()(Atom *pAtom) { pAtom->SetCyclicFlag(b); }
01001 };
01002
01004 // Atom list functions (implemented as STL algorithms)
01006
01007 // Unary
01008
01009 // Generic template version of GetNumAtoms, passing in your own predicate
01010 template <class Predicate>
01011 unsigned int GetNumAtomsWithPredicate(const AtomList &atomList,
01012                                       const Predicate &pred) {
01013   return std::count_if(atomList.begin(), atomList.end(), pred);
01014 }
01015
01016 // Generic template version of GetAtomList, passing in your own predicate
01017 template <class Predicate>
01018 AtomList GetAtomListWithPredicate(const AtomList &atomList,
01019                                   const Predicate &pred) {
01020   AtomList newAtomList;
01021   std::copy_if(atomList.begin(), atomList.end(),
01022                std::back_inserter(newAtomList), pred);
01023   return newAtomList;
01024 }
01025
01026 // Generic template version of FindAtomInList, passing in your own predicate
01027 template <class Predicate>
01028 AtomListIter FindAtomInList(AtomList &atomList, const Predicate &pred) {
01029   return std::find_if(atomList.begin(), atomList.end(), pred);
01030 }
01031
01032 // Selected atoms
01033 void SetAtomSelectionFlagsInList(AtomList &atomList, bool bSelected = true);
01034 void InvertAtomSelectionFlags(AtomList &atomList); // DM 08 Jan 1999
01035
01036 inline unsigned int GetNumSelectedAtomsInList(const AtomList &atomList) {
01037   return GetNumAtomsWithPredicate(atomList, isAtomSelected());
01038 }
01039
01040 inline AtomList GetSelectedAtomsFromList(const AtomList &atomList) {
01041   return GetAtomListWithPredicate(atomList, isAtomSelected());
01042 }
01043
01044 // Cyclic atoms
01045 void SetAtomCyclicFlagsInList(AtomList &atomList, bool bCyclic = true);
01046
01047 inline unsigned int GetNumCyclicAtomsInList(const AtomList &atomList) {
01048   return GetNumAtomsWithPredicate(atomList, isAtomCyclic());
01049 }
01050 inline AtomList GetCyclicAtomsFromList(const AtomList &atomList) {
01051   return GetAtomListWithPredicate(atomList, isAtomCyclic());
01052 }
01053
01054 // Hydrogen bond acceptor atoms
01055 inline unsigned int GetNumHBondAcceptorAtomsInList(const AtomList &atomList) {
01056   return GetNumAtomsWithPredicate(atomList, isAtomHBondAcceptor());
01057 }
01058 inline AtomList GetHBondAcceptorAtomsFromList(const AtomList &atomList) {
```

```
01059   return GetAtomListWithPredicate(atomList, isAtomHBondAcceptor());
01060 }
01061
01062 // Hydrogen bond donor atoms
01063 inline unsigned int GetNumHBondDonorAtomsInList(const AtomList &atomList) {
01064   return GetNumAtomsWithPredicate(atomList, isAtomHBondDonor());
01065 }
01066 inline AtomList GetHBondDonorAtomsFromList(const AtomList &atomList) {
01067   return GetAtomListWithPredicate(atomList, isAtomHBondDonor());
01068 }
01069
01070 //(Formally) charged atoms
01071 inline unsigned int GetNumChargedAtomsInList(const AtomList &atomList) {
01072   return GetNumAtomsWithPredicate(atomList, isAtomCharged());
01073 }
01074 inline AtomList GetChargedAtomsFromList(const AtomList &atomList) {
01075   return GetAtomListWithPredicate(atomList, isAtomCharged());
01076 }
01077
01078 // Planar atoms
01079 inline unsigned int GetNumPlanarAtomsInList(const AtomList &atomList) {
01080   return GetNumAtomsWithPredicate(atomList, isAtomPlanar());
01081 }
01082 inline AtomList GetPlanarAtomsFromList(const AtomList &atomList) {
01083   return GetAtomListWithPredicate(atomList, isAtomPlanar());
01084 }
01085
01086 // RNA anionic atoms
01087 // inline UInt GetNumRNAAnionicAtoms(const AtomList& atomList) {
01088 //  return
01089 //  GetNumAtomsWithPredicate(atomList,isAtomRNAAnionic());
01090 //}
01091 // inline AtomList GetRNAAnionicAtomList(const AtomList& atomList) {
01092 //  return
01093 //  GetAtomListWithPredicate(atomList,isAtomRNAAnionic());
01094 //}
01095
01096 // Ligand anionic atoms
01097 // DM 22 Dec 1998
01098 // inline UInt GetNumLigandAnionicAtoms(const AtomList& atomList) {
01099 //  return
01100 //  GetNumAtomsWithPredicate(atomList,isAtomLigandAnionic());
01101 //}
01102 // inline AtomList GetLigandAnionicAtomList(const AtomList& atomList) {
01103 //  return
01104 //  GetAtomListWithPredicate(atomList,isAtomLigandAnionic());
01105 //}
01106
01107 // Ligand cationic atoms
01108 // DM 22 Dec 1998 -renamed from Get..RNACationic..
01109 // inline UInt GetNumLigandCationicAtoms(const AtomList& atomList) {
01110 //  return
01111 //  GetNumAtomsWithPredicate(atomList,isAtomLigandCationic());
01112 //}
01113 // inline AtomList GetLigandCationicAtomList(const AtomList& atomList) {
01114 //  return
01115 //  GetAtomListWithPredicate(atomList,isAtomLigandCationic());
01116 //}
01117
01118 // Guanidinium carbons
01119 // inline UInt GetNumGuanidiniumCarbonAtoms(const AtomList& atomList) {
01120 //  return
01121 //  GetNumAtomsWithPredicate(atomList,isAtomGuanidiniumCarbon());
01122 //}
01123 // inline AtomList GetGuanidiniumCarbonAtomList(const AtomList& atomList)
01124 // {
01125 //  return
01126 //  GetAtomListWithPredicate(atomList,isAtomGuanidiniumCarbon());
01127 //}
01128
01129 // BondedAtoms
01130 // Helper function to return the atom pointer for the "other" atom in the bond
01131 // i.e. return (bondPair.second) ? bondPair.first->GetAtom2Ptr() :
01132 // bondPair.first->GetAtom1Ptr();
01133 AtomPtr GetBondedAtomPtr(std::pair<Bond *, bool> bondBoolPair);
01134
01135 // This pair of functions is a little different to the above
01136 // They convert a BondMap to an AtomList. The BondMap is assumed to
01137 // contain all bonds to a particular atom GetBondedAtomPtr is used to get the
01138 // "other" atom in the pair.
01139 unsigned int GetNumBondedAtoms(const BondMap &bondMap);
01140 AtomList GetBondedAtomList(const BondMap &bondMap);
01141
01142 // If spAtom is a regular Atom, these two functions just behave like the two
01143 // above If spAtom can be dynamically_casted to an PseudoAtom, these return
01144 // the constituent atom list for the pseudoatom
01145 unsigned int GetNumBondedAtoms(const Atom *pAtom);
```

```
01146 RBTDLL_EXPORT AtomList GetBondedAtomList(const Atom *pAtom);
01147
01148 // Binary
01149
01150 // Atoms with atomic no = nAtomicNo
01151 inline unsigned int GetNumAtomsWithAtomicNo_eq(const AtomList &atomList,
01152                                                int nAtomicNo) {
01153   return GetNumAtomsWithPredicate(atomList, isAtomicNo_eq(nAtomicNo));
01154 }
01155 inline AtomList GetAtomListWithAtomicNo_eq(const AtomList &atomList,
01156                                            int nAtomicNo) {
01157   return GetAtomListWithPredicate(atomList, isAtomicNo_eq(nAtomicNo));
01158 }
01159
01160 // Atoms with FFType = strFFType
01161 inline unsigned int GetNumAtomsWithFFType_eq(const AtomList &atomList,
01162                                              std::string strFFType) {
01163   return GetNumAtomsWithPredicate(atomList, isFFType_eq(std::move(strFFType)));
01164 }
01165 inline AtomList GetAtomListWithFFType_eq(const AtomList &atomList,
01166                                          std::string strFFType) {
01167   return GetAtomListWithPredicate(atomList, isFFType_eq(std::move(strFFType)));
01168 }
01169
01170 // Atoms with AtomName = strAtomName
01171 inline unsigned int GetNumAtomsWithAtomName_eq(const AtomList &atomList,
01172                                                std::string strAtomName) {
01173   return GetNumAtomsWithPredicate(atomList,
01174                                   isAtomName_eq(std::move(strAtomName)));
01175 }
01176 inline AtomList GetAtomListWithAtomName_eq(const AtomList &atomList,
01177                                            std::string strAtomName) {
01178   return GetAtomListWithPredicate(atomList,
01179                                   isAtomName_eq(std::move(strAtomName)));
01180 }
01181
01182 // DM 1 Feb 1999
01183 // Attempts to match atoms in atomList2 with those in atomList1, where the match
01184 // is performed using isAtom_eq (tests subunit ID, subunit name and atom name
01185 // for equality) and NOT isAtomPtr_eq (which tests the Atom* pointers
01186 // themselves
01187 // for equality, i.e. same objects) Returns list of Atom smart pointers to
01188 // atoms in atomList1 for which a match is found.
01189 unsigned int GetNumMatchingAtoms(const AtomList &atomList1,
01190                                  const AtomList &atomList2);
01191 AtomList GetMatchingAtomList(const AtomList &atomList1,
01192                              const AtomList &atomList2);
01193
01194 // DM 30 Mar 1999
01195 // These versions match atoms against a full atom name specifier of the form
01196 // given by GetFullAtomName(): <segment name>:<subunit name>_<subunit id>:<atom
01197 // name>  e.g. A:U_23:O4 All fields are optional, and if missing, will match all
01198 // atoms So, for example, these are all valid full names: N7     (equiv. to
01199 // ::N7)
01200 // - matches N7's in all subunits, all segments A::O4                    -
01201 // matches O4's in all subunits in segment A U:O4 (equiv. to :U:O4)  - matches
01202 // O4's in subunits named U in all segments U_23:O4                 - matches
01203 // O4's in subunit U23 in all segments _23:                    - matches all
01204 // atoms in
01205 // subunit ID=23 in all segments
01206 unsigned int GetNumMatchingAtoms(const AtomList &atomList,
01207                                  const std::string &strFullName);
01208 AtomList GetMatchingAtomList(const AtomList &atomList,
01209                              const std::string &strFullName);
01210
01211 // DM 15 Apr 1999 - as above, but match against a list of full atom name
01212 // specifiers Returns total list (i.e. all matches OR'd). Does not remove
01213 // duplicates.
01214 unsigned int GetNumMatchingAtoms(const AtomList &atomList,
01215                                  const std::vector<std::string> &fullNameList);
01216 AtomList GetMatchingAtomList(const AtomList &atomList,
01217                              const std::vector<std::string> &fullNameList);
01218
01220 // DM 07 Jan 1999
01221 // Actions on atom lists
01223 // Translate all atoms in the list by the supplied vector
01224 inline void TranslateAtoms(const AtomList &atomList, const Vector &v) {
01225   std::for_each(atomList.begin(), atomList.end(), TranslateAtom(v));
01226 }
01227
01228 // Translate all selected atoms in the list by the supplied vector
01229 inline void TranslateSelectedAtoms(const AtomList &atomList, const Vector &v) {
01230   std::for_each(atomList.begin(), atomList.end(), TranslateAtomIfSelected(v));
01231 }
01232
01233 // Rotate all atoms in the list by the supplied quaternion
01234 inline void RotateAtomsUsingQuat(const AtomList &atomList, const Quat &q) {
```

```
01235   std::for_each(atomList.begin(), atomList.end(), RotateAtomUsingQuat(q));
01236 }
01237
01238 // Rotate all selected atoms in the list by the supplied quaternion
01239 inline void RotateSelectedAtomsUsingQuat(const AtomList &atomList,
01240                                          const Quat &q) {
01241   std::for_each(atomList.begin(), atomList.end(),
01242                 RotateAtomUsingQuatIfSelected(q));
01243 }
01244
01245 // Save coords by number for all atoms in the list
01246 void SaveAtomCoords(const AtomList &atomList, unsigned int coordNum = 0);
01247 // Revert to numbered coords for all atoms in the list
01248 void RevertAtomCoords(const AtomList &atomList, unsigned int coordNum = 0);
01249
01250 // DM 9 Nov 1999
01251 // Returns total atomic mass (molecular weight) for all atoms in the list
01252 double GetTotalAtomicMass(const AtomList &atomList);
01253
01254 // Returns center of mass of atoms in the list
01255 Coord GetCenterOfAtomicMass(const AtomList &atomList);
01256
01257 // DM 20 May 1999 - returns the coords of all atoms in the list
01258 CoordList GetCoordList(const AtomList &atomList);
01259 // DM 09 Aug 2001 - returns coordList via argument
01260 RBTDLL_EXPORT void GetCoordList(const AtomList &atomList, CoordList &coordList);
01261
01262 // Streams an atom list in Quanta CSD file format (for easy color coding of
01263 // selected atoms in Quanta) nFormat  = 0 => Receptor atom format: "zone 1 #
01264 // pick O5T = col 2" nFormat != 0 => Ligand atom format "pick N1 .and. segm H =
01265 // col 2" std::ostream should have been opened before calling
01266 // PrintQuantaCSDFormat
01267 void PrintQuantaCSDFormat(const AtomList &atomList, std::ostream &s,
01268                           int nColor = 2, int nFormat = 0);
01269
01270 // DM 25 Feb 1999
01271 // Modified DM 6 Apr 1999 to operate using the GroupCharge attribute
01272 // Scans an atom list and sets the group charge to zero on any
01273 // cationic-ionic pairs which are 1-2 or 1-3 connected
01274 //(e.g. on -OOC.CR.NH3+ would zero the C and N)
01275 void RemoveZwitterions(AtomList &atomList);
01276
01277 } // namespace rxdock
01278
01279 #endif //_RBTATOM_H_
```

6.6 AtomFuncs.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Common functions involving atoms and bonds
00014
00015 #ifndef _RBTATOMFUNCS_H_
00016 #define _RBTATOMFUNCS_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/Bond.h"
00020
00021 namespace rxdock {
00022
00023 // DM 31 Oct 2000
00024 // Given a bond, determines if it is in a ring (cutdown version of ToSpin)
00025 bool FindCyclic(BondPtr spBond, AtomList &atomList, BondList &bondList);
00026
00027 // DM 4 Dec 1998
00028 // Given a bond, set the selection flags for all atoms which are connected to
00029 // atom 2 of the bond Returns true if bond is in a ring (i.e. if atom 1's flag
00030 // gets set also) DM 8 Feb 2000 - standalone version (formerly only available as
00031 // Model method) WARNING - no check that spBond is actually present in
00032 // bondList, or that atom and bond lists are consistent
00033 bool ToSpin(BondPtr spBond, AtomList &atomList, BondList &bondList);
00034
00035 // DM 7 Dec 1998
```

```
00036 // Set the atom and bond cyclic flags for all atoms and bonds in the model
00037 // DM 8 Feb 2000 - standalone version (formerly only available as Model
00038 // method)
00039 void SetAtomAndBondCyclicFlags(AtomList &atomList, BondList &bondList);
00040
00041 // Find the smallest ring containing the given atom
00042 // Assumes SetAtomAndBondCyclicFlags has already been called
00043 // Note: if the atom is a member of two equally sized rings, only one will be
00044 // returned 30 Oct 2000 - Standalone version
00045 AtomList FindRing(AtomPtr spAtom, BondList &bondList);
00046
00047 // 30 Oct 2000 (DM) - Find all rings, standalone version
00048 void FindRings(AtomList &atomList, BondList &bondList, AtomListList &ringList);
00049
00050 } // namespace rxdock
00051
00052 #endif //_RBTATOMFUNCS_H_
```

6.7 BaseBiMolTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base class for bimolecular transforms. Overrides Update()
00014
00015 #ifndef _RBTBASEBIMOLTRANSFORM_H_
00016 #define _RBTBASEBIMOLTRANSFORM_H_
00017
00018 #include "rxdock/BaseTransform.h"
00019 #include "rxdock/Model.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class BaseBiMolTransform : public BaseTransform {
00028 public:
00029   // Class type string
00030   static const std::string _CT;
00032   // Constructors/destructors
00033   virtual ~BaseBiMolTransform();
00034
00035   friend void to_json(json &j, const BaseBiMolTransform &baseBiMolTrans);
00036   friend void from_json(const json &j, BaseBiMolTransform &baseBiMolTrans);
00037
00039   // Public methods
00041
00042   ModelPtr GetReceptor() const;
00043   ModelPtr GetLigand() const;
00044   ModelList GetSolvent() const;
00045
00046   // Override Observer pure virtual
00047   // Notify observer that subject has changed
00048   virtual void Update(Subject *theChangedSubject);
00049
00050 protected:
00052   // Protected methods
00054   BaseBiMolTransform(const std::string &strClass, const std::string &strName);
00055
00056   // PURE VIRTUAL - Derived classes must override
00057   virtual void SetupReceptor() = 0; // Called by Update when receptor is changed
00058   virtual void SetupLigand() = 0;   // Called by Update when ligand is changed
00059   virtual void SetupSolvent() {}    // Called by Update when ligand is changed
00060   virtual void
00061   SetupTransform() = 0; // Called by Update when either model has changed
00062
00063 private:
00065   // Private methods
00067
00068 protected:
00070   // Protected data
00072
```

```
00073 private:
00075   // Private data
00077   ModelPtr m_spReceptor;
00078   ModelPtr m_spLigand;
00079   ModelList m_solventList;
00080 };
00081
00082 void to_json(json &j, const BaseBiMolTransform &baseBiMolTrans);
00083 void from_json(const json &j, BaseBiMolTransform &baseBiMolTrans);
00084
00085 } // namespace rxdock
00086
00087 #endif //_RBTBASEBIMOLTRANSFORM_H_
```

6.8 BaseFileSink.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract base class for text file-based output. Knows how to Open a file,
00014 // Write a text cache to the file, and Close the file.
00015 // It is up to derived classes to provide a Render method to populate the cache
00016 // with text records of the appropriate format.
00017
00018 #ifndef _RBTBASEFILESINK_H_
00019 #define _RBTBASEFILESINK_H_
00020
00021 #include <fstream>
00022
00023 #include "rxdock/Config.h"
00024
00025 #include <nlohmann/json.hpp>
00026
00027 using json = nlohmann::json;
00028
00029 namespace rxdock {
00030
00031 class BaseFileSink {
00032 public:
00034   // Constructors/destructors
00035   // BaseFileSink(const char* fileName);
00036   BaseFileSink(const std::string &fileName);
00037
00038   virtual ~BaseFileSink(); // Default destructor
00039
00040   friend void to_json(json &j, const BaseFileSink &baseFileSink);
00041   friend void from_json(const json &j, BaseFileSink &baseFileSink);
00042
00044   // Public methods
00046   std::string GetFileName() const { return m_strFileName; }
00047   // void SetFileName(const char* fileName);
00048   void SetFileName(const std::string &fileName);
00049   bool StatusOK() { return Status().isOK(); }
00050   Error Status();
00051
00052   // PURE VIRTUAL - MUST BE OVERRIDDEN IN DERIVED CLASSES
00053   virtual void Render() = 0;
00054
00055 protected:
00057   // Protected methods
00059   // Write the cache to the file
00060   // DM 11 Feb 1999 - add flag to allow the cache to be written without clearing
00061   // it
00062   void Write(bool bClearCache = true);
00063   // Add a complete line to the cache
00064   void AddLine(const std::string &fileRec);
00065   // Replace a complete line in the cache
00066   void ReplaceLine(const std::string &fileRec, unsigned int nRec);
00067   // Is cache empty
00068   bool isCacheEmpty() const { return m_lineRecs.empty(); }
00069
00070   // DM 06 Apr 1999 - append attribute is for derived class use only
00071   bool GetAppend() const {
00072     return m_bAppend;
```

```
00073   } // Get append status (true=append, false=overwrite)
00074   void SetAppend(bool bAppend) {
00075     m_bAppend = bAppend;
00076   } // Set append status (true=append, false=overwrite)
00077
00078 private:
00080   // Private methods
00082
00083   BaseFileSink();                     // Disable default constructor
00084   BaseFileSink(const BaseFileSink &); // Copy constructor disabled by default
00085   BaseFileSink &
00086   operator=(const BaseFileSink &); // Copy assignment disabled by default
00087
00088   void Open(bool bAppend = false);
00089   void Close();
00090   void ClearCache();
00091
00092 protected:
00094   // Protected data
00096
00097 private:
00099   // Private data
00101   std::vector<std::string> m_lineRecs;
00102   std::string m_strFileName;
00103   std::ofstream m_fileOut;
00104   bool m_bAppend; // If true, Write() appends to file rather than overwriting
00105 };
00106
00107 void to_json(json &j, const BaseFileSink &baseFileSink);
00108 void from_json(const json &j, BaseFileSink &baseFileSink);
00109
00110 // Useful typedefs
00111 typedef SmartPtr<BaseFileSink> BaseFileSinkPtr; // Smart pointer
00112
00113 } // namespace rxdock
00114
00115 #endif //_RBTBASEFILESINK_H_
```

6.9 BaseFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract base class for file-based input
00014
00015 #ifndef _RBTBASEFILESOURCE_H_
00016 #define _RBTBASEFILESOURCE_H_
00017
00018 #include <fstream>
00019
00020 #include "rxdock/Config.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 // useful typedefs
00029 typedef std::string FileRec;
00030 typedef std::vector<FileRec> FileRecList;
00031 typedef FileRecList::iterator FileRecListIter;
00032
00033 // Max line length expected in file
00034 const int MAXLINELENGTH = 255;
00035
00036 class BaseFileSource {
00037 public:
00038   // Constructors
00039   // BaseFileSource(const char* fileName);
00040   BaseFileSource(const std::string &fileName);
00041   BaseFileSource(const std::string &fileName, const std::string &strRecDelim);
00042
00043   // Default destructor
```

```
00044   virtual ~BaseFileSource();
00045
00046   friend void to_json(json &j, const BaseFileSource &baseFileSource);
00047   friend void from_json(const json &j, BaseFileSource &baseFileSource);
00048
00049   // Public methods
00050
00051   RBTDLL_EXPORT std::string GetFileName();
00052   // void SetFileName(const char* fileName);
00053   void SetFileName(const std::string &fileName);
00054
00055   // Status and StatusOK parse the file to check for errors
00056   bool StatusOK();
00057   RBTDLL_EXPORT Error Status();
00058
00059   // FileStatus and FileStatusOK just try and read the file
00060   RBTDLL_EXPORT bool FileStatusOK();
00061   Error FileStatus();
00062
00063   // Multi-record methods
00064   bool isMultiRecordSupported() { return m_bMultiRec; }
00065   RBTDLL_EXPORT void NextRecord();
00066   void Rewind();
00067   RBTDLL_EXPORT std::size_t GetEstimatedNumRecords();
00068
00069 protected:
00071   // Protected member functions
00072   // PURE VIRTUAL - MUST BE OVERRIDDEN IN DERIVED CLASSES
00073   virtual void Parse() = 0;
00074   // chance to give false when record delimiter at the beginning
00075   void Read(bool aDelimiterAtEnd = true);
00077
00078   // Protected data
00079 protected:
00080   bool m_bParsedOK;       // For use by Parse
00081   FileRecList m_lineRecs; // Allow direct access to cache from derived
00082                           // classes
00083
00084   // Private member functions
00085 private:
00086   BaseFileSource(); // Disable default constructor
00087   BaseFileSource(
00088       const BaseFileSource &); // Copy constructor disabled by default
00089   BaseFileSource &
00090   operator=(const BaseFileSource &); // Copy assignment disabled by default
00091   void Open();
00092   void Close();
00093   void ClearCache();
00094
00095   // Private data
00096 private:
00097   std::string m_strFileName;
00098   std::size_t m_fileSize;
00099   bool m_bReadOK; // For use by Read
00100   std::size_t m_numReads;
00101   std::size_t m_bytesRead;
00102   std::ifstream m_fileIn;
00103   char *m_szBuf;    // Line buffer
00104   bool m_bFileOpen; // Keep track of whether we've opened the file or not
00105   bool m_bMultiRec; // Is file multi-record ?
00106   std::string m_strRecDelim; // Record delimiter
00107 };
00108
00109 void to_json(json &j, const BaseFileSource &baseFileSource);
00110 void from_json(const json &j, BaseFileSource &baseFileSource);
00111
00112 } // namespace rxdock
00113
00114 #endif //_RBTBASEFILESOURCE_H_
```

6.10 BaseGrid.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
```

```
00012
00013 // Interface to all grid classes. Provides conversion methods between different
00014 // addressing methods: (iXYZ),(iX,iY,iZ), and (x,y,z) coords. Underlying data
00015 // values are managed by subclasses.
00016
00017 #ifndef _RBTBASEGRID_H_
00018 #define _RBTBASEGRID_H_
00019
00020 #include "rxdock/Config.h"
00021 #include "rxdock/Coord.h"
00022
00023 #include <set>
00024
00025 namespace rxdock {
00026
00027 class BaseGrid {
00028 public:
00029   // Class type string
00030   static const std::string _CT;
00031
00033   // Constructors/destructors
00034   // Construct a NXxNYxNZ grid running from gridMin at gridStep resolution
00035   BaseGrid(const Coord &gridMin, const Vector &gridStep, unsigned int NX,
00036            unsigned int NY, unsigned int NZ, unsigned int NPad = 0);
00037   // Constructor reading all params from binary stream
00038   BaseGrid(json j);
00039   BaseGrid(const BaseGrid &);            // Copy constructor
00040   BaseGrid &operator=(const BaseGrid &); // Copy assignment
00041   virtual ~BaseGrid();                   // Default destructor
00042
00043   // Friend functions
00044   // Insertion operator (primarily for debugging)
00045   friend std::ostream &operator<<(std::ostream &s, const BaseGrid &grid);
00046   friend void to_json(json &j, const BaseGrid &grid);
00047   friend void from_json(const json &j, BaseGrid &grid);
00048
00050   // Virtual functions for reading/writing grid data to streams in
00051   // text and binary format
00052   // Subclasses should provide their own private OwnPrint
00053   // method to handle subclass data members, and override the public
00054   // Print method
00055   virtual void Print(std::ostream &ostr) const; // Text output
00056
00058   // Public methods
00060
00062   // Get attribute functions
00064
00065   unsigned int GetNX() const { return m_NX; }
00066   unsigned int GetNY() const { return m_NY; }
00067   unsigned int GetNZ() const { return m_NZ; }
00068   unsigned int GetN() const { return m_N; }
00069   unsigned int GetStrideX() const { return m_SX; }
00070   unsigned int GetStrideY() const { return m_SY; }
00071   unsigned int GetStrideZ() const { return m_SZ; }
00072   int GetnXMin() const { return m_nXMin; }
00073   int GetnYMin() const { return m_nYMin; }
00074   int GetnZMin() const { return m_nZMin; }
00075   int GetnXMax() const { return m_nXMax; }
00076   int GetnYMax() const { return m_nYMax; }
00077   int GetnZMax() const { return m_nZMax; }
00078
00079   const Coord &GetGridMin() const { return m_min; }
00080   const Coord &GetGridMax() const { return m_max; }
00081   const Vector &GetGridStep() const { return m_step; }
00082   Coord GetGridCenter() const { return (GetGridMax() + GetGridMin()) / 2.0; }
00083   Coord GetGridSize() const { return GetGridMax() - GetGridMin(); }
00084   unsigned int GetPad() const { return m_NPad; }
00085   const Coord &GetPadMin() const { return m_padMin; }
00086   const Coord &GetPadMax() const { return m_padMax; }
00087
00089   // Set attribute functions
00091
00092   // Set grid min without changing the grid step
00093   void SetGridMin(const Coord &gridMin);
00094   // Translate the grid by the given vector
00095   void TranslateGrid(const Vector &vec) { SetGridMin(GetGridMin() + vec); }
00096   // Set the center of the grid to the given coord
00097   void SetGridCenter(const Coord &gridCenter) {
00098     TranslateGrid(gridCenter - GetGridCenter());
00099   }
00100   // Set grid step
00101   void SetGridStep(const Vector &gridStep);
00102   // Set a pad region (cuboid inside the grid which defines the valid coords)
00103   void SetPad(unsigned int NPad = 0);
00104
00106   // Bounds checking and index conversion routines
00108
```

```
00109   // When checking coords, take account of pad region
00110   bool isValid(const Coord &c) const {
00111     return (c >= m_padMin) && (c < m_padMax);
00112   }
00113   // When checking 3-D indices, take account of pad region
00114   bool isValid(unsigned int iX, unsigned int iY, unsigned int iZ) const {
00115     return (iX >= m_NPad) && (iX < m_NX - m_NPad) && (iY >= m_NPad) &&
00116            (iY < m_NY - m_NPad) && (iZ >= m_NPad) && (iZ < m_NZ - m_NPad);
00117   }
00118   // When checking 1-D index, don't take account of pad region
00119   bool isValid(unsigned int iXYZ) const { return (iXYZ < m_N); }
00120
00121   // Return iX,iY,iZ indices for given coord
00122   unsigned int GetIX(const Coord &c) const {
00123     return static_cast<unsigned int>((c.xyz(0) - m_min.xyz(0)) / m_step.xyz(0));
00124   }
00125   unsigned int GetIY(const Coord &c) const {
00126     return static_cast<unsigned int>((c.xyz(1) - m_min.xyz(1)) / m_step.xyz(1));
00127   }
00128   unsigned int GetIZ(const Coord &c) const {
00129     return static_cast<unsigned int>((c.xyz(2) - m_min.xyz(2)) / m_step.xyz(2));
00130   }
00131
00132   // Return iX,iY,iZ indices for given coord
00133   unsigned int GetIX(double x) const {
00134     return static_cast<unsigned int>((x - m_min.xyz(0)) / m_step.xyz(0));
00135   }
00136   unsigned int GetIY(double y) const {
00137     return static_cast<unsigned int>((y - m_min.xyz(1)) / m_step.xyz(1));
00138   }
00139   unsigned int GetIZ(double z) const {
00140     return static_cast<unsigned int>((z - m_min.xyz(2)) / m_step.xyz(2));
00141   }
00142
00143   // Return iX,iY,iZ indices for given iXYZ index
00144   unsigned int GetIX(unsigned int iXYZ) const { return iXYZ / m_SX; }
00145   unsigned int GetIY(unsigned int iXYZ) const { return (iXYZ % m_SX) / m_SY; }
00146   unsigned int GetIZ(unsigned int iXYZ) const { return (iXYZ % m_SY) / m_SZ; }
00147
00148   // Return iXYZ index for given iX,iY,iZ indices
00149   unsigned int GetIXYZ(unsigned int iX, unsigned int iY,
00150                        unsigned int iZ) const {
00151     return iX * m_SX + iY * m_SY + iZ * m_SZ;
00152   }
00153   // Return iXYZ index for given coord
00154   unsigned int GetIXYZ(const Coord &c) const {
00155     return GetIXYZ(GetIX(c), GetIY(c), GetIZ(c));
00156   }
00157
00158   // Returns real-world coordinate for given iX,iY,iZ indices
00159   Coord GetCoord(unsigned int iX, unsigned int iY, unsigned int iZ) const {
00160     return (Coord(m_nXMin, m_nYMin, m_nZMin) + Coord(iX, iY, iZ)) * m_step;
00161   }
00162   // Returns real-world X coordinate for given iX index
00163   double GetXCoord(unsigned int iX) const {
00164     return (m_nXMin + static_cast<int>(iX)) * m_step.xyz(0);
00165   }
00166   // Returns real-world Y coordinate for given iY index
00167   double GetYCoord(unsigned int iY) const {
00168     return (m_nYMin + static_cast<int>(iY)) * m_step.xyz(1);
00169   }
00170   // Returns real-world Z coordinate for given iZ index
00171   double GetZCoord(unsigned int iZ) const {
00172     return (m_nZMin + static_cast<int>(iZ)) * m_step.xyz(2);
00173   }
00174
00175   // Returns real-world coordinate for given iXYZ index
00176   Coord GetCoord(unsigned int iXYZ) const {
00177     return GetCoord(GetIX(iXYZ), GetIY(iXYZ), GetIZ(iXYZ));
00178   }
00179   // Returns list of real-world coordinates for given set of iXYZ indices
00180   CoordList GetCoordList(const std::set<unsigned int> &iXYZSet) const;
00181
00182   // DM 17 May 1999 - returns the set of valid grid points within a sphere of a
00183   // given center and radius DM 17 Jul 2000 - use std::vector<UInt> and
00184   // return by reference, for performance boost
00185   void GetSphereIndices(const Coord &c, double radius,
00186                         std::vector<unsigned int> &sIndices) const;
00187
00188 protected:
00190   // Protected methods
00192
00193   // Protected method for writing data members for this class to text stream
00194   void OwnPrint(std::ostream &ostr) const;
00195
00196 private:
00198   // Private methods
```

```
00200   BaseGrid(); // Disable default constructor
00201
00202   // Helper function called by copy constructor and assignment operator
00203   void CopyGrid(const BaseGrid &);
00204
00205 protected:
00207   // Protected data
00209
00210 private:
00212   // Private data
00214   unsigned int
00215       m_NX; // Number of grid points in X direction (fixed in constructor)
00216   unsigned int
00217       m_NY; // Number of grid points in Y direction (fixed in constructor)
00218   unsigned int
00219       m_NZ; // Number of grid points in Z direction (fixed in constructor)
00220   unsigned int m_N;     // Total number of grid points (fixed in constructor)
00221   unsigned int m_SX;    // Stride of X
00222   unsigned int m_SY;    // Stride of Y
00223   unsigned int m_SZ;    // Stride of Z
00224   Coord m_min;          // Minimum grid coords (real-world units)
00225   Coord m_max;          // Maximum grid coords (real-world units)
00226   Vector m_step;        // Grid step in X,Y,Z (real-world units)
00227   unsigned int m_NPad; // Defines a zero-padded border around the grid
00228   Coord m_padMin;       // Minimum pad coords (used to define an active region of
00229                         // the grid)
00230   Coord m_padMax;       // Maximum pad coords (used to define an active region of
00231                         // the grid)
00232   int m_nXMin;          // Minimum X grid point (in units of grid step)
00233   int m_nXMax;          // Maximum X grid point (in units of grid step)
00234   int m_nYMin;          // Minimum Y grid point (in units of grid step)
00235   int m_nYMax;          // Maximum Y grid point (in units of grid step)
00236   int m_nZMin;          // Minimum Z grid point (in units of grid step)
00237   int m_nZMax;          // Maximum Z grid point (in units of grid step)
00238 };
00239
00240 // Useful typedefs
00241 typedef SmartPtr<BaseGrid> BaseGridPtr; // Smart pointer
00242
00243 void to_json(json &j, const BaseGrid &grid);
00244 void from_json(const json &j, BaseGrid &grid);
00245
00246 } // namespace rxdock
00247
00248 #endif //_RBTBASEGRID_H_
```

6.11 BaseIdxSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base class for indexed-grid-based scoring functions
00014 // Provides protected methods for subclasses to use to create indexing grids
00015
00016 #ifndef _RBTBASEIDXSF_H_
00017 #define _RBTBASEIDXSF_H_
00018
00019 #include "rxdock/BaseSF.h"
00020 #include "rxdock/InteractionGrid.h"
00021 #include "rxdock/NonBondedGrid.h"
00022 #include "rxdock/NonBondedHHSGrid.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
00030 class BaseIdxSF : public virtual BaseSF {
00031 public:
00032   // Class type string
00033   static const std::string _CT;
00034   // Parameter names
00035   static const std::string _GRIDSTEP;
```

```
00036   static const std::string _BORDER;
00037
00039   // Constructors/destructors
00040   virtual ~BaseIdxSF();
00041
00043   // Public methods
00045   double GetGridStep() const;
00046   void SetGridStep(double step);
00047   double GetBorder() const;
00048   void SetBorder(double border);
00049
00050   friend void to_json(json &j, const BaseIdxSF &baseIdxSF);
00051   friend void from_json(const json &j, BaseIdxSF &baseIdxSF);
00052
00053 protected:
00055   // Protected methods
00057   BaseIdxSF();
00058
00059   InteractionGridPtr CreateInteractionGrid() const;
00060   NonBondedGridPtr CreateNonBondedGrid() const;
00061   NonBondedHHSGridPtr CreateNonBondedHHSGrid() const;
00062   double GetMaxError() const;
00063   // DM 12 Apr 2002
00064   // Returns the maximum range of the scoring function,
00065   // corrected for max grid error, and grid border around docking site
00066   // This should be used by subclasses for selecting the receptor atoms to index
00067   // GetCorrectedRange() = GetRange() + GetMaxError() + GetBorder()
00068   double GetCorrectedRange() const;
00069
00070   // As this has a virtual base class we need a separate OwnParameterUpdated
00071   // which can be called by concrete subclass ParameterUpdated methods
00072   // See Stroustrup C++ 3rd edition, p395, on programming virtual base classes
00073   void OwnParameterUpdated(const std::string &strName);
00074
00075 private:
00077   // Private methods
00079
00080 protected:
00082   // Protected data
00084
00085 private:
00087   // Private data
00089   double m_gridStep;
00090   double m_border;
00091 };
00092
00093 void to_json(json &j, const BaseIdxSF &baseIdxSF);
00094 void from_json(const json &j, BaseIdxSF &baseIdxSF);
00095
00096 } // namespace rxdock
00097
00098 #endif //_RBTBASEIDXSF_H_
```

6.12 BaseInterSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Virtual base class for intermolecular scoring functions. Overrides Update()
00014
00015 #ifndef _RBTBASEINTERSF_H_
00016 #define _RBTBASEINTERSF_H_
00017
00018 #include "rxdock/BaseSF.h"
00019 #include "rxdock/Model.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class BaseInterSF : public virtual BaseSF {
00028 public:
```

```
00029   // Class type string
00030   static const std::string _CT;
00031
00033   // Constructors/destructors
00034   virtual ~BaseInterSF();
00035
00036   friend void to_json(json &j, const BaseInterSF &baseInterSF);
00037   friend void from_json(const json &j, BaseInterSF &baseInterSF);
00038
00040   // Public methods
00042
00043   ModelPtr GetReceptor() const;
00044   ModelPtr GetLigand() const;
00045   ModelList GetSolvent() const;
00046
00047   // Override Observer pure virtual
00048   // Notify observer that subject has changed
00049   virtual void Update(Subject *theChangedSubject);
00050
00051 protected:
00053   // Protected methods
00055   BaseInterSF();
00056
00057   // PURE VIRTUAL - Derived classes must override
00058   virtual void SetupReceptor() = 0; // Called by Update when receptor is changed
00059   virtual void SetupLigand() = 0;   // Called by Update when ligand is changed
00060   virtual void SetupSolvent() {}    // Called by Update when solvent is changed
00061   virtual void
00062   SetupScore() = 0; // Called by Update when either model has changed
00063
00064 private:
00066   // Private methods
00068
00069 protected:
00071   // Protected data
00073
00074 private:
00076   // Private data
00078   ModelPtr m_spReceptor;
00079   ModelPtr m_spLigand;
00080   ModelList m_solventList;
00081 };
00082
00083 void to_json(json &j, const BaseInterSF &baseInterSF);
00084 void from_json(const json &j, BaseInterSF &baseInterSF);
00085
00086 } // namespace rxdock
00087
00088 #endif //_RBTBASEINTERSF_H_
```

6.13 BaseIntraSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Virtual base class for intramolecular scoring functions. Overrides Update()
00014
00015 #ifndef _RBTBASEINTRASF_H_
00016 #define _RBTBASEINTRASF_H_
00017
00018 #include "rxdock/BaseSF.h"
00019 #include "rxdock/Model.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class BaseIntraSF : public virtual BaseSF {
00028 public:
00029   // Class type string
00030   static const std::string _CT;
00031
```

```
00033   // Constructors/destructors
00034   virtual ~BaseIntraSF();
00035
00036   friend void to_json(json &j, const BaseIntraSF &baseIntraSF);
00037   friend void from_json(const json &j, BaseIntraSF &baseIntraSF);
00038
00040   // Public methods
00042
00043   ModelPtr GetLigand() const;
00044
00045   // Override Observer pure virtual
00046   // Notify observer that subject has changed
00047   virtual void Update(Subject *theChangedSubject);
00048
00049   // Override BaseSF::ScoreMap to provide additional raw descriptors
00050   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00051
00052 protected:
00054   // Protected methods
00056   BaseIntraSF();
00057
00058   // PURE VIRTUAL - Derived classes must override
00059   virtual void SetupScore() = 0; // Called by Update when model has changed
00060
00061 private:
00063   // Private methods
00065
00066 protected:
00068   // Protected data
00070
00071 private:
00073   // Private data
00075   ModelPtr m_spLigand;
00076   // 26 Mar 2003 (DM) Remember the raw score for the initial ligand conformation
00077   // This becomes the zero point for all subsequent score reporting
00078   // i.e. all intramolecular scores are reported relative to the initial score
00079   double m_zero;
00080 };
00081
00082 void to_json(json &j, const BaseIntraSF &baseIntraSF);
00083 void from_json(const json &j, BaseIntraSF &baseIntraSF);
00084
00085 } // namespace rxdock
00086
00087 #endif //_RBTBASEINTRASF_H_
```

6.14 BaseMolecularDataSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract base class (interface) for molecular data sources
00014
00015 #ifndef _RBTBASEMOLECULARDATASOURCE_H_
00016 #define _RBTBASEMOLECULARDATASOURCE_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/Bond.h"
00020 #include "rxdock/Config.h"
00021 #include "rxdock/Variant.h"
00022
00023 namespace rxdock {
00024
00025 class BaseMolecularDataSource {
00026 public:
00028   // Constructors/destructors
00029
00030   // Parameterised constructor
00031   // strName is a descriptive text label for the type of data source
00032   // Derived classes should use this constructor to set a suitable name
00033   BaseMolecularDataSource(std::string strName) : m_strName(strName) {}
00034
00035   virtual ~BaseMolecularDataSource() {} // Default destructor
00036
```

```
00038   // Public methods
00040   std::string GetName() {
00041     return m_strName;
00042   } // Returns source identifier string
00043
00044   virtual bool isTitleListSupported() { return false; }
00045   virtual bool isAtomListSupported() { return false; }
00046   virtual bool isCoordinatesSupported() { return false; }
00047   virtual bool isBondListSupported() { return false; }
00048
00049   // Pure virtual - derived classes must override
00050
00051   // Reset source so that next time it is used, a new set of atom and bond
00052   // objects are created
00053   virtual void Reset() = 0;
00054
00055   virtual int GetNumTitles() = 0;
00056   virtual int GetNumAtoms() = 0;
00057   virtual int GetNumBonds() = 0;
00058   virtual int GetNumSegments() = 0;
00059
00060   virtual std::vector<std::string> GetTitleList() = 0;
00061   virtual AtomList GetAtomList() = 0;
00062   virtual BondList GetBondList() = 0;
00063   virtual SegmentMap GetSegmentMap() = 0;
00064
00065   // DM 12 May 1999 - support for data records (e.g. SD file)
00066   // Does source support data records (default=false)
00067   virtual bool isDataSupported() { return false; }
00068   // Get number of data fields
00069   virtual int GetNumData() = 0;
00070   // Get list of field names as string list
00071   virtual std::vector<std::string> GetDataFieldList() = 0;
00072   // Get all data as map of key=field name, value=variant (double,string or
00073   // string list)
00074   virtual StringVariantMap GetDataMap() = 0;
00075   // Query as to whether a particular data field name is present
00076   virtual bool isDataFieldPresent(const std::string &strDataField) = 0;
00077   // Get a particular data value
00078   virtual Variant GetDataValue(const std::string &strDataField) = 0;
00079
00080 private:
00082   // Private methods
00084   BaseMolecularDataSource(); // Disable default constructor
00085   BaseMolecularDataSource(
00086       const BaseMolecularDataSource &); // Copy constructor disabled by default
00087   BaseMolecularDataSource &operator=(
00088       const BaseMolecularDataSource &); // Copy assignment disabled by default
00089
00090 private:
00092   // Private data
00094   std::string m_strName; // Source identifier string
00095 };
00096
00097 } // namespace rxdock
00098
00099 #endif //_RBTBASEMOLECULARDATASOURCE_H_
```

6.15 BaseMolecularFileSink.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract base class for molecular file-based output. Provides hooks for
00014 // handling multiple models/conformations being rendered.
00015 // Derived classes must provide a Render method.
00016
00017 #ifndef _RBTBASEMOLECULARFILESINK_H_
00018 #define _RBTBASEMOLECULARFILESINK_H_
00019
00020 #include "rxdock/BaseFileSink.h"
00021 #include "rxdock/Model.h"
00022
00023 #include <nlohmann/json.hpp>
```

```
00024
00025 using json = nlohmann::json;
00026
00027 namespace rxdock {
00028
00029 class BaseMolecularFileSink : public BaseFileSink {
00030 public:
00032   // Constructors/destructors
00033   // BaseMolecularFileSink(const char* fileName, ModelPtr spModel);
00034   BaseMolecularFileSink(const std::string &fileName, ModelPtr spModel,
00035                         bool bUseModelSegmentNames = true);
00036
00037   virtual ~BaseMolecularFileSink(); // Default destructor
00038
00039   friend void to_json(json &j, const BaseMolecularFileSink &baseBiMolFSink);
00040   friend void from_json(const json &j, BaseMolecularFileSink &baseBiMolFSink);
00041
00043   // Public methods
00045
00046   // Derived classes should override
00047   // Controls whether file type can support the writing of multiple
00048   // conformations/models to a single file
00049   virtual bool isMultiConfSupported() { return false; }
00050
00051   // Get/set the Model that is linked to the sink
00052   ModelPtr GetModel() const;
00053   // DM 21 Apr 1999 - if bUseModelSegmentNames is true, the model segment names
00054   // are rendered if bUseModelSegmentNames is false, incremental numeric segment
00055   // IDs are rendered
00056   RBTDLL_EXPORT void SetModel(ModelPtr spModel,
00057                               bool bUseModelSegmentNames = false);
00058
00059   // DM 21 Apr 1999 - change the segment ID counter
00060   // Set to nSegmentId-1 as the counter gets incremented before each Render
00061   void SetNextSegmentId(int nSegmentId) {
00062     m_nSegmentId = nSegmentId - 1;
00063     m_bUseModelSegmentNames = false;
00064   }
00065
00066   // Get/set the multi-conformer status
00067   // With MultiConf = false, each call to Render should immediately Write the
00068   // cache to the file With MultiConf = true, each call to Render should
00069   // continue to fill the cache with atom records The cache is only written by
00070   // an explicit call to WriteMultiConf, or by the base FileSink destructor
00071   bool GetMultiConf() const;
00072   void SetMultiConf(bool bMultiConf);
00073   // Force writing the file following several multi-conf Renders
00074   void WriteMultiConf();
00075
00076   ModelList GetSolvent() const;
00077   void SetSolvent(ModelList solventList);
00078
00079 protected:
00081   // Protected methods
00083   void Reset(); // Reset the atom, residue and segment numbering
00084
00085 private:
00087   // Private methods
00089
00090   BaseMolecularFileSink(); // Disable default constructor
00091   BaseMolecularFileSink(
00092       const BaseMolecularFileSink &); // Copy constructor disabled by default
00093   BaseMolecularFileSink &operator=(
00094       const BaseMolecularFileSink &); // Copy assignment disabled by default
00095
00096 protected:
00098   // Protected data
00100   int m_nAtomId;    // Next available atom number
00101   int m_nSubunitId; // Next available subunit number
00102   int m_nSegmentId; // Next available segment number
00103   // DM 21 Apr 1999 - if true, output the atom segment names in the model
00104   // If false, output incremental segment IDs
00105   bool m_bUseModelSegmentNames;
00106
00107 private:
00109   // Private data
00111   ModelPtr m_spModel;
00112   ModelList m_solventList;
00113   bool m_bMultiConf;
00114 };
00115
00116 void to_json(json &j, const BaseMolecularFileSink &baseBiMolFSink);
00117 void from_json(const json &j, BaseMolecularFileSink &baseBiMolFSink);
00118
00119 // Useful typedefs
00120 typedef SmartPtr<BaseMolecularFileSink> MolecularFileSinkPtr; // Smart pointer
00121
```

```
00122 } // namespace rxdock
00123
00124 #endif //_RBTBASEMOLECULARFILESINK_H_
```

6.16 BaseMolecularFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract base class for molecular file-based data sources
00014
00015 #ifndef _RBTBASEMOLECULARFILESOURCE_H_
00016 #define _RBTBASEMOLECULARFILESOURCE_H_
00017
00018 #include "rxdock/BaseFileSource.h"
00019 #include "rxdock/BaseMolecularDataSource.h"
00020 #include "rxdock/ParameterFileSource.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class BaseMolecularFileSource : public BaseFileSource,
00029                                 public BaseMolecularDataSource {
00030 public:
00031   // Constructors
00032   // BaseMolecularFileSource(const char* fileName, const char* sourceName);
00033   // Single-record version
00034   RBTDLL_EXPORT BaseMolecularFileSource(const std::string &fileName,
00035                                         const std::string &sourceName);
00036   // Multi-record version, with record delimiter passed as an argument
00037   BaseMolecularFileSource(const std::string &fileName,
00038                           const std::string &strRecDelim,
00039                           const std::string &sourceName);
00040
00041   // Destructor
00042   virtual ~BaseMolecularFileSource();
00043
00044   friend void to_json(json &j, const BaseMolecularFileSource &baseMolFilSrc);
00045   friend void from_json(const json &j, BaseMolecularFileSource &baseMolFilSrc);
00046
00047   // Pure virtual methods from BaseMolecularDataSource
00048   void Reset();
00049   int GetNumTitles();
00050   int GetNumAtoms();
00051   int GetNumBonds();
00052   int GetNumSegments();
00053
00054   std::vector<std::string> GetTitleList();
00055   AtomList GetAtomList();
00056   BondList GetBondList();
00057   SegmentMap GetSegmentMap();
00058
00059   // DM 12 May 1999 - support for data records (e.g. SD file)
00060   // Get number of data fields
00061   virtual int GetNumData();
00062   // Get list of field names as string list
00063   virtual std::vector<std::string> GetDataFieldList();
00064   // Get all data as map of key=field name, value=variant (double,string or
00065   // string list)
00066   virtual StringVariantMap GetDataMap();
00067   // Query as to whether a particular data field name is present
00068   virtual bool isDataFieldPresent(const std::string &strDataField);
00069   // Get a particular data value
00070   virtual Variant GetDataValue(const std::string &strDataField);
00071
00072   // These methods allow filtering of the data source by segment name
00073   // So for example, we could just read the segment named TAR from the source
00074   // The filter modifies the behaviour of GetAtomList, GetBondList etc
00075   SegmentMap GetSegmentFilterMap();
00076   RBTDLL_EXPORT void SetSegmentFilterMap(const SegmentMap &segmentFilterMap);
00077   void ClearSegmentFilterMap();
```

```
00078   bool isSegmentFilterMapDefined();
00079
00080 protected:
00081   // Protected methods
00082   void ClearMolCache();            // Clear the atom, bond and title lists
00083   void RemoveAtom(AtomPtr spAtom); // DM 16 Feb 2000 - moved from MdlFileSource
00084   void
00085   RenumberAtomsAndBonds(); // DM 30 Oct 2000 - now independent from RemoveAtom
00086   // Moved from MOL2FileSource - sets partial charges on a per-residue basis
00087   // Automatically determines protonation states of ASP/GLU/HIS using Mandatory
00088   // and Forbidden atoms in spParamSource
00089   void SetupPartialIonicGroups(AtomList &atoms,
00090                                ParameterFileSourcePtr spParamSource);
00091
00092 private:
00093   // Private methods
00094   BaseMolecularFileSource(); // Disable default constructor
00095   BaseMolecularFileSource(
00096       const BaseMolecularFileSource &); // Copy constructor disabled by default
00097   BaseMolecularFileSource &operator=(
00098       const BaseMolecularFileSource &); // Copy assignment disabled by default
00099
00100   // Get methods with segment filter applied
00101   int GetNumAtomsWithFilter();
00102   int GetNumBondsWithFilter();
00103   AtomList GetAtomListWithFilter();
00104   BondList GetBondListWithFilter();
00105
00106 protected:
00107   // Protected data
00108   std::vector<std::string> m_titleList;
00109   AtomList m_atomList;
00110   BondList m_bondList;
00111   SegmentMap m_segmentMap;
00112   StringVariantMap m_dataMap; // DM 12 May 1999 - for storing data records
00113
00114 private:
00115   // Private data
00116   SegmentMap m_segmentFilterMap;
00117 };
00118
00119 void to_json(json &j, const BaseMolecularFileSource &baseMolFilSrc);
00120 void from_json(const json &j, BaseMolecularFileSource &baseMolFilSrc);
00121
00122 // useful typedefs
00123 typedef SmartPtr<BaseMolecularFileSource>
00124     MolecularFileSourcePtr; // Smart pointer
00125
00126 } // namespace rxdock
00127
00128 #endif //_RBTBASEMOLECULARFILESOURCE_H_
```

6.17 BaseObject.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base class for all workspace observers which have named parameters and
00014 // handle arbitrary requests (i.e. scoring functions and transforms)
00015
00016 #ifndef _RBTBASEOBJECT_H_
00017 #define _RBTBASEOBJECT_H_
00018
00019 #include "rxdock/Config.h"
00020 #include "rxdock/Observer.h"
00021 #include "rxdock/ParamHandler.h"
00022 #include "rxdock/RequestHandler.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
```

```
00030 class WorkSpace; // forward definition
00031
00032 class BaseObject : public ParamHandler, public Observer, public RequestHandler {
00033 public:
00034   // Class type string
00035   static const std::string _CT;
00036   // Parameter names
00037   static const std::string _CLASS;
00038   static const std::string _NAME;
00039   static const std::string _ENABLED;
00040
00041   friend void to_json(json &j, const BaseObject &baseObject);
00042   friend void from_json(const json &j, BaseObject &baseObject);
00043
00045   // Constructors/destructors
00046   virtual ~BaseObject();
00047
00049   // Public methods
00051   // Class name (e.g. ConstSF)
00052   std::string GetClass() const;
00053   // Short name (e.g. HBOND)
00054   std::string GetName() const;
00055   void SetName(const std::string &);
00056   // Fully qualified name (should be overridden by subclasses which can be
00057   // aggregated to prefix the name with the parent's name)
00058   virtual std::string GetFullName() const;
00059   void Enable();
00060   void Disable();
00061   bool isEnabled() const;
00062
00063   // WorkSpace handling methods
00064   // Register scoring function with a workspace
00065   // Base class version just registers itself
00066   virtual void Register(WorkSpace *);
00067   // Unregister with a workspace
00068   // Base class version just unregisters itself
00069   virtual void Unregister();
00070   // Get workspace pointer
00071   WorkSpace *GetWorkSpace() const;
00072
00073   // Override Observer method
00074   // Notify observer that subject is about to be deleted
00075   virtual void Deleted(Subject *theDeletedSubject);
00076
00077   // Request Handling method
00078   virtual void HandleRequest(RequestPtr spRequest);
00079
00080   // Virtual function for dumping parameters to an output stream
00081   // Called by operator «
00082   virtual void Print(std::ostream &s) const;
00083
00084 protected:
00086   // Protected methods
00088   BaseObject(const std::string &strClass, const std::string &strName);
00089   BaseObject();
00090
00091   // DM 25 Oct 2000 - track changes to parameter values in local data members
00092   // ParameterUpdated is invoked by ParamHandler::SetParameter
00093   void ParameterUpdated(const std::string &strName);
00094
00095 private:
00097   // Private methods
00099   BaseObject(const BaseObject &); // Copy constructor disabled by default
00100   BaseObject &
00101   operator=(const BaseObject &); // Copy assignment disabled by default
00102
00103 protected:
00105   // Protected data
00107
00108 private:
00110   // Private data
00112   WorkSpace *m_workspace;
00113   bool m_enabled;
00114 };
00115
00116 void to_json(json &j, const BaseObject &baseObject);
00117 void from_json(const json &j, BaseObject &baseObject);
00118
00119 // Useful typedefs
00120 typedef std::vector<BaseObject *> BaseObjectList; // Vector of smart pointers
00121 typedef BaseObjectList::iterator BaseObjectListIter;
00122 typedef BaseObjectList::const_iterator BaseObjectListConstIter;
00123
00124 } // namespace rxdock
00125
00126 #endif //_RBTBASEOBJECT_H_
```

6.18 BaseSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base class for all scoring functions
00014 #ifndef _RBTBASESF_H_
00015 #define _RBTBASESF_H_
00016
00017 #include "rxdock/BaseObject.h"
00018 #include "rxdock/Config.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 class SFAgg; // forward declaration
00027
00028 class BaseSF : public BaseObject {
00029 public:
00030   // Class type string
00031   static const std::string _CT;
00032   // Parameter names
00033   static const std::string _WEIGHT;
00034   // DM 09 Apr 2002 - max distance range over which scoring function operates
00035   // Subclasses are free not to use if not required
00036   static const std::string _RANGE;
00037   // DM 17 Jan 2006 - hardcoded name of scoring function branch under which to
00038   // save "system" scoring function terms (e.g. intra-receptor, intra-solvent).
00039   // This is a dirty (and hopefully temporary) solution!
00040   static const std::string _SYSTEM_SF;
00041   static const std::string _INTRA_SF; // Ditto, for ligand intramolecular terms
00042
00044   // Constructors/destructors
00045   virtual ~BaseSF();
00046
00047   friend void to_json(json &j, const BaseSF &baseSF);
00048   friend void from_json(const json &j, BaseSF &baseSF);
00049
00050   // Give aggregates access to BaseSF private methods/data
00051   friend class SFAgg;
00052
00054   // Public methods
00056   // Fully qualified name, prefixed by all ancestors (e.g.
00057   // rxdock.score.inter.hbond)
00058   std::string GetFullName() const;
00059   double GetWeight() const;
00060   void SetWeight(double);
00061
00062   double GetRange() const;
00063   RBTDLL_EXPORT void SetRange(double);
00064
00065   // Main public method - returns current weighted score
00066   RBTDLL_EXPORT double Score() const;
00067   // Returns all child component scores as a string-variant map
00068   // Key = fully qualified component name, value = weighted score
00069   //(for saving in a Model's data fields)
00070   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00071
00072   // Aggregate handling methods
00073   virtual void Add(BaseSF *);
00074   virtual void Remove(BaseSF *);
00075   virtual bool isAgg() const;
00076   virtual unsigned int GetNumSF() const;
00077   virtual BaseSF *GetSF(unsigned int iSF) const;
00078   void Orphan(); // Force removal from the parent aggregate
00079   BaseSF *GetParentSF() const;
00080
00081 protected:
00083   // Protected methods
00085   BaseSF(const std::string &strClass, const std::string &strName);
00086   BaseSF();
00087   // PURE VIRTUAL - DERIVED CLASSES MUST OVERRIDE
00088   virtual double RawScore() const = 0;
00089   // DM 25 Oct 2000 - track changes to parameter values in local data members
00090   // ParameterUpdated is invoked by ParamHandler::SetParameter
```

```
00091   void ParameterUpdated(const std::string &strName);
00092   // Helper method for ScoreMap
00093   void AddToParentMapEntry(StringVariantMap &scoreMap, double rs) const;
00094
00095 private:
00097   // Private methods
00099   BaseSF(const BaseSF &);            // Copy constructor disabled by default
00100   BaseSF &operator=(const BaseSF &); // Copy assignment disabled by
00101                                      // default
00102
00103 protected:
00105   // Protected data
00107
00108 private:
00110   // Private data
00112   BaseSF *m_parent;
00113   double m_weight;
00114   double m_range;
00115 };
00116
00117 void to_json(json &j, const BaseSF &baseSF);
00118 void from_json(const json &j, BaseSF &baseSF);
00119
00120 // Useful typedefs
00121 typedef std::vector<BaseSF *> BaseSFList; // Vector of smart pointers
00122 typedef BaseSFList::iterator BaseSFListIter;
00123 typedef BaseSFList::const_iterator BaseSFListConstIter;
00124
00125 } // namespace rxdock
00126
00127 #endif //_RBTBASESF_H_
```

6.19 BaseTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract base class for all classes which manipulate molecular coords
00014
00015 #ifndef _RBTBASETRANSFORM_H_
00016 #define _RBTBASETRANSFORM_H_
00017
00018 #include "rxdock/BaseObject.h"
00019 #include "rxdock/Config.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class TransformAgg; // forward declaration
00028
00029 class BaseTransform : public BaseObject {
00030 public:
00031   // Class type string
00032   static const std::string _CT;
00033   // Parameter names
00034
00036   // Constructors/destructors
00037   virtual ~BaseTransform();
00038
00039   friend void to_json(json &j, const BaseTransform &bt);
00040   friend void from_json(const json &j, BaseTransform &bt);
00041
00042   // Give aggregates access to BaseSF private methods/data
00043   friend class TransformAgg;
00044
00046   // Public methods
00048   // Fully qualified name, prefixed by all ancestors
00049   std::string GetFullName() const;
00050
00051   // Main public method - actually apply the transform
00052   // Not virtual. Base class method checks if transform is enabled,
```

```
00053   // sends SFRequests to reconfigure the scoring function
00054   // then calls private virtual method Execute() to apply the transform
00055   void Go();
00056
00057   // Aggregate handling methods
00058   virtual void Add(BaseTransform *);
00059   virtual void Remove(BaseTransform *);
00060   virtual bool isAgg() const;
00061   virtual unsigned int GetNumTransforms() const;
00062   virtual BaseTransform *GetTransform(unsigned int) const;
00063   void Orphan(); // Force removal from the parent aggregate
00064   BaseTransform *GetParentTransform() const;
00065
00066   // Scoring function request handling
00067   // Transforms can store up a list of requests to send to the workspace
00068   // scoring function each time Go() is executed
00069   void AddSFRequest(RequestPtr);
00070   void ClearSFRequests();
00071   void SendSFRequests();
00072
00073 protected:
00075   // Protected methods
00077   BaseTransform(const std::string &strClass, const std::string &strName);
00078
00079 private:
00081   // Private methods
00083   BaseTransform();
00084   BaseTransform(const BaseTransform &); // Copy constructor disabled by default
00085   BaseTransform &
00086   operator=(const BaseTransform &); // Copy assignment disabled by default
00087
00088   // PURE VIRTUAL - subclasses should override. Applies the transform
00089   virtual void Execute() = 0;
00090
00091 protected:
00093   // Protected data
00095
00096 private:
00098   // Private data
00100   BaseTransform *m_parent;
00101   RequestList m_SFRequests;
00102 };
00103
00104 void to_json(json &j, const BaseTransform &bt);
00105 void from_json(const json &j, BaseTransform &bt);
00106
00107 // Useful typedefs
00108 typedef std::vector<BaseTransform *>
00109     BaseTransformList; // Vector of smart pointers
00110 typedef BaseTransformList::iterator BaseTransformListIter;
00111 typedef BaseTransformList::const_iterator BaseTransformListConstIter;
00112
00113 } // namespace rxdock
00114
00115 #endif //_RBTBASETRANSFORM_H_
```

6.20 BaseUniMolTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base class for unimolecular transforms. Overrides Update()
00014
00015 #ifndef _RBTBASEUNIMOLTRANSFORM_H_
00016 #define _RBTBASEUNIMOLTRANSFORM_H_
00017
00018 #include "rxdock/BaseTransform.h"
00019 #include "rxdock/Model.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
```

```
00026
00027 class BaseUniMolTransform : public BaseTransform {
00028 public:
00029   // Class type string
00030   static const std::string _CT;
00032   // Constructors/destructors
00033   virtual ~BaseUniMolTransform();
00034
00035   friend void to_json(json &j, const BaseUniMolTransform &baseUniMolTrans);
00036   friend void from_json(const json &j, BaseUniMolTransform &baseUniMolTrans);
00037
00039   // Public methods
00041
00042   ModelPtr GetLigand() const;
00043
00044   // Override Observer pure virtual
00045   // Notify observer that subject has changed
00046   virtual void Update(Subject *theChangedSubject);
00047
00048 protected:
00050   // Protected methods
00052   BaseUniMolTransform(const std::string &strClass, const std::string &strName);
00053
00054   // PURE VIRTUAL - Derived classes must override
00055   virtual void SetupTransform() = 0; // Called by Update when model has changed
00056
00057 private:
00059   // Private methods
00061
00062 protected:
00064   // Protected data
00066
00067 private:
00069   // Private data
00071   ModelPtr m_spLigand;
00072 };
00073
00074 void to_json(json &j, const BaseUniMolTransform &baseUniMolTrans);
00075 void from_json(const json &j, BaseUniMolTransform &baseUniMolTrans);
00076
00077 } // namespace rxdock
00078
00079 #endif //_RBTBASEUNIMOLTRANSFORM_H_
```

6.21 BiMolWorkSpace.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Subclass of WorkSpace for simulations involving
00014 // Receptor = Model #0
00015 // Ligand   = Model #1
00016 // Solvent  = Model #2 upwards
00017 #ifndef _RBTBIMOLWORKSPACE_H_
00018 #define _RBTBIMOLWORKSPACE_H_
00019
00020 #include "rxdock/Population.h"
00021 #include "rxdock/WorkSpace.h"
00022
00023 namespace rxdock {
00024
00025 class BiMolWorkSpace : public WorkSpace {
00026   enum eModelID {
00027     RECEPTOR = 0,
00028     LIGAND = 1,
00029     SOLVENT = 2 // Flexible waters are represented as separate models, from 2
00030   };
00031
00032 public:
00034   // Constructors/destructors
00035
00036   RBTDLL_EXPORT BiMolWorkSpace();
00037   virtual ~BiMolWorkSpace();
00038
```

```
00040   // Public methods
00042
00043   // Model handling
00044   RBTDLL_EXPORT ModelPtr GetReceptor() const;
00045   RBTDLL_EXPORT ModelPtr GetLigand() const;
00046   RBTDLL_EXPORT ModelList GetSolvent() const;
00047   RBTDLL_EXPORT bool hasSolvent() const;
00048   RBTDLL_EXPORT void SetReceptor(ModelPtr spReceptor);
00049   RBTDLL_EXPORT void SetLigand(ModelPtr spLigand);
00050   RBTDLL_EXPORT void SetSolvent(ModelList solventList);
00051   RBTDLL_EXPORT void RemoveSolvent();
00052   RBTDLL_EXPORT void
00053   UpdateModelCoordsFromChromRecords(BaseMolecularFileSource *pSource);
00054
00055   // Model I/O
00056   // Saves ligand to file sink
00057   virtual void Save(bool bSaveScores = true);
00058   virtual void SaveHistory(bool bSaveScores = true);
00059
00060 protected:
00062   // Protected methods
00064
00065 private:
00067   // Private methods
00069
00070   BiMolWorkSpace(
00071       const BiMolWorkSpace &); // Copy constructor disabled by default
00072   BiMolWorkSpace &
00073   operator=(const BiMolWorkSpace &); // Copy assignment disabled by default
00074   void SaveLigand(MolecularFileSinkPtr spSink, bool bSaveScores = true);
00075
00076 protected:
00078   // Protected data
00080
00081 private:
00083   // Private data
00085 };
00086
00087 // Useful typedefs
00088 typedef SmartPtr<BiMolWorkSpace> BiMolWorkSpacePtr; // Smart pointer
00089
00090 } // namespace rxdock
00091
00092 #endif //_RBTBIMOLWORKSPACE_H_
```

6.22 Bond.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Handles all the properties of a bond
00014
00015 #ifndef _RBTBOND_H_
00016 #define _RBTBOND_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/Config.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class Bond {
00028 public:
00030   // Constructors/destructors
00032
00033   // Default constructor
00034   Bond();
00035
00036   // Cconstructor supplying all parameters
00037   Bond(int nBondId, AtomPtr &spAtom1, AtomPtr &spAtom2,
00038        int nFormalBondOrder = 1);
```

```
00039
00040   Bond(json j);
00041
00042   virtual ~Bond(); // Default destructor
00043
00044   Bond(const Bond &bond); // Copy constructor
00045
00046   Bond &operator=(const Bond &bond); // Copy assignment
00047
00049   // Stream functions
00051
00052   // Insertion operator (primarily for debugging)
00053   // Note: needs to be friend so can access bond private data
00054   // without using the accessor functions
00055   RBTDLL_EXPORT friend std::ostream &operator<<(std::ostream &s,
00056                                                 const Bond &bond);
00057
00058   friend void to_json(json &j, const Bond &bond);
00059   friend void from_json(const json &j, Bond &bond);
00060
00062   // Public accessor functions
00064
00065   // Bond ID
00066   int GetBondId() const { return m_nBondId; }
00067   void SetBondId(const int nBondId) { m_nBondId = nBondId; }
00068
00069   // Atom pointers
00070   AtomPtr GetAtom1Ptr() const { return m_spAtom1; }
00071   AtomPtr GetAtom2Ptr() const { return m_spAtom2; }
00072   void SetAtom1Ptr(AtomPtr &spAtom1) { m_spAtom1 = spAtom1; }
00073   void SetAtom2Ptr(AtomPtr &spAtom2) { m_spAtom2 = spAtom2; }
00074
00075   // Formal Bond order
00076   int GetFormalBondOrder() const { return m_nFormalBondOrder; }
00077   void SetFormalBondOrder(const int nFormalBondOrder) {
00078     m_nFormalBondOrder = nFormalBondOrder;
00079   }
00080
00081   // Partial Bond order
00082   double GetPartialBondOrder() const { return m_dPartialBondOrder; }
00083   void SetPartialBondOrder(const double dPartialBondOrder) {
00084     m_dPartialBondOrder = dPartialBondOrder;
00085   }
00086
00087   // CyclicFlag - flag to indicate bond is in a ring (set by Model::FindRing)
00088   bool GetCyclicFlag() const { return m_bCyclic; }
00089   void SetCyclicFlag(bool bCyclic = true) { m_bCyclic = bCyclic; }
00090
00091   // SelectionFlag - general purpose flag can be set/cleared by various search
00092   // algorithms (e.g. FindRings)
00093   bool GetSelectionFlag() const { return m_bSelected; }
00094   void SetSelectionFlag(bool bSelected = true) { m_bSelected = bSelected; }
00095
00097   // Public methods
00099   // Returns bond length
00100   double Length() const;
00101
00102 protected:
00104   // Protected methods
00106
00107 private:
00109   // Private methods
00111
00112 protected:
00114   // Protected data
00116
00117 private:
00119   // Private data
00121   int m_nBondId;              // Original bond ID in PSF file
00122   AtomPtr m_spAtom1;          // Smart pointer to atom 1
00123   AtomPtr m_spAtom2;          // Smart pointer to atom 2
00124   int m_nFormalBondOrder;     // Formal bond order (1,2,3, no aromatic bond
00125                               // orders of 1.5)
00126   double m_dPartialBondOrder; // Partial bond order (1.0, 1.5, 2.0, 3.0 etc)
00127   bool m_bCyclic;             // Is the bond in a ring ?
00128   bool m_bSelected; // Can be set/cleared by various search algorithms (e.g.
00129                     // FindRings)
00130 };
00131
00132 // Useful typedefs
00133 typedef SmartPtr<Bond> BondPtr;        // Smart pointer
00134 typedef std::vector<BondPtr> BondList; // Vector of smart pointers
00135 typedef BondList::iterator BondListIter;
00136 typedef BondList::const_iterator BondListConstIter;
00137
00139 // Non-member functions (in rxdock namespace)
00141
```

```
00142 std::ostream &operator«(std::ostream &s, const Bond &bond);
00143
00144 void to_json(json &j, const Bond &bond);
00145 void from_json(const json &j, Bond &bond);
00146
00148 // Predicates (for use by STL algorithms)
00150 // DM 08 May 2002 - convert to regular Bond* functions
00151 // More universal as BondPtr parameters will be automatically degraded to
00152 // regular pointers anyway
00153
00154 // Is bond selected ?
00155 class isBondSelected : public std::function<bool(Bond *)> {
00156 public:
00157   explicit isBondSelected() = default;
00158   bool operator()(Bond *pBond) const { return pBond->GetSelectionFlag(); }
00159 };
00160
00161 // Is bond cyclic ?
00162 class isBondCyclic : public std::function<bool(Bond *)> {
00163 public:
00164   explicit isBondCyclic() = default;
00165   bool operator()(Bond *pBond) const { return pBond->GetCyclicFlag(); }
00166 };
00167
00168 // Is bond rotatable ?
00169 class isBondRotatable : public std::function<bool(Bond *)> {
00170 public:
00171   explicit isBondRotatable() = default;
00172   RBTDLL_EXPORT bool operator()(Bond *) const;
00173 };
00174
00175 // Is bond to a terminal NH3+ group?
00176 class isBondToNH3 : public std::function<bool(Bond *)> {
00177 public:
00178   explicit isBondToNH3() = default;
00179   RBTDLL_EXPORT bool operator()(Bond *) const;
00180 };
00181
00182 // Is bond to a terminal OH group?
00183 class isBondToOH : public std::function<bool(Bond *)> {
00184 public:
00185   explicit isBondToOH() = default;
00186   RBTDLL_EXPORT bool operator()(Bond *pBond) const;
00187 };
00188
00189 // DM 1 April 1999
00190 // Is bond2 equal to bond1 (checks if underlying regular pointers match)
00191 // Note: this is a binary rather than unary predicate
00192 // class isBondPtr_eq : public std::function<bool(Bond *, Bond *)> {
00193 // public:
00194 //  explicit isBondPtr_eq() {}
00195 //  Bool operator() (BondPtr spBond1, BondPtr spBond2) const {return
00196 //  spBond1.Ptr() == spBond2.Ptr();}
00197 //};
00198
00199 // Is bond.Ptr() equal to Bond* (checks if underlying regular pointers match)
00200 class isBondPtr_eq : public std::function<bool(Bond *)> {
00201   Bond *p;
00202
00203 public:
00204   explicit isBondPtr_eq(Bond *pp) : p(pp) {}
00205   bool operator()(Bond *pBond) const { return pBond == p; }
00206 };
00207
00208 // DM 2 Aug 1999
00209 // Is bond2 equal to bond1 (checks if bond ID, atom1 and atom2 match)
00210 class isBond_eq : public std::function<bool(Bond *)> {
00211   Bond *p;
00212   isAtom_eq bIsAtomEqual;
00213
00214 public:
00215   explicit isBond_eq(Bond *pp) : p(pp) {}
00216   bool operator()(Bond *pBond) const {
00217     return ((pBond->GetBondId() == p->GetBondId()) &&
00218             (bIsAtomEqual(pBond->GetAtom1Ptr(), p->GetAtom1Ptr())) &&
00219             (bIsAtomEqual(pBond->GetAtom2Ptr(), p->GetAtom2Ptr())));
00220   }
00221 };
00222
00223 // DM 7 June 1999
00224 // Is bond an amide bond?
00225 class isBondAmide : public std::function<bool(Bond *)> {
00226 public:
00227   explicit isBondAmide() = default;
00228   RBTDLL_EXPORT bool operator()(Bond *) const;
00229 };
00230
```

```
00232 // Bond list functions (implemented as STL algorithms)
00234
00235 // Generic template version of GetNumBonds, passing in your own predicate
00236 template <class Predicate>
00237 unsigned int GetNumBondsWithPredicate(const BondList &bondList,
00238                                       const Predicate &pred) {
00239   return static_cast<unsigned int>(
00240       std::count_if(bondList.begin(), bondList.end(), pred));
00241 }
00242
00243 // Generic template version of GetBondList, passing in your own predicate
00244 template <class Predicate>
00245 BondList GetBondListWithPredicate(const BondList &bondList,
00246                                   const Predicate &pred) {
00247   BondList newBondList;
00248   std::copy_if(bondList.begin(), bondList.end(),
00249                std::back_inserter(newBondList), pred);
00250   return newBondList;
00251 }
00252
00253 // Generic template version of FindBond, passing in your own predicate
00254 template <class Predicate>
00255 BondListIter FindBond(BondList &bondList, const Predicate &pred) {
00256   return std::find_if(bondList.begin(), bondList.end(), pred);
00257 }
00258
00259 // Selected bonds
00260 void SetBondSelectionFlagsInList(BondList &bondList, bool bSelected = true);
00261 inline unsigned int GetNumSelectedBondsInList(const BondList &bondList) {
00262   return GetNumBondsWithPredicate(bondList, isBondSelected());
00263 }
00264
00265 inline BondList GetSelectedBondsFromList(const BondList &bondList) {
00266   return GetBondListWithPredicate(bondList, isBondSelected());
00267 }
00268
00269 // Cyclic bonds
00270 void SetBondCyclicFlagsInList(BondList &bondList, bool bCyclic = true);
00271 inline unsigned int GetNumCyclicBondsInList(const BondList &bondList) {
00272   return GetNumBondsWithPredicate(bondList, isBondCyclic());
00273 }
00274 inline BondList GetCyclicBondsFromList(const BondList &bondList) {
00275   return GetBondListWithPredicate(bondList, isBondCyclic());
00276 }
00277
00278 // Rotatable bonds
00279 inline unsigned int GetNumRotatableBonds(const BondList &bondList) {
00280   return GetNumBondsWithPredicate(bondList, isBondRotatable());
00281 }
00282 inline BondList GetRotatableBondList(const BondList &bondList) {
00283   return GetBondListWithPredicate(bondList, isBondRotatable());
00284 }
00285
00286 } // namespace rxdock
00287
00288 #endif //_RBTBOND_H_
```

6.23 Cavity.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Struct for holding a coordinate list representing an active site cavity
00014 // Properties are read-only once set in the constructor
00015
00016 #ifndef _RBTCAVITY_H_
00017 #define _RBTCAVITY_H_
00018
00019 #include "rxdock/Config.h"
00020 #include "rxdock/Coord.h"
00021 #include "rxdock/PrincipalAxes.h"
00022 #include "rxdock/RealGrid.h"
00023
00024 namespace rxdock {
```

```
00025
00026 class Cavity {
00027 public:
00029   // Constructors/destructors
00030
00031   // Constructor
00032   Cavity(const CoordList &coordList, const Vector gridStep)
00033       : m_coordList(coordList), m_gridStep(gridStep),
00034         m_prAxes(GetPrincipalAxesOfAtoms(coordList)),
00035         m_minCoord(Min(coordList)), m_maxCoord(Max(coordList)) {}
00036
00037   // Constructor reading from binary stream
00038   Cavity(json j) { j.get_to(*this); }
00039
00040   // Destructor
00041   virtual ~Cavity() {}
00042
00044   // Stream functions
00046
00047   // Insertion operator
00048   friend std::ostream &operator«(std::ostream &s, const Cavity &cavity) {
00049     cavity.Print(s);
00050     return s;
00051   }
00052
00053   // Virtual function for dumping cavity details to an output stream
00054   // Derived classes can override if required
00055   virtual void Print(std::ostream &s) const {
00056     s « "Size=" « m_coordList.size() « " points; Vol=" « GetVolume()
00057       « " A^3; Min=" « m_minCoord « "; Max=" « m_maxCoord
00058       « "; Center=" « m_prAxes.com « "; Extent=" « m_maxCoord - m_minCoord;
00059   }
00060
00061   // DM 4 Apr 2002 - return a grid with all cavity points set to 1.0
00062   RealGridPtr GetGrid() const {
00063     Vector extent = m_maxCoord - m_minCoord;
00064     // Add +3 to allow a small border
00065     Eigen::Vector3d nXYZ = extent.xyz.array() / m_gridStep.xyz.array();
00066     unsigned int nX = static_cast<unsigned int>(nXYZ(0)) + 3;
00067     unsigned int nY = static_cast<unsigned int>(nXYZ(1)) + 3;
00068     unsigned int nZ = static_cast<unsigned int>(nXYZ(2)) + 3;
00069     RealGridPtr spGrid = RealGridPtr(
00070         new RealGrid(m_minCoord - m_gridStep, m_gridStep, nX, nY, nZ));
00071     for (CoordListConstIter iter = m_coordList.begin();
00072          iter != m_coordList.end(); iter++) {
00073       spGrid->SetValue(*iter, 1.0);
00074     }
00075     return spGrid;
00076   }
00077
00079   // Public methods
00081   const Coord &GetCenterOfMass() const { return m_prAxes.com; }
00082   const PrincipalAxes &GetPrincipalAxes() const { return m_prAxes; }
00083   int GetNumCoords() const { return m_coordList.size(); }
00084   const CoordList &GetCoordList() const { return m_coordList; }
00085   const Coord &GetMinCoord() const { return m_minCoord; }
00086   const Coord &GetMaxCoord() const { return m_maxCoord; }
00087   const Vector &GetGridStep() const { return m_gridStep; }
00088   double GetVolume() const {
00089     return m_coordList.size() * m_gridStep.xyz.prod();
00090   }
00091
00092   friend void to_json(json &j, const Cavity &c) {
00093     j = json{{"grid-step", c.m_gridStep}, {"coordinates", c.m_coordList}};
00094   }
00095
00096   friend void from_json(const json &j, Cavity &c) {
00097     j.at("grid-step").get_to(c.m_gridStep);
00098     j.at("coordinates").get_to(c.m_coordList);
00099   }
00100
00101 protected:
00103   // Protected methods
00105
00106 private:
00108   // Private methods
00110   Cavity();                           // Disable default constructor
00111   Cavity(const Cavity &);             // Copy constructor disabled by default
00112   Cavity &operator=(const Cavity &); // Copy assignment disabled by
00113                                       // default
00114
00115 protected:
00117   // Protected data
00119
00120 private:
00122   // Private data
00124   CoordList m_coordList;
```

```
00125   Vector m_gridStep;
00126   PrincipalAxes m_prAxes;
00127   Coord m_minCoord;
00128   Coord m_maxCoord;
00129 };
00130
00131 // Useful typedefs
00132 typedef SmartPtr<Cavity> CavityPtr;        // Smart pointer
00133 typedef std::vector<CavityPtr> CavityList; // Vector of smart pointers
00134 typedef CavityList::iterator CavityListIter;
00135 typedef CavityList::const_iterator CavityListConstIter;
00136
00138 // Comparison functions for sorting CavityPtr containers
00139 // For use by STL sort algorithms
00141 // Less than operator for sorting CavityPtrs into ascending order by center
00142 // of mass distance from a given coord
00143 class CavityPtrCmp_Distance {
00144   const Coord &c;
00145
00146 public:
00147   CavityPtrCmp_Distance(const Coord &cc) : c(cc) {}
00148   bool operator()(CavityPtr spCav1, CavityPtr spCav2) const {
00149     return Length2(spCav1->GetCenterOfMass() - c) <
00150            Length2(spCav2->GetCenterOfMass() - c);
00151   }
00152 };
00153 // Less than operator for sorting CavityPtrs into descending order by volume
00154 class CavityPtrCmp_Volume {
00155 public:
00156   CavityPtrCmp_Volume() {}
00157   bool operator()(CavityPtr spCav1, CavityPtr spCav2) const {
00158     return spCav1->GetNumCoords() > spCav2->GetNumCoords();
00159   }
00160 };
00162 // Predicate functions for CavityPtr
00163 // For use by STL algorithms
00165 class isCavityNearCoord : public std::function<bool(CavityPtr)> {
00166   const Coord &c;
00167   double r2; // radius squared (to avoid taking square roots)
00168 public:
00169   explicit isCavityNearCoord(const Coord &cc, double rr) : c(cc), r2(rr * rr) {}
00170   bool operator()(CavityPtr spCavity) const {
00171     return Length2(spCavity->GetCenterOfMass() - c) <= r2;
00172   }
00173 };
00174
00176 // Functions objects for performing actions on cavities
00177 // For use by STL algorithms
00179 inline const Coord &ExtractCavityMinCoord(CavityPtr spCav) {
00180   return spCav->GetMinCoord();
00181 }
00182 inline const Coord &ExtractCavityMaxCoord(CavityPtr spCav) {
00183   return spCav->GetMaxCoord();
00184 }
00185
00186 } // namespace rxdock
00187
00188 #endif //_RBTCAVITY_H_
```

6.24 CavityFillSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Post-filtering scoring function for calculating percentage cavity unfilled
00014 //(mode=0) or percentage ligand unbound (mode=1)
00015
00016 #ifndef _RBTCAVITYFILLSF_H_
00017 #define _RBTCAVITYFILLSF_H_
00018
00019 #include "rxdock/BaseInterSF.h"
00020 #include "rxdock/FFTGrid.h"
00021
00022 #include <nlohmann/json.hpp>
```

```
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class CavityFillSF : public BaseInterSF {
00029 public:
00030   // Class type string
00031   static const std::string _CT;
00032   // Parameter names
00033   CavityFillSF(const std::string &strName = "cavity-fill");
00034   virtual ~CavityFillSF();
00035
00036   friend void to_json(json &j, const CavityFillSF &sf);
00037   friend void from_json(const json &j, CavityFillSF &sf);
00038
00039 protected:
00040   virtual void SetupReceptor();
00041   virtual void SetupLigand();
00042   virtual void SetupScore();
00043   virtual double RawScore() const;
00044   void ParameterUpdated(const std::string &strName);
00045
00046 private:
00047   FFTGridPtr m_spGrid;
00048   AtomList m_ligAtomList;
00049 };
00050
00051 void to_json(json &j, const CavityFillSF &sf);
00052 void from_json(const json &j, CavityFillSF &sf);
00053
00054 } // namespace rxdock
00055
00056 #endif //_RBTCAVITYFILLSF_H_
```

6.25 CavityGridSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Penalty function for keeping ligand and solvent within the docking volume
00014 #ifndef _RBTCAVITYGRIDSF_H_
00015 #define _RBTCAVITYGRIDSF_H_
00016
00017 #include "rxdock/BaseInterSF.h"
00018 #include "rxdock/FlexDataVisitor.h"
00019 #include "rxdock/RealGrid.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class CavityGridSF : public BaseInterSF {
00028 public:
00029   // FlexDataVisitor class to generate list of movable non-hydrogen atoms
00030   // i.e. those atoms that have the potential to move outside of the cavity
00031   class RBTDLL_EXPORT HeavyAtomFactory : public FlexDataVisitor {
00032   public:
00033     HeavyAtomFactory() {}
00034     HeavyAtomFactory(ModelList modelList);
00035     virtual void VisitReceptorFlexData(ReceptorFlexData *);
00036     virtual void VisitLigandFlexData(LigandFlexData *);
00037     virtual void VisitSolventFlexData(SolventFlexData *);
00038     AtomRList GetAtomList() const { return m_atomList; }
00039
00040     friend void to_json(json &j, const HeavyAtomFactory &heavyAtomFactory);
00041     friend void from_json(const json &j, HeavyAtomFactory &heavyAtomFactory);
00042
00043   private:
00044     AtomRList m_atomList;
00045   };
00046
```

```
00047   void to_json(json &j, const HeavyAtomFactory &heavyAtomFactory);
00048   void from_json(const json &j, HeavyAtomFactory &heavyAtomFactory);
00049
00050   // Class type string
00051   static const std::string _CT;
00052   // Parameter names
00053   static const std::string _RMAX;
00054   static const std::string
00055       _QUADRATIC; // True = quadratic penalty function; false = linear
00056
00057   CavityGridSF(const std::string &strName = "cavity");
00058   virtual ~CavityGridSF();
00059
00060   friend void to_json(json &j, const CavityGridSF &cavGridSF);
00061   friend void from_json(const json &j, CavityGridSF &cavGridSF);
00062
00063 protected:
00064   virtual void SetupReceptor();
00065   virtual void SetupLigand();
00066   virtual void SetupSolvent();
00067   virtual void SetupScore();
00068   virtual double RawScore() const;
00069   // DM 25 Oct 2000 - track changes to parameter values in local data members
00070   // ParameterUpdated is invoked by ParamHandler::SetParameter
00071   void ParameterUpdated(const std::string &strName);
00072
00073 private:
00074   RealGridPtr m_spGrid;
00075   double m_maxDist;     // Max distance of any grid point from the cavity
00076   AtomRList m_atomList; // combined list of all movable heavy atoms
00077   double m_rMax;
00078   bool m_bQuadratic; // synchronised with QUADRATIC named parameter
00079 };
00080
00081 void to_json(json &j, const CavityGridSF &cavGridSF);
00082 void from_json(const json &j, CavityGridSF &cavGridSF);
00083
00084 } // namespace rxdock
00085
00086 #endif //_RBTCAVITYGRIDSF_H_
```

6.26 CellTokenIter.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Defines the interface for a token iterator to be used by the parser
00014
00015 #ifndef _RBT_CELLTOKENITER_H_
00016 #define _RBT_CELLTOKENITER_H_
00017
00018 #include "rxdock/Config.h"
00019 #include "rxdock/Context.h"
00020 #include "rxdock/Token.h"
00021 #include "rxdock/geneticprogram/GPChromosome.h"
00022 #include <stack>
00023
00024 #include "rxdock/TokenIter.h"
00025
00026 #include <nlohmann/json.hpp>
00027
00028 using json = nlohmann::json;
00029
00030 namespace rxdock {
00031
00032 class CellTokenIter : public TokenIter {
00033 public:
00034   static const std::string _CT;
00036   // Constructors
00038   CellTokenIter(const CellTokenIter &);
00039   CellTokenIter(const geneticprogram::GPChromosomePtr, ContextPtr);
00040
00042   // Destructor
00044   virtual ~CellTokenIter();
```

```
00045
00046   friend void to_json(json &j, const CellTokenIter &callTokIt);
00047   friend void from_json(const json &j, CellTokenIter &callTokIt);
00048
00049   virtual void copy(const CellTokenIter &);
00050   // void First();
00051   void Next(ContextPtr);
00052   // Bool IsDone() const;
00053   TokenPtr Current();
00054
00056   // Private methods
00058
00059 private:
00060   CellTokenIter(); // Default constructor disabled
00061   geneticprogram::GPChromosomePtr chrom;
00062   TokenPtr current;
00063   std::stack<int> cells;
00064   ContextPtr contextp;
00065 };
00066
00067 void to_json(json &j, const CellTokenIter &callTokIt);
00068 void from_json(const json &j, CellTokenIter &callTokIt);
00069
00070 // Useful typedefs
00071 typedef SmartPtr<CellTokenIter> CellTokenIterPtr; // Smart pointer
00072 typedef std::vector<CellTokenIterPtr>
00073     CellTokenIterList; // Vector of smart pointers
00074 typedef CellTokenIterList::iterator CellTokenIterListIter;
00075 typedef CellTokenIterList::const_iterator CellTokenIterListConstIter;
00076
00077 } // namespace rxdock
00078
00079 #endif //_CellTokenIter_H_
```

6.27 CharmmDataSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Class for providing Charmm force field data
00014
00015 #ifndef _RBTCHARMMDATASOURCE_H_
00016 #define _RBTCHARMMDATASOURCE_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/Config.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // Added by DM, 8 Dec 1998 for HybridState lookup method
00028 typedef std::map<std::string, Atom::eHybridState> StringHybridStateMap;
00029 typedef StringHybridStateMap::iterator StringHybridStateMapIter;
00030 typedef StringHybridStateMap::const_iterator StringHybridStateMapConstIter;
00031
00032 class CharmmDataSource {
00033 public:
00035   // Constructors/destructors
00036
00037   // DM 30 Apr 1999 - pass in masses.rtf file name as parameter
00038   CharmmDataSource(
00039       const std::string &strMassesFile = GetDataFileName("data", "masses.rtf"));
00040
00041   ~CharmmDataSource(); // Default destructor
00042
00043   friend void to_json(json &j, const CharmmDataSource &charDatSrc);
00044   friend void from_json(const json &j, CharmmDataSource &charDatSrc);
00045
00047   // Public methods
00049   std::string AtomTypeString(int nAtomType);
00050   int ImplicitHydrogens(const std::string &strFFType);
```

```
00051   int AtomicNumber(const std::string &strFFType);
00052   int FormalCharge(const std::string &strFFType); // DM 24 Mar 1999 - changed
00053                                                   // from double to int
00054   Atom::eHybridState HybridState(const std::string &strFFType); // DM 8 Dec 1998
00055
00056 protected:
00058   // Protected methods
00060
00061 private:
00063   // Private methods
00065
00066   CharmmDataSource(
00067       const CharmmDataSource &); // Copy constructor disabled by default
00068
00069   CharmmDataSource &
00070   operator=(const CharmmDataSource &); // Copy assignment disabled by default
00071
00072   // DM 8 Dec 1998 Searches for Hybridisation state string in the masses.rtf
00073   // comment field Valid strings are (RBT::SP), (RBT::SP2), (RBT::SP3),
00074   // (RBT::AROM), (RBT::TRI) (brackets are important)
00075   Atom::eHybridState
00076   ConvertCommentStringToHybridState(const std::string &strComment);
00077
00078 protected:
00080   // Protected data
00082
00083 private:
00085   // Private data
00087   std::map<int, std::string> m_atomTypes;
00088   std::map<std::string, int> m_implicitHydrogens;
00089   std::map<std::string, int> m_atomicNumber;
00090   std::map<std::string, int> m_formalCharge;
00091   StringHybridStateMap m_hybridState; // DM 8 Dec 1998
00092 };
00093
00094 void to_json(json &j, const CharmmDataSource &charDatSrc);
00095 void from_json(const json &j, CharmmDataSource &charDatSrc);
00096
00097 // useful typedefs
00098 typedef SmartPtr<CharmmDataSource> CharmmDataSourcePtr; // Smart pointer
00099
00100 } // namespace rxdock
00101
00102 #endif //_RBTCHARMMDATASOURCE_H_
```

6.28 CharmmTypesFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // File reader for Charmm MASSES.RTF atom type file
00014 // Provides the atom type list as a vector of structs
00015
00016 #ifndef _RBTCHARMMTYPESFILESOURCE_H_
00017 #define _RBTCHARMMTYPESFILESOURCE_H_
00018
00019 #include "rxdock/BaseFileSource.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // Simple struct for holding the type info in masses.rtf
00028 class CharmmType {
00029 public:
00030   CharmmType()
00031       : nAtomType(0), strAtomType(""), mass(0.0), element(""), comment("") {}
00032   int nAtomType;
00033   std::string strAtomType;
00034   double mass;
00035   std::string element;
00036   std::string comment;
```

```
00037 };
00038
00039 void to_json(json &j, const CharmmType &charmmType);
00040 void from_json(const json &j, CharmmType &charmmType);
00041
00042 typedef std::vector<CharmmType> CharmmTypeList;
00043 typedef CharmmTypeList::iterator CharmmTypeListIter;
00044
00045 class CharmmTypesFileSource : public BaseFileSource {
00046 public:
00047   // Constructors
00048   CharmmTypesFileSource(const char *fileName);
00049   CharmmTypesFileSource(const std::string fileName);
00050
00051   // Destructor
00052   virtual ~CharmmTypesFileSource();
00053
00054   friend void to_json(json &j,
00055                       const CharmmTypesFileSource &charmmTypesFileSource);
00056   friend void from_json(const json &j,
00057                         CharmmTypesFileSource &charmmTypesFileSource);
00058
00060   // Public methods
00061   int GetNumTypes();
00062   CharmmTypeList GetTypeList();
00063
00064 protected:
00065   // Protected methods
00066
00067 private:
00068   // Private methods
00069   CharmmTypesFileSource(); // Disable default constructor
00070   CharmmTypesFileSource(
00071       const CharmmTypesFileSource &); // Copy constructor disabled by default
00072   CharmmTypesFileSource &operator=(
00073       const CharmmTypesFileSource &); // Copy assignment disabled by default
00074
00075   // Pure virtual in BaseFileSource - needs to be defined here
00076   virtual void Parse();
00077   void ClearTypesCache();
00078
00079 protected:
00080   // Protected data
00081
00082 private:
00083   // Private data
00084   CharmmTypeList m_typesList;
00085 };
00086
00087 void to_json(json &j, const CharmmTypesFileSource &charmmTypesFileSource);
00088 void from_json(const json &j, CharmmTypesFileSource &charmmTypesFileSource);
00089
00090 // useful typedefs
00091 typedef SmartPtr<CharmmTypesFileSource>
00092     CharmmTypesFileSourcePtr; // Smart pointer
00093
00094 } // namespace rxdock
00095
00096 #endif //_RBTCHARMMTYPESFILESOURCE_H_
```

6.29 Chrom.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Aggregate class for managing a vector of ChromElement objects
00014 // Also implements the ChromElement interface itself
00015 // to delegate requests to each element in the vector
00016 // NOTE: Chrom destructor is responsible for deleting all managed elements
00017 #ifndef RBTCHROM_H_
00018 #define RBTCHROM_H_
00019
00020 #include "rxdock/ChromElement.h"
00021 #include "rxdock/Error.h"
```

```
00022 #include "rxdock/Model.h"
00023
00024 namespace rxdock {
00025
00026 class Chrom : public ChromElement {
00027 public:
00028   // Class type string
00029   static const std::string _CT;
00030   // Constructor for an empty chromosome
00031   RBTDLL_EXPORT Chrom();
00032   // Constructor for a combined chromosome for each model
00033   // in the list
00034   Chrom(const ModelList &modelList);
00035   virtual ~Chrom();
00036
00037   // pure virtual from ChromElement
00038   virtual void Reset();
00039   virtual void Randomise();
00040   virtual void Mutate(double relStepSize);
00041   virtual void SyncFromModel();
00042   virtual void SyncToModel();
00043   virtual ChromElement *clone() const;
00044   virtual int GetLength() const;
00045   virtual int GetXOverLength() const;
00046   virtual void GetVector(std::vector<double> &v) const;
00047   virtual void GetVector(XOverList &v) const;
00048   virtual void SetVector(const std::vector<double> &v, int &i);
00049   virtual void SetVector(const XOverList &v, int &i);
00050   virtual void GetStepVector(std::vector<double> &v) const;
00051   virtual double CompareVector(const std::vector<double> &v, int &i) const;
00052   virtual void Print(std::ostream &s) const;
00053
00054   // Aggregate methods
00055   // Appends a new chromosome element to the vector
00056   // Chrom destructor is responsible for deleting the new element
00057   // Null operation if pChromElement is nullptr
00058   virtual void Add(ChromElement *pChromElement);
00059
00060 protected:
00061 private:
00062   ChromElementList m_elementList;
00063   // We need to store the model list so that
00064   // we can call UpdatePseudoAtoms() on each model following
00065   // a SyncToModel
00066   ModelList m_modelList;
00067 };
00068
00069 } // namespace rxdock
00070
00071 #endif /*RBTCHROM_H_*/
```

6.30 ChromDihedralElement.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Chromosome element for a single dihedral angle
00014 #ifndef RBTCHROMDIHEDRALELEMENT_H_
00015 #define RBTCHROMDIHEDRALELEMENT_H_
00016
00017 #include "rxdock/ChromDihedralRefData.h"
00018 #include "rxdock/ChromElement.h"
00019 #include "rxdock/Rand.h"
00020
00021 namespace rxdock {
00022
00023 class ChromDihedralElement : public ChromElement {
00024 public:
00025   // Class type string
00026   static const std::string _CT;
00027   // Sole constructor
00028   // If the tetheredAtoms list is empty, then
00029   //  the end of the bond with the fewest pendant atoms is rotated (other half
00030   //  remains fixed)
```

```
00031   // else if the tetheredAtoms list is not empty, then
00032   //  the end of the bond with the fewest tethered atoms is rotated (other half
00033   //  remains fixed)
00034   RBTDLL_EXPORT ChromDihedralElement(
00035       BondPtr spBond,          // Rotatable bond
00036       AtomList tetheredAtoms, // Tethered atom list
00037       double stepSize,         // maximum mutation step size (degrees)
00038       ChromElement::eMode mode = ChromElement::FREE, // sampling mode
00039       double maxDihedral =
00040           0.0); // max deviation from reference (tethered mode only)
00041
00042   virtual ~ChromDihedralElement();
00043   virtual void Reset();
00044   virtual void Randomise();
00045   virtual void Mutate(double relStepSize);
00046   virtual void SyncFromModel();
00047   virtual void SyncToModel();
00048   virtual ChromElement *clone() const;
00049   virtual int GetLength() const { return 1; }
00050   virtual int GetXOverLength() const { return 1; }
00051   virtual void GetVector(std::vector<double> &v) const;
00052   virtual void GetVector(XOverList &v) const;
00053   virtual void SetVector(const std::vector<double> &v, int &i);
00054   virtual void SetVector(const XOverList &v, int &i);
00055   virtual void GetStepVector(std::vector<double> &v) const;
00056   virtual double CompareVector(const std::vector<double> &v, int &i) const;
00057   virtual void Print(std::ostream &s) const;
00058
00059   // Returns a standardised dihedral angle in the range [-180, +180}
00060   // This function operates in degrees
00061   static double StandardisedValue(double dihedralAngle);
00062
00063 protected:
00064   // For use by clone()
00065   ChromDihedralElement(ChromDihedralRefDataPtr spRefData, double value);
00066   ChromDihedralElement();
00067   void CorrectTetheredDihedral();
00068
00069 private:
00070   ChromDihedralRefDataPtr m_spRefData; // Fixed reference data
00071   double m_value;                      // The genotype value
00072 };
00073
00074 } // namespace rxdock
00075
00076 #endif /*RBTCHROMDIHEDRALELEMENT_H_*/
```

6.31 ChromDihedralRefData.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages the fixed reference data for a single dihedral angle chromosome
00014 // element Also provides methods to map the genotype (dihedral angle value) onto
00015 // the phenotype (model coords) A single instance is designed to be shared
00016 // between all clones of a given element
00017 #ifndef RBTCHROMDIHEDRALREFDATA_H_
00018 #define RBTCHROMDIHEDRALREFDATA_H_
00019
00020 #include "rxdock/Atom.h"
00021 #include "rxdock/Bond.h"
00022 #include "rxdock/ChromElement.h"
00023
00024 namespace rxdock {
00025
00026 class ChromDihedralRefData {
00027 public:
00028   // Class type string
00029   static const std::string _CT;
00030   // Sole constructor
00031   // If the tetheredAtoms list is empty, then
00032   //  the end of the bond with the fewest pendant atoms is rotated (other half
00033   //  remains fixed)
00034   // else if the tetheredAtoms list is not empty, then
```

```
00035   //  the end of the bond with the fewest tethered atoms is rotated (other half
00036   //  remains fixed)
00037   ChromDihedralRefData(
00038       BondPtr spBond,         // Rotatable bond
00039       AtomList tetheredAtoms, // Tethered atom list
00040       double stepSize,        // maximum mutation step size (degrees)
00041       ChromElement::eMode mode = ChromElement::FREE, // sampling mode
00042       double maxDihedral =
00043           0.0); // max deviation from reference (tethered mode only)
00044   virtual ~ChromDihedralRefData();
00045
00046   // Gets the maximum step size for this bond
00047   double GetStepSize() const { return m_stepSize; }
00048   // Gets the sampling mode for this bond
00049   ChromElement::eMode GetMode() const { return m_mode; }
00050   // Gets the maximum deviation from the reference dihedral (tethered mode only)
00051   double GetMaxDihedral() const { return m_maxDihedral; }
00052   // Gets the current dihedral angle (phenotype) for this bond
00053   // from the model coords
00054   double GetModelValue() const;
00055   // Sets the phenotype (model coords) for this bond
00056   // to a given dihedral angle
00057   void SetModelValue(double dihedralAngle);
00058   // Gets the initial dihedral angle for this bond
00059   //(initialised from model coords in ChromDihedralRefData constructor)
00060   double GetInitialValue() const { return m_initialValue; }
00061
00062 private:
00063   // Sets up the movable atom list for this bond
00064   void Setup(BondPtr spBond, const AtomList &tetheredAtoms);
00065   Atom *m_atom1;
00066   Atom *m_atom2;
00067   Atom *m_atom3;
00068   Atom *m_atom4;
00069   AtomRList m_rotAtoms;
00070   double m_stepSize;
00071   double m_initialValue;
00072   ChromElement::eMode m_mode;
00073   double m_maxDihedral; // max deviation from reference (tethered mode only)
00074 };
00075
00076 typedef SmartPtr<ChromDihedralRefData> ChromDihedralRefDataPtr; // Smart pointer
00077
00078 } // namespace rxdock
00079
00080 #endif /*RBTCHROMDIHEDRALREFDATA_H_*/
```

6.32 ChromElement.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract interface for all chromosome elements (degrees of freedom)
00014 #ifndef RBTCHROMELEMENT_H_
00015 #define RBTCHROMELEMENT_H_
00016
00017 #include "rxdock/Config.h"
00018 #include "rxdock/Rand.h"
00019
00020 namespace rxdock {
00021
00022 // Typedefs for crossover data.
00023 // To prevent splitting (for example) an orientation or position vector in two
00024 // during crossover, we convert the chromosome to a vector of vector of doubles
00025 // An orientation vector (3 Euler angles) is a single XOverElement
00026 // GetVector(std::vector<double>&) returns a flat vector of doubles (for use by
00027 // Simplex) GetVector(XOverList&) returns a vector of XOverElements (for
00028 // use by crossover)
00029 typedef std::vector<double> XOverElement;
00030 typedef XOverElement::iterator XOverElementIter;
00031 typedef XOverElement::const_iterator XOverElementConstIter;
00032 typedef std::vector<XOverElement> XOverList;
00033 typedef XOverList::iterator XOverListIter;
00034 typedef XOverList::const_iterator XOverListConstIter;
```

```
00035
00036 class ChromElement {
00037 public:
00038   // Class type string
00039   static const std::string _CT;
00040   // Threshold used to assess equality of two chromosome elements
00041   static double _THRESHOLD;
00042
00043   RBTDLL_EXPORT static double &GetThreshold();
00044
00045   // General enum for defining types of chromosome element flexibility.
00046   // For use by subclasses of ChromElement if desired.
00047   // There is no functionality in the base class that uses this enum.
00048   enum eMode { FIXED = 0, TETHERED = 1, FREE = 2 };
00049   // Static methods to convert from mode enum to string and vice versa
00050   static eMode StrToMode(const std::string &modeStr); // case insensitive
00051   static std::string RBTDLL_EXPORT
00052   ModeToStr(eMode mode); // returns "FIXED", "TETHERED" or "FREE"ss
00053
00054   virtual ~ChromElement();
00055   //
00056   // INTERFACE METHODS - must be implemented in subclasses
00057   //
00058   // Resets the chromosome element to its initial value (extracted from
00059   // model coords at time of construction)
00060   virtual void Reset() = 0;
00061   // Randomises the chromosome element within the context of any restraints
00062   virtual void Randomise() = 0;
00063   // Mutates the chromosome element
00064   // relStepSize = fraction (0.0->1.0) of maximum defined step size
00065   virtual void Mutate(double relStepSize) = 0;
00066   // Updates the chromosome element to match the current model coords
00067   virtual void SyncFromModel() = 0;
00068   // Updates the model coords to match the chromosome element
00069   virtual void SyncToModel() = 0;
00070   // Creates an independent clone
00071   // The current value of the cloned element should match the current value
00072   // of this element.
00073   virtual ChromElement *clone() const = 0;
00074   // Gets the number of double values needed to represent this
00075   // chromosome element
00076   virtual int GetLength() const = 0;
00077   // Gets the number of vectors of double values needed to represent this
00078   // chromosome element for crossover purposes
00079   // e.g. an orientation vector consists of 3 doubles, but should be crossed
00080   // over as a single intact entity
00081   virtual int GetXOverLength() const = 0;
00082   // Converts chromosome element to a vector of double values,
00083   // Number of double values appended should match GetLength().
00084   // v = vector of doubles to append to.
00085   virtual void GetVector(std::vector<double> &v) const = 0;
00086   // Converts chromosome element to a vector of XOverElements
00087   // where each XOverElement is itself a vector of double values
00088   // v = vector of XOverElements to append to.
00089   virtual void GetVector(XOverList &v) const = 0;
00090   // Updates chromosome element from a vector of double values,
00091   // Number of double values read should match GetLength().
00092   // v = vector of doubles to extract from
00093   // i = index of next vector element to read (should be updated by method)
00094   virtual void SetVector(const std::vector<double> &v, int &i) = 0;
00095   // Updates chromosome element from a vector of XOverElements,
00096   // Number of XOverElements read should match GetXOverLength().
00097   // v = vector of XOverElements to extract from
00098   // i = index of next vector element to read (should be updated by method)
00099   virtual void SetVector(const XOverList &v, int &i) = 0;
00100   // Gets the vector of absolute step sizes that correspond to each double
00101   // value.
00102   virtual void GetStepVector(std::vector<double> &v) const = 0;
00103   // Gets the maximum relative difference between this element and another
00104   // element as represented by the vector. Differences should be normalised by
00105   // the step size. This method is provided to allow better comparisons between
00106   // values that are difficult to compare by simple numerical differences. e.g.
00107   // Dihedral angles are cyclical, therefore -180 and + 179 only differ by 1
00108   // deg.
00109   virtual double CompareVector(const std::vector<double> &v, int &i) const = 0;
00110   //
00111   // IMPLEMENTED VIRTUAL METHODS
00112   //
00113   // Invalid operation in base class
00114   virtual void Add(ChromElement *pChromElement);
00115   // Prints details of element to stream (null implementation in base class)
00116   virtual void Print(std::ostream &s) const {}
00117   //
00118   // NON-VIRTUAL METHODS
00119   //
00120   Rand &GetRand() const { return m_rand; }
00121   void CauchyMutate(double mean, double variance);
```

```
00122   // Compares two chromosome elements. Returns -1 if the comparison is invalid
00123   //(unequal lengths), else returns the maximum relative pair-wise difference
00124   // as returned by CompareVector.
00125   RBTDLL_EXPORT double Compare(const ChromElement &c) const;
00126   // Returns true if chromosome elements have near-equal values.
00127   // i.e. if Compare(c) < threshold
00128   // Returns false if comparison is invalid (unequal lengths)
00129   bool Equals(const ChromElement &c, double threshold) const;
00130   // Convenience method that calls SetVector(const std::vector<double>& v,
00131   // Int& i) with i initialised to zero
00132   void SetVector(const std::vector<double> &v);
00133   // Convenience method that calls SetVector(const XOverList& v, Int& i)
00134   // with i initialised to zero
00135   void SetVector(const XOverList &v);
00136   // operator== and operator!= are implemented by calling Equals with the
00137   // static _THRESHOLD value
00138   RBTDLL_EXPORT friend bool operator==(const ChromElement &c1,
00139                                        const ChromElement &c2);
00140   RBTDLL_EXPORT friend bool operator!=(const ChromElement &c1,
00141                                        const ChromElement &c2);
00142   // operator« is implemented by calling Print()
00143   RBTDLL_EXPORT friend std::ostream &operator«(std::ostream &s,
00144                                                const ChromElement &c);
00145
00146 protected:
00147   RBTDLL_EXPORT ChromElement();
00148   RBTDLL_EXPORT ChromElement(const ChromElement &c);
00149   RBTDLL_EXPORT ChromElement &operator=(const ChromElement &c);
00150   // Helper method to check that index i is in range [0,v.size()}
00151   // and that v has sufficient elements remaining to satisfy
00152   // GetLength()
00153   bool VectorOK(const std::vector<double> &v, unsigned int i) const;
00154   // Helper method to check that index i is in range [0,v.size()}
00155   // and that v has sufficient elements remaining to satisfy
00156   // GetXOverLength()
00157   bool VectorOK(const XOverList &v, unsigned int i) const;
00158
00159 private:
00160   Rand &m_rand; // Reference to singleton random number generator
00161 };
00162
00163 typedef SmartPtr<ChromElement> ChromElementPtr;
00164 typedef std::vector<ChromElement *>
00165     ChromElementList; // Vector of regular pointers
00166 typedef ChromElementList::iterator ChromElementListIter;
00167 typedef ChromElementList::const_iterator ChromElementListConstIter;
00168
00169 bool operator==(const ChromElement &c1, const ChromElement &c2);
00170 bool operator!=(const ChromElement &c1, const ChromElement &c2);
00171 std::ostream &operator«(std::ostream &s, const ChromElement &c);
00172
00173 // 2-point crossover
00174 RBTDLL_EXPORT void Crossover(ChromElement *pChr1, ChromElement *pChr2,
00175                              ChromElement *pChr3, ChromElement *pChr4);
00176
00177 } // namespace rxdock
00178
00179 #endif /*RBTCHROMELEMENT_H_*/
```

6.33 ChromFactory.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Chromosome factory class
00014 // Creates an Chrom aggregate by visiting a number of
00015 // FlexData subclasses.
00016 // It is the users responsibility to delete the
00017 // chromosome returned by GetChrom() when no longer required.
00018 // No smart pointers are used within ChromFactory.
00019 #ifndef _RBTCHROMFACTORY_H_
00020 #define _RBTCHROMFACTORY_H_
00021
00022 #include "rxdock/FlexDataVisitor.h"
```

```
00023 #include "rxdock/ModelMutator.h"
00024
00025 namespace rxdock {
00026
00027 class ChromElement;
00028
00029 class RBTDLL_EXPORT ChromFactory : public FlexDataVisitor {
00030 public:
00031   ChromFactory();
00032   virtual void VisitReceptorFlexData(ReceptorFlexData *);
00033   virtual void VisitLigandFlexData(LigandFlexData *);
00034   virtual void VisitSolventFlexData(SolventFlexData *);
00035
00036   ChromElement *GetChrom() const;
00037   // Temporary solution whilst we replace the ModelMutator class
00038   ModelMutatorPtr GetModelMutator() const { return m_spMutator; }
00039
00040 private:
00041   ChromElement *m_pChrom;
00042   ModelMutatorPtr m_spMutator;
00043 };
00044
00045 } // namespace rxdock
00046
00047 #endif //_RBTCHROMFACTORY_H_
```

6.34 ChromOccupancyElement.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Chromosome element for solvent model occupancy
00014 #ifndef RBTCHROMOCCUPANCYELEMENT_H_
00015 #define RBTCHROMOCCUPANCYELEMENT_H_
00016
00017 #include "rxdock/ChromElement.h"
00018 #include "rxdock/ChromOccupancyRefData.h"
00019 #include "rxdock/Rand.h"
00020
00021 namespace rxdock {
00022
00023 class ChromOccupancyElement : public ChromElement {
00024 public:
00025   // Class type string
00026   static const std::string _CT;
00027   // Sole constructor
00028   RBTDLL_EXPORT ChromOccupancyElement(Model *pModel, double stepSize,
00029                                       double threshold);
00030
00031   virtual ~ChromOccupancyElement();
00032   virtual void Reset();
00033   virtual void Randomise();
00034   virtual void Mutate(double relStepSize);
00035   virtual void SyncFromModel();
00036   virtual void SyncToModel();
00037   virtual ChromElement *clone() const;
00038   virtual int GetLength() const { return 1; }
00039   virtual int GetXOverLength() const { return 1; }
00040   virtual void GetVector(std::vector<double> &v) const;
00041   virtual void GetVector(XOverList &v) const;
00042   virtual void SetVector(const std::vector<double> &v, int &i);
00043   virtual void SetVector(const XOverList &v, int &i);
00044   virtual void GetStepVector(std::vector<double> &v) const;
00045   virtual double CompareVector(const std::vector<double> &v, int &i) const;
00046   virtual void Print(std::ostream &s) const;
00047
00048   // Returns a standardised occupancy value in the range [0,1]
00049   static double StandardisedValue(double occupancy);
00050
00051 protected:
00052   // For use by clone()
00053   ChromOccupancyElement(ChromOccupancyRefDataPtr spRefData, double value);
00054   ChromOccupancyElement();
00055
```

```
00056 private:
00057   ChromOccupancyRefDataPtr m_spRefData; // Fixed reference data
00058   double m_value;                       // The genotype value
00059 };
00060
00061 } // namespace rxdock
00062
00063 #endif /*RBTCHROMOCCUPANCYELEMENT_H_*/
```

6.35 ChromOccupancyRefData.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages the fixed reference data for a model occupancy chromosome element
00014 // A single instance is designed to be shared between all clones of a given
00015 // element
00016 #ifndef RBTCHROMOCCUPANCYREFDATA_H_
00017 #define RBTCHROMOCCUPANCYREFDATA_H_
00018
00019 #include "rxdock/Config.h"
00020
00021 namespace rxdock {
00022
00023 class Model;
00024
00025 class ChromOccupancyRefData {
00026 public:
00027   // Class type string
00028   static const std::string _CT;
00029   // Sole constructor
00030   ChromOccupancyRefData(
00031       Model *pModel,
00032       double stepSize,   // mutation step size
00033       double threshold); // threshold for enabling/disabling atoms
00034   virtual ~ChromOccupancyRefData();
00035
00036   double GetStepSize() const { return m_stepSize; }
00037   double GetModelValue() const;
00038   void SetModelValue(double occupancy);
00039   double GetInitialValue() const { return m_initialValue; }
00040
00041 private:
00042   Model *m_pModel;
00043   double m_stepSize;
00044   double m_threshold;
00045   double m_initialValue;
00046 };
00047
00048 typedef SmartPtr<ChromOccupancyRefData>
00049     ChromOccupancyRefDataPtr; // Smart pointer
00050
00051 } // namespace rxdock
00052
00053 #endif /*RBTCHROMOCCUPANCYREFDATA_H_*/
```

6.36 ChromPositionElement.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
```

```
00013 // Chromosome element for model centre of mass and orientation
00014 #ifndef RBTCHROMPOSITIONELEMENT_H_
00015 #define RBTCHROMPOSITIONELEMENT_H_
00016
00017 #include "rxdock/ChromElement.h"
00018 #include "rxdock/ChromPositionRefData.h"
00019 #include "rxdock/Rand.h"
00020
00021 namespace rxdock {
00022
00023 class ChromPositionElement : public ChromElement {
00024 public:
00025   // Class type string
00026   static const std::string _CT;
00027
00028   RBTDLL_EXPORT
00029   ChromPositionElement(const Model *pModel, const DockingSite *pDockSite,
00030                        double transStepSize, // Angstroms
00031                        double rotStepSize,   // Radians
00032                        ChromElement::eMode transMode = ChromElement::FREE,
00033                        ChromElement::eMode rotMode = ChromElement::FREE,
00034                        double maxTrans = 0.0, // Angstroms
00035                        double maxRot = 0.0);  // radians
00036   virtual ~ChromPositionElement();
00037   virtual void Reset();
00038   virtual void Randomise();
00039   virtual void Mutate(double relStepSize);
00040   virtual void SyncFromModel();
00041   virtual void SyncToModel();
00042   virtual ChromElement *clone() const;
00043   virtual int GetLength() const { return m_spRefData->GetLength(); }
00044   virtual int GetXOverLength() const { return m_spRefData->GetXOverLength(); }
00045   virtual void GetVector(std::vector<double> &v) const;
00046   virtual void GetVector(XOverList &v) const;
00047   virtual void SetVector(const std::vector<double> &v, int &i);
00048   virtual void SetVector(const XOverList &v, int &i);
00049   virtual void GetStepVector(std::vector<double> &v) const;
00050   virtual double CompareVector(const std::vector<double> &v, int &i) const;
00051   virtual void Print(std::ostream &s) const;
00052
00053   // Returns a standardised rotation angle in the range [-M_PI, +M_PI}
00054   // This function operates in radians
00055   static double StandardisedValue(double rotationAngle);
00056
00057 protected:
00058   // For use by clone()
00059   ChromPositionElement(ChromPositionRefDataPtr spRefData, const Coord &com,
00060                        const Euler &orientation);
00061   ChromPositionElement();
00062   void RandomiseCOM();
00063   void RandomiseOrientation();
00064   void MutateCOM(double relStepSize);
00065   void MutateOrientation(double relStepSize);
00066   void CorrectTetheredCOM();
00067   void CorrectTetheredOrientation();
00068
00069 private:
00070   ChromPositionRefDataPtr m_spRefData; // Fixed reference data
00071   Coord m_com;                         // Centre of mass genotype value
00072   Euler m_orientation;                 // Euler angle orientation genotype value
00073 };
00074
00075 } // namespace rxdock
00076
00077 #endif /*RBTCHROMPOSITIONELEMENT_H_*/
```

6.37 ChromPositionRefData.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages the fixed reference data for a position chromosome element
00014 // Also provides methods to map the genotype (COM and Euler angles) onto the
00015 // phenotype (model coords)
```

```
00016 // A single instance is designed to be shared between all clones of a given
00017 // element
00018 #ifndef RBTCHROMPOSITIONREFDATA_H_
00019 #define RBTCHROMPOSITIONREFDATA_H_
00020
00021 #include "rxdock/ChromElement.h"
00022 #include "rxdock/DockingSite.h"
00023 #include "rxdock/Euler.h"
00024 #include "rxdock/Model.h"
00025
00026 #include <nlohmann/json.hpp>
00027
00028 using json = nlohmann::json;
00029
00030 namespace rxdock {
00031
00032 class ChromPositionRefData {
00033 public:
00034   // Class type string
00035   static const std::string _CT;
00036   // Reference Cartesian axes
00037   static const PrincipalAxes CARTESIAN_AXES;
00038   ChromPositionRefData(const Model *pModel, const DockingSite *pDockSite,
00039                        double transStepSize, // Angstroms
00040                        double rotStepSize,   // radians
00041                        ChromElement::eMode transMode = ChromElement::FREE,
00042                        ChromElement::eMode rotMode = ChromElement::FREE,
00043                        double maxTrans = 0.0, // Angstroms
00044                        double maxRot = 0.0);  // radians
00045   virtual ~ChromPositionRefData();
00046
00047   int GetNumStartCoords() const { return m_startCoords.size(); }
00048   const Coord &GetStartCoord(int iCoord) const { return m_startCoords[iCoord]; }
00049   double GetTransStepSize() const { return m_transStepSize; }
00050   double GetRotStepSize() const { return m_rotStepSize; }
00051   ChromElement::eMode GetTransMode() const { return m_transMode; }
00052   ChromElement::eMode GetRotMode() const { return m_rotMode; }
00053   // Chromosome length, excluding FIXED modes (0, 3 or 6)
00054   int GetLength() const { return m_length; }
00055   // Chromosome length for crossover, excluding FIXED modes (0, 1 or 2)
00056   int GetXOverLength() const { return m_xOverLength; }
00057   bool IsTransFixed() const { return m_transMode == ChromElement::FIXED; }
00058   bool IsRotFixed() const { return m_rotMode == ChromElement::FIXED; }
00059   double GetMaxTrans() const { return m_maxTrans; }
00060   double GetMaxRot() const { return m_maxRot; }
00061   const Coord &GetInitialCOM() const { return m_initialCom; }
00062   const Euler &GetInitialOrientation() const { return m_initialOrientation; }
00063   const Quat &GetInitialQuat() const { return m_initialQuat; }
00064
00065   void GetModelValue(Coord &com, Euler &orientation) const;
00066   void SetModelValue(const Coord &com, const Euler &orientation);
00067
00068   friend void to_json(json &j, const ChromPositionRefData &chrposrdata);
00069   friend void from_json(const json &j, ChromPositionRefData &chrposrdata);
00070
00071 private:
00072   AtomList m_refAtoms;
00073   AtomRList m_movableAtoms;
00074   CoordList m_startCoords;
00075   double m_transStepSize;
00076   double m_rotStepSize;
00077   Coord m_initialCom;
00078   Euler m_initialOrientation;
00079   Quat m_initialQuat;
00080   ChromElement::eMode m_transMode;
00081   ChromElement::eMode m_rotMode;
00082   int m_length;
00083   int m_xOverLength;
00084   // Max distance allowed from starting coord
00085   // Only used if m_transMode == TETHERED
00086   double m_maxTrans;
00087   // Max rot allowed from starting orientation
00088   // Only used if m_rotMode == TETHERED
00089   double m_maxRot;
00090 };
00091
00092 void to_json(json &j, const ChromPositionRefData &chrposrdata);
00093 void from_json(const json &j, ChromPositionRefData &chrposrdata);
00094
00095 typedef SmartPtr<ChromPositionRefData> ChromPositionRefDataPtr; // Smart pointer
00096
00097 } // namespace rxdock
00098
00099 #endif /*RBTCHROMPOSITIONREFDATA_H_*/
```

6.38 Commands.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Defines the interface for a token to be used by the parser
00014
00015 #ifndef _RBT_COMMANDS_H_
00016 #define _RBT_COMMANDS_H_
00017
00018 namespace rxdock {
00019
00020 class Commands {
00021 public:
00022   enum CommName {
00023     ADD = 0,
00024     SUB = 1,
00025     MUL = 2,
00026     DIV = 3,
00027     IF = 4,
00028     LOG = 5,
00029     EXP = 6,
00030     RCTE = 7,
00031     AND = 8,
00032     NULLC = -1
00033   };
00034
00036   // Constructors
00038   Commands(const Commands &c) : name(c.name), nargs(c.nargs) {}
00039   inline Commands(int ncomm) {
00040     name = CommName(ncomm);
00041     switch (name) {
00042     case ADD:
00043     case SUB:
00044     case MUL:
00045     case DIV:
00046     case AND:
00047       nargs = 2;
00048       break;
00049     case IF:
00050       nargs = 3;
00051       break;
00052     case LOG:
00053     case EXP:
00054       nargs = 1;
00055       break;
00056     case RCTE:
00057       nargs = 0;
00058       break;
00059     case NULLC:
00060       nargs = 0;
00061       break;
00062     }
00063   }
00064
00065   inline CommName GetName() const { return name; }
00066
00067   inline int GetNArgs() const { return nargs; }
00068
00069   inline bool IsAdd() const { return (name == ADD); }
00070   inline bool IsSub() const { return (name == SUB); }
00071   inline bool IsMul() const { return (name == MUL); }
00072   inline bool IsDiv() const { return (name == DIV); }
00073   inline bool IsIf() const { return (name == IF); }
00074   inline bool IsLog() const { return (name == LOG); }
00075   inline bool IsExp() const { return (name == EXP); }
00076   inline bool IsRCte() const { return (name == RCTE); }
00077   inline bool IsAnd() const { return (name == AND); }
00078
00080   // Destructor
00082   virtual ~Commands() {}
00083
00084   virtual void copy(const Commands &c) {
00085     name = c.name;
00086     nargs = c.nargs;
00087   }
00088
00090   // Private methods
```

```
00092
00093 private:
00094   Commands(); // Default constructor disabled
00095   CommName name;
00096   int nargs;
00097 };
00098
00099 // Useful typedefs
00100 // typedef SmartPtr<Commands> CommandsPtr;  //Smart pointer
00101 // typedef std::vector<CommandsPtr> CommandsList;//Vector of smart
00102 // pointers typedef CommandsList::iterator CommandsListIter; typedef
00103 // CommandsList::const_iterator CommandsListConstIter;
00104
00105 // Compare class to compare different genomes inside a population
00106 // This is used to sort the genomes in a population depending
00107 // of the value of their scoring function.
00108
00109 } // namespace rxdock
00110
00111 #endif //_Commands_H_
```

6.39 Config.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Common includes for rxdock framework
00014
00015 #ifndef _RBTCONFIG_H_
00016 #define _RBTCONFIG_H_
00017
00018 #ifndef _RBTSMARTPOINTER_H_
00019 #include "rxdock/SmartPointer.h"
00020 #endif //_RBTSMARTPOINTER_H_
00021
00022 #ifndef _RBTDEBUG_H_
00023 #include "rxdock/Debug.h"
00024 #endif //_RBTDEBUG_H_
00025
00026 #ifndef _RBT_H_
00027 #include "rxdock/Rbt.h"
00028 #endif //_RBT_H_
00029
00030 #ifndef _RBTERROR_H_
00031 #include "rxdock/Error.h"
00032 #endif //_RBTERROR_H_
00033
00034 #endif //_RBTCONFIG_H_
```

6.40 Constraint.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base class for pharmacophore constraints
00014
00015 #ifndef _RBT_CONSTRAINT_H_
00016 #define _RBT_CONSTRAINT_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/Coord.h"
```

```
00020 #include "rxdock/Error.h"
00021 #include "rxdock/Model.h"
00022
00023 #include <nlohmann/json.hpp>
00024
00025 using json = nlohmann::json;
00026
00027 namespace rxdock {
00028
00029 class Constraint {
00030 public:
00031   static const std::string _CT;
00033   // Constructors
00035   Constraint(Coord, double);
00036   Constraint(const Constraint &);
00037   Constraint();
00038
00039   Constraint(json j);
00040
00041   friend void to_json(json &j, const Constraint &constraint);
00042   friend void from_json(const json &j, Constraint &constraint);
00043
00045   // Destructor
00047   virtual ~Constraint();
00048
00049   void copy(const Constraint &);
00050   std::ostream &Print(std::ostream &) const;
00051   friend std::ostream &operator«(std::ostream &, const Constraint &);
00052
00053   Coord GetCoords() const { return coord; }
00054   double GetTolerance() const { return tolerance; }
00055   virtual void AddAtomList(ModelPtr, bool bCheck = true) = 0;
00056   double Score() const;
00057
00059   // Private methods
00061 protected:
00062   Coord coord;
00063   double tolerance;
00064   AtomList m_atomList;
00065
00066 private:
00068   // Private data
00070 };
00071
00072 void to_json(json &j, const Constraint &constraint);
00073 void from_json(const json &j, Constraint &constraint);
00074
00075 // Useful typedefs
00076 typedef SmartPtr<Constraint> ConstraintPtr;        // Smart pointer
00077 typedef std::vector<ConstraintPtr> ConstraintList; // Vector of smart pointers
00078 typedef ConstraintList::iterator ConstraintListIter;
00079 typedef ConstraintList::const_iterator ConstraintListConstIter;
00080
00081 ConstraintPtr CreateConstraint(Coord &c, double &t, std::string &n,
00082                                bool bCount = true);
00083 void ZeroCounters();
00084 void ReadConstraint(std::istream &ifile, ConstraintPtr &cnt,
00085                     bool bCount = true);
00086 void ReadConstraintFromMoe(std::istream &ifile, ConstraintPtr &cnt,
00087                            bool bCount = true);
00088 void ReadStartMoe(std::istream &ifile);
00089 void ReadConstraints(std::istream &ii, ConstraintList &cl, bool bCount = true);
00090
00091 // 7 Feb 2005 (DM, Enspiral Discovery)
00092 // New constraint type - any heavy atom
00093 // Likely to be inefficient for large ligands, but has the advantage
00094 // of being simple to code and run
00095 class HeavyConstraint : public Constraint {
00096 public:
00097   HeavyConstraint(Coord c, double t) : Constraint(c, t) {}
00098   void AddAtomList(ModelPtr, bool bCheck = true);
00099   static unsigned int counter;
00100 };
00101
00102 class HBAConstraint : public Constraint {
00103 public:
00104   HBAConstraint(Coord c, double t) : Constraint(c, t) {}
00105   void AddAtomList(ModelPtr, bool bCheck = true);
00106   static unsigned int counter;
00107 };
00108
00109 class HBDConstraint : public Constraint {
00110 public:
00111   HBDConstraint(Coord c, double t) : Constraint(c, t) {}
00112   void AddAtomList(ModelPtr, bool bCheck = true);
00113   static unsigned int counter;
00114 };
```

```
00115
00116 class HydroConstraint : public Constraint {
00117 public:
00118   HydroConstraint(Coord c, double t) : Constraint(c, t) {}
00119   void AddAtomList(ModelPtr, bool bCheck = true);
00120   static unsigned int counter;
00121 };
00122
00123 class HydroAliphaticConstraint : public Constraint {
00124 public:
00125   HydroAliphaticConstraint(Coord c, double t) : Constraint(c, t) {}
00126   void AddAtomList(ModelPtr, bool bCheck = true);
00127   static unsigned int counter;
00128 };
00129
00130 class HydroAromaticConstraint : public Constraint {
00131 public:
00132   HydroAromaticConstraint(Coord c, double t) : Constraint(c, t) {}
00133   void AddAtomList(ModelPtr, bool bCheck = true);
00134   static unsigned int counter;
00135 };
00136
00137 class NegChargeConstraint : public Constraint {
00138 public:
00139   NegChargeConstraint(Coord c, double t) : Constraint(c, t) {}
00140   void AddAtomList(ModelPtr, bool bCheck = true);
00141   static unsigned int counter;
00142 };
00143
00144 class PosChargeConstraint : public Constraint {
00145 public:
00146   PosChargeConstraint(Coord c, double t) : Constraint(c, t) {}
00147   void AddAtomList(ModelPtr, bool bCheck = true);
00148   static unsigned int counter;
00149 };
00150
00151 class RingAromaticConstraint : public Constraint {
00152 public:
00153   RingAromaticConstraint(Coord c, double t) : Constraint(c, t) {}
00154   void AddAtomList(ModelPtr, bool bCheck = true);
00155   static unsigned int counter;
00156 };
00157
00158 } // namespace rxdock
00159
00160 #endif //_Constraint_H_
```

6.41 ConstSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Scoring function to penalise ligand and solvent binding
00014 // Ligand binding RawScore is always 1
00015 // Solvent binding RawScore is:
00016 // SOLVENT_PENALTY * the number of enabled solvent models
00017 #ifndef _RBTCONSTSF_H_
00018 #define _RBTCONSTSF_H_
00019
00020 #include "rxdock/BaseInterSF.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class ConstSF : public BaseInterSF {
00029 public:
00030   // Static data member for class type (i.e. "ConstSF")
00031   static const std::string _CT;
00032   static const std::string _SOLVENT_PENALTY;
00033
00034   ConstSF(const std::string &strName = "const");
```

```
00035   virtual ~ConstSF();
00036
00037   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00038
00039   friend void to_json(json &j, const ConstSF &consf);
00040   friend void from_json(const json &j, ConstSF &consf);
00041
00042 protected:
00043   virtual void SetupReceptor() {}
00044   virtual void SetupLigand() {}
00045   virtual void SetupScore() {}
00046   virtual double RawScore() const;
00047   void ParameterUpdated(const std::string &strName);
00048
00049 private:
00050   // The original constant score for ligand binding
00051   double InterScore() const;
00052   // The solvent binding penalty
00053   double SystemScore() const;
00054   double m_solventPenalty;
00055 };
00056 void to_json(json &j, const ConstSF &consf);
00057 void from_json(const json &j, ConstSF &consf);
00058
00059 } // namespace rxdock
00060
00061 #endif //_RBTCONSTSF_H_
```

6.42 Context.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Stores the information that is global to the interpreter
00014 // in this case, values of ctes, names of variables,...
00015 // It basically contains a map of variables, class Vble is also defined here
00016
00017 #ifndef _RBT_CONTEXT_H_
00018 #define _RBT_CONTEXT_H_
00019
00020 #include "rxdock/BaseSF.h"
00021 #include "rxdock/DockingSite.h"
00022 #include "rxdock/Model.h"
00023 #include "rxdock/Vble.h"
00024 #include "rxdock/geneticprogram/GPTypes.h"
00025 #include <fstream>
00026
00027 namespace rxdock {
00028
00029 typedef std::map<std::string, VblePtr> StringVbleMap; // Map of Vbles
00030 typedef std::map<int, VblePtr> IntVbleMap;            // Map of Vbles
00031 typedef StringVbleMap::iterator StringVbleMapIter;
00032 typedef IntVbleMap::iterator IntVbleMapIter;
00033
00034 class Context {
00035 public:
00036   static const std::string _CT;
00038   // Constructors
00040   Context(const Context &c);
00041   Context(); // default constructor disabled
00043              // Destructors
00045   virtual ~Context();
00046   virtual void Assign(std::string, ReturnType) = 0;
00047   virtual void Assign(int, ReturnType) = 0;
00048   virtual const Vble &GetVble(int) = 0;
00049   virtual const Vble &GetVble(std::string) = 0;
00050   virtual void SetVble(int key, const Vble &v) = 0;
00051   //    virtual String GetName(Int)=0;
00052   //    virtual ReturnType GetValue(Int)=0;
00053   //    virtual String GetName(String)=0;
00054   //    virtual ReturnType GetValue(String)=0;
00055 };
00056
00057 class CellContext : public Context {
```

```
00058 public:
00059   // static const std::string _CT;
00060   CellContext(std::ifstream &ifile);
00061   CellContext();
00062   CellContext(const CellContext &c);
00063   virtual ~CellContext();
00064   void Assign(int key, ReturnType val) {
00065     IntVbleMapIter it = vm.find(key);
00066     if (it != vm.end())
00067       vm[key]->SetValue(val);
00068     else {
00069       vm[key] = new Vble(std::to_string(val), val);
00070     }
00071   }
00072   void Assign(std::string s, ReturnType val) {
00073     throw Error(_WHERE_, "This is not a string context");
00074   }
00075   //    String GetName(Int key) { return vm[key].GetName();}
00076   //    ReturnType GetValue(Int key) {return vm[key].GetValue();}
00077   //    String GetName(String){return "";}
00078   //    ReturnType GetValue(String){return 0.0;}
00079   const Vble &GetVble(int key) { return *(vm[key]); }
00080   void SetVble(int key, const Vble &v) { *(vm[key]) = v; }
00081   const Vble &GetVble(std::string key) {
00082     throw Error(_WHERE_, "This is not a string context");
00083   }
00084   //    void Clear();
00085
00086 private:
00087   IntVbleMap vm;
00088   int ninputs;
00089 };
00090
00091 class StringContext : public Context {
00092
00093 public:
00094   // static const std::string _CT;
00095   StringContext();
00096   StringContext(SmartPtr<std::ifstream> ifile);
00097   StringContext(const StringContext &c);
00098   virtual ~StringContext();
00099   void Assign(std::string key, ReturnType val) {
00100     StringVbleMapIter it = vm.find(key);
00101     if (it != vm.end())
00102       vm[key]->SetValue(val);
00103     else {
00104       vm[key] = new Vble(key, val);
00105     }
00106   }
00107   void Assign(int i, ReturnType val) {
00108     throw Error(_WHERE_, "This is not a cell context");
00109   }
00110
00111   //    String GetName(Int){return "";}
00112   //    ReturnType GetValue(Int){return 0.0;}
00113   //    String GetName(String key) { return vm[key].GetName();}
00114   //    ReturnType GetValue(String key) {return vm[key].GetValue();}
00115   double Get(ModelPtr lig, std::string name);
00116   double Get(ModelPtr rec, DockingSitePtr site, std::string name);
00117   double Get(BaseSF *spSF, std::string name, ModelPtr lig);
00118   const Vble &GetVble(std::string key) { return *(vm[key]); }
00119   const Vble &GetVble(int key) {
00120     throw Error(_WHERE_, "This is not a cell context");
00121   }
00122   void SetVble(int key, const Vble &v) { *(vm[""]) = v; }
00123   void UpdateLigs(ModelPtr lig);
00124   void UpdateSite(ModelPtr rec, DockingSitePtr site);
00125   void UpdateScores(BaseSF *spSF, ModelPtr lig);
00126
00127 private:
00128   StringVbleMap vm;
00129 };
00130
00131 // Useful typedefs
00132 typedef SmartPtr<CellContext> CellContextPtr;        // Smart pointer
00133 typedef std::vector<CellContextPtr> CellContextList; // Vector of smart
00134                                                      // pointers
00135 typedef CellContextList::iterator CellContextListIter;
00136 typedef CellContextList::const_iterator CellContextListConstIter;
00137 typedef SmartPtr<StringContext> StringContextPtr; // Smart pointer
00138 typedef std::vector<StringContextPtr>
00139     StringContextList; // Vector of smart pointers
00140 typedef StringContextList::iterator StringContextListIter;
00141 typedef StringContextList::const_iterator StringContextListConstIter;
00142 typedef SmartPtr<Context> ContextPtr;        // Smart pointer
00143 typedef std::vector<ContextPtr> ContextList; // Vector of smart pointers
00144 typedef ContextList::iterator ContextListIter;
```

```
00145 typedef ContextList::const_iterator ContextListConstIter;
00146
00147 } // namespace rxdock
00148
00149 #endif //_Context_H_
```

6.43 Coord.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple class for handling cartesian coords (points) and 3-d vectors
00014 // The same class is used to represent both points and vectors, although the
00015 // Vector typedef can be used in argument lists to distinguish the two cases
00016
00017 #ifndef _RBTCOORD_H_
00018 #define _RBTCOORD_H_
00019
00020 #include <algorithm> //for min,max
00021 #include <cmath>     //for sqrt
00022 #include <iterator>  //for back_inserter
00023 #include <numeric>   //for accumulate
00024
00025 #ifdef __PGI
00026 #define EIGEN_DONT_VECTORIZE
00027 #endif
00028 #include <Eigen/Core>
00029 #include <Eigen/Geometry>
00030
00031 #include <nlohmann/json.hpp>
00032
00033 using json = nlohmann::json;
00034
00035 #ifndef M_PI
00036 #define M_PI 3.14159265358979323846
00037 #endif
00038
00039 #include "rxdock/Config.h"
00040
00041 extern std::istream &eatSeps(std::istream &is);
00042
00043 namespace rxdock {
00044
00045 class Coord {
00047   // Data members
00049   //(leave public so we don't need
00050   // to write accessor functions)
00051 public:
00052   Eigen::Vector3d xyz;
00053
00055   // Constructors/destructors:
00057 public:
00058   // Default constructor (initialise to zero)
00059   inline Coord() : xyz(0.0, 0.0, 0.0) {}
00060
00061   // Constructor with initial values
00062   inline Coord(double x1, double y1, double z1) : xyz(x1, y1, z1) {}
00063
00064   // Constructor using Eigen
00065   inline Coord(Eigen::Vector3d xyz1) : xyz(xyz1) {}
00066
00067   // Destructor
00068   virtual ~Coord() {}
00069
00070   // Copy constructor
00071   inline Coord(const Coord &coord) { xyz = coord.xyz; }
00072
00074   // Operator functions:
00076
00077   // Copy assignment (*this = coord)
00078   inline Coord &operator=(const Coord &coord) {
00079     if (this != &coord) { // beware of self-assignment
00080       xyz = coord.xyz;
00081     }
```

```
00082     return *this;
00083   }
00084
00085   // Copy assignment (*this = const)
00086   inline Coord &operator=(const double &d) {
00087     xyz.setConstant(d);
00088     return *this;
00089   }
00090
00091   //*this += coord
00092   inline void operator+=(const Coord &coord) { xyz += coord.xyz; }
00093
00094   //*this += const
00095   inline void operator+=(const double &d) { xyz += xyz.Constant(d); }
00096
00097   //*this -= coord
00098   inline void operator-=(const Coord &coord) { xyz -= coord.xyz; }
00099
00100  //*this -= const
00101  inline void operator-=(const double &d) { xyz -= xyz.Constant(d); }
00102
00103  //*this *= const
00104  inline void operator*=(const double &d) { xyz *= d; }
00105
00106  //*this /= const
00107  inline void operator/=(const double &d) { xyz /= d; }
00108
00110  // Friend functions:
00112
00113  // Insertion (output) operator
00114  friend std::ostream &operator«(std::ostream &s, const Coord &coord) {
00115    return s « "(" « coord.xyz(0) « "," « coord.xyz(1) « ","
00116             « coord.xyz(2) « ")";
00117  }
00118
00119  // Input operator
00120  // DM 15 Apr 1999
00121  // Valid formats (actually the comma can be any single non-whitespace
00122  // character):
00123  // x,y,z
00124  // (x,y,z)
00125  // BGD 06 Nov 2002
00126  // Trying to make it more versatil. The separator now is any
00127  // amount of white space or commas.
00128  // x   y,z
00129  // (x   y, z  )
00130  // uses the auxiliary function eatSeps
00131
00132  friend std::istream &operator»(std::istream &s, Coord &coord) {
00133    double x = 0, y = 0, z = 0;
00134    char c = 0;
00135    s » c;
00136    if (c == '(') {
00137      s » x;
00138      eatSeps(s);
00139      s » y;
00140      eatSeps(s);
00141      s » z;
00142      eatSeps(s);
00143      s » c;
00144      if (c != ')')
00145        s.clear(std::ios_base::badbit); // Missing closing bracket
00146    } else {
00147      s.putback(c);
00148      s » x;
00149      eatSeps(s);
00150      s » y;
00151      eatSeps(s);
00152      s » z;
00153    }
00154    if (s)
00155      coord = Coord(x, y, z);
00156    return s;
00157  }
00158
00159  // Equality
00160  inline friend bool operator==(const Coord &coord1, const Coord &coord2) {
00161    return coord1.xyz == coord2.xyz;
00162  }
00163
00164  // Non-equality
00165  inline friend bool operator!=(const Coord &coord1, const Coord &coord2) {
00166    return coord1.xyz != coord2.xyz;
00167  }
00168
00169  // Greater than
00170  inline friend bool operator>(const Coord &coord1, const Coord &coord2) {
```

```
00171     return (coord1.xyz.array() > coord2.xyz.array()).all();
00172   }
00173
00174   // Greater than or equal
00175   inline friend bool operator>=(const Coord &coord1, const Coord &coord2) {
00176     return (coord1.xyz.array() >= coord2.xyz.array()).all();
00177   }
00178
00179   // Less than
00180   inline friend bool operator<(const Coord &coord1, const Coord &coord2) {
00181     return (coord1.xyz.array() < coord2.xyz.array()).all();
00182   }
00183
00184   // Less than or equal
00185   inline friend bool operator<=(const Coord &coord1, const Coord &coord2) {
00186     return (coord1.xyz.array() <= coord2.xyz.array()).all();
00187   }
00188
00189   // Addition (coord + coord)
00190   inline friend Coord operator+(const Coord &coord1, const Coord &coord2) {
00191     return Coord(coord1.xyz + coord2.xyz);
00192   }
00193
00194   // Addition (coord + const)
00195   inline friend Coord operator+(const Coord &coord1, double d) {
00196     return Coord(coord1.xyz + coord1.xyz.Constant(d));
00197   }
00198
00199   // Addition (const + coord)
00200   inline friend Coord operator+(double d, const Coord &coord1) {
00201     return Coord(coord1.xyz.Constant(d) + coord1.xyz);
00202   }
00203
00204   // Subtraction (coord - coord)
00205   inline friend Coord operator-(const Coord &coord1, const Coord &coord2) {
00206     return Coord(coord1.xyz - coord2.xyz);
00207   }
00208
00209   // Subtraction (coord - const)
00210   inline friend Coord operator-(const Coord &coord1, double d) {
00211     return Coord(coord1.xyz - coord1.xyz.Constant(d));
00212   }
00213
00214   // Subtraction (const - coord)
00215   inline friend Coord operator-(double d, const Coord &coord1) {
00216     return Coord(coord1.xyz.Constant(d) - coord1.xyz);
00217   }
00218
00219   // Negation
00220   inline friend Coord operator-(const Coord &coord) {
00221     return Coord(-coord.xyz);
00222   }
00223
00224   // Dot product
00225   // DM 7 Jan 1999 - operator* replaced by Dot(coord) member function
00226   // and by Dot(coord1,coord2) non-member function
00227   // inline friend Double operator*(const Coord& coord1, const Coord&
00228   // coord2) {
00229   //  return coord1.xyz.dot(coord.xyz);
00230   //}
00231
00232   // Scalar product (coord * const)
00233   inline friend Coord operator*(const Coord &coord, const double &d) {
00234     return Coord(coord.xyz * d);
00235   }
00236
00237   // Scalar product (const * coord)
00238   inline friend Coord operator*(const double &d, const Coord &coord) {
00239     return Coord(d * coord.xyz);
00240   }
00241
00242   // Scalar product (coord * coord : component-wise multiplication)
00243   inline friend Coord operator*(const Coord &coord1, const Coord &coord2) {
00244     return Coord(coord1.xyz.array() * coord2.xyz.array());
00245   }
00246
00247   // Scalar division (coord / const)
00248   inline friend Coord operator/(const Coord &coord, const double &d) {
00249     return Coord(coord.xyz / d);
00250   }
00251
00252   friend void to_json(json &j, const Coord &c) {
00253     j = json{c.xyz(0), c.xyz(1), c.xyz(2)};
00254   }
00255
00256   friend void from_json(const json &j, Coord &c) {
00257     j.at(0).get_to(c.xyz(0));
```

```
00258     j.at(1).get_to(c.xyz(1));
00259     j.at(2).get_to(c.xyz(2));
00260   }
00261
00263   // Public methods
00265
00266   // Mostly applicable to vectors (rather than coords)
00267
00268   // Returns square of magnitude of vector (or distance from origin)
00269   inline double Length2() const { return xyz.squaredNorm(); }
00270
00271   // Returns magnitude of vector (or distance from origin)
00272   inline double Length() const { return xyz.norm(); }
00273
00274   // Returns unit vector in same direction
00275   // Member function (V2 = V1.Unit())
00276   // Checks for zero length
00277   inline Coord Unit() const {
00278     double l = Length();
00279     return (l > 0) ? *this / l : *this;
00280   }
00281
00282   // Cross product member function (V3 = V1.Cross(V2))
00283   inline Coord Cross(const Coord &v2) const { return Coord(xyz.cross(v2.xyz)); }
00284
00285   // Dot product member function (D = V1.Dot(V2)
00286   inline double Dot(const Coord &v2) const { return xyz.dot(v2.xyz); }
00287 };
00288
00289 // Useful typedefs
00290
00291 // DM 4 Mar 1999 - to make it easier to distinguish arguments that are
00292 // treated as -3D vectors from those are treated as 3-D points, use this typedef
00293 typedef Coord Vector;
00294
00295 typedef std::vector<Coord> CoordList; // Vector of coords
00296 typedef CoordList::iterator CoordListIter;
00297 typedef CoordList::const_iterator CoordListConstIter;
00298
00299 // DM 19/1/99 typedefs for a map of key=Int, value=Coord
00300 // typedef std::map<String,Coord> StringCoordMap;
00301 // typedef StringCoordMap::iterator StringCoordMapIter;
00302 // typedef StringCoordMap::const_iterator StringCoordMapConstIter;
00303 // DM 08 Feb 1999 - Changed from key=string to key=int
00304 typedef std::map<unsigned int, Coord> UIntCoordMap;
00305 typedef UIntCoordMap::iterator UIntCoordMapIter;
00306 typedef UIntCoordMap::const_iterator UIntCoordMapConstIter;
00307
00309 // Non-member functions (in rxdock namespace)
00311
00313 // VECTOR FUNCTIONS
00315
00316 // Returns square of magnitude of vector
00317 inline double Length2(const Vector &v1) { return v1.Length2(); }
00318
00319 // Returns square of distance between two coords
00320 inline double Length2(const Coord &c1, const Coord &c2) {
00321   return Length2(c2 - c1);
00322 }
00323
00324 // Returns magnitude of vector
00325 inline double Length(const Vector &v1) { return v1.Length(); }
00326
00327 // Returns distance between two coords
00328 inline double Length(const Coord &c1, const Coord &c2) {
00329   return Length(c2 - c1);
00330 }
00331
00332 // Returns unit vector in same direction
00333 // V2 = Unit(V1)
00334 inline Vector Unit(const Vector &v1) { return v1.Unit(); }
00335
00336 // Cross product (V3 = Cross(V1,V2))
00337 inline Vector Cross(const Vector &v11, const Vector &v2) {
00338   return v11.Cross(v2);
00339 }
00340
00341 // Dot product (D = Dot(V1,V2))
00342 inline double Dot(const Vector &v1, const Vector &v2) { return v1.Dot(v2); }
00343
00344 // Returns angle formed between 2 vectors
00345 inline double Angle(const Vector &v1, const Vector &v2) {
00346   // Calculate the product of the lengths
00347   double d1d2 = v1.Length() * v2.Length();
00348
00349   // Calculate the sin and cos
00350   double cos_theta = v1.Dot(v2) / d1d2;
```

```
00351   double sin_theta = (v1.Cross(v2)).Length() / d1d2;
00352
00353   // Get theta and convert to degrees
00354   double theta = std::atan2(sin_theta, cos_theta);
00355   return theta * 180.0 / M_PI;
00356 }
00357
00358 // Returns angle formed between 3 coords coord1..coord2..coord3
00359 inline double Angle(const Coord &coord1, const Coord &coord2,
00360                     const Coord &coord3) {
00361   // Determine vectors between coords and call vector version of Angle()
00362   return Angle(coord1 - coord2, coord3 - coord2);
00363 }
00364
00365 // DM 7 June 1999
00366 // Returns dihedral formed between 3 vectors
00367 inline double Dihedral(const Vector &v1, const Vector &v2, const Vector &v3) {
00368   // Calculate the cross products
00369   Vector A = v1.Cross(v2);
00370   Vector B = v2.Cross(v3);
00371   Vector C = v2.Cross(A);
00372   // Calculate the distances
00373   double rA = A.Length();
00374   double rB = B.Length();
00375   double rC = C.Length();
00376   // Calculate the sin and cos
00377   double cos_phi = A.Dot(B) / (rA * rB);
00378   double sin_phi = C.Dot(B) / (rC * rB);
00379   // Get phi and convert to degrees
00380   double phi = -std::atan2(sin_phi, cos_phi);
00381   return phi * 180.0 / M_PI;
00382 }
00383
00384 // Returns dihedral formed between 4 coords c1..c2..c3..c4
00385 inline double Dihedral(const Coord &c1, const Coord &c2, const Coord &c3,
00386                        const Coord &c4) {
00387   // Determine vectors between coords and call vector version of Dihedral()
00388   return Dihedral(c1 - c2, c2 - c3, c3 - c4);
00389 }
00390
00392 // COORDINATE FUNCTIONS
00394
00395 // DM 28 Jul 1999 - functions for use by STL transform
00396 inline double ExtractXCoord(const Coord &c) { return c.xyz(0); }
00397 inline double ExtractYCoord(const Coord &c) { return c.xyz(1); }
00398 inline double ExtractZCoord(const Coord &c) { return c.xyz(2); }
00399
00400 // Returns minimum of two coords (component-wise minimum)
00401 inline Coord Min(const Coord &coord1, const Coord &coord2) {
00402   return Coord(coord1.xyz.array().min(coord2.xyz.array()));
00403 }
00404
00405 // Returns maximum of two coords (component-wise maximum)
00406 inline Coord Max(const Coord &coord1, const Coord &coord2) {
00407   return Coord(coord1.xyz.array().max(coord2.xyz.array()));
00408 }
00409
00410 // DM 28 Jul 1999 - returns component-wise min coord for coord list (i.e.min x,
00411 // min y, min z)
00412 inline Coord Min(const CoordList &cl) {
00413   std::vector<double> cx, cy, cz;
00414   // Extract arrays of x,y,z components separately
00415   std::transform(cl.begin(), cl.end(), std::back_inserter(cx), ExtractXCoord);
00416   std::transform(cl.begin(), cl.end(), std::back_inserter(cy), ExtractYCoord);
00417   std::transform(cl.begin(), cl.end(), std::back_inserter(cz), ExtractZCoord);
00418   // Compose the min coord from the min of each of the components
00419   return Coord(*(std::min_element(cx.begin(), cx.end())),
00420                *(std::min_element(cy.begin(), cy.end())),
00421                *(std::min_element(cz.begin(), cz.end())));
00422 }
00423
00424 // DM 28 Jul 1999 - returns component-wise max coord for coord list (i.e. max x,
00425 // max y, max z)
00426 inline Coord Max(const CoordList &cl) {
00427   std::vector<double> cx, cy, cz;
00428   // Extract arrays of x,y,z components separately
00429   std::transform(cl.begin(), cl.end(), std::back_inserter(cx), ExtractXCoord);
00430   std::transform(cl.begin(), cl.end(), std::back_inserter(cy), ExtractYCoord);
00431   std::transform(cl.begin(), cl.end(), std::back_inserter(cz), ExtractZCoord);
00432   // Compose the max coord from the max of each of the components
00433   return Coord(*(std::max_element(cx.begin(), cx.end())),
00434                *(std::max_element(cy.begin(), cy.end())),
00435                *(std::max_element(cz.begin(), cz.end())));
00436 }
00437
00438 // Returns center of "mass" of coords in coord list
00439 inline Coord GetCenterOfAtomicMass(const CoordList &coordList) {
```

```
00440   Coord com;
00441   com = std::accumulate(coordList.begin(), coordList.end(), com);
00442   // Check for divide by zero
00443   return coordList.empty() ? com : com / static_cast<double>(coordList.size());
00444 }
00445
00446 } // namespace rxdock
00447
00448 #endif //_RBTCOORD_H_
```

6.44 CrdFileSink.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Class for writing Models to CRD files
00014
00015 #ifndef _RBTCRDFILESINK_H_
00016 #define _RBTCRDFILESINK_H_
00017
00018 #include "rxdock/BaseMolecularFileSink.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 class CrdFileSink : public BaseMolecularFileSink {
00027 public:
00029   // Constructors/destructors
00030   // CrdFileSink(const char* fileName, ModelPtr spModel);
00031   RBTDLL_EXPORT CrdFileSink(const std::string &fileName, ModelPtr spModel);
00032
00033   virtual ~CrdFileSink(); // Default destructor
00034
00035   friend void to_json(json &j, const CrdFileSink &crdFileSink);
00036   friend void from_json(const json &j, CrdFileSink &crdFileSink);
00037
00039   // Public methods
00041   //
00042   // Derived classes should override
00043   // Controls whether file type can support the writing of multiple
00044   // conformations/models to a single file
00045   virtual bool isMultiConfSupported() { return true; }
00046   //
00048   // Override public methods from BaseFileSink
00049   virtual void Render();
00050
00051 protected:
00053   // Protected methods
00055
00056 private:
00058   // Private methods
00060
00061   CrdFileSink();                    // Disable default constructor
00062   CrdFileSink(const CrdFileSink &); // Copy constructor disabled by
00063                                     // default
00064   CrdFileSink &
00065   operator=(const CrdFileSink &); // Copy assignment disabled by default
00066
00067 protected:
00069   // Protected data
00071
00072 private:
00074   // Private data
00076   unsigned int m_numAtomsLineRec;
00077 };
00078
00079 void to_json(json &j, const CrdFileSink &crdFileSink);
00080 void from_json(const json &j, CrdFileSink &crdFileSink);
00081
00082 // Useful typedefs
00083 typedef SmartPtr<CrdFileSink> CrdFileSinkPtr; // Smart pointer
```

```
00084
00085 } // namespace rxdock
00086
00087 #endif //_RBTCRDFILESINK_H_
```

6.45 CrdFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Handles retrieval of atomic (coordinate) info from Charmm CRD files
00014
00015 #ifndef _RBTCRDFILESOURCE_H_
00016 #define _RBTCRDFILESOURCE_H_
00017
00018 #include "rxdock/BaseMolecularFileSource.h"
00019
00020 namespace rxdock {
00021
00022 class CrdFileSource : public BaseMolecularFileSource {
00023 public:
00024   // Constructors
00025   CrdFileSource(const char *fileName);
00026   CrdFileSource(const std::string fileName);
00027
00028   // Default destructor
00029   virtual ~CrdFileSource();
00030
00032   // Override public methods from BaseMolecularDataSource
00033   virtual bool isTitleListSupported() { return true; }
00034   virtual bool isAtomListSupported() { return true; }
00035   virtual bool isCoordinatesSupported() { return true; }
00036   virtual bool isBondListSupported() { return false; }
00037
00038 protected:
00039   // Pure virtual in BaseFileSource - needs to be defined here
00040   virtual void Parse();
00041
00042 private:
00043   // Private methods
00044   CrdFileSource();                      // Disable default constructor
00045   CrdFileSource(const CrdFileSource &); // Copy constructor disabled by default
00046   CrdFileSource &
00047   operator=(const CrdFileSource &); // Copy assignment disabled by default
00048 };
00049
00050 // useful typedefs
00051 typedef SmartPtr<CrdFileSource> CrdFileSourcePtr; // Smart pointer
00052
00053 } // namespace rxdock
00054
00055 #endif //_RBTCRDFILESOURCE_H_
```

6.46 CSVFileSink.h File Reference

```
#include ¨rxdock/BaseMolecularFileSink.h¨
#include ¨rxdock/ElementFileSource.h¨
```

Classes

- class rxdock::CSVFileSink

Typedefs

- typedef SmartPtr< CSVFileSink > **rxdock::CSVFileSinkPtr**

6.46.1 Detailed Description

Outputs Model(s) to a CSV file.

6.47 CSVFileSink.h

Go to the documentation of this file.

```
00001 //===-- CSVFileSink.h - Model output to a CSV file --------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef _RBTCSVFILESINK_H_
00019 #define _RBTCSVFILESINK_H_
00020
00021 #include "rxdock/BaseMolecularFileSink.h"
00022 #include "rxdock/ElementFileSource.h"
00023
00024 namespace rxdock {
00025
00026 class CSVFileSink : public BaseMolecularFileSink {
00027 public:
00028   RBTDLL_EXPORT CSVFileSink(const std::string &fileName, ModelPtr spModel,
00029                             std::size_t nAtoms, std::size_t nDataFields);
00030
00031   virtual ~CSVFileSink();
00032
00033   virtual void Render();
00034
00035 private:
00036   std::string RenderAtomList(const AtomList &atomList);
00037   std::string RenderData(const StringVariantMap &dataMap);
00038
00039   CSVFileSink();                    // Disable default constructor
00040   CSVFileSink(const CSVFileSink &); // Copy constructor disabled by
00041                                     // default
00042   CSVFileSink &
00043   operator=(const CSVFileSink &); // Copy assignment disabled by default
00044
00045   std::size_t m_nAtoms;
00046   std::size_t m_nDataFields;
00047   ElementFileSourcePtr m_spElementData;
00048 };
00049
00050 // Useful typedefs
00051 typedef SmartPtr<CSVFileSink> CSVFileSinkPtr; // Smart pointer
00052
00053 } // namespace rxdock
00054
00055 #endif //_RBTCSVFILESINK_H_
```

6.48 Smarts.h

```
00001 /*****************************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
```

```
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // SMARTS class. Defines a function that takes a model and a smarts string as
00014 // input, and returns a list of lists of atoms that match the pattern
00015
00016 #ifndef _RBT_SMARTS_H_
00017 #define _RBT_SMARTS_H_
00018
00019 #include "rxdock/Model.h"
00020
00021 namespace rxdock {
00022
00023 namespace daylight {
00024 AtomListList QueryModel(ModelPtr spModel, const std::string &strSmart,
00025                         std::string &strSmiles);
00026 }
00027
00028 // Map of (key=integer, value=integer)
00029 typedef std::map<int, int> IntIntMap;
00030 typedef IntIntMap::iterator IntIntMapIter;
00031 typedef IntIntMap::const_iterator IntIntMapConstIter;
00032
00033 } // namespace rxdock
00034
00035 #endif
```

6.49 Debug.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Tools for tracking of object construction and destruction for
00014 // debugging purposes
00015
00016 #ifndef _RBTDEBUG_H_
00017 #define _RBTDEBUG_H_
00018
00019 #ifdef _RBTOBJECTCOUNTER
00020
00021 #include <map>
00022
00023 namespace rxdock {
00024
00025 // Object counter class for debugging no. of object constructions and
00026 // destructions
00027 class ObjectCounter {
00028 public:
00029   ObjectCounter();
00030   ~ObjectCounter();
00031   friend std::ostream &operator«(std::ostream &s,
00032                                  const ObjectCounter &counter);
00033   Int nConstr;
00034   Int nCopyConstr;
00035   Int nDestr;
00036 };
00037
00038 // Useful typedefs
00039 typedef std::map<String, ObjectCounter> ObjectCounterMap;
00040 typedef ObjectCounterMap::iterator ObjectCounterMapIter;
00041
00042 // External declaration of the one and only object counter map
00043 extern ObjectCounterMap theObjectCounterMap;
00044
00045 } // namespace rxdock
00046
00047 #define _RBTOBJECTCOUNTER_CONSTR_(class) theObjectCounterMap[(class)].nConstr++
00048 #define _RBTOBJECTCOUNTER_COPYCONSTR_(class)                                \
00049   theObjectCounterMap[(class)].nCopyConstr++
00050 #define _RBTOBJECTCOUNTER_DESTR_(class) theObjectCounterMap[(class)].nDestr++
00051
```

```
00052 #define _RBTOBJECTCOUNTER_DUMP_(stream)                                          \
00053   (stream) « std::endl « "RBT OBJECT COUNTERS:" « std::endl « std::endl;   \
00054   for (ObjectCounterMapIter dumpIter = theObjectCounterMap.begin();          \
00055        dumpIter != theObjectCounterMap.end(); dumpIter++) {                  \
00056     (stream) « (*dumpIter).first « ": " « (*dumpIter).second « std::endl;  \
00057   }
00058
00059 #else //_RBTOBJECTCOUNTER
00060
00061 #define _RBTOBJECTCOUNTER_CONSTR_(class)
00062 #define _RBTOBJECTCOUNTER_COPYCONSTR_(class)
00063 #define _RBTOBJECTCOUNTER_DESTR_(class)
00064 #define _RBTOBJECTCOUNTER_DUMP_(stream)
00065
00066 #endif //_RBTOBJECTCOUNTER
00067
00068 #endif //_RBTDEBUG_H_
```

6.50 DihedralIntraSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Ligand intramolecular dihedral scoring function
00014 #ifndef _RBTDIHEDRALINTRASF_H_
00015 #define _RBTDIHEDRALINTRASF_H_
00016
00017 #include "rxdock/BaseIntraSF.h"
00018 #include "rxdock/DihedralSF.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 class DihedralIntraSF : public BaseIntraSF, public DihedralSF {
00027 public:
00028   // Class type string
00029   static const std::string _CT;
00030   // Parameter names
00031
00032   DihedralIntraSF(const std::string &strName = "dihedral");
00033   virtual ~DihedralIntraSF();
00034
00035   friend void to_json(json &j, const DihedralIntraSF &dihedralIntraSF);
00036   friend void from_json(const json &j, DihedralIntraSF &dihedralIntraSF);
00037
00038 protected:
00039   virtual void SetupScore();
00040   virtual double RawScore() const;
00041
00042   // Clear the dihedral list
00043   // As we are not using smart pointers, there is some memory management to do
00044   void ClearModel();
00045
00046 private:
00047   DihedralList m_dihList;
00048 };
00049
00050 void to_json(json &j, const DihedralIntraSF &dihedralIntraSF);
00051 void from_json(const json &j, DihedralIntraSF &dihedralIntraSF);
00052
00053 } // namespace rxdock
00054
00055 #endif //_RBTDIHEDRALINTRASF_H_
```

6.51 DihedralSF.h

```
00001 /**********************************************************************
```

```
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Base implementation class for all dihedral scoring functions. Provides
00014 // Dihedral struct for dihedral definitions and score, plus a method for
00015 // generating lists of dihedral definitions from a bond list.
00016 // This version hardwired to use Tripos 5.2 atom types and params.
00017
00018 #ifndef _RBTDIHEDRALSF_H_
00019 #define _RBTDIHEDRALSF_H_
00020
00021 #include "rxdock/Atom.h"
00022 #include "rxdock/BaseSF.h"
00023 #include "rxdock/Bond.h"
00024 #include "rxdock/ParameterFileSource.h"
00025 #include "rxdock/TriposAtomType.h"
00026
00027 #include <nlohmann/json.hpp>
00028
00029 using json = nlohmann::json;
00030
00031 namespace rxdock {
00032
00033 // Class for holding dihedral atom specifiers and force field params
00034 // Main method is operator() - calculates single dihedral score for current
00035 // conformation DM (9 Dec 2003): Extended to allow additional terms to be added
00036 // to the dihedral potential (AddTerm) Two main uses for AddTerm:
00037 //  i) to define a Fourier series expansion, rather than a simple cosine
00038 //  potential
00039 // ii) implicit hydrogen correction (ghost terms are added for the missing
00040 // hydrogen dihedrals
00041 //                                  defined as offsets from the real heavy-atom
00042 //                                  dihedral specifiers)
00043
00044 class DihedralElement {
00045 public:
00046   // Simple struct for storing the force field params for a single term of the
00047   // dihedral potential
00048   class prms {
00049   public:
00050     prms(double ss = 3.0, double kk = 0.2, double o = 0.0)
00051         : s(std::fabs(ss)), k(kk), sign((ss > 0.0) ? +1.0 : -1.0), offset(o) {}
00052     double s;      // rotational degeneracy
00053     double k;      // barrier height
00054     double sign;   //+1 or -1, to invert potential
00055     double offset; // dihedral offset in degrees (useful for ghost implicit
00056                    // hydrogens)
00057   };
00058
00059   // Constructor takes the real atom specifiers, plus the first term of the
00060   // potential
00061   DihedralElement(Atom *pAtom1, Atom *pAtom2, Atom *pAtom3, Atom *pAtom4,
00062                   const prms &dihprms);
00063
00064   friend void to_json(json &j, const DihedralElement &dihedralElem);
00065   friend void from_json(const json &j, DihedralElement &dihedralElem);
00066
00067   double operator()() const; // Calculate dihedral score for this interaction
00068   Atom *GetAtom1Ptr() const { return m_pAtom1; }
00069   Atom *GetAtom2Ptr() const { return m_pAtom2; }
00070   Atom *GetAtom3Ptr() const { return m_pAtom3; }
00071   Atom *GetAtom4Ptr() const { return m_pAtom4; }
00072   // Add additional terms to the potential
00073   void AddTerm(const prms &dihprms);
00074
00075 private:
00076   DihedralElement(); // Forbid default constructor
00077   Atom *m_pAtom1;
00078   Atom *m_pAtom2;
00079   Atom *m_pAtom3;
00080   Atom *m_pAtom4;
00081   std::vector<prms> m_prms;
00082 };
00083
00084 void to_json(json &j, const DihedralElement &dihedralElem);
00085 void from_json(const json &j, DihedralElement &dihedralElem);
00086
00087 // Useful typedefs
00088 typedef std::vector<DihedralElement *> DihedralList; // Vector of smart pointers
```

```
00089 typedef DihedralList::iterator DihedralListIter;
00090 typedef DihedralList::const_iterator DihedralListConstIter;
00091
00092 class DihedralSF : public virtual BaseSF {
00093 public:
00094   // Class type string
00095   static const std::string _CT;
00096   // Parameter names
00097   static const std::string _IMPL_H_CORR; // DM 6 Aug 2002 - option to correct
00098                                          // barrier heights for implicit H's
00099
00100   virtual ~DihedralSF();
00101
00102   friend void to_json(json &j, const DihedralSF &dihedralSF);
00103   friend void from_json(const json &j, DihedralSF &dihedralSF);
00104
00105 protected:
00106   DihedralSF();
00107   // Creates a vector of pointers to dihedral objects from a supplied bond list
00108   // Dihedral params are set using Tripos 5.2 atom types + params
00109   // NOTE: It is the responsibility of the subclass to delete the dihedral
00110   // objects which are created
00111   DihedralList CreateDihedralList(const BondList &bondList);
00112
00113 private:
00114   // Lookup the dihedral params for a given set of types
00115   DihedralElement::prms FindDihedralParams(TriposAtomType::eType t1,
00116                                            TriposAtomType::eType t2,
00117                                            TriposAtomType::eType t3,
00118                                            TriposAtomType::eType t4);
00119
00120   // Determines the list of bonded atoms for pAtom1, and the list of dihedral
00121   // offsets for any implicit hydrogens pAtom2 is the other central atom in the
00122   // bond, is excluded from the returned list of bonded atoms
00123   void CalcBondedAtoms(Atom *pAtom1, Atom *pAtom2, AtomList &bondedAtoms,
00124                        std::vector<double> &offsets);
00125
00126   ParameterFileSourcePtr m_spDihedralSource;
00127   std::vector<std::string> m_centralPairs;
00128   TriposAtomType m_triposType;
00129 };
00130
00131 void to_json(json &j, const DihedralSF &dihedralSF);
00132 void from_json(const json &j, DihedralSF &dihedralSF);
00133
00134 } // namespace rxdock
00135
00136 #endif //_RBTDIHEDRALSF_H_
```

6.52 DihedralTargetSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Target intramolecular dihedral scoring function
00014 // for flexible receptors
00015 #ifndef _RBTDIHEDRALTARGETSF_H_
00016 #define _RBTDIHEDRALTARGETSF_H_
00017
00018 #include "rxdock/BaseInterSF.h"
00019 #include "rxdock/DihedralSF.h"
00020
00021 namespace rxdock {
00022
00023 class DihedralTargetSF : public BaseInterSF, public DihedralSF {
00024 public:
00025   // Class type string
00026   static const std::string _CT;
00027   // Parameter names
00028
00029   DihedralTargetSF(const std::string &strName = "dihedral");
00030   virtual ~DihedralTargetSF();
00031
00032 protected:
```

```
00033   virtual void SetupReceptor();
00034   virtual void SetupLigand();
00035   virtual void SetupScore();
00036   virtual double RawScore() const;
00037
00038   // Clear the dihedral list
00039   // As we are not using smart pointers, there is some memory management to do
00040   void ClearReceptor();
00041
00042 private:
00043   DihedralList m_dihList;
00044 };
00045
00046 } // namespace rxdock
00047
00048 #endif //_RBTDIHEDRALTARGETSF_H_
```

6.53 DirectorySource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTDIRECTORYSOURCE_H_
00014 #define _RBTDIRECTORYSOURCE_H_
00015
00016 #include "rxdock/Config.h"
00017 #include "rxdock/FileError.h" // for exception types
00018 #include <dirent.h>           // for scandir()
00019 #include <fstream>            // ditto
00020 #include <sys/stat.h>         // for stat()
00021 #include <sys/types.h>
00022
00023 namespace rxdock {
00024
00025 const unsigned int PATH_SIZE = 1024; // max path width
00026 const unsigned int LINE_WIDTH = 256; // max num of chars in a line
00027 const unsigned int ALL = 0;          // read all files in directory
00028
00035 class DirectorySource {
00036 protected:
00037   struct stat fStat; // stat struct for file access check
00038 #ifdef COMMENT
00039 #ifdef linux
00040   struct dirent **fNameList; // file name list in directory
00041 #endif
00042 #ifdef sun                   // for gcc
00043   struct direct **fNameList; // file name list in directory
00044 #endif
00045 #endif                                // COMMENT
00046   int fNum;                           // number of files in the dir
00047   struct dirent **fNameList;          // file name list in directory
00048   std::ifstream inFile;               // an actual file to read
00049   std::string thePath;                // leading path
00050   void CheckDirectory(std::string &); // to check directory access
00051
00052 public:
00053   DirectorySource(const std::string &);
00054   virtual ~DirectorySource();
00055
00056   static const std::string _CT;
00057
00064   // virtual void ReadFiles()=0;
00065   // virtual void ParseLines()=0;
00066 };
00067
00068 } // namespace rxdock
00069
00070 #endif // _RBTDIRECTORYSOURCE_H_
```

6.54 DockingError.h

```
00001 /**********************************************************************
```

```
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  *********************************************************************/
00012
00013 // rxdock docking exceptions, errors during the docking run (e.g. early
00014 // convergence of the GA population)
00015
00016 #ifndef _RBTDOCKINGERROR_H_
00017 #define _RBTDOCKINGERROR_H_
00018
00019 #include "rxdock/Error.h"
00020
00021 namespace rxdock {
00022
00023 const std::string IDS_DOCKING_ERROR = "RBT_DOCKING_ERROR";
00024
00025 // Unspecified model error
00026 class DockingError : public Error {
00027 public:
00028   DockingError(const std::string &strFile, int nLine,
00029                const std::string &strMessage = "")
00030       : Error(IDS_DOCKING_ERROR, strFile, nLine, strMessage) {}
00031   // Protected constructor to allow derived docking error classes
00032   // to set error name
00033 protected:
00034   DockingError(const std::string &strName, const std::string &strFile,
00035                int nLine, const std::string &strMessage = "")
00036       : Error(strName, strFile, nLine, strMessage) {}
00037 };
00038
00039 } // namespace rxdock
00040
00041 #endif //_RBTDOCKINGERROR_H_
```

6.55 DockingSite.h

```
00001 /*********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  *********************************************************************/
00012
00013 // The role of DockingSite is to manage the
00014 // list of cavities generated by the site mapper, and to store a 3-D grid of
00015 // precalculated distances to the nearest cavity coordinate. This grid can be
00016 // used by scoring function classes to lookup the list of atoms required to
00017 // calculate the score component.
00018
00019 #ifndef _RBTDOCKINGSITE_H_
00020 #define _RBTDOCKINGSITE_H_
00021
00022 #include "rxdock/Atom.h"
00023 #include "rxdock/Cavity.h"
00024 #include "rxdock/Config.h"
00025 #include "rxdock/Error.h"
00026 #include "rxdock/RealGrid.h"
00027
00028 namespace rxdock {
00029
00030 class DockingSite {
00031 public:
00032   // STL predicate for selecting atoms within a defined distance range from
00033   // nearest cavity coords Uses precalculated distance grid
00034   class isAtomInRange : public std::function<bool(AtomPtr)> {
00035   public:
00036     explicit isAtomInRange(RealGrid *pGrid, double minDist, double maxDist)
00037         : m_pGrid(pGrid), m_minDist(minDist), m_maxDist(maxDist) {}
00038     bool operator()(Atom *pAtom) const;
00039
00040   private:
```

```
00041     RealGrid *m_pGrid;
00042     double m_minDist;
00043     double m_maxDist;
00044   };
00045
00046 public:
00047   // Class type string
00048   static const std::string _CT;
00049
00050   RBTDLL_EXPORT DockingSite(const CavityList &cavList, double border);
00051   RBTDLL_EXPORT DockingSite(json j);
00052
00053   // Destructor
00054   virtual ~DockingSite();
00055
00056   // Insertion operator
00057   RBTDLL_EXPORT friend std::ostream &operator<<(std::ostream &s,
00058                                                 const DockingSite &site);
00059
00060   RBTDLL_EXPORT friend void to_json(json &j, const DockingSite &site);
00061   friend void from_json(const json &j, DockingSite &site);
00062
00063   // Virtual function for dumping docking site details to an output stream
00064   // Derived classes can override if required
00065   virtual void Print(std::ostream &s) const;
00066
00067   // Public methods
00068   RBTDLL_EXPORT RealGridPtr GetGrid();
00069   double GetBorder() const { return m_border; }
00070   Coord GetMinCoord() const { return m_minCoord; }
00071   Coord GetMaxCoord() const { return m_maxCoord; }
00072   CavityList GetCavityList() const { return m_cavityList; }
00073   int GetNumCavities() const { return m_cavityList.size(); }
00074   RBTDLL_EXPORT double
00075   GetVolume() const; // returns total volume of all cavities in A^3
00076
00077   // Returns the combined coord lists of all the cavities
00078   void GetCoordList(CoordList &retVal) const;
00079
00080   // Filters an atom list according to distance from the cavity coords
00081   // Only returns atoms within minDist and maxDist from cavity
00082   RBTDLL_EXPORT AtomList GetAtomList(const AtomList &atomList, double minDist,
00083                                      double maxDist);
00084   // Filters an atom list according to distance from the cavity coords
00085   // Only returns atoms within maxDist from cavity
00086   // This version does not require the cavity grid
00087   RBTDLL_EXPORT AtomList GetAtomList(const AtomList &atomList, double maxDist);
00088   // Returns the count of atoms within minDist and maxDist from cavity
00089   unsigned int GetNumAtoms(const AtomList &atomList, double minDist,
00090                            double maxDist);
00091
00092   // private methods
00093 private:
00094   DockingSite();                    // default constructor disabled
00095   DockingSite(const DockingSite &); // Copy constructor disabled by
00096                                     // default
00097   DockingSite &
00098   operator=(const DockingSite &); // Copy assignment disabled by default
00099
00100   void CreateGrid();
00101
00102   // Private data
00103 private:
00104   CavityList m_cavityList; // Cavity list
00105   Coord m_minCoord;
00106   Coord m_maxCoord;
00107   RealGridPtr m_spGrid;
00108   double m_border;
00109 };
00110
00111 // Useful typedefs
00112 typedef SmartPtr<DockingSite> DockingSitePtr; // Smart pointer
00113
00114 std::ostream &operator<<(std::ostream &s, const DockingSite &site);
00115
00116 RBTDLL_EXPORT void to_json(json &j, const DockingSite &site);
00117 void from_json(const json &j, DockingSite &site);
00118
00119 } // namespace rxdock
00120
00121 #endif //_RBTDOCKINGSITE_H_
```

6.56 ElementFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // File reader for rxdock elemental data file (atomic no,element name,vdw radii)
00014 // Provides the data as a vector of structs
00015
00016 #ifndef _RBTELEMENTFILESOURCE_H_
00017 #define _RBTELEMENTFILESOURCE_H_
00018
00019 #include "rxdock/BaseFileSource.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // Simple struct for holding the element data
00028 class ElementData {
00029 public:
00030   ElementData()
00031       : atomicNo(0), element(""), minVal(0), maxVal(0), commonVal(0), mass(0.0),
00032         vdwRadius(0.0) {}
00033   ElementData(json j) { j.get_to(*this); }
00034
00035   friend void to_json(json &j, const ElementData &e) {
00036     j = json{{"atomic-number", e.atomicNo}, {"element-name", e.element},
00037              {"minimum-value", e.minVal},   {"maximum-value", e.maxVal},
00038              {"common-value", e.commonVal}, {"atomic-mass", e.mass},
00039              {"vdw-radius", e.vdwRadius}};
00040   }
00041
00042   friend void from_json(const json &j, ElementData &e) {
00043     j.at("atomic-number").get_to(e.atomicNo);
00044     j.at("element-name").get_to(e.element);
00045     j.at("minimum-value").get_to(e.minVal);
00046     j.at("maximum-value").get_to(e.maxVal);
00047     j.at("common-value").get_to(e.commonVal);
00048     j.at("atomic-mass").get_to(e.mass);
00049     j.at("vdw-radius").get_to(e.vdwRadius);
00050   }
00051
00052   int atomicNo;
00053   std::string element;
00054   int minVal;
00055   int maxVal;
00056   int commonVal;
00057   double mass;
00058   double vdwRadius; // Regular vdW radius
00059 };
00060
00061 // Map with element data indexed by element name
00062 typedef std::map<std::string, ElementData> StringElementDataMap;
00063 typedef StringElementDataMap::iterator StringElementDataMapIter;
00064 typedef StringElementDataMap::const_iterator StringElementDataMapConstIter;
00065 // Map with element data indexed by atomic number
00066 typedef std::map<int, ElementData> IntElementDataMap;
00067 typedef IntElementDataMap::iterator IntElementDataMapIter;
00068 typedef IntElementDataMap::const_iterator IntElementDataMapConstIter;
00069
00070 class ElementFileSource : public BaseFileSource {
00071 public:
00072   // Constructors
00073   // ElementFileSource(const char* fileName);
00074   ElementFileSource(const std::string &fileName);
00075
00076   // Destructor
00077   virtual ~ElementFileSource();
00078
00080   // Public methods
00081   RBTDLL_EXPORT std::string GetTitle();
00082   std::string GetVersion();
00083   unsigned int GetNumElements();
00084   std::vector<std::string> GetElementNameList(); // List of element names
00085   std::vector<int> GetAtomicNumberList();        // List of atomic numbers
00086   // Get element data for a given element name, throws error if not found
```

```
00087   ElementData GetElementData(const std::string &strElementName);
00088   // Get element data for a given atomic number, throws error if not found
00089   ElementData GetElementData(int nAtomicNumber);
00090   // Check if given element name is present
00091   bool isElementNamePresent(const std::string &strElementName);
00092   // Check if given atomic number is present
00093   bool isAtomicNumberPresent(int nAtomicNumber);
00094
00095   // Get vdw radius increment for hydrogen-bonding donors
00096   double GetHBondRadiusIncr();
00097   // Get vdw radius increment for atoms with implicit hydrogens
00098   double GetImplicitRadiusIncr();
00099
00100 protected:
00101   // Protected methods
00102
00103 private:
00104   // Private methods
00105   ElementFileSource(); // Disable default constructor
00106   ElementFileSource(
00107       const ElementFileSource &); // Copy constructor disabled by default
00108   ElementFileSource &
00109   operator=(const ElementFileSource &); // Copy assignment disabled by default
00110
00111   // Pure virtual in BaseFileSource - needs to be defined here
00112   virtual void Parse();
00113   void ClearElementDataCache();
00114
00115 protected:
00116   // Protected data
00117
00118 private:
00119   // Private data
00120   std::string m_inputFileName;
00121   std::string m_strTitle;
00122   std::string m_strVersion;
00123   double m_dHBondRadiusIncr;    // Increment to add to vdW radius for H-bonding
00124                                 // hydrogens
00125   double m_dImplicitRadiusIncr; // Increment to add to vdW radius for atoms
00126                                 // with implicit hydrogens
00127   StringElementDataMap m_elementNameMap;
00128   IntElementDataMap m_atomicNumberMap;
00129 };
00130
00131 // useful typedefs
00132 typedef SmartPtr<ElementFileSource> ElementFileSourcePtr; // Smart pointer
00133
00134 } // namespace rxdock
00135
00136 #endif //_RBTELEMENTFILESOURCE_H_
```

6.57 Error.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // rxdock exception base class
00014 // Errors can be thrown for unspecified errors, or derived classes can be
00015 // used for particular exception types.
00016 // Has the following methods:
00017 //
00018 // Name      - exception name (set by constructor)
00019 // File      - source file where error occurred
00020 // Line      - source line no.
00021 // Message   - description of the error
00022 // isOK      - if true, status is OK, not an error
00023 // AddMessage- append new message to existing message, e.g. when rethrowing
00024 // Insertion operator defined for dumping exception details to a stream
00025
00026 #ifndef _RBTERROR_H_
00027 #define _RBTERROR_H_
00028
00029 #include <exception>
00030 #include <sstream>
```

```
00031
00032 // Useful macro for reporting errors
00033 //__FILE__ and __LINE__ are standard macros for source code file name
00034 // and line number respectively
00035 #define _WHERE_ __FILE__, __LINE__
00036
00037 namespace rxdock {
00038
00039 // General errors
00040 const std::string IDS_ERROR = "RBT_ERROR";
00041 const std::string IDS_OK = "RBT_OK";
00042 const std::string IDS_INVALID_REQUEST = "RBT_INVALID_REQUEST";
00043 const std::string IDS_BAD_ARGUMENT = "RBT_BAD_ARGUMENT";
00044 const std::string IDS_ASSERT = "RBT_ASSERT";
00045 const std::string IDS_BAD_RECEPTOR_FILE = "BAD_RECEPTOR_FILE";
00046
00047 // Template for throwing exceptions from assert failures
00048 //(The C++ Programming Language, 3rd Ed, p750)
00049 template <class X, class A> inline void Assert(A assertion) {
00050   if (!assertion) {
00051     throw X();
00052   }
00053 }
00055 // BASE ERROR CLASS
00056 // Note: This is a concrete class so can be instantiated
00058
00059 class Error : public std::exception {
00061   // Public methods
00062 public:
00063   // Default constructor
00064   // Use to create an "error" with status=OK, no line, no file, no message
00065   // All other constructors set the OK flag to false
00066   Error()
00067       : m_strName(IDS_OK), m_strFile(""), m_strMessage(""), m_nLine(0),
00068         m_bOK(true) {}
00069
00070   // Parameterised constructor
00071   // Use to create unspecified rxdock Errors
00072   Error(const std::string &strFile, int nLine,
00073         const std::string &strMessage = "")
00074       : Error(IDS_ERROR, strFile, nLine, strMessage) {}
00075
00076   // Default destructor
00077   virtual ~Error() {}
00078
00079   virtual const char *what() const noexcept { return m_strWhat.c_str(); }
00080
00081   // Attribute methods
00082   std::string File() const { return m_strFile; }       // Get filename
00083   int Line() const { return m_nLine; }                 // Get line number
00084   std::string Message() const { return m_strMessage; } // Get message
00085   std::string Name() const { return m_strName; }       // Get error name
00086   bool isOK() const { return m_bOK; } // If true, status is OK (not an error)
00087
00088   // Append new message to existing message
00089   void AddMessage(const std::string &strMessage) {
00090     m_strMessage += strMessage;
00091     m_strWhat += "\n" + strMessage;
00092   }
00093
00095   // Protected methods
00096 protected:
00097   // Protected constructor to allow derived error classes to set error name
00098   Error(const std::string &strName, const std::string &strFile, int nLine,
00099         const std::string &strMessage = "")
00100       : m_strName(strName), m_strFile(strFile), m_strMessage(strMessage),
00101         m_nLine(nLine), m_bOK(false) {
00102     std::ostringstream oss;
00103     oss « Name();
00104     if (!File().empty())
00105       oss « " at " « File() « ", line " « Line();
00106     if (!Message().empty())
00107       oss « std::endl « Message();
00108     oss.flush();
00109     m_strWhat = oss.str();
00110   }
00111
00113   // Private data
00114 private:
00115   std::string m_strName;
00116   std::string m_strFile;
00117   std::string m_strMessage;
00118   std::string m_strWhat;
00119   int m_nLine;
00120   bool m_bOK;
00121 };
00122
```

```
00124 // DERIVED ERROR CLASSES - General
00126
00127 // Invalid request - object does not supported the request
00128 class InvalidRequest : public Error {
00129 public:
00130   InvalidRequest(const std::string &strFile, int nLine,
00131                  const std::string &strMessage = "")
00132       : Error(IDS_INVALID_REQUEST, strFile, nLine, strMessage) {}
00133 };
00134 // Bad argument - e.g. empty atom list
00135 class BadArgument : public Error {
00136 public:
00137   BadArgument(const std::string &strFile, int nLine,
00138               const std::string &strMessage = "")
00139       : Error(IDS_BAD_ARGUMENT, strFile, nLine, strMessage) {}
00140 };
00141 // Assertion failure
00142 // Assert template doesn't pass params to exception class so can't pass file and
00143 // line number
00144 class Assertion : public Error {
00145 public:
00146   Assertion() : Error(IDS_ASSERT, "unspecified file", 0, "Assertion failed") {}
00147 };
00148
00149 // bad receptor type (ie not PSF/CRD either mol2
00150 class BadReceptorFile : public Error {
00151 public:
00152   BadReceptorFile(const std::string &strFile, int nLine,
00153                   const std::string &strMessage = "")
00154       : Error(IDS_BAD_RECEPTOR_FILE, strFile, nLine, strMessage) {}
00155 };
00156
00157 } // namespace rxdock
00158
00159 #endif //_RBTERROR_H_
```

6.58 Euler.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple class to manage Euler angles
00014 // Main purpose is to provides conversion methods
00015 // to/from quaternion representation
00016 //
00017 // We adopt a modified "NASA Standard Aeroplane" convention
00018 // in which the y and z axes are swapped relative to the definition in
00019 // http://www.euclideanspace.com/maths/geometry/rotations/euler/index.htm
00020 //(and related links).
00021 // This should be equivalent to the VRML convention.
00022 //
00023 // An object's orientation is defined in terms of:
00024 //
00025 // 1) heading (-PI to +PI) (rotation around z) (y in NASA Aeroplane)
00026 // 2) attitude (-PI/2 to +PI/2) (rotation around y) (z in NASA Aeroplane)
00027 // 3) bank (-PI to +PI) (rotation around x)
00028 //
00029 // See http://www.euclideanspace.com/maths/geometry/rotations/euler/index.htm
00030 // and related links for definitions and conversion equations.
00031 //
00032 // NOTE: This class is designed to provide fast conversion between Euler angles
00033 // and quaternions. It is the callers responsibility to ensure that:
00034 // a) heading, attitude and bank angles are within range
00035 // b) quaternions passed to the constructor and SetQuat are of unit length
00036 // Run-time exceptions may occur if these conditions are not met.
00037 #ifndef RBTEULER_H_
00038 #define RBTEULER_H_
00039
00040 #include "rxdock/Quat.h"
00041
00042 #include <nlohmann/json.hpp>
00043
00044 using json = nlohmann::json;
00045
```

```
00046 namespace rxdock {
00047
00048 class Euler {
00049 public:
00050   // Constructor accepting values for heading, attitude and bank (radians).
00051   // NOTE: For performance reasons, no checks are made that the values
00052   // are within range
00053   Euler(double heading = 0.0, double attitude = 0.0, double bank = 0.0)
00054       : m_heading(heading), m_attitude(attitude), m_bank(bank) {}
00055   // Constructor accepting a quaternion.
00056   // Quaternion is converted to Euler angle representation
00057   // NOTE: For performance reasons, q is assumed to be of unit length
00058   // and no further checks are made.
00059   Euler(const Quat &q) { FromQuat(q); }
00060
00061   // Returns true if all three Euler angles are within their
00062   // standard ranges
00063   bool isStandardised() const {
00064     return (m_heading >= -M_PI) && (m_heading <= M_PI) &&
00065            (m_attitude >= -M_PI / 2.0) && (m_attitude <= M_PI / 2.0) &&
00066            (m_bank >= -M_PI) && (m_bank <= M_PI);
00067   }
00068   // Standardises the Euler angles by conversion to/from quaternion
00069   void Standardise() {
00070     if (!isStandardised())
00071       FromQuat(ToQuat());
00072   }
00073
00074   // Gets the current heading angle (radians)
00075   double GetHeading() const { return m_heading; }
00076   // Gets the current attitude angle (radians)
00077   double GetAttitude() const { return m_attitude; }
00078   // Gets the current bank angle (radians)
00079   double GetBank() const { return m_bank; }
00080   // Gets the equivalent quaternion representation
00081   // NOTE: For performance reasons, no checks are made that the current
00082   // Euler angles are within range prior to the conversion.
00083   RBTDLL_EXPORT Quat ToQuat() const;
00084   // Updates the Euler angles from a quaternion representation
00085   // NOTE: For performance reasons, q is assumed to be of unit length
00086   // and no further checks are made.
00087   RBTDLL_EXPORT void FromQuat(const Quat &q);
00088   // Convenience method to rotate orientation by a quaternion
00089   void Rotate(const Quat &q) { FromQuat(q * ToQuat()); }
00090   void Rotate(const Vector &axis, double theta) { Rotate(Quat(axis, theta)); }
00091
00092   // Insertion operator
00093   friend std::ostream &operator«(std::ostream &s, const Euler &euler) {
00094     return s « "Heading = " « euler.m_heading
00095              « ", Attitude = " « euler.m_attitude
00096              « ", Bank = " « euler.m_bank;
00097   }
00098
00099   friend void to_json(json &j, const Euler &eul);
00100   friend void from_json(const json &j, Euler &eul);
00101
00102 private:
00103   double m_heading;  // heading angle (radians)
00104   double m_attitude; // attitude angle (radians)
00105   double m_bank;     // bank angle (radians)
00106 };
00107
00108 void to_json(json &j, const Euler &eul);
00109 void from_json(const json &j, Euler &eul);
00110
00111 } // namespace rxdock
00112
00113 #endif /*RBTEULER_H_*/
```

6.59 FFTGrid.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
```

```
00013 // Subclass of RealGrid with extra methods for finding peaks
00014 // The original FFT functionality has long since been abandoned
00015
00016 #ifndef _RBTFFTGRID_H_
00017 #define _RBTFFTGRID_H_
00018
00019 #include "rxdock/RealGrid.h"
00020
00021 namespace rxdock {
00022
00023 // struct for holding FFT Peak info
00024 class FFTPeak {
00025 public:
00026   // Constructor
00027   FFTPeak() : index(0), coord(0.0, 0.0, 0.0), height(0.0), volume(0) {}
00028
00029   // Data attributes - public to avoid need for accessor functions
00030   unsigned int index;  // Location of peak maximum - iXYZ index into grid
00031   Coord coord;         // Location of peak maximum - real world vector/coord
00032                        // corresponding to displacement
00033   double height;       // Maximum value of peak
00034   unsigned int volume; // Number of grid points in peak (=points.size())
00035   std::set<unsigned int> points; // Set of iXYZ peak indices
00036 };
00037
00038 typedef SmartPtr<FFTPeak> FFTPeakPtr; // Smart pointer
00039 // Multi-map of key=peak height,value=FFTPeakPtr
00040 typedef std::multimap<double, FFTPeakPtr> FFTPeakMap;
00041 typedef FFTPeakMap::iterator FFTPeakMapIter;
00042 typedef FFTPeakMap::reverse_iterator FFTPeakMapRIter;
00043 typedef FFTPeakMap::const_iterator FFTPeakMapConstIter;
00044 typedef FFTPeakMap::const_reverse_iterator FFTPeakMapConstRIter;
00045
00046 class FFTGrid : public RealGrid {
00047 public:
00048   // Class type string
00049   static const std::string _CT;
00050
00052   // Constructors/destructors
00053   // Construct a NXxNYxNZ grid running from gridMin at gridStep resolution
00054   FFTGrid(const Coord &gridMin, const Coord &gridStep, unsigned int NX,
00055           unsigned int NY, unsigned int NZ, unsigned int NPad = 0);
00056
00057   // Constructor reading all params from binary stream
00058   FFTGrid(json j);
00059
00060   ~FFTGrid(); // Default destructor
00061
00062   FFTGrid(const FFTGrid &); // Copy constructor
00063   // Copy constructors taking base class arguments
00064   FFTGrid(const RealGrid &);
00065   FFTGrid(const BaseGrid &);
00066
00067   FFTGrid &operator=(const FFTGrid &); // Copy assignment
00068   // Copy assignment taking base class arguments
00069   FFTGrid &operator=(const RealGrid &);
00070   FFTGrid &operator=(const BaseGrid &);
00071
00073   // Public methods
00075
00077   // Virtual functions for reading/writing grid data to streams in
00078   // text and binary format
00079   // Subclasses should provide their own private OwnPrint
00080   // method to handle subclass data members, and override the public
00081   // Print method
00082   virtual void Print(std::ostream &ostr) const; // Text output
00083
00084   // Find the coords of all (separate) peaks above the threshold value
00085   // whose volumes are not less than minVol
00086   // Returns a map of FFTPeaks
00087   FFTPeakMap FindPeaks(double threshold, unsigned int minVol = 1);
00088   // Returns the grid point with the maximum value in FFTPeak format
00089   // Just a wrapper around FindMaxValue() (see below)
00090   FFTPeak FindMaxPeak() const;
00091
00092 protected:
00094   // Protected methods
00096   // Protected method for writing data members for this class to text stream
00097   void OwnPrint(std::ostream &ostr) const;
00098
00099 private:
00101   // Private methods
00103   FFTGrid(); // Disable default constructor
00104
00105   // Helper function called by copy constructor and assignment operator
00106   void CopyGrid(const FFTGrid &);
00107
```

```
00108 protected:
00110   // Protected data
00112
00113 private:
00115   // Private data
00117 };
00118
00119 // Useful typedefs
00120 typedef SmartPtr<FFTGrid> FFTGridPtr; // Smart pointer
00121
00122 } // namespace rxdock
00123
00124 #endif //_RBTFFTGRID_H_
```

6.60 FileError.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // rxdock file-based exceptions
00014
00015 #ifndef _RBTFILEERROR_H_
00016 #define _RBTFILEERROR_H_
00017
00018 #include "rxdock/Error.h"
00019
00020 namespace rxdock {
00021
00022 const std::string IDS_FILE_ERROR = "RBT_FILE_ERROR";
00023 const std::string IDS_FILE_READ_ERROR = "RBT_FILE_READ_ERROR";
00024 const std::string IDS_FILE_WRITE_ERROR = "RBT_FILE_WRITE_ERROR";
00025 const std::string IDS_FILE_PARSE_ERROR = "RBT_FILE_PARSE_ERROR";
00026 const std::string IDS_FILE_MISSING_PARAMETER = "RBT_FILE_MISSING_PARAMETER";
00027 const std::string IDS_DIR_NOACCESS = "RBT_DIR_NOACCESS";
00028 const std::string IDS_NO_FILEINDIR = "RBT_NO_FILEINDIR";
00029 const std::string IDS_NOENV = "RBT_NO_ENVIRONMENT";
00030 const std::string IDS_STRINGTOOLONG = "RBT_SRINGTOOLONG";
00031
00032 // Unspecified file error
00033 class FileError : public Error {
00034 public:
00035   FileError(const std::string &strFile, int nLine,
00036             const std::string &strMessage = "")
00037       : Error(IDS_FILE_ERROR, strFile, nLine, strMessage) {}
00038   // Protected constructor to allow derived file error classes to set error name
00039 protected:
00040   FileError(const std::string &strName, const std::string &strFile, int nLine,
00041             const std::string &strMessage = "")
00042       : Error(strName, strFile, nLine, strMessage) {}
00043 };
00044
00045 // File read error
00046 class FileReadError : public FileError {
00047 public:
00048   FileReadError(const std::string &strFile, int nLine,
00049                 const std::string &strMessage = "")
00050       : FileError(IDS_FILE_READ_ERROR, strFile, nLine, strMessage) {}
00051 };
00052
00053 // File write error
00054 class FileWriteError : public FileError {
00055 public:
00056   FileWriteError(const std::string &strFile, int nLine,
00057                  const std::string &strMessage = "")
00058       : FileError(IDS_FILE_WRITE_ERROR, strFile, nLine, strMessage) {}
00059 };
00060
00061 // File parse error
00062 class FileParseError : public FileError {
00063 public:
00064   FileParseError(const std::string &strFile, int nLine,
00065                  const std::string &strMessage = "")
00066       : FileError(IDS_FILE_PARSE_ERROR, strFile, nLine, strMessage) {}
00067 };
```

```
00068
00069 // Missing parameter error
00070 class FileMissingParameter : public FileError {
00071 public:
00072   FileMissingParameter(const std::string &strFile, int nLine,
00073                        const std::string &strMessage = "")
00074       : FileError(IDS_FILE_MISSING_PARAMETER, strFile, nLine, strMessage) {}
00075 };
00076
00077 // Exceptions for DirectorySource and derived classes:
00078 //
00079 // Directory access error
00080 class DirIsNotAccessible : public FileError {
00081 public:
00082   DirIsNotAccessible(const std::string &strFile, int nLine,
00083                      const std::string &strMessage = "")
00084       : FileError(IDS_DIR_NOACCESS, strFile, nLine, strMessage) {}
00085 };
00086
00087 // Directory contains no file with that extension
00088 class NoFileInDir : public FileError {
00089 public:
00090   NoFileInDir(const std::string &strFile, int nLine,
00091               const std::string &strMessage = "")
00092       : FileError(IDS_NO_FILEINDIR, strFile, nLine, strMessage) {}
00093 };
00094
00095 // Misc exceptions
00096 //
00097 // If path is missing
00098 class EnvNotDefined : public FileError {
00099 public:
00100   EnvNotDefined(const std::string &strFile, int nLine,
00101                 const std::string &strMessage = "")
00102       : FileError(IDS_NOENV, strFile, nLine, strMessage) {}
00103 };
00104
00105 // If string is too long
00106 class StringTooLong : public FileError {
00107 public:
00108   StringTooLong(const std::string &strFile, int nLine,
00109                 const std::string &strMessage = "")
00110       : FileError(IDS_STRINGTOOLONG, strFile, nLine, strMessage) {}
00111 };
00112
00113 } // namespace rxdock
00114
00115 #endif //_RBTFILEERROR_H_
```

6.61 Filter.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Reads a file that contains a list of filters that determine
00014 // when to terminate with a given ligand, and another filter to determine
00015 // when to write the current conformation to the output.
00016
00017 #ifndef _RBT_FILTER_H_
00018 #define _RBT_FILTER_H_
00019
00020 #include "rxdock/BaseObject.h"
00021 #include "rxdock/Context.h"
00022 #include "rxdock/FilterExpression.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
00030 class FilterExpressionVisitor;
00031
00032 class Filter : public BaseObject {
```

```
00033 public:
00034   static const std::string _CT;
00035   RBTDLL_EXPORT Filter(std::string strfilter, bool filter = false);
00037   // Destructor
00039   virtual ~Filter();
00040
00041   friend void to_json(json &j, const Filter &filter);
00042   friend void from_json(const json &j, Filter &filter);
00043
00044   // Override Observer pure virtual
00045   // Notify observer that subject has changed
00046   virtual void Update(Subject *theChangedSubject);
00047
00048   void SetupReceptor();       // Called by Update when receptor is changed
00049   void SetupLigand();         // Called by Update when ligand is changed
00050   void SetupScore();          // Called by Update when either model has changed
00051   RBTDLL_EXPORT bool Write(); // Output conformation?
00052   RBTDLL_EXPORT bool Terminate(); // Finished with ligand?
00053   ModelPtr GetReceptor() const;
00054   ModelPtr GetLigand() const;
00055   void SetMaxNRuns(int n) { maxnruns = n; }
00056
00058   // Private methods
00060
00061 protected:
00062   Filter(); // Default constructor disabled
00063
00064 private:
00065   void ReadFilters();
00066
00067   int filteridx, nTermFilters, nWriteFilters, nruns, maxnruns;
00068   FilterExpressionList terminationFilters;
00069   FilterExpressionList writtingFilter;
00070   ModelPtr m_spReceptor;
00071   ModelPtr m_spLigand;
00072   ContextPtr contextp;
00073 };
00074
00075 void to_json(json &j, const Filter &filter);
00076 void from_json(const json &j, Filter &filter);
00077
00078 // Useful typedefs
00079 typedef SmartPtr<Filter> FilterPtr; // Smart pointer
00080 typedef std::vector<FilterPtr> FilterList;
00081 // Vector of smart pointers
00082 typedef FilterList::iterator FilterListIter;
00083 typedef FilterList::const_iterator FilterListConstIter;
00084
00085 } // namespace rxdock
00086
00087 #endif //_Filter_H_
```

6.62 FilterExpression.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Takes a list of tokens and parsers them to create a Filter Expression
00014
00015 #ifndef _RBT_FILTEREXPRESSION_H_
00016 #define _RBT_FILTEREXPRESSION_H_
00017
00018 #include "rxdock/Error.h"
00019 #include "rxdock/FilterExpressionVisitor.h"
00020 #include "rxdock/TokenIter.h"
00021 #include "rxdock/geneticprogram/GPTypes.h"
00022
00023 #include <iostream>
00024
00025 #include <nlohmann/json.hpp>
00026
00027 using json = nlohmann::json;
00028
00029 namespace rxdock {
```

```
00030
00031 class FilterExpressionVisitor;
00032
00033 class FilterExpression {
00034 public:
00035   static const std::string _CT;
00036   virtual void Print() = 0;
00037   virtual void Accept(FilterExpressionVisitor &) = 0;
00038   virtual int GetNOps() = 0;
00039   virtual SmartPtr<FilterExpression> GetOp(int) = 0;
00041   // Destructor
00043   virtual ~FilterExpression();
00044
00045   friend void to_json(json &j, const FilterExpression &filterExpression);
00046   friend void from_json(const json &j, FilterExpression &filterExpression);
00047
00048   ReturnType GetValue() { return value; }
00049   void SetValue(ReturnType v) { value = v; }
00050
00052   // Private methods
00054
00055 protected:
00056   FilterExpression(); // Default constructor disabled
00057   ReturnType value;
00058 };
00059
00060 void to_json(json &j, const FilterExpression &filterExpression);
00061 void from_json(const json &j, FilterExpression &filterExpression);
00062
00063 // Useful typedefs
00064 typedef SmartPtr<FilterExpression> FilterExpressionPtr; // Smart pointer
00065 typedef std::vector<FilterExpressionPtr> FilterExpressionList;
00066 // Vector of smart pointers
00067 typedef FilterExpressionList::iterator FilterExpressionListIter;
00068 typedef FilterExpressionList::const_iterator FilterExpressionListConstIter;
00069
00070 class FilterVbleExp : public FilterExpression {
00071 public:
00072   FilterVbleExp(const Vble &);
00073   virtual ~FilterVbleExp();
00074   void Accept(FilterExpressionVisitor &visitor);
00075   void Print() { std::cout « "v" « vble.GetName() « " "; }
00076   int GetNOps() { return 0; }
00077   const Vble &GetVble() const { return vble; }
00078   FilterExpressionPtr GetOp(int i) {
00079     throw Error(_WHERE_, "Vble Expressions don't have operands");
00080   }
00081
00082 private:
00083   const Vble &vble;
00084 };
00085
00086 class FilterLogExp : public FilterExpression {
00087 public:
00088   FilterLogExp(FilterExpressionPtr);
00089   virtual ~FilterLogExp();
00090   void Accept(FilterExpressionVisitor &visitor);
00091   void Print() {
00092     std::cout « "log ";
00093     operand->Print();
00094   }
00095   virtual int GetNOps() { return 1; }
00096   FilterExpressionPtr GetOp(int i) {
00097     if (i == 0)
00098       return operand;
00099     else
00100       throw Error(_WHERE_, "Log Expressions only have 1 operand");
00101   }
00102
00103 private:
00104   FilterExpressionPtr operand;
00105 };
00106
00107 class FilterExpExp : public FilterExpression {
00108 public:
00109   FilterExpExp(FilterExpressionPtr);
00110   virtual ~FilterExpExp();
00111   void Accept(FilterExpressionVisitor &visitor);
00112   void Print() {
00113     std::cout « "exp ";
00114     operand->Print();
00115   }
00116   virtual int GetNOps() { return 1; }
00117   FilterExpressionPtr GetOp(int i) {
00118     if (i == 0)
00119       return operand;
00120     else
```

```
00121       throw Error(_WHERE_, "Exp Expressions only have 1 operand");
00122   }
00123
00124 private:
00125   FilterExpressionPtr operand;
00126 };
00127
00128 class FilterAddExp : public FilterExpression {
00129 public:
00130   FilterAddExp(FilterExpressionPtr, FilterExpressionPtr);
00131   virtual ~FilterAddExp();
00132   void Accept(FilterExpressionVisitor &visitor);
00133   void Print() {
00134     std::cout << "add ";
00135     operand1->Print();
00136     operand2->Print();
00137   }
00138   virtual int GetNOps() { return 2; }
00139   FilterExpressionPtr GetOp(int i) {
00140     if (i == 0)
00141       return operand1;
00142     if (i == 1)
00143       return operand2;
00144     else
00145       throw Error(_WHERE_, "Add Expressions only have 2 operands");
00146   }
00147
00148 private:
00149   FilterExpressionPtr operand1, operand2;
00150 };
00151
00152 class FilterSubExp : public FilterExpression {
00153 public:
00154   FilterSubExp(FilterExpressionPtr, FilterExpressionPtr);
00155   virtual ~FilterSubExp();
00156   void Accept(FilterExpressionVisitor &visitor);
00157   void Print() {
00158     std::cout << "sub ";
00159     operand1->Print();
00160     operand2->Print();
00161   }
00162   virtual int GetNOps() { return 2; }
00163   FilterExpressionPtr GetOp(int i) {
00164     if (i == 0)
00165       return operand1;
00166     if (i == 1)
00167       return operand2;
00168     else
00169       throw Error(_WHERE_, "Sub Expressions only have 2 operands");
00170   }
00171
00172 private:
00173   FilterExpressionPtr operand1, operand2;
00174 };
00175
00176 class FilterMulExp : public FilterExpression {
00177 public:
00178   FilterMulExp(FilterExpressionPtr, FilterExpressionPtr);
00179   virtual ~FilterMulExp();
00180   void Accept(FilterExpressionVisitor &visitor);
00181   void Print() {
00182     std::cout << "mul ";
00183     operand1->Print();
00184     operand2->Print();
00185   }
00186   virtual int GetNOps() { return 2; }
00187   FilterExpressionPtr GetOp(int i) {
00188     if (i == 0)
00189       return operand1;
00190     if (i == 1)
00191       return operand2;
00192     else
00193       throw Error(_WHERE_, "Mul Expressions only have 2 operands");
00194   }
00195
00196 private:
00197   FilterExpressionPtr operand1, operand2;
00198 };
00199
00200 class FilterDivExp : public FilterExpression {
00201 public:
00202   FilterDivExp(FilterExpressionPtr, FilterExpressionPtr);
00203   virtual ~FilterDivExp();
00204   void Accept(FilterExpressionVisitor &visitor);
00205   void Print() {
00206     std::cout << "div ";
00207     operand1->Print();
```

```
00208     operand2->Print();
00209   }
00210   virtual int GetNOps() { return 2; }
00211   FilterExpressionPtr GetOp(int i) {
00212     if (i == 0)
00213       return operand1;
00214     if (i == 1)
00215       return operand2;
00216     else
00217       throw Error(_WHERE_, "Div Expressions only have 2 operands");
00218   }
00219
00220 private:
00221   FilterExpressionPtr operand1, operand2;
00222 };
00223
00224 class FilterAndExp : public FilterExpression {
00225 public:
00226   FilterAndExp(FilterExpressionPtr, FilterExpressionPtr);
00227   virtual ~FilterAndExp();
00228   void Accept(FilterExpressionVisitor &visitor);
00229   void Print() {
00230     std::cout << "and ";
00231     operand1->Print();
00232     operand2->Print();
00233   }
00234   virtual int GetNOps() { return 2; }
00235   FilterExpressionPtr GetOp(int i) {
00236     if (i == 0)
00237       return operand1;
00238     if (i == 1)
00239       return operand2;
00240     else
00241       throw Error(_WHERE_, "And Expressions only have 2 operands");
00242   }
00243
00244 private:
00245   FilterExpressionPtr operand1, operand2;
00246 };
00247
00248 class FilterIfExp : public FilterExpression {
00249 public:
00250   FilterIfExp(FilterExpressionPtr, FilterExpressionPtr, FilterExpressionPtr);
00251   virtual ~FilterIfExp();
00252   void Accept(FilterExpressionVisitor &visitor);
00253   void Print() {
00254     std::cout << "if ";
00255     operand1->Print();
00256     operand2->Print();
00257     operand3->Print();
00258   }
00259   virtual int GetNOps() { return 3; }
00260   FilterExpressionPtr GetOp(int i) {
00261     if (i == 0)
00262       return operand1;
00263     if (i == 1)
00264       return operand2;
00265     if (i == 2)
00266       return operand3;
00267     else
00268       throw Error(_WHERE_, "Add Expressions only have 3 operands");
00269   }
00270
00271 private:
00272   FilterExpressionPtr operand1, operand2, operand3;
00273 };
00274
00275 } // namespace rxdock
00276
00277 #endif //_FilterExpression_H_
```

6.63 FilterExpressionVisitor.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
```

```
00011  ***************************************************************************/
00012
00013 // Visitors will be created to
00014 // execute the functions represented by the expressions, print them etc.
00015
00016 #ifndef _RBT_FILTEREXPRESSIONVISITOR_H_
00017 #define _RBT_FILTEREXPRESSIONVISITOR_H_
00018
00019 #include "rxdock/Context.h"
00020 #include "rxdock/FilterExpression.h"
00021
00022 namespace rxdock {
00023
00024 class FilterVbleExp;
00025 class FilterAddExp;
00026 class FilterSubExp;
00027 class FilterMulExp;
00028 class FilterDivExp;
00029 class FilterAndExp;
00030 class FilterLogExp;
00031 class FilterExpExp;
00032 class FilterIfExp;
00033
00034 class FilterExpressionVisitor {
00035 public:
00036   virtual void VisitVbleExp(FilterVbleExp *) = 0;
00037   virtual void VisitAddExp(FilterAddExp *) = 0;
00038   virtual void VisitSubExp(FilterSubExp *) = 0;
00039   virtual void VisitMulExp(FilterMulExp *) = 0;
00040   virtual void VisitDivExp(FilterDivExp *) = 0;
00041   virtual void VisitAndExp(FilterAndExp *) = 0;
00042   virtual void VisitLogExp(FilterLogExp *) = 0;
00043   virtual void VisitExpExp(FilterExpExp *) = 0;
00044   virtual void VisitIfExp(FilterIfExp *) = 0;
00046   // Destructor
00048   virtual ~FilterExpressionVisitor();
00049
00051   // Protected methods
00053
00054 protected:
00055   FilterExpressionVisitor(); // Default constructor disabled
00056 };
00057
00058 class PrintVisitor : public FilterExpressionVisitor {
00059 public:
00060   PrintVisitor(ContextPtr);
00061   ~PrintVisitor();
00062   virtual void VisitVbleExp(FilterVbleExp *);
00063   virtual void VisitAddExp(FilterAddExp *);
00064   virtual void VisitSubExp(FilterSubExp *);
00065   virtual void VisitMulExp(FilterMulExp *);
00066   virtual void VisitDivExp(FilterDivExp *);
00067   virtual void VisitAndExp(FilterAndExp *);
00068   virtual void VisitLogExp(FilterLogExp *);
00069   virtual void VisitExpExp(FilterExpExp *);
00070   virtual void VisitIfExp(FilterIfExp *);
00071
00072 private:
00073   ContextPtr contextp;
00074   std::ostringstream strbuf;
00075 };
00076
00077 class PrettyPrintVisitor : public FilterExpressionVisitor {
00078 public:
00079   PrettyPrintVisitor(ContextPtr);
00080   ~PrettyPrintVisitor();
00081   virtual void VisitVbleExp(FilterVbleExp *);
00082   virtual void VisitAddExp(FilterAddExp *);
00083   virtual void VisitSubExp(FilterSubExp *);
00084   virtual void VisitMulExp(FilterMulExp *);
00085   virtual void VisitDivExp(FilterDivExp *);
00086   virtual void VisitAndExp(FilterAndExp *);
00087   virtual void VisitLogExp(FilterLogExp *);
00088   virtual void VisitExpExp(FilterExpExp *);
00089   virtual void VisitIfExp(FilterIfExp *);
00090
00091 private:
00092   ContextPtr contextp;
00093   std::ostringstream strbuf;
00094 };
00095
00096 class EvaluateVisitor : public FilterExpressionVisitor {
00097 public:
00098   EvaluateVisitor(ContextPtr);
00099   virtual void VisitVbleExp(FilterVbleExp *);
00100   virtual void VisitAddExp(FilterAddExp *);
00101   virtual void VisitSubExp(FilterSubExp *);
```

```
00102   virtual void VisitMulExp(FilterMulExp *);
00103   virtual void VisitDivExp(FilterDivExp *);
00104   virtual void VisitAndExp(FilterAndExp *);
00105   virtual void VisitLogExp(FilterLogExp *);
00106   virtual void VisitExpExp(FilterExpExp *);
00107   virtual void VisitIfExp(FilterIfExp *);
00108
00109 private:
00110   ContextPtr contextp;
00111 };
00112
00113 } // namespace rxdock
00114
00115 #endif
```

6.64 FlexAtomFactory.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // FlexDataVisitor class to generate lists of fixed, tethered and free atoms
00014 // from a model or list of models
00015 #ifndef _RBTFLEXATOMFACTORY_H_
00016 #define _RBTFLEXATOMFACTORY_H_
00017
00018 #include "rxdock/FlexDataVisitor.h"
00019 #include "rxdock/Model.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // The main use of FlexAtomFactory is to aid with scoring function
00028 // performance enhancements by partitioning model atoms into subsets that are:
00029 // - fixed (never move)
00030 // - tethered (move, but within a predictable radius)
00031 // - free (move freely)
00032 //
00033 // For receptor models, the majority of atoms are FIXED
00034 // The flexible OH/NH3+ terminal atoms are defined as TETHERED
00035 //
00036 // Two categorisations are defined for ligands and solvent
00037 // depending on the internal flexibility of the model
00038 //
00039 // A) If model is flexible (Model::isFlexible()), all atoms are defined as
00040 // FREE as it is too difficult to take the impact of internal dihedral motion
00041 // into account
00042 //
00043 // B) If model is not flexible (e.g. explicit solvent), the mapping is:
00044 // transMode == FIXED && rotMode == FIXED => atoms are FIXED
00045 // transMode == FREE => atoms are FREE
00046 // else atoms are TETHERED
00047 //
00048 // On exit:
00049 // User2Value of each tethered atom is set to the maximum displacement
00050 // possible for translational and rotational moves. The value is a
00051 // conservative overestimate as the allowed volume mapped on by each tethered
00052 // atom can be complex. Here we define the sphere radius that includes all
00053 // allowed coordinates.
00054 // The User2Value for fixed and free atoms is set to zero.
00055 class RBTDLL_EXPORT FlexAtomFactory : public FlexDataVisitor {
00056 public:
00057   FlexAtomFactory() {}
00058   // Constructor accepting a single model
00059   FlexAtomFactory(Model *);
00060   // Constructor accepting a list of models
00061   FlexAtomFactory(ModelList);
00062   // Helper method to visit a single model
00063   void Visit(Model *);
00064   // Clears the partitioned atom lists
00065   void Clear();
00066   // Implement the base class abstract methods
```

```
00067   virtual void VisitReceptorFlexData(ReceptorFlexData *);
00068   virtual void VisitLigandFlexData(LigandFlexData *);
00069   virtual void VisitSolventFlexData(SolventFlexData *);
00070   // Get methods for the partitioned atom lists
00071   AtomRList GetFixedAtomList() const { return m_fixedAtomList; }
00072   AtomRList GetTetheredAtomList() const { return m_tetheredAtomList; }
00073   AtomRList GetFreeAtomList() const { return m_freeAtomList; }
00074
00075   friend void to_json(json &j, const FlexAtomFactory &flexAtomFactory);
00076   friend void from_json(const json &j, FlexAtomFactory &flexAtomFactory);
00077
00078 private:
00079   AtomRList m_fixedAtomList;
00080   AtomRList m_tetheredAtomList;
00081   AtomRList m_freeAtomList;
00082 };
00083
00084 void to_json(json &j, const FlexAtomFactory &flexAtomFactory);
00085 void from_json(const json &j, FlexAtomFactory &flexAtomFactory);
00086
00087 } // namespace rxdock
00088
00089 #endif //_RBTFLEXATOMFACTORY_H_
```

6.65 FlexData.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract base class for all model flexibility types
00014 // Provides Accept method for FlexDataVisitors
00015 //(subclasses must implement)
00016 #ifndef _RBTFLEXDATA_
00017 #define _RBTFLEXDATA_
00018
00019 #include "rxdock/FlexDataVisitor.h"
00020 #include "rxdock/ParamHandler.h"
00021
00022 namespace rxdock {
00023
00024 class Model;
00025 class DockingSite;
00026
00027 class FlexData : public ParamHandler {
00028 public:
00029   virtual ~FlexData() {}
00030   Model *GetModel() const { return m_pModel; }
00031   void SetModel(Model *pModel) { m_pModel = pModel; }
00032   DockingSite *GetDockingSite() const { return m_pDockSite; }
00033   virtual void Accept(FlexDataVisitor &) = 0;
00034
00035 protected:
00036   FlexData(DockingSite *pDockSite)
00037       : m_pModel(nullptr), m_pDockSite(pDockSite) {}
00038
00039 private:
00040   Model *m_pModel;
00041   DockingSite *m_pDockSite;
00042 };
00043
00044 typedef SmartPtr<FlexData> FlexDataPtr;
00045
00046 } // namespace rxdock
00047
00048 #endif //_RBTFLEXDATA_
```

6.66 FlexDataVisitor.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
```

```
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Abstract interface for all visitors of FlexData subclasses
00014 #ifndef _RBTFLEXDATAVISITOR_H_
00015 #define _RBTFLEXDATAVISITOR_H_
00016
00017 namespace rxdock {
00018
00019 class ReceptorFlexData;
00020 class LigandFlexData;
00021 class SolventFlexData;
00022
00023 class FlexDataVisitor {
00024 public:
00025   virtual ~FlexDataVisitor() {}
00026   virtual void VisitReceptorFlexData(ReceptorFlexData *) = 0;
00027   virtual void VisitLigandFlexData(LigandFlexData *) = 0;
00028   virtual void VisitSolventFlexData(SolventFlexData *) = 0;
00029 };
00030
00031 } // namespace rxdock
00032
00033 #endif //_RBTFLEXDATAVISITOR_H_
```

6.67 GATransform.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Evolves an existing population using a GA
00014 #ifndef _RBTGATRANSFORM_H_
00015 #define _RBTGATRANSFORM_H_
00016
00017 #include "rxdock/BaseBiMolTransform.h"
00018 #include "rxdock/Population.h"
00019 #include "rxdock/Rand.h"
00020
00021 namespace rxdock {
00022
00023 class GATransform : public BaseBiMolTransform {
00024 public:
00025   static const std::string _CT;
00026   // New individuals to create each cycle, as fraction of population size
00027   static const std::string _NEW_FRACTION;
00028   // Probability of crossover (1-probability of mutation)
00029   static const std::string _PCROSSOVER;
00030   // If true, perform Cauchy mutation after each crossover
00031   static const std::string _XOVERMUT;
00032   // If true, mutations are from Cauchy distribution; if false, from rect.
00033   // distribution
00034   static const std::string _CMUTATE;
00035   // Relative step size for mutations (relative to absolute step sizes defined
00036   // for each chromosome element)
00037   static const std::string _STEP_SIZE;
00038   // Two genomes are considered equal if the maximum relative difference
00039   // between chromosome elements is less than _EQUALITY_THRESHOLD
00040   static const std::string _EQUALITY_THRESHOLD;
00041   // Maximum number of cycles
00042   static const std::string _NCYCLES;
00043   // Terminate if the best score does not improve over _NCONVERGENCE
00044   // consecutive cycles
00045   static const std::string _NCONVERGENCE;
00046   // Output the best pose every _HISTORY_FREQ cycles.
00047   static const std::string _HISTORY_FREQ;
00048
00050   // Constructors/destructors
```

```
00052   RBTDLL_EXPORT GATransform(const std::string &strName = "GAGENRW");
00053   virtual ~GATransform();
00054
00055 protected:
00057   // Protected methods
00059   virtual void SetupReceptor(); // Called by Update when receptor is changed
00060   virtual void SetupLigand();   // Called by Update when ligand is changed
00061   virtual void
00062   SetupTransform(); // Called by Update when either model has changed
00063   virtual void Execute();
00064
00065 private:
00067   // Private methods
00069   GATransform(const GATransform &); // Copy constructor disabled by
00070                                     // default
00071   GATransform &
00072   operator=(const GATransform &); // Copy assignment disabled by default
00073
00074 private:
00075   Rand &m_rand;
00076 };
00077
00078 } // namespace rxdock
00079
00080 #endif //_RBTGATRANSFORM_H_
```

6.68 Cell.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Cell Class. Cell represents a function or input variable of the CGP
00014
00015 #ifndef _RBT_CELL_H_
00016 #define _RBT_CELL_H_
00017
00018 #include "rxdock/geneticprogram/GPTypes.h"
00019
00020 #include <loguru.hpp>
00021
00022 namespace rxdock {
00023
00024 class Cell {
00025 public:
00027   // Constructors
00029   Cell() {
00030     evaluated = false;
00031     name = "";
00032     constant = false;
00033   }
00034   Cell(const Cell &c)
00035       : evaluated(c.evaluated), constant(c.constant), name(c.name),
00036         result(c.result) {}
00037
00039   // Destructor
00041   virtual ~Cell() {}
00042
00043   Cell &operator=(const Cell &c) {
00044     evaluated = c.evaluated;
00045     constant = c.constant;
00046     name = c.name;
00047     result = c.result;
00048     return *this;
00049   }
00050
00051   bool Evaluated() const { return evaluated; }
00052   bool Named() const { return (name != ""); }
00053   bool Constant() const { return constant; }
00054   void Clear() {
00055     if (!constant) {
00056       evaluated = false;
00057       name = "";
00058     }
00059   }
```

```
00060   ReturnType GetResult() const { return result; }
00061   std::string GetName() const { return name; }
00062   void SetName(std::string s) { name = s; }
00063   void SetResult(ReturnType r) {
00064     result = r;
00065     evaluated = true;
00066   }
00067   void SetConstant(ReturnType r) {
00068     result = r;
00069     evaluated = true;
00070     constant = true;
00071     name = std::to_string(r);
00072   }
00073   void ResetConstant() {
00074     evaluated = false;
00075     name = "";
00076     if (constant)
00077       constant = false;
00078     else
00079       LOG_F(WARNING, "Cell::ResetConstant: no constant");
00080   }
00081
00082 private:
00083   bool evaluated, constant;
00084   std::string name;
00085   ReturnType result;
00086   //    friend std::ostream& operator«(std::ostream& s, const Cell &p);
00087 };
00088
00089 // Useful typedefs
00090 typedef SmartPtr<Cell> CellPtr;        // Smart pointer
00091 typedef std::vector<CellPtr> CellList; // Vector of smart pointers
00092 typedef CellList::iterator CellListIter;
00093 typedef CellList::const_iterator CellListConstIter;
00094
00095 } // namespace rxdock
00096
00097 #endif //_Cell_H_
```

6.69 Command.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Command Class. Command represents a function or input variable of the CGP
00014
00015 #ifndef _RBT_COMMAND_H_
00016 #define _RBT_COMMAND_H_
00017
00018 #include "rxdock/geneticprogram/GPTypes.h"
00019
00020 #include <cmath>
00021
00022 namespace rxdock {
00023
00024 class Command {
00025 public:
00026   virtual ReturnType Execute() = 0;
00027   virtual int GetNArgs() = 0;
00028   virtual std::string GetName() = 0;
00029   virtual void SetArg(int i, ReturnType f) = 0;       // { *(Arg[i]) = f;};
00030   virtual void SetNameArg(int i, std::string &n) = 0; // { NameArg[i] = n;};
00031   static void Clear() { ntabs = 0; }
00032
00034   // Destructor
00036   virtual ~Command() {}
00037   static int ntabs;
00038   static bool inside;
00039
00040 protected:
00042   // Constructors
00044   Command() {} // Default constructor disabled
00045   std::string tabs(int n) {
00046     std::string st = "";
```

```
00047     for (int i = 0; i < n; i++)
00048       st += "  ";
00049     return st;
00050   }
00051
00052 private:
00053   //    friend std::ostream& operator«(std::ostream& s, const Command &p);
00054 };
00055
00056 // Useful typedefs
00057 typedef SmartPtr<Command> CommandPtr;        // Smart pointer
00058 typedef std::vector<CommandPtr> CommandList; // Vector of smart pointers
00059 typedef CommandList::iterator CommandListIter;
00060 typedef CommandList::const_iterator CommandListConstIter;
00061 class AddCommand : public Command {
00062 public:
00063   int GetNArgs() { return 2; }
00064   std::string GetName() { return ("(" + NameArg[0] + "+ " + NameArg[1] + ")"); }
00065   AddCommand() {} //{Arg = ReturnType(2);}
00066   void SetArg(int i, ReturnType f) { Arg[i] = f; }
00067   void SetNameArg(int i, std::string &n) { NameArg[i] = n; }
00068   ReturnType Execute() { return Arg[0] + Arg[1]; }
00069
00070 private:
00071   ReturnType Arg[2];
00072   std::string NameArg[2];
00073 };
00074
00075 class SubCommand : public Command {
00076 public:
00077   int GetNArgs() { return 2; }
00078   std::string GetName() { return ("(" + NameArg[0] + "- " + NameArg[1] + ")"); }
00079   void SetArg(int i, ReturnType f) { Arg[i] = f; }
00080   void SetNameArg(int i, std::string &n) { NameArg[i] = n; }
00081   SubCommand() {}
00082   ReturnType Execute() { return Arg[0] - Arg[1]; }
00083
00084 private:
00085   ReturnType Arg[2];
00086   std::string NameArg[2];
00087 };
00088
00089 class MulCommand : public Command {
00090 public:
00091   int GetNArgs() { return 2; }
00092   std::string GetName() { return ("(" + NameArg[0] + "* " + NameArg[1] + ")"); }
00093   void SetArg(int i, ReturnType f) { Arg[i] = f; }
00094   void SetNameArg(int i, std::string &n) { NameArg[i] = n; }
00095   MulCommand() {}
00096   ReturnType Execute() { return Arg[0] * Arg[1]; }
00097
00098 private:
00099   ReturnType Arg[2];
00100   std::string NameArg[2];
00101 };
00102
00103 class DivCommand : public Command {
00104 public:
00105   int GetNArgs() { return 2; }
00106   std::string GetName() {
00107     return ("(" + NameArg[0] + "div " + NameArg[1] + ")");
00108   }
00109   DivCommand() {}
00110   void SetArg(int i, ReturnType f) { Arg[i] = f; }
00111   void SetNameArg(int i, std::string &n) { NameArg[i] = n; }
00112   ReturnType Execute() {
00113     if (std::fabs(Arg[1]) < 0.000001)
00114       return Arg[0];
00115     return Arg[0] / Arg[1];
00116   }
00117
00118 private:
00119   ReturnType Arg[2];
00120   std::string NameArg[2];
00121 };
00122
00123 class IfCommand : public Command {
00124 public:
00125   int GetNArgs() { return 3; }
00126   std::string GetName() {
00127     //  return ("\niff(" + NameArg[0] + "," + NameArg[1] + "," +
00128     //          NameArg[2] + ")");};
00129     if (inside)
00130       return ("iff(" + NameArg[0] + " , " + NameArg[1] + " , " + NameArg[2] +
00131               ")");
00132     return ("\n" + tabs(ntabs - 1) + "if " + NameArg[0] + "> 0 then\n" +
00133             tabs(ntabs) + NameArg[1] + "\n" + tabs(ntabs - 1) + "else \n" +
```

```
00134             tabs(ntabs) + NameArg[2] + "\n" + tabs(ntabs - 1) + "end\n");
00135   }
00136   IfCommand() {}
00137   void SetArg(int i, ReturnType f) { Arg[i] = f; }
00138   void SetNameArg(int i, std::string &n) { NameArg[i] = n; }
00139   ReturnType Execute() {
00140     if (Arg[0] > 0.0)
00141       return Arg[1];
00142     return Arg[2];
00143   }
00144
00145 private:
00146   ReturnType Arg[3];
00147   std::string NameArg[3];
00148 };
00149
00150 class LogCommand : public Command {
00151 public:
00152   int GetNArgs() { return 1; }
00153   std::string GetName() { return ("log(" + NameArg[0] + ")"); }
00154   LogCommand() {}
00155   void SetArg(int i, ReturnType f) { Arg[i] = f; }
00156   void SetNameArg(int i, std::string &n) { NameArg[i] = n; }
00157   ReturnType Execute() {
00158     if (std::fabs(Arg[0]) < 0.000001)
00159       return 0;
00160     return (std::log(std::fabs(Arg[0])));
00161   }
00162
00163 private:
00164   ReturnType Arg[1];
00165   std::string NameArg[1];
00166 };
00167
00168 class ExpCommand : public Command {
00169 public:
00170   int GetNArgs() { return 1; }
00171   std::string GetName() { return ("exp(" + NameArg[0] + ")"); }
00172   ExpCommand() {}
00173   void SetArg(int i, ReturnType f) { Arg[i] = f; }
00174   void SetNameArg(int i, std::string &n) { NameArg[i] = n; }
00175   ReturnType Execute() {
00176     if (Arg[0] > 200)
00177       return std::exp(200);
00178     if (Arg[0] < -200)
00179       return 0;
00180     return std::exp(Arg[0]);
00181   }
00182
00183 private:
00184   ReturnType Arg[2];
00185   std::string NameArg[2];
00186 };
00187
00188 } // namespace rxdock
00189
00190 #endif //_Command_H_
```

6.70 GPChromosome.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // GP chromosome. Keeps the list of integers that represent a chromosome and
00014 // a list of cells that represent a program.
00015
00016 #ifndef _RBT_GPCHROMOSOME_H
00017 #define _RBT_GPCHROMOSOME_H
00018
00019 #include "rxdock/Commands.h"
00020 #include "rxdock/geneticprogram/Cell.h"
00021 #include "rxdock/geneticprogram/GPTypes.h"
00022
00023 namespace rxdock {
```

```
00024
00025 namespace geneticprogram {
00026
00027 class GPChromosome {
00028 public:
00029   static const std::string _CT;
00031   // Constructors
00033   GPChromosome(int, int, int, int, int, int);
00034   GPChromosome(const GPChromosome &);
00035
00037   // Destructor
00039   virtual ~GPChromosome();
00040
00041   void Clear();
00042   std::ostream &Print(std::ostream &) const;
00043   friend std::ostream &operator«(std::ostream &s, const GPChromosome &p);
00044   friend std::istream &operator»(std::istream &s, GPChromosome &p);
00045   const int &operator[](int idx) const { return chrom[idx]; }
00046   int &operator[](int idx) { return chrom[idx]; }
00047   GPChromosome &operator=(const GPChromosome &c);
00048   int size() const { return chrom.size(); }
00049   CellPtr Cells(int idx) const { return (cells[idx]); }
00050   void SetConstant(ReturnType, int);
00051   void ResetConstant(int);
00052   int GetStartingCell() { return (chrom[chrom.size() - 1]); }
00053   ReturnType GetRCte(int ncell) { return cells[ncell]->GetResult(); }
00054   Commands GetCommand(int ncell) {
00055     if (ncell < nProgramInputs)
00056       throw Error(_WHERE_, "This cell does not have a command");
00057     Commands comm(chrom[(nFunctionsInputs + 1) * (ncell - nProgramInputs) +
00058                         nFunctionsInputs]);
00059     return comm;
00060   }
00061   int GetArgument(int ncell, int narg) {
00062     if (ncell < nProgramInputs)
00063       throw Error(_WHERE_, "This cell does not have arguments");
00064     return (chrom[(nFunctionsInputs + 1) * (ncell - nProgramInputs) + narg]);
00065   }
00066   bool IsProgramInput(int ncell) { return (ncell < nProgramInputs); }
00067
00068 private:
00070   // Private data
00072   std::vector<int> chrom;
00073   CellList cells;
00074   int nProgramInputs, nFunctionsInputs, nProgramOutputs, nRows, nColumns;
00075 };
00076
00077 // Useful typedefs
00078 typedef SmartPtr<GPChromosome> GPChromosomePtr; // Smart pointer
00079
00080 std::ostream &operator«(std::ostream &s, const GPChromosome &p);
00081 std::istream &operator»(std::istream &s, GPChromosome &p);
00082
00083 } // namespace geneticprogram
00084
00085 } // namespace rxdock
00086
00087 #endif //_RBTGPCHROMOSOME_H_
```

6.71 GPFFCHK1.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // GPFFCHK1 Class. GPFFCHK1 represents a fitness function
00014 // valid for the CGP. It is used to train with the CHK1 set.
00015
00016 #ifndef _RBT_GPFFCHK1_H_
00017 #define _RBT_GPFFCHK1_H_
00018
00019 #include "rxdock/Context.h"
00020 #include "rxdock/Rand.h"
00021 #include "rxdock/geneticprogram/GPFitnessFunction.h"
00022 #include "rxdock/geneticprogram/GPGenome.h"
```

```
00023 #include "rxdock/geneticprogram/GPTypes.h"
00024 // #include <cmath>
00025
00026 namespace rxdock {
00027
00028 namespace geneticprogram {
00029
00030 class GPFFCHK1 : public GPFitnessFunction {
00031 public:
00032   static const std::string _CT;
00033   GPFFCHK1(ContextPtr c) : contextp(c) { _RBTOBJECTCOUNTER_CONSTR_(_CT); }
00034   virtual ~GPFFCHK1() { _RBTOBJECTCOUNTER_DESTR_(_CT); }
00035
00036   void ReadTables(std::istream &, ReturnTypeArray &, ReturnTypeArray &);
00037   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00038                           double, bool);
00039   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00040                           bool);
00041
00042 private:
00043   void CreateRandomCtes(int);
00044   std::vector<double> ctes;
00045   CellContextPtr contextp;
00046   GPFFCHK1(); // default constructor disabled
00047   ReturnTypeArray inputTestTable, inputTrainingTable;
00048   ReturnTypeArray sfTestTable, sfTrainingTable;
00049 };
00050
00051 // Useful typedefs
00052 typedef SmartPtr<GPFFCHK1> GPFFCHK1Ptr;        // Smart pointer
00053 typedef std::vector<GPFFCHK1Ptr> GPFFCHK1List; // Vector of smart pointers
00054 typedef GPFFCHK1List::iterator GPFFCHK1ListIter;
00055 typedef GPFFCHK1List::const_iterator GPFFCHK1ListConstIter;
00056
00057 } // namespace geneticprogram
00058
00059 } // namespace rxdock
00060
00061 #endif //_GPFFCHK1
```

6.72 GPFFCrossDock.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // GPFFCrossDock Class. GPFFCrossDock represents a fitness function
00014 // valid for the CGP. It is used to train with the CrossDock set.
00015
00016 #ifndef _RBT_GPFFCROSSDOCK_H_
00017 #define _RBT_GPFFCROSSDOCK_H_
00018
00019 #include "rxdock/Rand.h"
00020 #include "rxdock/geneticprogram/GPFitnessFunction.h"
00021 #include "rxdock/geneticprogram/GPGenome.h"
00022 #include "rxdock/geneticprogram/GPTypes.h"
00023 // #include <cmath>
00024
00025 namespace rxdock {
00026
00027 namespace geneticprogram {
00028
00029 class GPFFCrossDock : public GPFitnessFunction {
00030 public:
00031   static const std::string _CT;
00032
00033   void ReadTables(std::istream &, ReturnTypeArray &, ReturnTypeArray &);
00034   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00035                           double, bool);
00036   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00037                           bool);
00038
00039 private:
00040   void CreateRandomCtes(int);
00041   std::vector<double> ctes;
```

```
00042 };
00043
00044 // Useful typedefs
00045 typedef SmartPtr<GPFFCrossDock> GPFFCrossDockPtr; // Smart pointer
00046 typedef std::vector<GPFFCrossDockPtr>
00047     GPFFCrossDockList; // Vector of smart pointers
00048 typedef GPFFCrossDockList::iterator GPFFCrossDockListIter;
00049 typedef GPFFCrossDockList::const_iterator GPFFCrossDockListConstIter;
00050
00051 } // namespace geneticprogram
00052
00053 } // namespace rxdock
00054
00055 #endif //_GPFFCrossDock
```

6.73 GPFFGold.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // GPFFGold Class. GPFFGold represents a fitness function
00014 // valid for the CGP. It is used to train with the Gold set.
00015
00016 #ifndef _RBT_GPFFGOLD_H_
00017 #define _RBT_GPFFGOLD_H_
00018
00019 #include "rxdock/Rand.h"
00020 #include "rxdock/geneticprogram/GPFitnessFunction.h"
00021 #include "rxdock/geneticprogram/GPGenome.h"
00022 #include "rxdock/geneticprogram/GPTypes.h"
00023 // #include <cmath>
00024
00025 namespace rxdock {
00026
00027 namespace geneticprogram {
00028
00029 class GPFFGold : public GPFitnessFunction {
00030 public:
00031   static const std::string _CT;
00032
00033   void ReadTables(std::istream &, ReturnTypeArray &, ReturnTypeArray &);
00034   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00035                           double, bool);
00036   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00037                           bool);
00038 };
00039
00040 // Useful typedefs
00041 typedef SmartPtr<GPFFGold> GPFFGoldPtr;        // Smart pointer
00042 typedef std::vector<GPFFGoldPtr> GPFFGoldList; // Vector of smart pointers
00043 typedef GPFFGoldList::iterator GPFFGoldListIter;
00044 typedef GPFFGoldList::const_iterator GPFFGoldListConstIter;
00045
00046 } // namespace geneticprogram
00047
00048 } // namespace rxdock
00049
00050 #endif //_GPFFGold
```

6.74 GPFFHSP90.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
```

```
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // GPFFHSP90 Class. GPFFHSP90 represents a fitness function
00014 // valid for the CGP. It is used to train with the HSP90 set.
00015
00016 #ifndef _RBT_GPFFHSP90_H_
00017 #define _RBT_GPFFHSP90_H_
00018
00019 #include "rxdock/Context.h"
00020 #include "rxdock/Rand.h"
00021 #include "rxdock/geneticprogram/GPFitnessFunction.h"
00022 #include "rxdock/geneticprogram/GPGenome.h"
00023 #include "rxdock/geneticprogram/GPTypes.h"
00024
00025 namespace rxdock {
00026
00027 namespace geneticprogram {
00028
00029 class GPFFHSP90 : public GPFitnessFunction {
00030 public:
00031   static const std::string _CT;
00032   GPFFHSP90(ContextPtr c) : contextp(c) { _RBTOBJECTCOUNTER_CONSTR_(_CT); }
00033   virtual ~GPFFHSP90() { _RBTOBJECTCOUNTER_DESTR_(_CT); }
00034
00035   void ReadTables(std::istream &, ReturnTypeArray &, ReturnTypeArray &);
00036   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00037                           double, bool);
00038   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00039                           bool);
00040
00041 private:
00042   void CreateRandomCtes(int);
00043   std::vector<double> ctes;
00044   CellContextPtr contextp;
00045   GPFFHSP90(); // default constructor disabled
00046   ReturnTypeArray inputTestTable, inputTrainingTable;
00047   ReturnTypeArray sfTestTable, sfTrainingTable;
00048 };
00049
00050 // Useful typedefs
00051 typedef SmartPtr<GPFFHSP90> GPFFHSP90Ptr;        // Smart pointer
00052 typedef std::vector<GPFFHSP90Ptr> GPFFHSP90List; // Vector of smart pointers
00053 typedef GPFFHSP90List::iterator GPFFHSP90ListIter;
00054 typedef GPFFHSP90List::const_iterator GPFFHSP90ListConstIter;
00055
00056 } // namespace geneticprogram
00057
00058 } // namespace rxdock
00059
00060 #endif //_GPFFHSP90
```

6.75 GPFFSpike.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // GPFFSpike Class. GPFFSpike represents a fitness function
00014 // valid for the CGP. It is used to train with the Spike set.
00015
00016 #ifndef _RBT_GPFFSPIKE_H_
00017 #define _RBT_GPFFSPIKE_H_
00018
00019 #include "rxdock/Context.h"
00020 #include "rxdock/Rand.h"
00021 #include "rxdock/geneticprogram/GPFitnessFunction.h"
00022 #include "rxdock/geneticprogram/GPGenome.h"
00023 #include "rxdock/geneticprogram/GPTypes.h"
00024 // #include <cmath>
00025
00026 namespace rxdock {
00027
00028 namespace geneticprogram {
00029
```

```
00030 class GPFFSpike : public GPFitnessFunction {
00031 public:
00032   static const std::string _CT;
00033   GPFFSpike(ContextPtr c) : contextp(c) { _RBTOBJECTCOUNTER_CONSTR_(_CT); }
00034   virtual ~GPFFSpike() { _RBTOBJECTCOUNTER_DESTR_(_CT); }
00035
00036   void ReadTables(std::istream &, ReturnTypeArray &, ReturnTypeArray &);
00037   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00038                           double, bool);
00039   double CalculateFitness(GPGenomePtr, ReturnTypeArray &, ReturnTypeArray &,
00040                           bool);
00041
00042 private:
00043   void CreateRandomCtes(int);
00044   std::vector<double> ctes;
00045   CellContextPtr contextp;
00046   GPFFSpike(); // default constructor disabled
00047 };
00048
00049 // Useful typedefs
00050 typedef SmartPtr<GPFFSpike> GPFFSpikePtr;        // Smart pointer
00051 typedef std::vector<GPFFSpikePtr> GPFFSpikeList; // Vector of smart pointers
00052 typedef GPFFSpikeList::iterator GPFFSpikeListIter;
00053 typedef GPFFSpikeList::const_iterator GPFFSpikeListConstIter;
00054
00055 } // namespace geneticprogram
00056
00057 } // namespace rxdock
00058
00059 #endif //_GPFFSpike
```

6.76 GPFitnessFunction.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // GPFitnessFunction Class. GPFitnessFunction represents a fitness function
00014 // valid for the CGP
00015
00016 #ifndef _RBT_GPFITNESSFUNCTION_H_
00017 #define _RBT_GPFITNESSFUNCTION_H_
00018
00019 #include "rxdock/Context.h"
00020 #include "rxdock/Rand.h"
00021 #include "rxdock/geneticprogram/GPGenome.h"
00022 #include "rxdock/geneticprogram/GPTypes.h"
00023 // #include <cmath>
00024
00025 namespace rxdock {
00026
00027 namespace geneticprogram {
00028
00029 class GPFitnessFunction {
00030 public:
00031   static const std::string _CT;
00033   // Constructors
00035   GPFitnessFunction();
00036   GPFitnessFunction(const GPFitnessFunction &);
00037
00038   GPFitnessFunction &operator=(const GPFitnessFunction &);
00039
00041   // Destructor
00043   virtual ~GPFitnessFunction();
00044   virtual void ReadTables(std::istream &, ReturnTypeArray &,
00045                           ReturnTypeArray &) = 0;
00046   double GetObjective() const;
00047   double GetFitness() const;
00048   void SetFitness(double);
00049   virtual double CalculateFitness(GPGenomePtr, ReturnTypeArray &,
00050                                   ReturnTypeArray &, double, bool) = 0;
00051   virtual double CalculateFitness(GPGenomePtr, ReturnTypeArray &,
00052                                   ReturnTypeArray &, bool) = 0;
00053
00054 protected:
```

```
00055   Rand &m_rand;
00056   double objective; // raw score
00057   double fitness;   // scaled score
00058   ReturnTypeArray inputTable, SFTable;
00059
00060 private:
00061 };
00062
00063 // Useful typedefs
00064 typedef SmartPtr<GPFitnessFunction> GPFitnessFunctionPtr; // Smart pointer
00065 typedef std::vector<GPFitnessFunctionPtr>
00066     GPFitnessFunctionList; // Vector of smart pointers
00067 typedef GPFitnessFunctionList::iterator GPFitnessFunctionListIter;
00068 typedef GPFitnessFunctionList::const_iterator GPFitnessFunctionListConstIter;
00069
00070 } // namespace geneticprogram
00071
00072 } // namespace rxdock
00073
00074 #endif //_GPFitnessFunction
```

6.77 GPGenome.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // GPGenome Class. GPGenome represents a genome in the CGP
00014
00015 #ifndef _RBT_GPGENOME_H_
00016 #define _RBT_GPGENOME_H_
00017
00018 #include "rxdock/Rand.h"
00019 #include "rxdock/geneticprogram/GPChromosome.h"
00020 #include "rxdock/geneticprogram/GPTypes.h"
00021 // #include <cmath>
00022
00023 namespace rxdock {
00024
00025 namespace geneticprogram {
00026
00027 class GPGenome {
00028 public:
00029   static const std::string _CT;
00031   // Constructors
00033   GPGenome();
00034   GPGenome(const GPGenome &);
00035   GPGenome(std::string);
00036   GPGenome(std::istream &);
00037
00038   GPGenome &operator=(const GPGenome &);
00039
00041   // Destructor
00043   virtual ~GPGenome();
00044   static void SetStructure(int, int, int, int, int, int, int, int);
00045   static int GetNIP() { return npi; }
00046   static void SetNIP(int n) { npi = n; }
00047   static int GetNIF() { return nfi; }
00048   static int GetNN() { return nn; }
00049   static int GetNO() { return no; }
00050   static int GetNSFI() { return nsfi; }
00051   static void SetNSFI(int n) { nsfi = n; }
00052   GPChromosomePtr GetChrom() const { return (new GPChromosome(*chrom)); }
00053   void Initialise();
00054   void Mutate(double);
00055   void SetFitness(double f) { fitness = f; }
00056   double GetFitness() const { return fitness; }
00057   void UniformCrossover(const GPGenome &, const GPGenome &);
00058   void Crossover(GPGenome &);
00059   friend std::ostream &operator<<(std::ostream &s, const GPGenome &p);
00060   std::ostream &Print(std::ostream &) const;
00061
00062 protected:
00063 private:
00064   void MutateGene(int);
```

```
00065
00066   static int npi, nfi, nsfi, no, nn, nf, nr, nc, l;
00067   // defines structures and constraints of each genome
00068   Rand &m_rand;
00069   double fitness;        // scaled score. For now same than score
00070   GPChromosomePtr chrom; // list of integers that represent a chrom
00071 };
00072
00073 // Useful typedefs
00074 typedef SmartPtr<GPGenome> GPGenomePtr;        // Smart pointer
00075 typedef std::vector<GPGenomePtr> GPGenomeList; // Vector of smart pointers
00076 typedef GPGenomeList::iterator GPGenomeListIter;
00077 typedef GPGenomeList::const_iterator GPGenomeListConstIter;
00078
00079 std::ostream &operator<<(std::ostream &s, const GPGenome &p);
00080
00081 } // namespace geneticprogram
00082
00083 } // namespace rxdock
00084
00085 #endif //_GPGenome_H_
```

6.78 GPParser.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // GPParser Class. Parses a genome from a cartesian GP
00014
00015 #ifndef _RBT_GPPARSER_H_
00016 #define _RBT_GPPARSER_H_
00017
00018 #include "rxdock/geneticprogram/Cell.h"
00019 #include "rxdock/geneticprogram/Command.h"
00020 #include "rxdock/geneticprogram/GPChromosome.h"
00021 #include "rxdock/geneticprogram/GPTypes.h"
00022
00023 namespace rxdock {
00024
00025 namespace geneticprogram {
00026
00027 class GPParser {
00028 public:
00029   static const std::string _CT;
00031   // Constructors
00033   GPParser(int, int, int, int);
00034   GPParser(const GPParser &);
00035
00037   // Destructor
00039   virtual ~GPParser();
00040
00041   void Clear();
00042   std::ostream &Print(std::ostream &) const;
00043   friend std::ostream &operator<<(std::ostream &s, const GPParser &p);
00044   ReturnTypeList Parse(GPChromosomePtr, ReturnTypeList &);
00045   std::string PrintParse(std::istream &, GPChromosomePtr, bool, bool);
00046   bool IsConstantInt(int i) { return (i == CINT); }
00047   bool IsConstantFloat(int i) { return (i == CFLOAT); }
00048
00049 private:
00051   // Private data
00053   ReturnType Eval(GPChromosomePtr, int);
00054   ReturnType ParseOutput(GPChromosomePtr, int);
00055   std::string PrintEval(GPChromosomePtr, int, bool, bool);
00056   std::string PrintParseOutput(GPChromosomePtr, int, bool, bool);
00057   // GPChromosomePtr chrom;
00058   // CellList cells;
00059   CommandList commands;
00060   int nProgramInputs, nFunctionsInputs, nProgramOutputs;
00061   int ADD, SUB, MUL, DIV, IF, CINT, CFLOAT;
00062 };
00063
00064 // Useful typedefs
00065 typedef SmartPtr<GPParser> GPParserPtr; // Smart pointer
```

```
00066
00067 } // namespace geneticprogram
00068
00069 } // namespace rxdock
00070
00071 #endif //_GPParser_H_
```

6.79 GPPopulation.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Population class. Defines a population of CGP genomes that represent
00014 // functions.
00015
00016 #ifndef _RBTGPPOPULATION_H_
00017 #define _RBTGPPOPULATION_H_
00018
00019 #include "rxdock/Error.h"
00020 #include "rxdock/geneticprogram/GPFFGold.h"
00021 #include "rxdock/geneticprogram/GPFitnessFunction.h"
00022 #include "rxdock/geneticprogram/GPGenome.h"
00023
00024 namespace rxdock {
00025
00026 namespace geneticprogram {
00027
00028 class GPPopulation {
00029 public:
00030   static const std::string _CT;
00032   // Constructors
00034   GPPopulation(int, int, GPFitnessFunctionPtr, ReturnTypeArray &,
00035                ReturnTypeArray &);
00036   void SetTrainingSets(ReturnTypeArray &it, ReturnTypeArray sf) {
00037     ittrain = it;
00038     sfttrain = sf;
00039   }
00040
00042   // Destructors
00044   GPPopulation(const GPPopulation &);
00045   virtual ~GPPopulation();
00046   void Initialise(double, bool);
00047   void ScaleFitness();
00048   void Eval(double, bool);
00049   GPGenomePtr Select(std::string) const;
00050
00051   void SelectionUpdate(std::string);
00052
00053   void GAstep(std::string, double, double, double, double, double, bool);
00054   void EPstep(std::string, double, double, double, double, double, bool);
00055   GPGenomePtr Best() const;
00056   std::ostream &Print(std::ostream &) const;
00057   friend std::ostream &operator<<(std::ostream &, const GPPopulation &);
00058   void Swap(SmartPtr<GPPopulation>, int);
00059   void QSort(GPGenomeList &);
00060   void QSort();
00061   void MergePops();
00062   static bool Gen_eq(GPGenome *, GPGenome *);
00063   int GetSize();
00064   int GetNrepl();
00065
00066 private:
00067   GPGenomePtr RwSelect() const;
00068   GPGenomePtr RkSelect() const;
00069   GPGenomePtr TSelect(double) const;
00070   GPGenomeList pop;    // population: array to pointers of GPGenomes
00071   GPGenomeList newpop; // array where the new individuals created are stored
00072   int popsize;         // size of population
00073   int nrepl;           // Number of new individuals that get
00074                        // created at each generation
00075   double total;        // total addition of the score values
00076   double ave;          // average
00077   double stdev;        // standard deviation
00078   double c;            // Sigma Truncation Multiplier
```

```
00079   double *psum;        // stores the partial sums of the fitness values. Used
00080                        // when selecting individuals from the population
00081   Rand &m_rand;        // keep a reference to the singleton
00082                        // random number generator
00083   int bestInd;         // keeps the index of the best individual
00084   GPFitnessFunctionPtr ff;
00085   ReturnTypeArray ittrain, sfttrain;
00086 };
00087 typedef SmartPtr<GPPopulation> GPPopulationPtr;
00088 typedef std::vector<GPPopulationPtr> GPPopulationList;
00089 typedef GPPopulationList::iterator GPPopulationListIter;
00090
00091 // Compare class to compare different populations. Use to find the
00092 // genome with best scoring function between all the populations
00093 class PopCmp {
00094 public:
00095   bool operator()(const GPPopulationPtr &x, const GPPopulationPtr &y) const {
00096     return (x->Best()->GetFitness() < y->Best()->GetFitness());
00097   }
00098 };
00099
00100 } // namespace geneticprogram
00101
00102 } // namespace rxdock
00103
00104 #endif //_GPPopulation_H_
```

6.80 GPTypes.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBT_GPTYPES_H_
00014 #define _RBT_GPTYPES_H_
00015
00016 #include "rxdock/Config.h"
00017
00018 namespace rxdock {
00019
00020 // As the chromosome is a list of integers, I don't think I need a
00021 // new class to define it. If later I find out it is necessary, I
00022 // just need to create a class called GPChromosome with the
00023 // operator[] and size() defined on it
00024 typedef float ReturnType;
00025 typedef SmartPtr<ReturnType> ReturnTypePtr;          // Smart pointer
00026 typedef std::vector<ReturnTypePtr> ReturnTypeList;   // Vector of smart pointers
00027 typedef std::vector<ReturnTypeList> ReturnTypeArray; // Vector of smart
00028                                                      // pointers
00029 typedef ReturnTypeList::iterator ReturnTypeListIter;
00030 typedef ReturnTypeList::const_iterator ReturnTypeListConstIter;
00031
00032 } // namespace rxdock
00033
00034 #endif
```

6.81 Genome.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Genome for roulette wheel selection.
```

```
00014 // Manages a chromosome element, an associated raw score from a scoring function
00015 // and a scaled fitness value for roulette wheel selection
00016 #ifndef _RBT_GENOME_H_
00017 #define _RBT_GENOME_H_
00018
00019 #include "rxdock/ChromElement.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class BaseSF;
00028
00029 class Genome {
00030 public:
00031   static const std::string _CT;
00032   // Sole constructor accepting an existing chromosome.
00033   // A clone of the existing chromosome is stored,
00034   // and is deleted in the Genome destructor
00035   Genome(ChromElement *pChr);
00036   // Copy constructor and Assignment operator.
00037   // The underlying chromosome is cloned during the copy.
00038   Genome(const Genome &);
00039   Genome &operator=(const Genome &);
00040   Genome *clone() const;
00041   virtual ~Genome();
00042
00043   friend void to_json(json &j, const Genome &genome);
00044   friend void from_json(const json &j, Genome &genome);
00045
00046   // Gets the underlying chromosome
00047   ChromElement *GetChrom() const { return m_chrom; }
00048
00049   // Sets the raw score from the scoring function.
00050   // Model coordinates are updated to reflect the current chromosome element
00051   // values. Scores are negated so that higher positive scores are better. pSF
00052   // is a pointer to a scoring function object. If pSF is null, a zero score is
00053   // set.
00054   void SetScore(BaseSF *pSF);
00055   // Gets the stored raw score (without re-evaluation of the scoring function).
00056   double GetScore() const { return m_score; }
00057
00058   // ROULETTE WHEEL SELECTION METHODS
00059   // The following three methods should be used only in the context of
00060   // genome lists sorted into ascending order by score.
00061   // See Population::EvaluateRWFitness for example of use.
00062   //
00063   // Sets the fitness value used for roulette wheel selection
00064   // sigmaOffset is the calculated offset for scaling using sigma truncation
00065   //(Goldberg page 124)
00066   // partialSum is the partial sum of all previous fitness values in the sorted
00067   // genome list.
00068   double SetRWFitness(double sigmaOffset, double partialSum);
00069   // Normalises the roulette wheel fitness values to lie between 0 and 1
00070   // total is the total sum of absolute fitness values returned by the last call
00071   // to SetRWFitness
00072   void NormaliseRWFitness(double total);
00073   // Gets the fitness value for roulette wheel selection
00074   double GetRWFitness() const { return m_RWFitness; }
00075
00076   // Tests equality based on chromosome element values.
00077   // Does not take score into account.
00078   bool Equals(const Genome &g, double threshold) const {
00079     return m_chrom->Equals(*(g.m_chrom), threshold);
00080   }
00081   friend bool operator==(const Genome &g1, const Genome &g2) {
00082     return g1.Equals(g2, ChromElement::_THRESHOLD);
00083   }
00084
00085   void Print(std::ostream &) const;
00086   friend std::ostream &operator<<(std::ostream &s, const Genome &g);
00087
00088 private:
00089   Genome(); // Default constructor disabled
00090
00091 private:
00093   // Private data
00095   ChromElement *m_chrom;
00096   double m_score;     // raw value of the scoring function
00097   double m_RWFitness; // scaled value of the raw score suitable for use
00098                       // with roulette wheel selection
00099 };
00100
00101 void to_json(json &j, const Genome &genome);
00102 void from_json(const json &j, Genome &genome);
```

```
00103
00104 // Useful typedefs
00105 typedef SmartPtr<Genome> GenomePtr;        // Smart pointer
00106 typedef std::vector<GenomePtr> GenomeList; // Vector of smart pointers
00107 typedef GenomeList::iterator GenomeListIter;
00108 typedef GenomeList::const_iterator GenomeListConstIter;
00109
00110 // Compare class to compare different genomes inside a population
00111 // This is used to sort the genomes in a population depending
00112 // of the value of their scoring function.
00113 class GenomeCmp_Score {
00114 public:
00115   bool operator()(const Genome *pG1, const Genome *pG2) const {
00116     return (pG1->GetScore() > pG2->GetScore());
00117   }
00118 };
00119
00120 // STL binary predicate to compare two Genome* for equality.
00121 // Tests equality based on chromosome element values.
00122 // Does not take score into account.
00123 // Uses operator== which ultimately calls ChromElement::Equals()
00124 class isGenome_eq : public std::function<bool(Genome *, Genome *)> {
00125 private:
00126   double m_threshold; // equality threshold
00127 public:
00128   explicit isGenome_eq(double threshold) : m_threshold(threshold) {}
00129   bool operator()(const Genome *pG1, const Genome *pG2) const {
00130     return pG1->Equals(*pG2, m_threshold);
00131   }
00132 };
00133
00134 } // namespace rxdock
00135
00136 #endif //_RBT_GENOME_H_
```

6.82 InteractionGrid.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Grid for indexing interactions involving more than 2 atoms
00014
00015 #ifndef _RBTINTERACTIONGRID_H_
00016 #define _RBTINTERACTIONGRID_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/BaseGrid.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // simple container for up to 3 atoms, to hold one half of an interaction
00028 // i.e. receptor atoms or ligand atoms.
00029 class InteractionCenter {
00030 public:
00031   enum eLP { NONE, PLANE, LONEPAIR };
00032   InteractionCenter(Atom *pAtom1 = nullptr, Atom *pAtom2 = nullptr,
00033                     Atom *pAtom3 = nullptr, eLP LP = NONE)
00034       : m_pAtom1(pAtom1), m_pAtom2(pAtom2), m_pAtom3(pAtom3), m_LP(LP) {
00035     _RBTOBJECTCOUNTER_CONSTR_("InteractionCenter");
00036   }
00037   ~InteractionCenter() { _RBTOBJECTCOUNTER_DESTR_("InteractionCenter"); }
00038
00039   friend void to_json(json &j, const InteractionCenter &interactionCenter);
00040   friend void from_json(const json &j, InteractionCenter &interactionCenter);
00041
00042   Atom *GetAtom1Ptr() const { return m_pAtom1; }
00043   Atom *GetAtom2Ptr() const { return m_pAtom2; }
00044   Atom *GetAtom3Ptr() const { return m_pAtom3; }
00045   eLP LP() const { return m_LP; }
00046   // Returns list of constituent atoms
```

```
00047   //(deconvolutes pseudoatoms into their constituent Atom* lists)
00048   AtomRList GetAtomList() const;
00049   bool isSelected() const;
00050
00051 private:
00052   // Could be a useful general function
00053   // If pAtom is a pseudo atom, then pushes all the constituent atoms onto list
00054   // else, push pAtom onto the list
00055   void AccumulateAtomList(const Atom *pAtom, AtomRList &atomList) const;
00056   Atom *m_pAtom1;
00057   Atom *m_pAtom2;
00058   Atom *m_pAtom3;
00059   eLP m_LP;
00060 };
00061
00062 void to_json(json &j, const InteractionCenter &interactionCenter);
00063 void from_json(const json &j, InteractionCenter &interactionCenter);
00064
00065 typedef std::vector<InteractionCenter *>
00066     InteractionCenterList; // Vector of regular pointers
00067 typedef InteractionCenterList::iterator InteractionCenterListIter;
00068 typedef InteractionCenterList::const_iterator InteractionCenterListConstIter;
00069
00070 // Less than operator for sorting InteractionCenter* by pointer value
00071 class InteractionCenterCmp {
00072 public:
00073   bool operator()(const InteractionCenter *pIC1,
00074                   const InteractionCenter *pIC2) const {
00075     return pIC1 < pIC2;
00076   }
00077 };
00078 // Is interaction center selected ?
00079 class isInteractionCenterSelected
00080     : public std::function<bool(InteractionCenter *)> {
00081 public:
00082   explicit isInteractionCenterSelected() {}
00083   bool operator()(const InteractionCenter *pIC) const {
00084     return pIC->isSelected();
00085   }
00086 };
00087
00088 // Is the distance between interaction centers less than a given value ?
00089 // Function checks d**2 to save performing a sqrt
00090 class isInteractionD_lt : public std::function<bool(InteractionCenter *)> {
00091   double d_sq;
00092   Atom *a;
00093
00094 public:
00095   explicit isInteractionD_lt(const InteractionCenter *pIC1, double dd)
00096       : d_sq(dd * dd), a(pIC1->GetAtom1Ptr()) {}
00097   bool operator()(const InteractionCenter *pIC2) const {
00098     return Length2(pIC2->GetAtom1Ptr()->GetCoords(), a->GetCoords()) < d_sq;
00099   }
00100 };
00101
00102 // Select/deselect the interaction center (selects all constituent atoms)
00103 class SelectInteractionCenter {
00104   bool b;
00105
00106 public:
00107   explicit SelectInteractionCenter(bool bb) : b(bb) {}
00108   void operator()(InteractionCenter *pIC);
00109 };
00110
00111 // A map of interaction centers indexed by unsigned int
00112 // Used to store the receptor atom lists at each grid point
00113 // DM 11 Jul 2000 - use map of regular Atom* list (not AtomPtr smart
00114 // pointer list)
00115
00116 // DM 3 Nov 2000 - replace map by vector for faster lookup
00117 // typedef std::map<UInt,InteractionCenterList> InteractionListMap;
00118 typedef std::vector<InteractionCenterList> InteractionListMap;
00119
00120 typedef InteractionListMap::iterator InteractionListMapIter;
00121 typedef InteractionListMap::const_iterator InteractionListMapConstIter;
00122
00123 class InteractionGrid : public BaseGrid {
00124 public:
00125   // Class type string
00126   static const std::string _CT;
00128   // Constructors/destructors
00129   // Construct a NXxNYxNZ grid running from gridMin at gridStep resolution
00130   InteractionGrid(const Coord &gridMin, const Coord &gridStep, unsigned int NX,
00131                   unsigned int NY, unsigned int NZ, unsigned int NPad = 0);
00132
00133   // Constructor reading all params from binary stream
00134   InteractionGrid(json j);
```

```
00135
00136   ~InteractionGrid(); // Default destructor
00137
00138   /*friend void to_json(json &j, const InteractionGrid &interactionGrid);
00139   friend void from_json(const json &j, InteractionGrid &interactionGrid);*/
00140
00141   // Copy constructor
00142   InteractionGrid(const InteractionGrid &);
00143   // Copy constructor taking base class argument
00144   InteractionGrid(const BaseGrid &);
00145   // Copy assignment
00146   InteractionGrid &operator=(const InteractionGrid &);
00147   // Copy assignment taking base class argument
00148   InteractionGrid &operator=(const BaseGrid &);
00149
00151   // Virtual functions for reading/writing grid data to streams in
00152   // text and binary format
00153   // Subclasses should provide their own private OwnPrint
00154   // method to handle subclass data members, and override the public
00155   // Print method
00156   virtual void Print(std::ostream &ostr) const; // Text output
00157
00159   // Public methods
00161
00163   // Get attribute functions
00165   const InteractionCenterList &GetInteractionList(unsigned int iXYZ) const;
00166   const InteractionCenterList &GetInteractionList(const Coord &c) const;
00167
00169   // Set attribute functions
00171   void SetInteractionLists(InteractionCenter *pIntn, double radius);
00172   void ClearInteractionLists();
00173   void UniqueInteractionLists();
00174
00175 protected:
00177   // Protected methods
00179   // Protected method for writing data members for this class to text stream
00180   void OwnPrint(std::ostream &ostr) const;
00181
00182 private:
00184   // Private methods
00186   InteractionGrid(); // Disable default constructor
00187
00188   // Helper function called by copy constructor and assignment operator
00189   void CopyGrid(const InteractionGrid &);
00190   // DM 3 Nov 2000 - create InteractionListMap of the appropriate size
00191   void CreateMap();
00192
00193 protected:
00195   // Protected data
00197
00198 private:
00200   // Private data
00202   InteractionListMap m_intnMap; // Used to store the interaction center lists
00203                                 // at each grid point
00204   const InteractionCenterList m_emptyList; // Dummy list used by GetAtomList
00205 };
00206 /*
00207 void to_json(json &j, const InteractionGrid &interactionGrid);
00208 void from_json(const json &j, InteractionGrid &interactionGrid);*/
00209
00210 // Useful typedefs
00211 typedef SmartPtr<InteractionGrid> InteractionGridPtr; // Smart pointer
00212
00213 } // namespace rxdock
00214
00215 #endif //_RBTINTERACTIONGRID_H_
```

6.83 InteractionTemplate.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTINTERACTIONTEMPLATE_H_
00014 #define _RBTINTERACTIONTEMPLATE_H_
```

```
00015 #include "rxdock/BaseGrid.h"
00016 #include "rxdock/Config.h"
00017
00018 #include <loguru.hpp>
00019
00020 namespace rxdock {
00021
00022 template <class T> class InteractionNode {
00023   T property;
00024
00025 public:
00026   InteractionNode() { _RBTOBJECTCOUNTER_CONSTR_("InteractionNode"); }
00027   InteractionNode(T aProperty) {
00028     _RBTOBJECTCOUNTER_CONSTR_("InteractionNode");
00029     property = aProperty;
00030   }
00031   ~InteractionNode() { _RBTOBJECTCOUNTER_DESTR_("InteractionNode"); }
00032
00033   T GetProperty() const { return property; }
00034   void SetProperty(T aProperty) { property = aProperty; }
00035 };
00036
00037 template <typename T> // instead InteractionCenterList
00038 class InteractionNodeList : public std::vector<InteractionNode<T> *> {
00039 }; // std:vector ctors
00040
00041 template <typename T> // instead InteractionListMap
00042 class InteractionNodeListMap : public std::vector<InteractionNodeList<T» {
00043 }; // std:vector ctors
00044
00045 template <typename T>
00046 class InteractionNodeListMapIter : public InteractionNodeListMap<T>::iterator {
00047 };
00048
00049 template <typename T>
00050 class InteractionNodeListMapConstIter
00051     : public InteractionNodeListMap<T>::const_iterator {};
00052 // Most of the stuff is copied shamelessly from InteractionGrid.[cxx,h]
00053 template <class T> class InteractionGridTemplate : public BaseGrid {
00054 public:
00055   static std::string _CT;
00056
00057   InteractionGridTemplate(const Coord &gridMin, const Coord &gridStep,
00058                           unsigned int NX, unsigned int NY, unsigned int NZ,
00059                           unsigned int NPad = 0) {
00060     _CT = "InteractionGridTemplate";
00061     CreateMap();
00062     _RBTOBJECTCOUNTER_CONSTR_("InteractionGridTemplate");
00063   }
00064
00065   // Constructor reading params from binary stream
00066   InteractionGridTemplate(std::istream &istr) {
00067     _CT = "InteractionGridTemplate";
00068     CreateMap();
00069     _RBTOBJECTCOUNTER_CONSTR_("InteractionGridTemplate");
00070   }
00071
00072   ~InteractionGridTemplate() {
00073     ClearInteractionLists();
00074     _RBTOBJECTCOUNTER_DESTR_("InteractionGridTemplate");
00075   }
00076
00077   // Copy constructor
00078   InteractionGridTemplate(const InteractionGridTemplate &aGrid)
00079       : BaseGrid(aGrid) {
00080     CreateMap();
00081     CopyGrid(aGrid);
00082     _RBTOBJECTCOUNTER_COPYCONSTR_("InteractionGridTemplate");
00083   }
00084
00085   // Copy constructor taking base class argument
00086   InteractionGridTemplate(const BaseGrid &) {
00087     CreateMap();
00088     _RBTOBJECTCOUNTER_COPYCONSTR_("InteractionGridTemplate");
00089   }
00090
00091   // Copy assignment
00092   InteractionGridTemplate &operator=(const InteractionGridTemplate &aGrid) {
00093     if (this != &aGrid) {
00094       ClearInteractionLists();
00095       BaseGrid::Operator = (aGrid);
00096       CopyGrid(aGrid);
00097     }
00098     return *this;
00099   }
00100
00101   // Copy assignment taking base class argument
```

```
00102   InteractionGridTemplate &operator=(const BaseGrid &aGrid) {
00103     if (this != &aGrid) {
00104       ClearInteractionLists();
00105       BaseGrid::Operator = (aGrid);
00106     }
00107     return *this;
00108   }
00109
00110   virtual void Print(std::ostream &ostr) const { // Text output
00111     BaseGrid::Print(ostr);
00112     OwnPrint(ostr);
00113   }
00114
00115   const InteractionNodeList<T> &GetInteractionList(unsigned int iXYZ) const {
00116     if (isValid(iXYZ)) {
00117       return m_intnMap[iXYZ];
00118     } else {
00119       return m_emptyList;
00120     }
00121   }
00122
00123   const InteractionNodeList<T> &GetInteractionList(const Coord &c) const {
00124     if (isValid(c)) {
00125       return m_intnMap[GetIXYZ(c)];
00126     } else {
00127       LOG_F(1, "InteractionGridTemplate::GetInteractionList: {} is off grid",
00128             c);
00129       return m_emptyList;
00130     }
00131   }
00132
00133   void SetInteractionLists(T *pIntn, double radius) {
00134     Atom *pAtom = pIntn->GetProperty().atom;
00135     if (nullptr == pAtom)
00136       return;
00137     const Coord &c = pAtom->GetCoords();
00138     std::vector<unsigned int> sphereIndices;
00139     GetSphereIndices(c, radius, sphereIndices);
00140     for (std::vector<unsigned int>::const_iterator sphereIter =
00141              sphereIndices.begin();
00142          sphereIter != sphereIndices.end(); sphereIter++) {
00143       m_intnMap[*sphereIter].push_back(pIntn);
00144     }
00145   }
00146
00147   void ClearInteractionLists() {
00148     for (InteractionNodeListMap<T>::iterator iter = m_intnMap.begin();
00149          iter != m_intnMap.end(); iter++) {
00150       (*iter).clear();
00151     }
00152   }
00153
00154 protected:
00155   void OwnPrint(std::ostream &ostr) const {
00156     ostr « std::endl « "Class\t" « _CT « std::endl;
00157     ostr « "No. of entries in the map: " « m_intnMap.size() « std::endl;
00158   }
00159
00160 private:
00161   InteractionGridTemplate();
00162
00163   void CopyGrid(const InteractionGridTemplate<T> &aGrid) {
00164     m_intnMap = aGrid.m_intnMap;
00165   }
00166
00167   void CreateMap() { m_intnMap = InteractionNodeListMap<T>(GetN()); }
00168
00169   InteractionNodeListMap<T> m_intnMap;      // Used to store the interaction
00170                                             // center lists at each grid point
00171   const InteractionNodeList<T> m_emptyList; // Dummy list used by GetAtomList
00172 };
00173
00174 template <typename T> // instead InteractionGridPtr
00175 class InteractionGridTemplatePtr : public SmartPtr<InteractionGridTemplate<T» {
00176 };
00177
00178 } // namespace rxdock
00179
00180 #endif //  _RBTINTERACTIONTEMPLATE_H_
```

6.84 LigandError.h

```
00001 /**********************************************************************
```

```
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // rxdock ligand exceptions, the model is correct, but the ligand is not
00014 // appropiate for this case. (e.g. doesn't comply with all the pharmacophore
00015 // constraints)
00016
00017 #ifndef _RBTLIGANDERROR_H_
00018 #define _RBTLIGANDERROR_H_
00019
00020 #include "rxdock/Error.h"
00021
00022 namespace rxdock {
00023
00024 const std::string IDS_LIGAND_ERROR = "RBT_LIGAND_ERROR";
00025
00026 // Unspecified ligand error
00027 class LigandError : public Error {
00028 public:
00029   LigandError(const std::string &strFile, int nLine,
00030               const std::string &strMessage = "")
00031       : Error(IDS_LIGAND_ERROR, strFile, nLine, strMessage) {}
00032   // Protected constructor to allow derived ligand error classes to set error
00033   // name
00034 protected:
00035   LigandError(const std::string &strName, const std::string &strFile, int nLine,
00036               const std::string &strMessage = "")
00037       : Error(strName, strFile, nLine, strMessage) {}
00038 };
00039
00040 } // namespace rxdock
00041
00042 #endif //_RBTLIGANDERROR_H_
```

6.85 LigandFlexData.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Manages flexibility data for ligands
00014 #ifndef _RBTLIGANDFLEXDATA_
00015 #define _RBTLIGANDFLEXDATA_
00016
00017 #include "rxdock/FlexData.h"
00018
00019 namespace rxdock {
00020
00021 class LigandFlexData : public FlexData {
00022 public:
00023   // Translation mutation step length (A)
00024   static const std::string &_TRANS_STEP;
00025   // Whole-body orientation mutation step length (deg)
00026   static const std::string &_ROT_STEP;
00027   // Dihedral angle mutation step length (deg)
00028   static const std::string &_DIHEDRAL_STEP;
00029   // Enum for translation mode (0=FIXED, 1=TETHERED, 2=FREE)
00030   static const std::string &_TRANS_MODE;
00031   // Enum for rotation mode (0=FIXED, 1=TETHERED, 2=FREE)
00032   static const std::string &_ROT_MODE;
00033   // Enum for dihedral mode (0=FIXED, 1=TETHERED, 2=FREE)
00034   static const std::string &_DIHEDRAL_MODE;
00035   // Max allowed distance from initial COM (for tethered translation only)
00036   static const std::string &_MAX_TRANS;
00037   // Max allowed rotation from initial orientation
00038   //(for tethered rotation only)
00039   static const std::string &_MAX_ROT;
```

```
00040   // Max allowed dihedral rotation from initial dihedrals
00041   //(for tethered dihedrals only)
00042   static const std::string &_MAX_DIHEDRAL;
00043
00044   RBTDLL_EXPORT static const std::string &GetTransMode();
00045   RBTDLL_EXPORT static const std::string &GetRotMode();
00046
00047   RBTDLL_EXPORT LigandFlexData(DockingSite *pDockSite);
00048   virtual void Accept(FlexDataVisitor &v) { v.VisitLigandFlexData(this); }
00049
00050 private:
00051   LigandFlexData();
00052 };
00053
00054 } // namespace rxdock
00055
00056 #endif //_RBTLIGANDFLEXDATA_
```

6.86 LigandSiteMapper.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Reference ligand site mapper function class
00014
00015 #ifndef _RBTLIGANDSITEMAPPER_H_
00016 #define _RBTLIGANDSITEMAPPER_H_
00017
00018 #include "rxdock/SiteMapper.h"
00019
00020 namespace rxdock {
00021
00022 class LigandSiteMapper : public SiteMapper {
00023 public:
00024   // Static data member for class type
00025   static const std::string _CT;
00026   // Parameter names
00027   static const std::string _REF_MOL;
00028   static const std::string _VOL_INCR;
00029   static const std::string _SMALL_SPHERE;
00030   static const std::string _GRIDSTEP;
00031   static const std::string _RADIUS;
00032   static const std::string _MIN_VOLUME;
00033   static const std::string _MAX_CAVITIES;
00034
00035   LigandSiteMapper(const std::string &strName = "LIGAND_MAPPER");
00036   virtual ~LigandSiteMapper();
00037
00038   // Override SiteMapper pure virtual
00039   // This is the function which actually does the mapping
00040   virtual CavityList operator()();
00041 };
00042
00043 // Useful typedefs
00044 typedef SmartPtr<LigandSiteMapper> LigandSiteMapperPtr; // Smart pointer
00045
00046 } // namespace rxdock
00047
00048 #endif //_RBTLIGANDSITEMAPPER_H_
```

6.87 MdlFileSink.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
```

```
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Class for writing Model's to MDL SD and MOL files
00014
00015 #ifndef _RBTMDLFILESINK_H_
00016 #define _RBTMDLFILESINK_H_
00017
00018 #include "rxdock/BaseMolecularFileSink.h"
00019 #include "rxdock/ElementFileSource.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // DM 19 June 2006
00028 // Map to keep track of logical atom IDs as they are rendered to file
00029 // This avoids the need for the atoms in the Model to have consecutive
00030 // atom IDs. The rendering of enabled solvent models is now supported
00031 // correctly.
00032 // Key = Atom* pointer
00033 // Value = logical atom ID as rendered to file
00034 // Logical atom IDs are consecutive for all atoms rendered
00035 // The actual atom ID (Atom::GetAtomId()) is unchanged
00036 typedef std::map<Atom *, unsigned int, AtomPtrCmp_Ptr> AtomIdMap;
00037
00038 class MdlFileSink : public BaseMolecularFileSink {
00039 public:
00041   // Constructors/destructors
00042   RBTDLL_EXPORT MdlFileSink(const std::string &fileName, ModelPtr spModel);
00043
00044   virtual ~MdlFileSink(); // Default destructor
00045
00046   friend void to_json(json &j, const MdlFileSink &mdlFileSink);
00047   friend void from_json(const json &j, MdlFileSink &mdlFileSink);
00048
00050   // Public methods
00052   //
00053   //
00055   // Override public methods from BaseFileSink
00056   virtual void Render();
00057
00058 protected:
00060   // Protected methods
00062
00063 private:
00065   // Private methods
00067   void RenderAtomList(const AtomList &atomList);
00068   void RenderBondList(const BondList &bondList);
00069   void RenderData(const StringVariantMap &dataMap);
00070
00071   MdlFileSink();                    // Disable default constructor
00072   MdlFileSink(const MdlFileSink &); // Copy constructor disabled by
00073                                     // default
00074   MdlFileSink &
00075   operator=(const MdlFileSink &); // Copy assignment disabled by default
00076
00077 protected:
00079   // Protected data
00081
00082 private:
00084   // Private data
00086   ElementFileSourcePtr m_spElementData;
00087   // DM 27 Apr 1999 - default behaviour is for the first Render to overwrite any
00088   // existing file then subsequent Renders to append.
00089   bool m_bFirstRender;
00090   AtomIdMap m_atomIdMap; // Keep track of logical atom IDs as rendered to
00091                          // file
00092 };
00093
00094 void to_json(json &j, const MdlFileSink &mdlFileSink);
00095 void from_json(const json &j, MdlFileSink &mdlFileSink);
00096
00097 // Useful typedefs
00098 typedef SmartPtr<MdlFileSink> MdlFileSinkPtr; // Smart pointer
00099
00100 void RenumberScaffold(AtomList &atomList);
00101
00102 } // namespace rxdock
00103
00104 #endif //_RBTMDLFILESINK_H_
```

6.88 MdlFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Handles retrieval of molecular info from MDL Mol and SD files.
00014
00015 #ifndef _RBTMDLFILESOURCE_H_
00016 #define _RBTMDLFILESOURCE_H_
00017
00018 #include "rxdock/BaseMolecularFileSource.h"
00019 #include "rxdock/ElementFileSource.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 const std::string IDS_MDL_RECDELIM = "$$$$";
00028
00029 class MdlFileSource : public BaseMolecularFileSource {
00030 public:
00031   // Constructors
00032   // MdlFileSource(const char* fileName);
00033   RBTDLL_EXPORT MdlFileSource(const std::string &fileName,
00034                               bool bPosIonisable = true,
00035                               bool bNegIonisable = true,
00036                               bool bImplHydrogens = true);
00037
00038   // Default destructor
00039   virtual ~MdlFileSource();
00040
00041   friend void to_json(json &j, const MdlFileSource &mdlFileSrc);
00042   friend void from_json(const json &j, MdlFileSource &mdlFileSrc);
00043
00045   // Override public methods from BaseMolecularDataSource
00046   virtual bool isTitleListSupported() { return true; }
00047   virtual bool isAtomListSupported() { return true; }
00048   virtual bool isCoordinatesSupported() { return true; }
00049   virtual bool isBondListSupported() { return true; }
00050   // DM 12 May 1999 - support for data records
00051   virtual bool isDataSupported() { return true; }
00052
00053 protected:
00054   // Pure virtual in BaseFileSource - needs to be defined here
00055   virtual void Parse();
00056
00057 private:
00058   // Private methods
00059   MdlFileSource();                      // Disable default constructor
00060   MdlFileSource(const MdlFileSource &); // Copy constructor disabled by default
00061   MdlFileSource &
00062   operator=(const MdlFileSource &); // Copy assignment disabled by default
00063
00064   // Sets up all the atomic attributes that are not explicitly stored in the MDL
00065   // file
00066   void SetupAtomParams();
00067
00068   // Helper functions for SetupAtomParams()
00069
00070   // Sets hybridisation state and checks for Nsp3 next to sp2 special case
00071   void SetupHybridState();
00072
00073   // Checks for valency within range
00074   // Adds implicit hydrogens to C,N,S
00075   // Defines vdW radius, correcting for extended atoms and H-bond donor
00076   // hydrogens Defines formal "force-field" type string
00077   void SetupTheRest();
00078
00079   void SetupPosIonisableGroups();
00080   void SetupNegIonisableGroups();
00081
00082   void AddHydrogen(AtomPtr spAtom);
00083
00084   // SetupIonicGroups sets the "group charge" attribute of the atoms. The idea
00085   // is that the "group charge" is file-format independent, and eliminates the
00086   // need for the scoring function setup to have any knowledge of the input file
```

```
00087   // format. For example, PSF files represent COO- as OC-C-OC (both oxygens
00088   // charged) whereas SD files represent it as the formal resonance structure
00089   // O=C-O-. For rDock we actually want the negative charge on the central
00090   // carbon!
00091   //
00092   // It is envisaged each molecular file source will have a version of
00093   // SetupIonicGroups to convert from the native representation to the rDock
00094   // representation. The FormalCharge and PartialCharge attributes should be
00095   // left untouched to allow the model to be rewritten back in the same format.
00096   void SetupIonicGroups();
00097   void SetupNSP3Plus(); // Helper function to set up protonated amines
00098   void
00099   SetupNSP2Plus(); // Helper function to set up guanidines, imidazoles, amidines
00100   void SetupOTRIMinus(); // Helper function to set up deprotonated acids
00101
00102   // DM 4 June 1999 - Search for the separate fragments in the record
00103   // and set distinct segment names for each one
00104   void SetupSegmentNames();
00105
00106   // DM 2 Aug 1999 - remove all non-polar hydrogens
00107   void RemoveNonPolarHydrogens();
00108
00109   // DM 8 Feb 2000 - setup atom and bond cyclic flags (previously in Model)
00110   // Set the atom and bond cyclic flags for all atoms and bonds in the source
00111   void SetCyclicFlags();
00112
00113   // Private data
00114   ElementFileSourcePtr
00115       m_spElementData;   // Elemental data source for vdW radii etc
00116   bool m_bPosIonisable;  // If true, protonate Nsp3, and Nsp2 in imidazole and
00117                          // guanidinium
00118   bool m_bNegIonisable;  // If true, deprotonate Otri-Hsp3
00119   bool m_bImplHydrogens; // If true, non-hydrogen bonding hydrogens are removed
00120 };
00121
00122 void to_json(json &j, const MdlFileSource &mdlFileSrc);
00123 void from_json(const json &j, MdlFileSource &mdlFileSrc);
00124
00125 // useful typedefs
00126 typedef SmartPtr<MdlFileSource> MdlFileSourcePtr; // Smart pointer
00127
00128 } // namespace rxdock
00129
00130 #endif //_RBTMDLFILESOURCE_H_
```

6.89 Model.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTMODEL_H_
00014 #define _RBTMODEL_H_
00015
00016 #include "rxdock/Atom.h"
00017 #include "rxdock/Bond.h"
00018 #include "rxdock/Config.h"
00019 #include "rxdock/ModelMutator.h"
00020 #include "rxdock/PrincipalAxes.h"
00021 #include "rxdock/PseudoAtom.h"
00022 #include "rxdock/Variant.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
00030 class BaseMolecularFileSource;
00031 class FlexData;
00032 class ChromElement;
00033
00034 class Model {
00035 public:
00037   // Constructors
```

```
00039   // Read topology from MolecularFileSource, and coordinates if supported by
00040   // source
00041   RBTDLL_EXPORT Model(BaseMolecularFileSource *pMolSource);
00042
00043   //(Fairly) temporary constructor taking arbitrary atom and bond lists
00044   // Use with caution
00045   RBTDLL_EXPORT Model(AtomList &atomList, BondList &bondList);
00046
00047   Model(json j);
00048
00049   // Default destructor
00050   virtual ~Model();
00051
00052   friend void to_json(json &j, const Model &model);
00053   friend void from_json(const json &j, Model &model);
00054
00056   // Friend functions/classes
00058   // DM 30 Jul 2001 - allow access to atom list by helper class for flexible
00059   // models
00060   friend class ModelMutator;
00061
00063   // Public accessor functions
00065
00066   // Only Get functions are currently provided
00067   // Models are designed to be CreateOnce (from file), ReadMany
00068
00069   // Model name
00070   std::string GetName() const { return m_strName; }
00071
00072   // Titles
00073   int GetNumTitles() const { return m_titleList.size(); }
00074   std::vector<std::string> GetTitleList() const { return m_titleList; }
00075   void SetTitle(std::size_t index, const std::string &strTitle) {
00076     m_titleList[index] = strTitle;
00077   }
00078
00079   // Atoms
00080   int GetNumAtoms() const { return m_atomList.size(); }
00081   AtomList GetAtomList() const { return m_atomList; }
00082
00083   // Bonds
00084   int GetNumBonds() const { return m_bondList.size(); }
00085   BondList GetBondList() const { return m_bondList; }
00086
00087   // Segments
00088   int GetNumSegments() const { return m_segmentMap.size(); }
00089   SegmentMap GetSegmentMap() const { return m_segmentMap; }
00090
00091   // Rings
00092   int GetNumRings() const { return m_ringList.size(); }
00093   AtomListList GetRingAtomLists() const { return m_ringList; }
00094
00095   // DM 12 May 1999 - support for associated model data (e.g. SD file, or
00096   // generated by rbdock etc) Get number of data fields
00097   int GetNumData() const { return m_dataMap.size(); }
00098   // Get list of field names as string list
00099   std::vector<std::string> GetDataFieldList() const;
00100   // Get all data as map of key=field name, value=variant (double,string or
00101   // string list)
00102   StringVariantMap GetDataMap() const { return m_dataMap; }
00103   // Query as to whether a particular data field name is present
00104   bool isDataFieldPresent(const std::string &strDataField) const;
00105   // Get a particular data value
00106   RBTDLL_EXPORT Variant GetDataValue(const std::string &strDataField) const;
00107   // Set a data value (replaces existing value if field name already exists)
00108   RBTDLL_EXPORT void SetDataValue(const std::string &strDataField,
00109                                   const Variant &dataValue);
00110   // Removes a data field completely from the data map
00111   void ClearDataField(const std::string &strDataField);
00112   // Removes all data fields starting with a given prefix from the data map
00113   RBTDLL_EXPORT void ClearAllDataFields(const std::string &strDataFieldPrefix);
00114   // Removes all data fields from the data map
00115   void ClearAllDataFields();
00116
00117   // DM 11 Jul 2000 - pseudoatom handling
00118   PseudoAtomPtr AddPseudoAtom(const AtomList &atomList);
00119   void ClearPseudoAtoms();
00120   void UpdatePseudoAtoms(); // Updates pseudoatom coords
00121   unsigned int GetNumPseudoAtoms() const;
00122   PseudoAtomList GetPseudoAtomList() const;
00123
00124   // DM 1 Jul 2002 - tethered atom handling
00125   unsigned int GetNumTetheredAtoms() const;
00126   AtomList GetTetheredAtomList() const;
00127
00128   // DM 7 June 2006 - occupancy handling
00129   // The model occupancy is represented in two ways:
```

```
00130   // 1) Occupancy variable (0.0 to 1.0)
00131   // 2) Enabled state (true/false)
00132   // If occupancy >= threshold, enabled=true
00133   // If occupancy < threshold, enabled=false
00134   // The occupancy variable is managed by the occupancy chromosome element
00135   // Scoring functions read the enabled state to determined if a model is
00136   // considered as present or absent
00137   double GetOccupancy() const { return m_occupancy; }
00138   bool GetEnabled() const { return m_enabled; }
00139   // Sets the occupancy and enabled state simultaneously
00140   RBTDLL_EXPORT void SetOccupancy(double occupancy, double threshold = 0.5);
00141
00143   // Public methods
00145
00146   // Update coords from a data source
00147   void UpdateCoords(BaseMolecularFileSource *pMolSource);
00148
00149   // DM 07 Jan 1999
00150   // Translate molecule by the given vector
00151   void Translate(const Vector &vector);
00152   // Rotate molecule around the given axis (through the center of mass) by theta
00153   // degrees
00154   void Rotate(const Vector &axis, double thetaDeg);
00155   // DM 09 Feb 1999 - Rotate molecule around the given axis (through the given
00156   // coordinate) by theta degrees
00157   void Rotate(const Vector &axis, double thetaDeg, const Coord &center);
00158   // Rotate around a given bond by theta degrees (spins both ends of the bond in
00159   // opposite directions)
00160   RBTDLL_EXPORT void RotateBond(BondPtr spBond, double thetaDeg);
00161   // DM 09 Feb 1999 - Rotate around a given bond by theta degrees, keeping the
00162   // given atom fixed (only spins the other end of the bond)
00163   void RotateBond(BondPtr spBond, double thetaDeg, AtomPtr spFixedAtom);
00164   // DM 25 Feb 1999 - Rotate around a given bond by theta degrees, only spinning
00165   // one end of the bond If bSwap is false, spins the end bonded to atom2 in the
00166   // bond If bSwap is true, spins the end bonded to atom1 in the bond
00167   RBTDLL_EXPORT void RotateBond(BondPtr spBond, double thetaDeg, bool bSwap);
00168
00169   // DM 8 Feb 1999
00170   void SaveCoords(const std::string &coordName = "");
00171   RBTDLL_EXPORT void RevertCoords(const std::string &coordName = "");
00172   std::map<std::string, int> GetSavedCoordNames() const { return m_coordNames; }
00173   int GetNumSavedCoords() const { return m_coordNames.size(); }
00174   int GetCurrentCoords() const { return m_currentCoord; }
00175   RBTDLL_EXPORT void RevertCoords(int);
00176
00177   // Returns center of mass of model
00178   Coord GetCenterOfMass() const;
00179   // Translates the model so that the center of mass is at the given coord
00180   void SetCenterOfMass(const Coord &c) { Translate(c - GetCenterOfMass()); }
00181
00182   // DM 9 Nov 1999
00183   // Returns total atomic mass (molecular weight) for the model
00184   RBTDLL_EXPORT double GetTotalMass() const;
00185
00186   // DM 14 Apr 1999 - principal axes methods
00187   // Return principal axes and center of mass for the model
00188   PrincipalAxes GetPrincipalAxes() const;
00189   // Aligns the principal axes of the model to lie along alignAxes
00190   // If required (bAlignCOM=true), also aligns the center of mass with
00191   // alignAxes.com Default is to align with X,Y,Z Cartesian axes centered at
00192   // origin
00193   void AlignPrincipalAxes(const PrincipalAxes &alignAxes = PrincipalAxes(),
00194                           bool bAlignCOM = true);
00195
00196   // DM 19 Oct 2005 - new chromosome handling
00197   // Sets the flexibility type for this model
00198   // To revert to a rigid model, call SetFlexData(nullptr)
00199   RBTDLL_EXPORT void SetFlexData(FlexData *pFlexData);
00200   // Gets the flexibility type of this model
00201   // Returns nullptr for a rigid model
00202   RBTDLL_EXPORT FlexData *GetFlexData() const;
00203   // Returns a clone of the current chromosome for this model
00204   // The caller has the responsibility for mem management of the clone
00205   // Returns nullptr for a rigid model
00206   RBTDLL_EXPORT ChromElement *GetChrom() const;
00207
00208   bool isFlexible() const;
00209   const AtomRList &GetFlexIntns(Atom *pAtom) const;
00210   BondList GetFlexBonds() const;
00211   // Selects all atoms that are rotated by at least one rotable bond
00212   void SelectFlexAtoms();
00213   void SelectFlexAtoms(Atom *pAtom);
00214
00216   // Atom list functions (provided for convenience, as user could just as well
00217   // call the rxdock namespace functions with Model::GetAtomList)
00218   // e.g. AtomList atomList =
00219   // GetSelectedAtomList(spModel->GetAtomList); is equivalent to
```

```
00220   // AtomList atomList = spModel->GetSelectedAtomList();
00222
00223   // Unary
00224   // Generic template version of GetNumAtoms, passing in your own predicate
00225   // template<class Predicate> UInt GetNumAtoms(const Predicate& pred);
00226   // Generic template version of GetAtomList, passing in your own predicate
00227   // template<class Predicate> AtomList GetAtomList(const Predicate& pred);
00228
00229   // Selected atoms
00230   void SetAtomSelectionFlags(bool bSelected = true);
00231   unsigned int GetNumSelectedAtoms();
00232   AtomList GetSelectedAtomList();
00233
00234   // Cyclic atoms
00235   void SetAtomCyclicFlags(bool bCyclic = true);
00236   unsigned int GetNumCyclicAtoms();
00237   AtomList GetCyclicAtomList();
00238
00239   // DM 21 Jul 1999 User1 flag
00240   void SetAtomUser1Flags(bool bUser1 = true);
00241   // DM 29 Jan 2000 User1 double
00242   void SetAtomUser1Values(double dUser1 = 0.0);
00243   // DM 27 Jul 2000 User2 double
00244   void SetAtomUser2Values(double dUser2 = 0.0);
00245
00246   // Hydrogen bond acceptor atoms
00247   unsigned int GetNumHBondAcceptorAtoms();
00248   AtomList GetHBondAcceptorAtomList();
00249
00250   // Hydrogen bond donor atoms
00251   unsigned int GetNumHBondDonorAtoms();
00252   AtomList GetHBondDonorAtomList();
00253
00254   //(Formally) charged atoms
00255   unsigned int GetNumChargedAtoms();
00256   AtomList GetChargedAtomList();
00257
00258   // Planar atoms
00259   unsigned int GetNumPlanarAtoms();
00260   AtomList GetPlanarAtomList();
00261
00262   // Binary
00263
00264   // Atoms with atomic no = nAtomicNo
00265   unsigned int GetNumAtomsWithAtomicNo(int nAtomicNo);
00266   AtomList GetAtomListWithAtomicNo(int nAtomicNo);
00267
00268   // Atoms with FFType = strFFType
00269   unsigned int GetNumAtomsWithFFType(std::string strFFType);
00270   AtomList GetAtomListWithFFType(std::string strFFType);
00271
00273   // Bond list functions (provided for convenience, as user could just as well
00274   // call the rxdock namespace functions with Model::GetBondList)
00275   // e.g. BondList bondList =
00276   // GetSelectedBondList(spModel->GetBondList); is equivalent to
00277   // BondList bondList = spModel->GetSelectedBondList();
00279
00280   // Unary
00281
00282   // Generic template version of GetNumBonds, passing in your own predicate
00283   template <class Predicate> unsigned int GetNumBonds(const Predicate &pred);
00284   // Generic template version of GetBondList, passing in your own predicate
00285   template <class Predicate> BondList GetBondList(const Predicate &pred);
00286
00287   // Selected bonds
00288   void SetBondSelectionFlags(bool bSelected = true);
00289   unsigned int GetNumSelectedBonds();
00290   BondList GetSelectedBondList();
00291
00292   // Cyclic bonds
00293   void SetBondCyclicFlags(bool bCyclic = true);
00294   unsigned int GetNumCyclicBonds();
00295   BondList GetCyclicBondList();
00296
00297   // DM 10 Dec 1998 - at some point these functions should be implemented as
00298   // generic rxdock namespace functions operating on arbitrary atom and bond
00299   // lists
00300
00301   // D Morley, 2 Dec 1998 - go back to the old way, it's more convenient to get
00302   // all donors in the same list and separate them later void
00303   // GetHBondDonorLists(AtomList& donorList, AtomList& donorHList);
00304
00305   // Get min and max x,y,z coords
00306   // bInit = true => initialize min, max
00307   // bInit = false => don't initialize min, max (useful for accumulating min,max
00308   // over several Models)
00309   void GetMinMaxCoords(Coord &minCoord, Coord &maxCoord, bool bInit = true);
```

```
00310
00311   // Get map of (key=force field atom type string, value=no. of occurrences)
00312   std::map<std::string, int> GetAtomTypeMap();
00313
00314   // Get map of (key=force field bond type (atom type pair) string, value=no. of
00315   // occurrences)
00316   std::map<std::string, int> GetBondTypeMap();
00317
00318 protected:
00320   // Private methods
00322
00323 private:
00324   Model();                          // Disable default constructor
00325   Model(const Model &);             // Copy constructor disabled by default
00326   Model &operator=(const Model &); // Copy assignment disabled by default
00327
00328   // Create a new model from a data source
00329   void Create(BaseMolecularFileSource *pMolSource);
00330   void Clear();                      // Clear the current model
00331   void AddAtoms(AtomList &atomList); // Register an atom list with the model
00332
00334   // Private data
00336 private:
00337   std::string m_strName;                // Model name
00338   std::vector<std::string> m_titleList; // Title list (read from file)
00339   AtomList m_atomList;                  // atom list
00340   BondList m_bondList;                  // bond list
00341   SegmentMap m_segmentMap; // map of (key=segment name, value=atom count)
00342   AtomListList m_ringList; //(DM 7 Dec 1998) list of atom lists for each ring
00343   std::map<std::string, int>
00344       m_coordNames;           //(DM 8 Feb 1999) map of named coord sets
00345                               //(key=name, value=index into Atom coord map)
00346   int m_currentCoord;         // DM 11 Jul 2003 - which coord set is current
00347   StringVariantMap m_dataMap; // DM 12 May 1999 - associated data (e.g. from
00348                               // SD file or generated by rbdock)
00349   PseudoAtomList
00350       m_pseudoAtomList; // DM 11 Jul 2000 - store associated pseudoatoms
00351   ModelMutatorPtr
00352       m_spMutator; // DM 30 Jul 2001 - helper object for fast dihedral mutations
00353   FlexData *m_pFlexData;  // Stores the flexibility data for this model
00354   ChromElement *m_pChrom; // Reference chromosome. GetChrom() returns a clone
00355                           // of this object.
00356   double m_occupancy; // Occupancy value (0->1), in support of solvent occupancy
00357   bool m_enabled;     // Enabled state, depends on occupancy value and threshold
00358 };
00359
00360 void to_json(json &j, const Model &model);
00361 void from_json(const json &j, Model &model);
00362
00363 // Useful typedefs
00364 typedef SmartPtr<Model> ModelPtr;        // Smart pointer
00365 typedef std::vector<ModelPtr> ModelList; // Vector of smart pointers
00366 typedef ModelList::iterator ModelListIter;
00367 typedef ModelList::const_iterator ModelListConstIter;
00368
00369 } // namespace rxdock
00370
00371 #endif //_RBTMODEL_H_
```

6.90 ModelError.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // rxdock model exceptions (e.g. topological errors)
00014
00015 #ifndef _RBTMODELERROR_H_
00016 #define _RBTMODELERROR_H_
00017
00018 #include "rxdock/Error.h"
00019
00020 namespace rxdock {
00021
00022 const std::string IDS_MODEL_ERROR = "RBT_MODEL_ERROR";
```

```
00023
00024 // Unspecified model error
00025 class ModelError : public Error {
00026 public:
00027   ModelError(const std::string &strFile, int nLine,
00028              const std::string &strMessage = "")
00029       : Error(IDS_MODEL_ERROR, strFile, nLine, strMessage) {}
00030   // Protected constructor to allow derived model error classes to set error
00031   // name
00032 protected:
00033   ModelError(const std::string &strName, const std::string &strFile, int nLine,
00034              const std::string &strMessage = "")
00035       : Error(strName, strFile, nLine, strMessage) {}
00036 };
00037
00038 } // namespace rxdock
00039
00040 #endif //_RBTMODELERROR_H_
```

6.91 ModelMutator.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Legacy class, largely replaced by ChromElement subclasses
00014 // The only remaining purpose of ModelMutator is to generate lists
00015 // of flexible intramolecular interactions.
00016 // i.e. intramolecular atom pairs across rotatable bonds
00017 #ifndef _RBTMODELMUTATOR_H_
00018 #define _RBTMODELMUTATOR_H_
00019
00020 #include "rxdock/Atom.h"
00021 #include "rxdock/Bond.h"
00022 #include "rxdock/PrincipalAxes.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
00030 class Model; // forward declaration
00031
00032 // Only check request assertions in debug build
00033 #ifdef _NDEBUG
00034 const Bool MUT_CHECK = true;
00035 #else
00036 const bool MUT_CHECK = true;
00037 #endif //_NDEBUG
00038
00039 class ModelMutator {
00040 public:
00042   // Constructors/destructors
00043   ModelMutator(Model *pModel, const BondList &rotBonds,
00044                const AtomList &tetheredAtoms);
00045
00046   friend void to_json(json &j, const ModelMutator &modelMutator);
00047   friend void from_json(const json &j, ModelMutator &modelMutator);
00048
00049   virtual ~ModelMutator(); // Default destructor
00050
00052   // Public methods
00054   const AtomRListList &GetFlexIntns() const;
00055   const AtomRListList &GetFlexAtoms() const;
00056   BondList GetFlexBonds() const;
00057
00058 protected:
00060   // Protected methods
00062
00063 private:
00065   // Private methods
00067
00068   ModelMutator(const ModelMutator &); // Copy constructor disabled by default
00069   ModelMutator &
```

```
00070   operator=(const ModelMutator &); // Copy assignment disabled by default
00071
00072   void Setup();
00073
00074 protected:
00076   // Protected data
00078
00079 private:
00081   // Private data
00083   Model *m_pModel;       // Parent model
00084   BondList m_rotBonds;   // A record of the defined rotatable bonds passed to
00085                          // the constructor
00086   AtomRList m_dih1Atoms; // Atom 1 in each rotable bond dihedral definition
00087   AtomRList m_dih2Atoms; // Atom 2 in each rotable bond dihedral definition
00088   AtomRList m_dih3Atoms; // Atom 3 in each rotable bond dihedral definition
00089   AtomRList m_dih4Atoms; // Atom 4 in each rotable bond dihedral definition
00090   AtomRListList
00091       m_rotAtoms; // List of atom lists on smallest side of each rotable bond
00092   // DM 25 Apr 2002
00093   // m_flexIntns is a vector of size N(atoms)
00094   // Each element is an AtomList (i.e. std::vector<AtomPtr>)
00095   //
00096   //*** The vector is indexed by AtomId
00097   //*** We assume all atom IDs in the molecule are assigned sequentially from 1
00098   //***
00099   //*** For molecules read from MdlFileSource and PsfFileSource this is
00100   // guaranteed
00101   //
00102   // m_flexIntns[i-1] is a vector of all the atoms which can interact with atom
00103   // i across at least one rotable bond: (these interaction distances are
00104   // conformation dependent) This is a symmetric matrix - For a flexible
00105   // interaction i,j, atom j will appear in the AtomList for atom i, and vice
00106   // versa. Atoms which are NOT in this list are therefore in the same rigid
00107   // fragment as atom i: (these interaction distances are fixed and can
00108   // therefore
00109   // be ignored by any scoring function)
00110   AtomRListList m_flexIntns;
00111
00112   // DM 2 Jul 2002 - if model has tethered atoms we need to modify the behaviour
00113   // of ModelMutator Could be subclassed if necessary
00114   AtomList m_tetheredAtoms;
00115 };
00116
00117 void to_json(json &j, const ModelMutator &modelMutator);
00118 void from_json(const json &j, ModelMutator &modelMutator);
00119
00120 // Useful typedefs
00121 typedef SmartPtr<ModelMutator> ModelMutatorPtr; // Smart pointer
00122
00123 } // namespace rxdock
00124
00125 #endif //_RBTMODELMUTATOR_H_
```

6.92 MOEGrid.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // MOE grid description at ${MOE}/html/svl/manual/gridman.htm
00014 // here you can read:
00015 // [...]
00016 // Creating a grid consists of creating both a shape vector and a data vector.
00017 // The shape vector stores gridpoint positions along each dimension. The number
00018 // of dimensions must be between 2 and 24 with each dimension containing at
00019 // least 2 points. Each element of the shape vector is a flat vector of strictly
00020 // increasing grid positions along the corresponding dimension. The values of
00021 // the field at the gridpoints are stored in row-major order, i.e., with the
00022 // index of the last dimension changing most rapidly.
00023 // [...]
00024 // Generating a shape vector consists of creating coordinate vectors along each
00025 // of the n dimensions of an n-dimensional lattice. The intervals between the
00026 // coordinates need not be regular. To generate the shape vector, you need the
00027 // origin of the lattice and the length of each side of the lattice, along with
00028 // the coordinate spacings in each dimension.
```

```
00029 // [...]
00030 // The values of the field at the gridpoints are stored in a data vector. There
00031 // must be one data value for every gridpoint. The length of the data vector
00032 // must therefore equal the product of the number of gridpoints along each
00033 // dimension.
00034
00035 #ifndef _RBTMOEGRID_H_
00036 #define _RBTMOEGRID_H_
00037
00038 #include "rxdock/Coord.h"
00039 #include "rxdock/Rbt.h"
00040 #include <cstdlib>
00041 #include <fstream>
00042 #include <vector>
00043
00044 #include "rxdock/BiMolWorkSpace.h"
00045
00046 namespace rxdock {
00047
00048 // single point in a multi-dimensional grid
00049 // the coords contains N values where N is the dimension of the grid
00050 class MOEGridPoint {
00051   std::vector<double> coords;
00052   double value;
00053
00054 public:
00055   // default ctor with a 1D gridpoint at 0.0 value 1.0
00056   // see also default ctor for the shape and data vector below
00057   RBTDLL_EXPORT MOEGridPoint();
00058   // parametrized ctor when supplying data as a vector + value pair
00059   MOEGridPoint(std::vector<double>, double);
00060   // parametrized ctor getting coord values from Coord (default 3D
00061   // grid for convenience)
00062   MOEGridPoint(Coord &, double);
00063   // data access methods
00064   double GetValue() { return value; }
00065   const std::vector<double> &GetCoords() { return coords; }
00066   void SetCoords(double a_val) { coords.push_back(a_val); }
00067   void SetValue(double a_val) { value = a_val; }
00068 };
00069
00070 // MOE grid shape vector
00071 class MOEGridShape {
00072   // data that is provided usually in the SVL code
00073   // a shape vector is defined by
00074   //        i)      starting coordinates
00075   //        ii)     X,Y,Z extension (in absolute coordinates)
00076   //        iii)    grid stepsize
00077   // data vector defined by
00078   //        i)      a 1D vector containing the same number
00079   //                of elements than the total number of the
00080   //                shape vector
00081
00082   std::vector<double> origin;  // the starting point
00083   std::vector<double> extents; // the final points
00084   std::vector<int> i_extents;  // extents in integers: number of
00085                                // points at the edges (ie 20 X 30 X 40 grid)
00086   double step_size;            // grid step size
00087   long data_size;              // total number of points
00088
00089 public:
00090   // default ctor allocates a 1D grid starting at 0.0 to 1.0
00091   // with 1.0 stepsize (2 elements at least needed for MOE)
00092   MOEGridShape();
00093
00094   // parametrized ctor receives grid as the origin vector, the extent
00095   // vector and stepsize
00096   RBTDLL_EXPORT MOEGridShape(std::vector<double>, std::vector<double>, double);
00097
00098   const std::vector<double> &GetOrigin() { return origin; }
00099   const std::vector<double> &GetExtents() { return extents; }
00100   const std::vector<int> &GetIExtents() { return i_extents; }
00101   double GetStepSize() { return step_size; }
00102   long GetDataSize() { return data_size; }
00103 };
00104
00105 typedef std::vector<MOEGridPoint> MOEGridData;
00106
00107 // MOE grid data class for IO
00108 //
00109 // the MOE grid file format (in the present stage at 06/03/2003)
00110 class MOEGrid {
00111   MOEGridShape myShape;
00112   MOEGridData myData;
00113   std::string stream_name; // filename
00114   Coord min, max;          // common extent min and max coords
00115   Coord c_min, c_max;      // current extent min and max coords
```

```
00116 public:
00117   // default ctor
00118   RBTDLL_EXPORT MOEGrid();
00119   // parameterised ctor for a single grid
00120   // pass shape, data and outfilename
00121   MOEGrid(MOEGridShape &, MOEGridData &, std::string);
00122   // or pass parameters using these functions (when it is about
00123   // to save multiple grids into a single file)
00124   void SetShape(MOEGridShape &a_shape) { myShape = a_shape; }
00125   void SetData(MOEGridData &a_data) { myData = a_data; }
00126   void SetOutputFileName(std::string &a_filename) { stream_name = a_filename; }
00127   // methods to calculate grid extents
00128   RBTDLL_EXPORT void
00129   CalculateCommonExtents(std::vector<std::string> strPrmFiles); // for all
00130   void GetDockingSiteExtents(std::string &a_strPrmFile);        // for a single
00131   // get methods
00132   Coord GetMinExtents() { return min; }
00133   Coord GetMaxExtents() { return max; }
00134
00135   // iofstream methods
00136   RBTDLL_EXPORT long WriteGrid(
00137       std::ios_base::openmode mode); // returns the number of points written
00138   long ReadGrid();                   // ditto (fields read)
00139 };
00140
00141 // the relevant SVL code in gridshow.svl
00142 //
00143 // basic double should be the IEEE 8 byte double MSB first in Linux
00144 // on big-endian try swab with dd and/or cpio
00145 // MOE grid format:
00146 //
00147 // Note: the internal grid format (subject to change without notice!)
00148 // Header for the grid-file only:
00149 //  GRID_SIGNATURE          8 bytes
00150 // Each grid:
00151 //  version number              int4m
00152 //  ndata (= length data)       int4m
00153 //  ndim (= length size)        int4m
00154 //  skip                int4m
00155 //  ... (reserved for future use)   byte * skip
00156 //  size (= app length shape)   int4m * ndim
00157 //  shape               ieee8m * (add size)
00158 //  data                ieee8m * ndata
00159 //
00160 // On reading, we check that ndata, ndim, and size are 24-bit non-negatives,
00161 // but we don't check "ndata == (mul size)"
00162 //
00163 // snippets from '$MOE/lib/svl/run/gridshow.svl'
00164 // const GRID_SIGNATURE = "SVLgrid\n";
00165 // const VERSION = 100;
00166 // const SKIP = 0;
00167 //
00168 // Check file signature and return non-zero if the file of file name fn
00169 // is a "grid file", i.e. a special file that contains exactly one grid
00170 // and nothing else
00171 //
00172 // local function read_gridsign fn =
00173 //    freadb [fn, 'char', length GRID_SIGNATURE] === GRID_SIGNATURE;
00174
00175 // Write grid to a file of given filename or filehandle
00176 //
00177 // local function writeb_fnum_grid [fn, [data, shape]]
00178 //    local ndata = length data;
00179 //    local ndim = length shape;
00180 //    local size = app length shape;
00181 //
00182 //    fwriteb [fn, 'int4m', [VERSION, ndata, ndim, SKIP]];
00183 //    fwriteb [fn, 'char', rep ["\0", SKIP]];
00184 //    fwriteb [fn, 'int4m', size];
00185 //    fwriteb [fn, 'IEEE8m', cat shape];
00186 //    fwriteb [fn, 'IEEE8m', data];
00187 // endfunction
00188 //
00189 // function grid_fwriteb [fn, grid]
00190 //    if type fn === 'num' then writeb_fnum_grid [fn, grid]; return; endif
00191 //    fn = fopenw fn;
00192 //    writeb_fnum_grid [fn, grid];
00193 //    fclose fn;
00194 // endfunction
00195 //
00196 // Create/rewrite "grid-file" of given name that willcontain a given grid
00197 //
00198 // function grid_writefile [fn, grid]
00199 //    fn = fopenw fn;
00200 //    fwriteb [fn, 'char', GRID_SIGNATURE];
00201 //    writeb_fnum_grid [fn, grid];
00202 //    fclose fn;
```

```
00203 // endfunction
00204
00205 //    fwriteb [fn, 'char', GRID_SIGNATURE];
00206 //    fwriteb [fn, 'int4m', [VERSION, ndata, ndim, SKIP]];
00207 //    fwriteb [fn, 'char', rep ["\0", SKIP]];
00208 //    fwriteb [fn, 'int4m', size];
00209 //    fwriteb [fn, 'IEEE8m', cat shape];
00210 //    fwriteb [fn, 'IEEE8m', data];
00211
00212 } // namespace rxdock
00213
00214 #endif //_RBTMOEGRID_H_
```

6.93 MOL2FileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTMOL2FILESOURCE_H_
00014 #define _RBTMOL2FILESOURCE_H_
00015
00016 #include "rxdock/BaseMolecularFileSource.h"
00017 #include "rxdock/ElementFileSource.h"
00018 #include "rxdock/ParameterFileSource.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 // Simple struct for holding MOL2 SUBSTRUCTURE records
00027 class MOL2Substructure {
00028 public:
00029   MOL2Substructure() : root_atom(0), chain("UNK"), sub_type("UNK") {}
00030   MOL2Substructure(const std::string &name, int rr, const std::string &cc,
00031                    const std::string &ss)
00032       : subst_name(name), root_atom(rr), chain(cc), sub_type(ss) {}
00033   std::string GetName() const { return subst_name; };
00034   int GetRoot() const { return root_atom; };
00035   std::string GetChain() const { return chain; };
00036   std::string GetType() const { return sub_type; };
00037
00038   friend void to_json(json &j, const MOL2Substructure &mol2Structure);
00039   friend void from_json(const json &j, MOL2Substructure &mol2Structure);
00040
00041 private:
00042   std::string subst_name;
00043   int root_atom;
00044   std::string chain;
00045   std::string sub_type;
00046 };
00047
00048 void to_json(json &j, const MOL2Substructure &mol2Structure);
00049 void from_json(const json &j, MOL2Substructure &mol2Structure);
00050
00051 // A MOL2SubstructureMap stores MOL2Substructures keyed by SUBSTRUCTURE ID
00052 // (subst_id)
00053 typedef std::map<int, MOL2Substructure> MOL2SubstructureMap;
00054 typedef MOL2SubstructureMap::iterator MOL2SubstructureMapIter;
00055 typedef MOL2SubstructureMap::const_iterator MOL2SubstructureMapConstIter;
00056
00057 class MOL2FileSource : public BaseMolecularFileSource {
00058 public:
00059   static const std::string _CT;
00060   // record delimiter strings
00061   static const std::string _TRIPOS_DELIM;
00062   static const unsigned int _TRIPOS_DELIM_SIZE;
00063   static const std::string _IDS_MOL2_RECDELIM;
00064
00065   MOL2FileSource(const std::string &fileName, bool bImplHydrogens = true);
00066   ~MOL2FileSource();
00067
00068   friend void to_json(json &j, const MOL2FileSource &mol2FileSrc);
```

```
00069   friend void from_json(const json &j, MOL2FileSource &mol2FileSrc);
00070
00071   bool isTitleListSupported() { return true; }
00072   bool isAtomListSupported() { return true; }
00073   bool isCoordinatesSupported() { return true; }
00074   bool isBondListSupported() { return true; }
00075   bool isDataSupported() { return false; }
00076
00077 protected:
00078   virtual void Parse();
00079
00080 private:
00081   MOL2FileSource();
00082   MOL2FileSource(const MOL2FileSource &);
00083   MOL2FileSource &operator=(const MOL2FileSource &);
00084
00085   // Helper functions to clean up atom attributes after parsing the file
00086   void SetupAtomParams();
00087   void FixImplicitHydrogenCount();
00088   void FixHybridState();
00089   void FixTriposTypes();
00090   void SetupVdWRadii();
00091   void RemoveNonPolarHydrogens();
00092
00093   // Splits subst_name into ID and name components (e.g. ALA27 => ALA and 27)
00094   void GetSSIDandName(const std::string &subst_name, int subst_id,
00095                       std::string &sID, std::string &sName);
00096
00097   // parser functions (switched by pointers)
00098   void ParseRecordMOLECULE(const std::string &aLine);
00099   void ParseRecordATOM(const std::string &aLine);
00100   void ParseRecordBOND(const std::string &aLine);
00101   void ParseRecordSUBSTRUCTURE(const std::string &aLine);
00102   void ParseRecordUNSUPPORTED(const std::string &aLine);
00103   // get delimiter tag
00104   std::string GetMOL2Tag(const std::string &aLine);
00105   // parse/tokenize second line (number of  atoms, bonds etc) in MOLECULE
00106   void ParseCountFields(const std::string &aLine);
00107   void Tokenize(const std::string &aString,
00108                 std::vector<std::string> &aTokensBuf);
00109
00110   unsigned int m_NL; // No. of lines
00111   // data fields in MOLECULE
00112   unsigned int nAtoms;         // number of atoms
00113   unsigned int nBonds;         //           bonds
00114   unsigned int nSubstructures; // substructures
00115   unsigned int nFeatures;      // features
00116   unsigned int nSets;          // sets
00117
00118   // helper data structures used to lookup chain names for each substructure
00119   // and to sort atom list into a more sensible order (ordered by substructure
00120   // ID)
00121   MOL2SubstructureMap m_ssInfo;
00122   std::map<int, AtomList> m_ssAtoms;
00123
00124   ElementFileSourcePtr
00125       m_spElementData; // Elemental data source for vdW radii etc
00126   ParameterFileSourcePtr
00127       m_spParamSource;    // Parameter file source for partial charges etc
00128   TriposAtomType m_typer; // Tripos atom typing object
00129   bool m_bImplHydrogens;
00130 };
00131
00132 void to_json(json &j, const MOL2FileSource &mol2FileSrc);
00133 void from_json(const json &j, MOL2FileSource &mol2FileSrc);
00134
00135 // usual smart pointer
00136 typedef SmartPtr<MOL2FileSource> MOL2FileSourcePtr;
00137 // function pointer (callback) for Parse()
00138 typedef void (MOL2FileSource::*fcPtr)(const std::string &);
00139
00140 } // namespace rxdock
00141
00142 #endif // _RBTMOL2FILESOURCE_H_
```

6.94 NMCriteria.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
```

```
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 #ifndef _RBTNMCRITERIA_H_
00014 #define _RBTNMCRITERIA_H_
00015
00016 #include <loguru.hpp>
00017
00018 #include <cmath>
00019 #include <limits>
00020
00021 namespace rxdock {
00022
00023 namespace neldermead {
00024
00025 class IterationCriterion {
00026 private:
00027   int m_maxIterations;
00028
00029 public:
00030   IterationCriterion(long max_iterations) : m_maxIterations(max_iterations) {}
00031
00032   template <class State> bool operator()(const State &state) {
00033     bool result = state.iteration < m_maxIterations;
00034     LOG_F(1, "IterationCriteria: {}", result);
00035     return result;
00036   }
00037 };
00038
00039 template <class DataType> class RelativeValueCriterion {
00040 private:
00041   DataType m_ftol;
00042
00043 public:
00044   RelativeValueCriterion(DataType ftol) : m_ftol(ftol) {}
00045
00046   template <class State> bool operator()(const State &state) {
00047     bool result =
00048         2 * std::abs(state.currentValue - state.formerValue) /
00049             (std::abs(state.currentValue) + std::abs(state.formerValue) +
00050              std::numeric_limits<DataType>::epsilon()) >
00051         m_ftol;
00052     LOG_F(1, "RelativeValueCriterion: {}", result);
00053     return result;
00054   }
00055 };
00056
00057 template <class Criteria1, class Criteria2> class AndCriteria {
00058 private:
00059   Criteria1 m_criteria1;
00060   Criteria2 m_criteria2;
00061
00062 public:
00063   AndCriteria(const Criteria1 &criteria1, const Criteria2 &criteria2)
00064       : m_criteria1(criteria1), m_criteria2(criteria2) {}
00065
00066   template <class State> bool operator()(const State &state) {
00067     return m_criteria1(state) && m_criteria2(state);
00068   }
00069 };
00070
00071 template <class Criteria1, class Criteria2>
00072 AndCriteria<Criteria1, Criteria2>
00073 CreateAndCriteria(const Criteria1 &criteria1, const Criteria2 &criteria2) {
00074   return AndCriteria<Criteria1, Criteria2>(criteria1, criteria2);
00075 }
00076
00077 } // namespace neldermead
00078
00079 } // namespace rxdock
00080
00081 #endif /* _RBTNMCRITERIA_H_ */
```

6.95 NmrRestraintFileSource.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
```

```
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // File reader for external NMR restraints. Supports NOE distance restraints
00014 // and STD ligand-receptor contact surface restraints.
00015 // NOE Syntax:
00016 //<ATOM_LIST_1> <ATOM_LIST_2> <MAX DISTANCE>
00017 //
00018 // STD Syntax:
00019 // STD <ATOM_LIST_1> <MAX DISTANCE> (this is the literal string "STD")
00020 //
00021 // Comment lines are supported (# first character)
00022 // Atom names in list should be comma-separated, with NO SPACES
00023 //
00024 // For both types of restraint, brackets can be used around each atom list to
00025 // denote various levels of ambiguity:
00026 //
00027 //(ATOM_LIST) = apply restraint to mean coords of atoms in list (MEAN)
00028 //[ATOM_LIST] = apply restraint to all atoms in list simultaneously (AND)
00029 // ATOM_LIST = restraint can be satisfied by any atom in list  (OR)
00030 //
00031 // e.g.
00032 // MOL_1:C3 (_22:C5,_23:N1) 5.0 #MOL_1:C3 within 5A of the MEAN of _22:C5,_23:N1
00033 // MOL_1:C3 [_22:C5,_23:N1] 5.0 #MOL_1:C3 within 5A of _22:C5 AND 5A of _23:N1
00034 // MOL_1:C3 _22:C5,_23:N1   5.0 #MOL_1:C3 within 5A of _22:C5 OR 5A of _23:N1
00035 // STD MOL_1:C3 5.0             #MOL_1:C3 within 5A of any part of the receptor
00036
00037 #ifndef _RBTNMRRESTRAINTFILESOURCE_H_
00038 #define _RBTNMRRESTRAINTFILESOURCE_H_
00039
00040 #include "rxdock/BaseFileSource.h"
00041 #include "rxdock/NoeRestraint.h"
00042
00043 #include <nlohmann/json.hpp>
00044
00045 using json = nlohmann::json;
00046
00047 namespace rxdock {
00048
00049 class NmrRestraintFileSource : public BaseFileSource {
00050 public:
00052   // Constructors/destructors
00053   NmrRestraintFileSource(const std::string &fileName);
00054
00055   // Destructor
00056   virtual ~NmrRestraintFileSource();
00057
00058   friend void to_json(json &j,
00059                       const NmrRestraintFileSource &nmrRestrainfFileSrc);
00060   friend void from_json(const json &j,
00061                         NmrRestraintFileSource &nmrRestrainfFileSrc);
00062
00064   // Public methods
00066   unsigned int GetNumNoeRestraints(); // Number of NOE restraints in file
00067   NoeRestraintNamesList GetNoeRestraintList(); // List of NOE restraints
00068   unsigned int GetNumStdRestraints(); // Number of STD restraints in file
00069   StdRestraintNamesList GetStdRestraintList(); // List of STD restraints
00070
00071 protected:
00073   // Protected methods
00075
00076 private:
00078   // Private methods
00080
00081   NmrRestraintFileSource(); // Disable default constructor
00082   NmrRestraintFileSource(
00083       const NmrRestraintFileSource &); // Copy constructor disabled by default
00084   NmrRestraintFileSource &operator=(
00085       const NmrRestraintFileSource &); // Copy assignment disabled by default
00086
00087   // Pure virtual in BaseFileSource - needs to be defined here
00088   virtual void Parse();
00089   void ClearRestraintCache();
00090
00091   // Returns NOE restraint type and modifies the atom name string accordingly
00092   // Returns UNDEFINED if the atom name string has bad syntax
00093   eNoeType NoeRestraintType(std::string &strAtomNames);
00094
00095 protected:
00097   // Protected data
00099
00100 private:
00102   // Private data
```

```
00104   NoeRestraintNamesList m_noeRestraintList;
00105   StdRestraintNamesList m_stdRestraintList;
00106 };
00107
00108 void to_json(json &j, const NmrRestraintFileSource &nmrRestrainfFileSrc);
00109 void from_json(const json &j, NmrRestraintFileSource &nmrRestrainfFileSrc);
00110
00111 // Useful typedefs
00112 typedef SmartPtr<NmrRestraintFileSource>
00113     NmrRestraintFileSourcePtr; // Smart pointer
00114 typedef std::vector<NmrRestraintFileSourcePtr>
00115     NmrRestraintFileSourceList; // Vector of smart pointers
00116 typedef NmrRestraintFileSourceList::iterator NmrRestraintFileSourceListIter;
00117 typedef NmrRestraintFileSourceList::const_iterator
00118     NmrRestraintFileSourceListConstIter;
00119
00120 } // namespace rxdock
00121
00122 #endif //_RBTNMRRESTRAINTFILESOURCE_H_
```

6.96 NmrSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Generic nmr restraint scoring function
00014
00015 #ifndef _RBTNMRSF_H_
00016 #define _RBTNMRSF_H_
00017
00018 #include "rxdock/BaseIdxSF.h"
00019 #include "rxdock/BaseInterSF.h"
00020 #include "rxdock/NoeRestraint.h"
00021
00022 namespace rxdock {
00023
00024 class NmrSF : public BaseInterSF, public BaseIdxSF {
00025 public:
00026   // Class type string
00027   static const std::string _CT;
00028   // Parameter names
00029   static const std::string _FILENAME; // Nmr restraint file name
00030   static const std::string
00031       _QUADRATIC; // True = quadratic penalty function; false = linear
00032
00033   NmrSF(const std::string &strName = "nmr");
00034   virtual ~NmrSF();
00035
00036 protected:
00037   virtual void SetupReceptor();
00038   virtual void SetupLigand();
00039   virtual void SetupScore();
00040   virtual double RawScore() const;
00041   void ParameterUpdated(const std::string &strName);
00042
00043 private:
00044   double NoeDistance(const NoeRestraintAtoms &noe) const;
00045   double StdDistance(const StdRestraintAtoms &std) const;
00046
00047   bool m_bQuadratic; // synchronised with QUADRATIC named parameter
00048   NonBondedGridPtr m_spGrid;
00049   AtomList m_ligAtomList;          // All ligand atoms
00050   NoeRestraintAtomsList m_noeList; // List of all NOE interactions
00051   StdRestraintAtomsList m_stdList; // List of all STD interactions
00052 };
00053
00054 } // namespace rxdock
00055
00056 #endif //_RBTNMRSF_H_
```

6.97 NMSimplex.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTNMSIMPLEX_H_
00014 #define _RBTNMSIMPLEX_H_
00015
00016 #ifdef __PGI
00017 #define EIGEN_DONT_VECTORIZE
00018 #endif
00019 #include <Eigen/Core>
00020
00021 #include "rxdock/NMState.h"
00022
00023 #include <loguru.hpp>
00024
00025 namespace rxdock {
00026
00027 namespace neldermead {
00028
00038 template <class DataType, class ParameterType, class Function, class Criterion>
00039 class Simplex {
00040 private:
00041   DataType m_delta;
00042   ParameterType m_deltas;
00043   bool use_deltas;
00044
00045   Eigen::Array<DataType, Eigen::Dynamic, Eigen::Dynamic> m_polytopePoints;
00046   Eigen::Array<DataType, Eigen::Dynamic, Eigen::Dynamic> m_polytopeValues;
00047
00048   State<DataType, ParameterType> m_state;
00049   Criterion m_criterion;
00050
00051   void InitializePolytope(const ParameterType &start_point, DataType delta,
00052                           Function &fun) // const
00053   {
00054     m_polytopePoints.resize(start_point.size(), start_point.size() + 1);
00055     m_polytopeValues.resize(1, start_point.size() + 1);
00056     m_polytopePoints.col(0) = start_point;
00057     m_polytopeValues(0, 0) = fun(start_point);
00058     Display(fun, m_polytopePoints.col(0));
00059     for (int i = 1; i < start_point.size() + 1; ++i) {
00060       m_polytopePoints.col(i) = start_point;
00061       m_polytopePoints(i - 1, i) += delta;
00062       m_polytopeValues(0, i) = fun(m_polytopePoints.col(i));
00063       Display(fun, m_polytopePoints.col(i));
00064     }
00065   }
00066
00067   void InitializePolytope(const ParameterType &start_point,
00068                           ParameterType deltas, Function &fun) // const
00069   {
00070     m_polytopePoints.resize(start_point.size(), start_point.size() + 1);
00071     m_polytopeValues.resize(1, start_point.size() + 1);
00072     m_polytopePoints.col(0) = start_point;
00073     m_polytopeValues(0, 0) = fun(start_point);
00074     Display(fun, m_polytopePoints.col(0));
00075     for (int i = 1; i < start_point.size() + 1; ++i) {
00076       m_polytopePoints.col(i) = start_point;
00077       m_polytopePoints(i - 1, i) += deltas(i - 1);
00078       m_polytopeValues(0, i) = fun(m_polytopePoints.col(i));
00079       Display(fun, m_polytopePoints.col(i));
00080     }
00081   }
00082
00083   void Display(Function &fun, const ParameterType &parameters) // const
00084   {
00085     LOG_F(1, "Point: {}", parameters);
00086     LOG_F(1, "Value: {}", fun(parameters));
00087   }
00088
00089   void FindBestWorstNearWorst(Function &fun, int &best, int &worst,
00090                               int &near_worst) // const
00091   {
00092     if (m_polytopeValues(0, 0) > m_polytopeValues(0, 1)) {
00093       worst = 0;
00094       near_worst = 1;
```

```
00095     } else {
00096       worst = 1;
00097       near_worst = 0;
00098     }
00099     best = near_worst;
00100     for (int i = 2; i < m_polytopeValues.cols(); ++i) {
00101       if (m_polytopeValues(0, i) < m_polytopeValues(0, best)) {
00102         best = i;
00103       }
00104       if (m_polytopeValues(0, i) > m_polytopeValues(0, worst)) {
00105         near_worst = worst;
00106         worst = i;
00107       } else if (m_polytopeValues(0, i) > m_polytopeValues(0, near_worst)) {
00108         near_worst = i;
00109       }
00110     }
00111     LOG_F(1, "Worst value is in position {} {}\n{}", worst,
00112           m_polytopeValues(0, worst), m_polytopePoints.col(worst));
00113     LOG_F(1, "Near-worst value is in position {} {}\n{}", near_worst,
00114           m_polytopeValues(0, near_worst), m_polytopePoints.col(near_worst));
00115     LOG_F(1, "Best value is in position {} {}\n{}", best,
00116           m_polytopeValues(0, best), m_polytopePoints.col(best));
00117   }
00118
00119   ParameterType CreateNewParameters(const ParameterType &sum,
00120                                     const ParameterType &discarded_point,
00121                                     DataType t) {
00122     DataType fac1 = (1 - t) / sum.size();
00123     DataType fac2 = fac1 - t;
00124     return sum * fac1 - discarded_point * fac2;
00125   }
00126
00127 public:
00128   Simplex(const Criterion &criterion)
00129       : m_delta(0), use_deltas(false), m_criterion(criterion) {}
00130
00131   void Optimize(Function &fun) // const
00132   {
00133     m_state.iteration = 0;
00134     m_state.currentValue = fun(m_state.currentParameters);
00135     m_state.formerValue = std::numeric_limits<DataType>::max();
00136     m_state.bestValue = std::numeric_limits<DataType>::max();
00137
00138     if (use_deltas) {
00139       InitializePolytope(m_state.currentParameters, m_deltas, fun);
00140     } else {
00141       InitializePolytope(m_state.currentParameters, m_delta, fun);
00142     }
00143
00144     while (m_criterion(m_state)) {
00145       LOG_F(1, "Starting iteration {}", m_state.iteration);
00146       int best, worst, near_worst;
00147       FindBestWorstNearWorst(fun, best, worst, near_worst);
00148       m_state.currentValue = m_polytopeValues(0, best);
00149       m_state.currentParameters = m_polytopePoints.col(best);
00150       m_state.formerValue = m_polytopeValues(0, worst);
00151       m_state.formerParameters = m_polytopePoints.col(worst);
00152
00153       if (m_state.currentValue < m_state.bestValue) {
00154         m_state.bestValue = m_state.currentValue;
00155         m_state.bestParameters = m_state.currentParameters;
00156       }
00157
00158       ParameterType new_parameters = CreateNewParameters(
00159           m_polytopePoints.rowwise().sum(), m_polytopePoints.col(worst), -1);
00160       DataType new_value = fun(new_parameters);
00161       LOG_F(1, "Trying normal");
00162       Display(fun, new_parameters);
00163       if (new_value < m_state.bestValue) {
00164         ParameterType expansion_parameters = CreateNewParameters(
00165             m_polytopePoints.rowwise().sum(), m_polytopePoints.col(worst), -2);
00166         DataType expansion_value = fun(expansion_parameters);
00167         LOG_F(1, "Trying expansion");
00168         Display(fun, expansion_parameters);
00169         if (expansion_value < m_state.bestValue) {
00170           m_state.currentValue = m_polytopeValues(0, worst) = expansion_value;
00171           m_state.currentParameters = m_polytopePoints.col(worst) =
00172               expansion_parameters;
00173         } else {
00174           m_state.currentValue = m_polytopeValues(0, worst) = new_value;
00175           m_state.currentParameters = m_polytopePoints.col(worst) =
00176               new_parameters;
00177         }
00178       } else if (new_value > m_polytopeValues(0, near_worst)) {
00179         // New point is not better than near worst
00180         ParameterType contraction_parameters = CreateNewParameters(
00181             m_polytopePoints.rowwise().sum(), m_polytopePoints.col(worst), -.5);
```

```
00182         DataType contraction_value = fun(contraction_parameters);
00183         LOG_F(1, "Trying contraction");
00184         Display(fun, contraction_parameters);
00185         if (contraction_value > new_value) {
00186           LOG_F(1, "Contraction around lowest");
00187           Display(fun, contraction_parameters);
00188           ParameterType best_parameters = m_polytopePoints.col(best);
00189           LOG_F(1, "Difference from the best {}",
00190                 ((m_polytopePoints.colwise() - best_parameters.array()) / 2));
00191           m_polytopePoints =
00192               ((m_polytopePoints.colwise() - best_parameters.array()) / 2)
00193                   .colwise() +
00194               best_parameters.array();
00195           for (int i = 0; i < m_polytopePoints.cols(); ++i) {
00196             m_polytopeValues(0, i) = fun(m_polytopePoints.col(i));
00197           }
00198         } else {
00199           m_state.currentValue = m_polytopeValues(0, worst) = contraction_value;
00200           m_state.currentParameters = m_polytopePoints.col(worst) =
00201               contraction_parameters;
00202         }
00203       } else {
00204         // New point, not the best, but better than the near worst
00205         m_state.currentValue = m_polytopeValues(0, worst) = new_value;
00206         m_state.currentParameters = m_polytopePoints.col(worst) =
00207             new_parameters;
00208       }
00209
00210       ++m_state.iteration;
00211     }
00212   }
00213
00217   const ParameterType &GetBestParameters() const {
00218     return m_state.bestParameters;
00219   }
00220
00224   DataType GetBestValue() const { return m_state.bestValue; }
00225
00226   void SetStartPoint(const ParameterType &point) {
00227     m_state.currentParameters = point;
00228   }
00229
00230   void SetDelta(DataType delta) {
00231     this->m_delta = delta;
00232     use_deltas = false;
00233   }
00234
00235   void SetDelta(ParameterType deltas) {
00236     this->m_deltas = deltas;
00237     use_deltas = true;
00238   }
00239 };
00240
00241 template <class Function, class Criterion>
00242 static Simplex<typename Function::DataType, typename Function::ParameterType,
00243                Function, Criterion>
00244 CreateSimplex(Function &fun, const Criterion &criterion) // const
00245 {
00246   return Simplex<typename Function::DataType, typename Function::ParameterType,
00247                  Function, Criterion>(criterion);
00248 }
00249
00250 } // namespace neldermead
00251
00252 } // namespace rxdock
00253
00254 #endif /* _RBTNMSIMPLEX_H_ */
```

6.98 NMState.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTNMSTATE_H_
```

```
00014 #define _RBTNMSTATE_H_
00015
00016 #include <nlohmann/json.hpp>
00017
00018 using json = nlohmann::json;
00019
00020 namespace rxdock {
00021
00022 namespace neldermead {
00023
00027 template <class DataType, class ParameterType> struct State {
00028   friend void to_json(json &j, const State &state) {
00029     j = json{{"best-parameters", state.bestParameters},
00030              {"best-value", state.bestValue},
00031              {"current-parameters", state.currentParameters},
00032              {"current-value", state.currentValue},
00033              {"former-parameters", state.formerParameters},
00034              {"former-value", state.formerValue},
00035              {"iteration", state.iteration}};
00036   }
00037
00038   friend void from_json(const json &j, State &state) {
00039     j.at("best-parameters").get_to(state.bestParameters);
00040     j.at("best-value").get_to(state.bestValue);
00041     j.at("current-parameters").get_to(state.currentParameters);
00042     j.at("current-value").get_to(state.currentValue);
00043     j.at("former-parameters").get_to(state.formerParameters);
00044     j.at("former-value").get_to(state.formerValue);
00045     j.at("iteration").get_to(state.iteration);
00046   }
00047
00048   ParameterType bestParameters;
00049   DataType bestValue;
00050
00051   ParameterType currentParameters;
00052   DataType currentValue;
00053
00054   ParameterType formerParameters;
00055   DataType formerValue;
00056
00057   long iteration;
00058 };
00059
00060 } // namespace neldermead
00061
00062 } // namespace rxdock
00063
00064 #endif /* _RBTNMSTATE_H_ */
```

6.99 NoeRestraint.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Miscellaneous helper classes/structs for handling NOE interactions
00014
00015 #ifndef _RBTNOERESTRAINT_H_
00016 #define _RBTNOERESTRAINT_H_
00017
00018 #include "rxdock/Atom.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 // NOE Restraint type
00027 // UNDEFINED: undefined restraint type
00028 // OR: The distance to each of the atoms in the list is calculated separately,
00029 //    and any of these can satisfy the restraint.
00030 //    In practice, this means only the shortest distance is scored
00031 // MEAN: The distance to the mean coords of the atoms in the list is calculated,
00032 //      and used to evaluate the restraint
```

```
00033 // OR: The distance to each of the atoms in the list is calculated separately,
00034 //    and all of these should satisfy the restraint.
00035 //    In practice, this means only the longest distance is scored
00036 enum eNoeType { NOE_UNDEFINED, NOE_OR, NOE_MEAN, NOE_AND };
00037
00038 // Returns a string representing the NOE restraint type
00039 std::string ConvertNoeRestraintTypeToString(const eNoeType type);
00040
00042 // Simple struct for holding one end of an NOE restraint
00043 // Has an atom name string list and a restraint type
00044 class NoeEndNames {
00045 public:
00046   NoeEndNames() {}
00047   bool isOK() const;
00048   // Public data
00049   std::vector<std::string> names;
00050   eNoeType type;
00051
00052   friend void to_json(json &j, const NoeEndNames &noeEndNames);
00053   friend void from_json(const json &j, NoeEndNames &noeEndNames);
00054 };
00055
00056 void to_json(json &j, const NoeEndNames &noeEndNames);
00057 void from_json(const json &j, NoeEndNames &noeEndNames);
00058
00059 // Insertion operator for the above
00060 std::ostream &operator<<(std::ostream &s, const NoeEndNames &n);
00061
00063 // As above, but with a real atom list in place of the string list
00064 class NoeEndAtoms {
00065 public:
00066   // Default constructor
00067   NoeEndAtoms() {}
00068   // Constructor accepting an NoeEndNames and an atom list to match against
00069   NoeEndAtoms(const NoeEndNames &n, const AtomList &atomList);
00070   bool isOK() const;
00071   // Public data
00072   AtomList atoms;
00073   eNoeType type;
00074
00075   friend void to_json(json &j, const NoeEndAtoms &noeEndAtoms);
00076   friend void from_json(const json &j, NoeEndAtoms &noeEndAtoms);
00077 };
00078
00079 void to_json(json &j, const NoeEndAtoms &noeEndAtoms);
00080 void from_json(const json &j, NoeEndAtoms &noeEndAtoms);
00081
00082 // Insertion operator for the above
00083 std::ostream &operator<<(std::ostream &s, const NoeEndAtoms &n);
00084
00086 // Simple struct for holding a complete NOE restraint definition
00087 // Has two NoeEndNames, and max distance
00088 class NoeRestraintNames {
00089 public:
00090   NoeRestraintNames() : maxDist(5.0) {}
00091   bool isOK() const;
00092   // Public data
00093   NoeEndNames from;
00094   NoeEndNames to;
00095   double maxDist;
00096
00097   friend void to_json(json &j, const NoeRestraintNames &noeRestraintNames);
00098   friend void from_json(const json &j, NoeRestraintNames &noeRestraintNames);
00099 };
00100
00101 void to_json(json &j, const NoeRestraintNames &noeRestraintNames);
00102 void from_json(const json &j, NoeRestraintNames &noeRestraintNames);
00103
00104 // Insertion operator for the above
00105 std::ostream &operator<<(std::ostream &s, const NoeRestraintNames &noe);
00106
00108 // As above, but with real NoeEndAtoms in place of NoeEndNames
00109 class NoeRestraintAtoms {
00110 public:
00111   // Default constructor
00112   NoeRestraintAtoms() : maxDist(5.0) {}
00113   // Constructor accepting an NoeRestraintNames and an atom list to match
00114   // against
00115   NoeRestraintAtoms(const NoeRestraintNames &n, const AtomList &atomList);
00116   bool isOK() const; // true if at least one atom at each end of the NOE and
00117                      // NOE type is not undefined
00118   bool isSimple()
00119       const; // true if exactly one atom at each end of the NOE (unambiguous)
00120   // Public data
00121   NoeEndAtoms from;
00122   NoeEndAtoms to;
00123   // Double minDist;
```

```
00124   double maxDist;
00125
00126   friend void to_json(json &j, const NoeRestraintAtoms &noeRestraintAtoms);
00127   friend void from_json(const json &j, NoeRestraintAtoms &noeRestraintAtoms);
00128 };
00129
00130 void to_json(json &j, const NoeRestraintAtoms &noeRestraintAtoms);
00131 void from_json(const json &j, NoeRestraintAtoms &noeRestraintAtoms);
00132
00133 // Insertion operator for the above
00134 std::ostream &operator«(std::ostream &s, const NoeRestraintAtoms &noe);
00135
00137 // Simple struct for holding a complete STD restraint definition
00138 // Has one NoeEndNames, and max distance (from receptor)
00139 class StdRestraintNames {
00140 public:
00141   StdRestraintNames() : maxDist(5.0) {}
00142   bool isOK() const;
00143   // Public data
00144   NoeEndNames from;
00145   double maxDist;
00146
00147   friend void to_json(json &j, const StdRestraintNames &stdRestraintNames);
00148   friend void from_json(const json &j, StdRestraintNames &stdRestraintNames);
00149 };
00150
00151 void to_json(json &j, const StdRestraintNames &stdRestraintNames);
00152 void from_json(const json &j, StdRestraintNames &stdRestraintNames);
00153
00154 // Insertion operator for the above
00155 std::ostream &operator«(std::ostream &s, const StdRestraintNames &std);
00156
00158 // As above, but with real NoeEndAtoms in place of NoeEndNames
00159 class StdRestraintAtoms {
00160 public:
00161   // Default constructor
00162   StdRestraintAtoms() : maxDist(5.0) {}
00163   // Constructor accepting an StdRestraintNames and an atom list to match
00164   // against
00165   StdRestraintAtoms(const StdRestraintNames &n, const AtomList &atomList);
00166   bool isOK() const; // true if at least one atom in the definition and type
00167                      // is not undefined
00168   bool isSimple()
00169       const; // true if exactly one atom at in the STD definition (unambiguous)
00170   // Public data
00171   NoeEndAtoms from;
00172   double maxDist;
00173
00174   friend void to_json(json &j, const StdRestraintAtoms &stdRestraintAtoms);
00175   friend void from_json(const json &j, StdRestraintAtoms &stdRestraintAtoms);
00176 };
00177
00178 void to_json(json &j, const StdRestraintAtoms &stdRestraintAtoms);
00179 void from_json(const json &j, StdRestraintAtoms &stdRestraintAtoms);
00180
00181 // Insertion operator for the above
00182 std::ostream &operator«(std::ostream &s, const StdRestraintAtoms &std);
00183
00184 // Useful typedefs
00185 typedef std::vector<NoeRestraintNames> NoeRestraintNamesList;
00186 typedef NoeRestraintNamesList::iterator NoeRestraintNamesListIter;
00187 typedef NoeRestraintNamesList::const_iterator NoeRestraintNamesListConstIter;
00188
00189 typedef std::vector<NoeRestraintAtoms> NoeRestraintAtomsList;
00190 typedef NoeRestraintAtomsList::iterator NoeRestraintAtomsListIter;
00191 typedef NoeRestraintAtomsList::const_iterator NoeRestraintAtomsListConstIter;
00192
00193 typedef std::vector<StdRestraintNames> StdRestraintNamesList;
00194 typedef StdRestraintNamesList::iterator StdRestraintNamesListIter;
00195 typedef StdRestraintNamesList::const_iterator StdRestraintNamesListConstIter;
00196
00197 typedef std::vector<StdRestraintAtoms> StdRestraintAtomsList;
00198 typedef StdRestraintAtomsList::iterator StdRestraintAtomsListIter;
00199 typedef StdRestraintAtomsList::const_iterator StdRestraintAtomsListConstIter;
00200
00201 } // namespace rxdock
00202
00203 #endif //_RBTNOERESTRAINT_H_
```

6.100 NonBondedGrid.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
```

```
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Grid for indexing lipophilic interactions
00014
00015 #ifndef _RBTNONBONDEDGRID_H_
00016 #define _RBTNONBONDEDGRID_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/BaseGrid.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // A map of atom vectors indexed by unsigned int
00028 // Used to store the receptor atom lists at each grid point
00029 // DM 11 Jul 2000 - use map of regular Atom* list (not AtomPtr smart
00030 // pointer list)
00031
00032 // DM 3 Nov 2000 - replace map by vector for faster lookup
00033 // typedef std::map<UInt,AtomRList> AtomListMap;
00034 typedef std::vector<AtomRList> AtomListMap;
00035
00036 typedef AtomListMap::iterator AtomListMapIter;
00037 typedef AtomListMap::const_iterator AtomListMapConstIter;
00038
00039 class NonBondedGrid : public BaseGrid {
00040 public:
00041   // Class type string
00042   static const std::string _CT;
00044   // Constructors/destructors
00045   // Construct a NXxNYxNZ grid running from gridMin at gridStep resolution
00046   NonBondedGrid(const Coord &gridMin, const Coord &gridStep, unsigned int NX,
00047                 unsigned int NY, unsigned int NZ, unsigned int NPad = 0);
00048
00049   // Constructor reading all params from binary stream
00050   NonBondedGrid(json j);
00051
00052   ~NonBondedGrid(); // Default destructor
00053
00054   /*friend void to_json(json &j, const NonBondedGrid &nonBondedGrid);
00055   friend void from_json(const json &j, NonBondedGrid &nonBondedGrid);*/
00056
00057   // Copy constructor
00058   NonBondedGrid(const NonBondedGrid &);
00059   // Copy constructor taking base class argument
00060   NonBondedGrid(const BaseGrid &);
00061   // Copy assignment
00062   NonBondedGrid &operator=(const NonBondedGrid &);
00063   // Copy assignment taking base class argument
00064   NonBondedGrid &operator=(const BaseGrid &);
00065
00067   // Virtual functions for reading/writing grid data to streams in
00068   // text and binary format
00069   // Subclasses should provide their own private OwnPrint
00070   // method to handle subclass data members, and override the public
00071   // Print method
00072   virtual void Print(std::ostream &ostr) const; // Text output
00074   // Public methods
00076
00078   // Get attribute functions
00080   // AtomList GetAtomList(UInt iXYZ) const;
00081   // AtomList GetAtomList(const Coord& c) const;
00082   const AtomRList &GetAtomList(unsigned int iXYZ) const;
00083   const AtomRList &GetAtomList(const Coord &c) const;
00084
00086   // Set attribute functions
00088   void SetAtomLists(Atom *pAtom, double radius);
00089   void ClearAtomLists();
00090   void UniqueAtomLists();
00091
00092 protected:
00094   // Protected methods
00096   // Protected method for writing data members for this class to text stream
00097   void OwnPrint(std::ostream &ostr) const;
00098
00099 private:
```

```
00101   // Private methods
00103   NonBondedGrid(); // Disable default constructor
00104
00105   // Helper function called by copy constructor and assignment operator
00106   void CopyGrid(const NonBondedGrid &);
00107   // DM 6 Nov 2000 - create AtomListMap of the appropriate size
00108   void CreateMap();
00109
00110 protected:
00112   // Protected data
00114
00115 private:
00117   // Private data
00119   AtomListMap
00120       m_atomMap; // Used to store the receptor atom lists at each grid point
00121   const AtomRList m_emptyList; // Dummy list used by GetAtomList
00122 };
00123
00124 /*void to_json(json &j, const NonBondedGrid &nonBondedGrid);
00125 void from_json(const json &j, NonBondedGrid &nonBondedGrid);*/
00126
00127 // Useful typedefs
00128 typedef SmartPtr<NonBondedGrid> NonBondedGridPtr; // Smart pointer
00129
00130 } // namespace rxdock
00131
00132 #endif //_RBTNONBONDEDGRID_H_
```

6.101 NonBondedHHSGrid.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTNONBONDEDHHSGRID_H_
00014 #define _RBTNONBONDEDHHSGRID_H_
00015
00016 #include "rxdock/BaseGrid.h"
00017 #include "rxdock/SATypes.h"
00018
00019 namespace rxdock {
00020
00021 // typedefs are in SATypes.h
00022 // rest of the class architecture from NonBondedGrid.h
00023 // but instead Atom HHS_Solvation is used as operand
00024
00025 class NonBondedHHSGrid : public BaseGrid {
00026 public:
00027   static const std::string _CT;
00028   NonBondedHHSGrid(const Coord &gridMin, const Coord &gridStep, unsigned int NX,
00029                    unsigned int NY, unsigned int NZ, unsigned int NPad = 0);
00030   NonBondedHHSGrid(json j);
00031   ~NonBondedHHSGrid();
00032
00033   NonBondedHHSGrid(const NonBondedHHSGrid &);
00034   NonBondedHHSGrid(const BaseGrid &);
00035   NonBondedHHSGrid &operator=(const NonBondedHHSGrid &);
00036   NonBondedHHSGrid &operator=(const BaseGrid &);
00037
00038   virtual void Print(std::ostream &ostr) const;
00039
00040   const HHS_SolvationRList &GetHHSList(unsigned int iXYZ) const;
00041   const HHS_SolvationRList &GetHHSList(const Coord &c) const;
00042
00043   void SetHHSLists(HHS_Solvation *pHHS, double radius);
00044   void ClearHHSLists(void);
00045
00046 protected:
00047   void OwnPrint(std::ostream &ostr) const;
00048
00049 private:
00050   NonBondedHHSGrid();
00051
00052   void CopyGrid(const NonBondedHHSGrid &);
00053   void CreateMap();
```

```
00054
00055   HHS_SolvationListMap m_hhsMap;
00056   const HHS_SolvationRList m_emptyList;
00057 };
00058
00059 typedef SmartPtr<NonBondedHHSGrid> NonBondedHHSGridPtr;
00060
00061 } // namespace rxdock
00062
00063 #endif // _RBTNONBONDEDHHSGRID_H_
```

6.102 NullTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Null transform which does nothing except fire off any requests to the
00014 // scoring function
00015
00016 #ifndef _RBTNULLTRANSFORM_H_
00017 #define _RBTNULLTRANSFORM_H_
00018
00019 #include "rxdock/BaseTransform.h"
00020
00021 namespace rxdock {
00022
00023 class NullTransform : public BaseTransform {
00024 public:
00025   // Static data member for class type
00026   static const std::string _CT;
00027
00029   // Constructors/destructors
00030   NullTransform(const std::string &strName = "NULL");
00031   virtual ~NullTransform();
00032
00034   // Public methods
00036
00037   virtual void Update(Subject *theChangedSubject); // Does nothing
00038
00039 protected:
00041   // Protected methods
00043   virtual void Execute(); // Does nothing
00044
00045 private:
00047   // Private methods
00049   NullTransform(const NullTransform &); // Copy constructor disabled by default
00050   NullTransform &
00051   operator=(const NullTransform &); // Copy assignment disabled by default
00052
00053 protected:
00055   // Protected data
00057
00058 private:
00060   // Private data
00062 };
00063
00064 // Useful typedefs
00065 typedef SmartPtr<NullTransform> NullTransformPtr; // Smart pointer
00066
00067 } // namespace rxdock
00068
00069 #endif //_RBTNULLTRANSFORM_H_
```

6.103 Observer.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
```

```
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Abstract Observer class for Subject-Observer design pattern, see Subject.h
00014 // Design considerations:
00015 // Subclasses should override the Update and Deleted methods, which are invoked
00016 // by the subject when the subject changes state or is about to be deleted,
00017 // respectively.
00018 // Observer subclasses also need to store a pointer to the concrete subject
00019 // subclass (e.g. WorkSpace), either via Register and Unregister methods or
00020 // via the constructor, so that subject subclass methods can be called to get
00021 // details of the change in state. No pointer to the base subject object is
00022 // stored in the base observer object.
00023
00024 #ifndef _RBTOBSERVER_H_
00025 #define _RBTOBSERVER_H_
00026
00027 #include <vector>
00028
00029 namespace rxdock {
00030
00031 class Subject;
00032
00033 class Observer {
00034 public:
00036   // Constructors/destructors
00037   virtual ~Observer(); // Default destructor
00038
00040   // Public methods
00042   // PURE VIRTUAL - DERIVED CLASSES MUST OVERRIDE
00043   // Notify observer that subject has changed
00044   virtual void Update(Subject *theChangedSubject) = 0;
00045   // Notify observer that subject is about to be deleted
00046   virtual void Deleted(Subject *theDeletedSubject) = 0;
00047
00048 protected:
00050   // Protected methods
00052   Observer();
00053
00054 private:
00056   // Private methods
00058   Observer(const Observer &);            // Copy constructor disabled by default
00059   Observer &operator=(const Observer &); // Copy assignment disabled by default
00060
00061 protected:
00063   // Protected data
00065
00066 private:
00068   // Private data
00070 };
00071
00072 // Useful typedefs
00073 typedef std::vector<Observer *> ObserverList;
00074 typedef ObserverList::iterator ObserverListIter;
00075 typedef ObserverList::const_iterator ObserverListConstIter;
00076
00077 } // namespace rxdock
00078
00079 #endif //_RBTOBSERVER_H_
```

6.104 CavitySearch.h File Reference

```
#include ¨rxdock/support/Export.h¨
#include <string>
```

Functions

- RBTDLL_EXPORT int **rxdock::operation::cavitySearch** (std::string strReceptorPrmFile, bool read↩ DockingSite, bool writeDockingSite, bool writeMOEGrid, bool writeInsightII, bool writePsfCrd, bool listAtoms, double listDistance, bool printSiteDescriptors, double border)

  Finds cavities on the receptor.

6.104.1 Detailed Description

CavitySearch operation.

6.105 CavitySearch.h

Go to the documentation of this file.

```
00001 //===-- CavitySearch.h - CavitySearch operation -----------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_OPERATION_CAVITYSEARCH_H
00019 #define RXDOCK_OPERATION_CAVITYSEARCH_H
00020
00021 #include "rxdock/support/Export.h"
00022
00023 #include <string>
00024
00025 namespace rxdock {
00026 namespace operation {
00027
00031 RBTDLL_EXPORT int cavitySearch(std::string strReceptorPrmFile,
00032                                bool readDockingSite, bool writeDockingSite,
00033                                bool writeMOEGrid, bool writeInsightII,
00034                                bool writePsfCrd, bool listAtoms,
00035                                double listDistance, bool printSiteDescriptors,
00036                                double border);
00037
00038 } // namespace operation
00039 } // namespace rxdock
00040
00041 #endif // RXDOCK_OPERATION_CAVITYSEARCH_H
```

6.106 Dock.h File Reference

```
#include ¨rxdock/support/Export.h¨
#include <string>
```

Functions

- RBTDLL_EXPORT int **rxdock::operation::dock** (std::string strLigandMdlFile, std::string strOutputMdlFile, bool bOutputCrd, std::string strOutputCrdFile, bool bOutputHistory, std::string strOutputHistoryFilePrefix, std::string strReceptorPrmFile, std::string strParamFile, bool bFilter, std::string strFilterFile, bool bDocking↩ Runs, std::size_t nDockingRuns, bool bPosIonise, bool bNegIonise, bool bExplH, bool bTarget, double d↩ TargetScore, bool bContinue, bool bSeed, std::size_t nSeed)

  Docks ligand(s) to the receptor.

6.106.1 Detailed Description

Dock operation.

6.107 Dock.h

Go to the documentation of this file.

```
00001 //===-- Dock.h - Dock operation ---------------------------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_OPERATION_DOCK_H
00019 #define RXDOCK_OPERATION_DOCK_H
00020
00021 #include "rxdock/support/Export.h"
00022
00023 #include <string>
00024
00025 namespace rxdock {
00026 namespace operation {
00027
00031 RBTDLL_EXPORT int
00032 dock(std::string strLigandMdlFile, std::string strOutputMdlFile,
00033      bool bOutputCrd, std::string strOutputCrdFile, bool bOutputHistory,
00034      std::string strOutputHistoryFilePrefix, std::string strReceptorPrmFile,
00035      std::string strParamFile, bool bFilter, std::string strFilterFile,
00036      bool bDockingRuns, std::size_t nDockingRuns, bool bPosIonise,
00037      bool bNegIonise, bool bExplH, bool bTarget, double dTargetScore,
00038      bool bContinue, bool bSeed, std::size_t nSeed);
00039
00040 } // namespace operation
00041 } // namespace rxdock
00042
00043 #endif // RXDOCK_OPERATION_DOCK_H
```

6.108 Tabularize.h File Reference

```
#include ¨rxdock/support/Export.h¨
#include <string>
```

Functions

- RBTDLL_EXPORT int **rxdock::operation::tabularize** (std::string inputSDFile, std::string outputCSVFile, std::size_t nAtoms, std::size_t nDataFields)

  Represents structure-data file (SDF) as a CSV table.

6.108.1 Detailed Description

Tabularize operation.

6.109 Tabularize.h

Go to the documentation of this file.

```
00001 //===-- Tabularize.h - Tabularize operation ---------------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_OPERATION_TABULARIZE_H
00019 #define RXDOCK_OPERATION_TABULARIZE_H
00020
00021 #include "rxdock/support/Export.h"
00022
00023 #include <string>
00024
00025 namespace rxdock {
00026 namespace operation {
00027
00031 RBTDLL_EXPORT int tabularize(std::string inputSDFile, std::string outputCSVFile,
00032                              std::size_t nAtoms, std::size_t nDataFields);
00033
00034 } // namespace operation
00035 } // namespace rxdock
00036
00037 #endif // RXDOCK_OPERATION_TABULARIZE_H
```

6.110 Transform.h File Reference

```
#include ¨rxdock/support/Export.h¨
#include <string>
```

Functions

- RBTDLL_EXPORT int **rxdock::operation::transform** (std::string inputSDFile, std::string outputSDFile, bool limitedRecPerMolecule, std::size_t maxNumRecPerMolecule, bool checkName, std::string name, bool checkData, std::string dataFieldName, std::string dataValue, bool changeName, std::string newName, std::size_t bufferSize, bool doSorting, std::string sortDataField, bool limitedRecords, std::size_t maxNum↩ Records)

  Transforms structure-data file (SDF) according to your requirements.

6.110.1 Detailed Description

Transform operation.

6.111 Transform.h

Go to the documentation of this file.

```
00001 //===-- Transform.h - Transform operation ----------------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_OPERATION_TRANSFORM_H
00019 #define RXDOCK_OPERATION_TRANSFORM_H
00020
00021 #include "rxdock/support/Export.h"
00022
00023 #include <string>
00024
00025 namespace rxdock {
00026 namespace operation {
00027
00031 RBTDLL_EXPORT int transform(std::string inputSDFile, std::string outputSDFile,
00032                             bool limitedRecPerMolecule,
00033                             std::size_t maxNumRecPerMolecule, bool checkName,
00034                             std::string name, bool checkData,
00035                             std::string dataFieldName, std::string dataValue,
00036                             bool changeName, std::string newName,
00037                             std::size_t bufferSize, bool doSorting,
00038                             std::string sortDataField, bool limitedRecords,
00039                             std::size_t maxNumRecords);
00040
00041 } // namespace operation
00042 } // namespace rxdock
00043
00044 #endif // RXDOCK_OPERATION_TRANSFORM_H
```

6.112 ParameterFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // File reader for rxdock parameter files
00014
00015 #ifndef _RBTPARAMETERFILESOURCE_H_
00016 #define _RBTPARAMETERFILESOURCE_H_
00017
00018 #include "rxdock/BaseFileSource.h"
00019 #include "rxdock/Variant.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class ParameterFileSource : public BaseFileSource {
00028 public:
00029   // Constructors
00030   ParameterFileSource(const char *fileName);
00031   RBTDLL_EXPORT ParameterFileSource(const std::string &fileName);
00032
00033   // Destructor
00034   virtual ~ParameterFileSource();
00035
00036   friend void to_json(json &j, const ParameterFileSource &parameterFileSrc);
00037   friend void from_json(const json &j, ParameterFileSource &parameterFileSrc);
00038
```

```
00040   // Public methods
00041   RBTDLL_EXPORT std::string GetTitle();
00042   std::string GetVersion();
00043   // DM 06 June 2000 - limit #parameters to those in current section
00044   unsigned int GetNumParameters();
00045   // DM 4 Feb 1999 Renamed from GetParameters()
00046   // std::map<std::string, double> GetParameterMap();
00047   // DM 12 Feb 1999 - only return the list of parameter names, not their values
00048   // DM 06 Jun 2000 - limits params to those in named section
00049   RBTDLL_EXPORT std::vector<std::string> GetParameterList();
00050   // DM 4 Feb 1999 Get a particular named parameter value as a double
00051   double GetParameterValue(const std::string &strParamName);
00052   // DM 12 Feb 1999 Get a particular named parameter value as a string
00053   RBTDLL_EXPORT std::string
00054   GetParameterValueAsString(const std::string &strParamName);
00055   // DM 11 Feb 1999 Check if parameter is present
00056   bool isParameterPresent(const std::string &strParamName);
00057
00058   // DM 11 Feb 1999 - section handling
00059   // Parameters can be grouped into named sections
00060   // such that the same parameter name can appear in multiple sections
00061   // Default namespace is the unnamed section.
00062   // Main use is for simulation protocols which may need to define a variable
00063   // number of phases - e.g. high temperature sampling, cooling phase,
00064   // minimisation phase and need the same parameters to appear in each
00065   int GetNumSections();                       // Number of named sections
00066   std::vector<std::string> GetSectionList(); // List of section names
00067   std::string GetSection() const;             // Get current section name
00068   RBTDLL_EXPORT void
00069   SetSection(const std::string &strSection = ""); // Set current section name
00070
00071 protected:
00072   // Protected methods
00073
00074 private:
00075   // Private methods
00076   ParameterFileSource(); // Disable default constructor
00077   ParameterFileSource(
00078       const ParameterFileSource &); // Copy constructor disabled by default
00079   ParameterFileSource &
00080   operator=(const ParameterFileSource &); // Copy assignment disabled by default
00081
00082   // Pure virtual in BaseFileSource - needs to be defined here
00083   virtual void Parse();
00084   void ClearParamsCache();
00085   // Add a new section name
00086   // Throws an error if section name is empty, or is a duplicate
00087   void AddSection(const std::string &strSection);
00088   // Returns the fully qualified parameter name (<section>::<parameter name>)
00089   // Checks if name already contains a section name, if so just returns the name
00090   // unchanged If not, prefixes the name with the current section name
00091   std::string GetFullParameterName(const std::string &strParamName);
00092
00093 protected:
00094   // Protected data
00095
00096 private:
00097   // Private data
00098   std::string m_inputFileName;
00099   std::string m_strTitle;
00100   std::string m_strVersion;
00101   StringVariantMap m_paramsMap;
00102   std::vector<std::string> m_sectionNames; // List(vector) of section names
00103   std::string m_strSection;                // Current section
00104   bool m_json;
00105 };
00106
00107 void to_json(json &j, const ParameterFileSource &parameterFileSrc);
00108 void from_json(const json &j, ParameterFileSource &parameterFileSrc);
00109
00110 // useful typedefs
00111 typedef SmartPtr<ParameterFileSource> ParameterFileSourcePtr; // Smart pointer
00112
00113 } // namespace rxdock
00114
00115 #endif //_RBTPARAMETERFILESOURCE_H_
```

6.113 ParamHandler.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
```

```
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Base implementation class for managing a collection of named parameters
00014 // Any class requiring parameter handling can derive from ParamHandler
00015 // Parameters are stored as Variants (double, string or stringlist)
00016 // Only derived classes can add and delete parameters from the collection
00017
00018 #ifndef _RBTPARAMHANDLER_H_
00019 #define _RBTPARAMHANDLER_H_
00020
00021 #include "rxdock/Config.h"
00022 #include "rxdock/Variant.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
00030 class ParamHandler {
00031 public:
00033   // Constructors/destructors
00034   virtual ~ParamHandler(); // Default destructor
00035
00036   friend void to_json(json &j, const ParamHandler &paramHandler);
00037   friend void from_json(const json &j, ParamHandler &paramHandler);
00038
00040   // Public methods
00042   // Get number of stored parameters
00043   unsigned int GetNumParameters() const;
00044   // Get a named parameter, throws error if name not found
00045   Variant GetParameter(const std::string &strName) const;
00046   // Check if named parameter is present
00047   bool isParameterValid(const std::string &strName) const;
00048   // Get list of all parameter names
00049   std::vector<std::string> GetParameterNames() const;
00050   // Get list of all parameter
00051   StringVariantMap GetParameters() const;
00052
00053   // Set named parameter to new value, throws error if name not found
00054   RBTDLL_EXPORT void SetParameter(const std::string &strName,
00055                                   const Variant &vValue);
00056
00057   // Virtual function for dumping parameters to an output stream
00058   // Called by operator «
00059   virtual void Print(std::ostream &s) const;
00060
00061 protected:
00063   // Protected methods
00065   ParamHandler(); // Default constructor
00066
00067   // Only derived classes can mess with the parameter list
00068   void AddParameter(const std::string &strName, const Variant &vValue);
00069   void DeleteParameter(const std::string &strName);
00070   void ClearParameters();
00071   // DM 25 Oct 2000 - ParameterUpdated is invoked whenever SetParameter is
00072   // called to allow derived classes to manage a data member which tracks the
00073   // param value Useful for performance purposes as there is quite an overhead
00074   // in finding a string in a map, then converting from a Variant to the native
00075   // datatype Base class version does nothing
00076   virtual void ParameterUpdated(const std::string &strName) {}
00077
00078 private:
00080   // Private methods
00082
00083   ParamHandler(const ParamHandler &); // Copy constructor disabled by default
00084   ParamHandler &
00085   operator=(const ParamHandler &); // Copy assignment disabled by default
00086
00087 protected:
00089   // Protected data
00091
00092 private:
00094   // Private data
00096   StringVariantMap m_parameters;
00097 };
00098
00099 void to_json(json &j, const ParamHandler &paramHandler);
00100 void from_json(const json &j, ParamHandler &paramHandler);
00101
00103 // Non-member functions
```

```
00104
00105 // Insertion operator (primarily for debugging)
00106 RBTDLL_EXPORT std::ostream &operator«(std::ostream &s, const ParamHandler &ph);
00107
00108 } // namespace rxdock
00109
00110 #endif //_RBTPARAMHANDLER_H_
```

6.114 Parser.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Parser class. Takes a list of tokens and parsers them to create a
00014 // Filter Expression
00015
00016 #ifndef _RBT_PARSER_H_
00017 #define _RBT_PARSER_H_
00018
00019 #include "rxdock/Context.h"
00020 #include "rxdock/FilterExpression.h"
00021 #include "rxdock/Token.h"
00022 #include "rxdock/TokenIter.h"
00023
00024 namespace rxdock {
00025
00026 class FilterExpression;
00027
00028 class Parser {
00029 public:
00030   static const std::string _CT;
00032   // Constructors
00034   Parser(const Parser &);
00035   Parser();
00036   FilterExpressionPtr Parse(TokenIterPtr, ContextPtr);
00037
00039   // Destructor
00041   virtual ~Parser();
00042
00044   // Private methods
00046
00047 private:
00048 };
00049
00050 // Useful typedefs
00051 typedef SmartPtr<Parser> ParserPtr;        // Smart pointer
00052 typedef std::vector<ParserPtr> ParserList; // Vector of smart pointers
00053 typedef ParserList::iterator ParserListIter;
00054 typedef ParserList::const_iterator ParserListConstIter;
00055
00056 // Compare class to compare different genomes inside a population
00057 // This is used to sort the genomes in a population depending
00058 // of the value of their scoring function.
00059
00060 } // namespace rxdock
00061
00062 #endif //_Parser_H_
```

6.115 PdbFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
```

```
00011  ***************************************************************************/
00012
00013 // Handles retrieval of atomic (coordinate) info
00014 // from XPLOR PDB files (with segment name in last column).
00015
00016 #ifndef _RBTPDBFILESOURCE_H_
00017 #define _RBTPDBFILESOURCE_H_
00018
00019 #include "rxdock/BaseMolecularFileSource.h"
00020
00021 namespace rxdock {
00022
00023 class PdbFileSource : public BaseMolecularFileSource {
00024 public:
00025   // Constructors
00026   PdbFileSource(const char *fileName);
00027   PdbFileSource(const std::string fileName);
00028
00029   // Default destructor
00030   virtual ~PdbFileSource();
00031
00033   // Override public methods from BaseMolecularDataSource
00034   virtual bool isTitleListSupported() { return true; }
00035   virtual bool isAtomListSupported() { return true; }
00036   virtual bool isCoordinatesSupported() { return true; }
00037   virtual bool isBondListSupported() { return false; }
00038
00039 protected:
00040   // Pure virtual in BaseFileSource - needs to be defined here
00041   virtual void Parse();
00042
00043 private:
00044   // Private methods
00045   PdbFileSource();                      // Disable default constructor
00046   PdbFileSource(const PdbFileSource &); // Copy constructor disabled by default
00047   PdbFileSource &
00048   operator=(const PdbFileSource &); // Copy assignment disabled by default
00049 };
00050
00051 // useful typedefs
00052 typedef SmartPtr<PdbFileSource> PdbFileSourcePtr; // Smart pointer
00053
00054 } // namespace rxdock
00055
00056 #endif //_RBTPDBFILESOURCE_H_
```

6.116 PharmaSF.h

```
00001 /***************************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***************************************************************************/
00012
00013 // Pharmacophore distance restraint scoring function
00014
00015 #ifndef _RBTPHARMASF_H_
00016 #define _RBTPHARMASF_H_
00017
00018 #include "rxdock/BaseInterSF.h"
00019 #include "rxdock/BaseMolecularFileSink.h"
00020 #include "rxdock/Constraint.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class PharmaSF : public BaseInterSF {
00029 public:
00030   // Class type string
00031   static const std::string _CT;
00032   // Parameter names
00033   static const std::string _CONSTRAINTS_FILE;
00034   static const std::string _OPTIONAL_FILE;
00035   static const std::string _NOPT;
```

```
00036   static const std::string _WRITE_ERRORS;
00037
00038   PharmaSF(const std::string &strName = "pharma");
00039   virtual ~PharmaSF();
00040
00041   friend void to_json(json &j, const PharmaSF &pharmaSF);
00042   friend void from_json(const json &j, PharmaSF &pharmaSF);
00043
00044   // Override BaseSF::ScoreMap to provide additional raw descriptors
00045   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00046
00047 protected:
00048   virtual void SetupReceptor();
00049   virtual void SetupLigand();
00050   virtual void SetupScore();
00051   virtual double RawScore() const;
00052   // DM 25 Oct 2000 - track changes to parameter values in local data members
00053   // ParameterUpdated is invoked by ParamHandler::SetParameter
00054   void ParameterUpdated(const std::string &strName);
00055
00056 private:
00057   ConstraintList m_constrList;
00058   ConstraintList m_optList;
00059   int m_nopt;
00060   MolecularFileSinkPtr m_spErrorFile;
00061   bool m_bWriteErrors;
00062   // Keep track of individual constraint scores for ScoreMap
00063   mutable std::vector<double> m_conScores; // Mandatory constraint scores
00064   mutable std::vector<double> m_optScores; // Optional constraint scores
00065 };
00066
00067 void to_json(json &j, const PharmaSF &pharmaSF);
00068 void from_json(const json &j, PharmaSF &pharmaSF);
00069
00070 } // namespace rxdock
00071
00072 #endif //_RBTPHARMASF_H_
```

6.117 Plane.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple class for representing 3-D planes in normal form
00014
00015 #ifndef _RBTPLANE_H_
00016 #define _RBTPLANE_H_
00017
00018 #include "rxdock/Coord.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 class Plane {
00028   // Data members
00030 private:
00031   Vector m_vnorm; // The vector normal to the plane (a,b,c)
00032   double m_d;     // Distance from plane to origin (in normal form)
00033
00034 public:
00036   // Constructors/destructors:
00038
00039   // Default constructor (initialise to zero)
00040   inline Plane() : m_vnorm(0.0, 0.0, 0.0), m_d(0.0) {}
00041
00042   // Constructor with initial values
00043   inline Plane(const Vector &v, double d) : m_vnorm(v), m_d(d) {
00044     Normalise(); // Convert to normal form
00045   }
00046
00047   // Constructor taking three coordinates which lie in the plane
```

```
00048   inline Plane(const Coord &c0, const Coord &c1, const Coord &c2) {
00049     // Determine the vectors from c0 to c1 and c0 to c2
00050     Vector v1 = c1 - c0;
00051     Vector v2 = c2 - c0;
00052     m_vnorm = v1.Cross(v2); // Vector normal to the plane
00053     m_d = -m_vnorm.Dot(c0);
00054     Normalise(); // Convert to normal form
00055   }
00056
00057   // Destructor
00058   virtual ~Plane() {}
00059
00060   friend void to_json(json &j, const Plane &plane) {
00061     j = json{{"vec-nor", plane.m_vnorm}, {"dis", plane.m_d}};
00062   }
00063
00064   friend void from_json(const json &j, Plane &plane) {
00065     j.at("vec-nor").get_to(plane.m_vnorm);
00066     j.at("dis").get_to(plane.m_d);
00067   }
00068
00070   // Operator functions:
00072
00074   // Friend functions:
00076
00077   // Insertion operator
00078   friend std::ostream &operator«(std::ostream &s, const Plane &plane) {
00079     return s « plane.m_vnorm.xyz(0) « "x + " « plane.m_vnorm.xyz(1) « "y + "
00080              « plane.m_vnorm.xyz(2) « "z + " « plane.m_d « " = 0";
00081   }
00082
00084   // Public methods
00086   inline double D() const { return m_d; }
00087   inline Vector VNorm() const { return m_vnorm; }
00088
00090   // Private methods
00092   inline void Normalise() {
00093     // Convert to normal form by dividing by +/- vnorm.Length()
00094     // where sign is opposite the sign of d when d<>0,
00095     // same as sign of c (vnorm.xyz(2)) when d==0 and c<>0,
00096     // and same as sign of b (vnorm.xyz(1)) otherwise
00097     double l = m_vnorm.Length();
00098     int iSign = (m_d < 0.0)              ? 1
00099                 : (m_d > 0.0)            ? -1
00100                : (m_vnorm.xyz(2) < 0.0) ? -1
00101                : (m_vnorm.xyz(2) > 0.0) ? 1
00102                : (m_vnorm.xyz(1) < 0.0) ? -1
00103                                         : 1;
00104     l *= iSign;
00105     // Check for divide by zero
00106     if (l != 0.0) {
00107       m_vnorm /= l;
00108       m_d /= l;
00109     }
00110   }
00111 };
00112
00114 // Non-member functions (in rxdock namespace)
00116
00117 // Since our planes are always in normal form, we don't need to divide
00118 // by the magnitude of the normal vector
00119 // DM 31 Mar 1999 - returned signed distance (not absolute value)
00120 inline double DistanceFromPointToPlane(const Coord &c0, const Plane &p) {
00121   // return std::fabs( c0.Dot(p.VNorm()) + p.D() );
00122   return c0.Dot(p.VNorm()) + p.D();
00123 }
00124
00125 // Angle between planes is just the angle between the normal vectors
00126 inline double AngleBetweenPlanes(const Plane &p0, const Plane &p1) {
00127   return Angle(p0.VNorm(), p1.VNorm());
00128 }
00129
00130 // Cos(Angle) between planes is just the dot product of the normal vectors
00131 inline double CosAngleBetweenPlanes(const Plane &p0, const Plane &p1) {
00132   return Dot(p0.VNorm(), p1.VNorm());
00133 }
00134
00135 } // namespace rxdock
00136
00137 #endif //_RBTPLANE_H_
```

6.118 PMF.h

```
00001 /*********************************************************************
```

```
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTPMF_H_
00014 #define _RBTPMF_H_
00015
00016 // #include <vector>        // STL vector
00017 #include <string>
00018
00019 namespace rxdock {
00020
00024 typedef struct {
00025   float distance;
00026   float density;
00027 } PMFValue;
00028
00029 typedef enum {
00030   CF = 1,
00031   CP,
00032   cF,
00033   cP,
00034   C3,
00035   CW,
00036   CO,
00037   CN,
00038   C0,
00039   NC, // 10
00040   NP,
00041   NA,
00042   ND,
00043   NR,
00044   N0,
00045   NS,
00046   OC,
00047   OA,
00048   OE,
00049   OR, // 20
00050   OS,
00051   OD,
00052   OW,
00053   P,
00054   SA,
00055   SD,
00056   HL,
00057   HH,
00058   Zn,
00059   CL, // 30
00060   Mn,
00061   Mg,
00062   F,
00063   Fe,
00064   Br,
00065   V,
00066   PMF_UNDEFINED
00067 } PMFType;
00068
00069 typedef PMFType PMFType;
00070
00071 std::string PMFType2Str(PMFType aType);
00072 PMFType PMFStr2Type(std::string anStr);
00073
00074 } // namespace rxdock
00075
00076 #endif // _RBTPMF_H_
```

6.119 PMFDirSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
```

```
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 #ifndef _RBTPMFDIRSOURCE_H_
00014 #define _RBTPMFDIRSOURCE_H_
00015
00016 #include "rxdock/DirectorySource.h"
00017 #include "rxdock/PMF.h"
00018
00019 namespace rxdock {
00020
00021 class PMFDirSource : public DirectorySource {
00022 public:
00023   PMFDirSource(const std::string &);
00024   virtual ~PMFDirSource() {}
00025
00026   virtual void ReadFiles(std::vector<std::vector<PMFValue» *aVect,
00027                          std::vector<std::string> *aNameVect,
00028                          std::vector<PMFValue> *aSlopeVect);
00029   virtual void ParseLines(std::vector<std::string> anStrVect,
00030                           std::vector<PMFValue> *aValueVect);
00031 };
00032
00033 } // namespace rxdock
00034
00035 #endif // _RBTPMFDIRSOURCE_H_
```

6.120 PMFGridSF.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 #ifndef _RBTPMFGRIDSF_H_
00014 #define _RBTPMFGRIDSF_H_
00015
00016 #include "rxdock/BaseInterSF.h"
00017 #include "rxdock/RealGrid.h"
00018
00019 #include <nlohmann/json.hpp>
00020
00021 using json = nlohmann::json;
00022
00023 namespace rxdock {
00024
00025 class PMFGridSF : public BaseInterSF {
00026   bool m_bSmoothed;
00027   AtomList theLigandList;            // vector to store the ligand
00028   std::vector<PMFType> theTypeList;  // store PMF used types here
00029   std::vector<RealGridPtr> theGrids; // grids with PMF data
00030
00031   void ReadGrids(json pmfGrids);
00032
00033 public:
00034   static const std::string _CT;   // class name
00035   static const std::string _GRID; // filename extension (.grd)
00036   static const std::string
00037       _SMOOTHED; // controls wether to smooth the grid values
00038
00039   PMFGridSF(const std::string &strName = "pmf-grid");
00040   virtual ~PMFGridSF();
00041
00042 protected:
00043   virtual void SetupReceptor();
00044   virtual void SetupLigand();
00045   virtual void SetupScore() {}
00046   virtual double RawScore() const;
00047   unsigned int GetCorrectedType(PMFType aType) const;
00048
00049 private:
00050   friend void to_json(json &j, const PMFGridSF &sf) {
00051     json atomList;
00052     for (const auto &cIter : sf.theLigandList) {
00053       json atom = *cIter;
```

```
00054       atomList.push_back(atom);
00055     }
00056     json grids;
00057     for (const auto &cIter : sf.theGrids) {
00058       json grid = *cIter;
00059       grids.push_back(grid);
00060     }
00061
00062     j = json{{"smoothed", sf.m_bSmoothed},
00063              {"atoms", atomList},
00064              {"pmf-types", sf.theTypeList},
00065              {"pmf-data", grids}};
00066   };
00067   friend void from_json(const json &j, PMFGridSF &sf) {
00068     j.at("smoothed").get_to(sf.m_bSmoothed);
00069
00070     sf.theLigandList.clear();
00071     sf.theLigandList.reserve(j.at("atoms").size());
00072     for (auto &atom : j.at("atoms")) {
00073       AtomPtr spAtom = AtomPtr(new Atom(atom));
00074       sf.theLigandList.push_back(spAtom);
00075     }
00076     j.at("pmf-types").get_to(sf.theTypeList);
00077
00078     sf.theGrids.clear();
00079     sf.theGrids.reserve(j.at("grids").size());
00080     for (auto &grid : j.at("grids")) {
00081       RealGridPtr spGrid = RealGridPtr(new RealGrid(grid));
00082       sf.theGrids.push_back(spGrid);
00083     }
00084   };
00085 };
00086
00087 } // namespace rxdock
00088
00089 #endif // _RBTPMFGRIDSF_H_
```

6.121 PMFIdxSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Indexed-grid-based PMF intermolecular scoring function
00014
00015 #ifndef _RBTPMFIDXSF_H_
00016 #define _RBTPMFIDXSF_H_
00017
00018 #include "rxdock/AnnotationHandler.h"
00019 #include "rxdock/BaseIdxSF.h"
00020 #include "rxdock/BaseInterSF.h"
00021 #include "rxdock/PMF.h"
00022 // for PMF pseudogrids where
00023 // X is distance, X and Y are receptor and ligand
00024 // distances respectively
00025 #include "rxdock/RealGrid.h"
00026
00027 namespace rxdock {
00028
00031 class PMFIdxSF : public BaseInterSF,
00032                  public BaseIdxSF,
00033                  public virtual AnnotationHandler {
00034   // by default "private" members
00038   AtomList theLigandList;   // ligand typing
00039   AtomList theReceptorList; // recepor typing
00040   // regular lists
00041   AtomRList theLigandRList;   // ligand
00042   AtomRList theReceptorRList; // recepor
00043   std::vector<RealGridPtr>
00044       theTypeGrid; // grids for PMF values for different atom types in receptor
00045   std::vector<PMFType> theLigandTypes; // type values in theTypeGrid
00046   NonBondedGridPtr theSurround;        // atoms arond a gridpoint
00047   RealGridPtr thePMFGrid;              // grid for X-distance Y
00048                                        // this is the representation of the PMFs
00049   RealGridPtr theSlopeGrid; // grid to store values where the plateaus starts
```

```
00050
00051 public:
00052   PMFIdxSF(const std::string &strName = "pmf");
00053   virtual ~PMFIdxSF();
00060   static const std::string _CT;
00061   static const std::string
00062       _PMFDIR;
00063   static const std::string _CC_CUTOFF;
00064   static const std::string _SLOPE;
00069   virtual void Update(Subject *theChangedSubject);
00070
00071 protected:
00076   virtual void SetupReceptor();
00080   virtual void SetupLigand();
00084   virtual void SetupScore();
00088   virtual double RawScore(void) const;
00092   double GetLinearCloseRangeValue(double aDist, PMFType aRecType,
00093                                   PMFType aLigType) const;
00094 };
00095
00096 } // namespace rxdock
00097
00098 #endif // _RBTPMFIDXSF_H_
```

6.122 PolarIdxSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Indexed-grid-based intermolecular scoring function for all attractive polar
00014 // interactions (HBD,HBA,metal,guanidinium carbon)
00015
00016 #ifndef _RBTPOLARIDXSF_H_
00017 #define _RBTPOLARIDXSF_H_
00018
00019 #include "rxdock/BaseIdxSF.h"
00020 #include "rxdock/BaseInterSF.h"
00021 #include "rxdock/PolarSF.h"
00022
00023 namespace rxdock {
00024
00025 class PolarIdxSF : public BaseInterSF, public BaseIdxSF, public PolarSF {
00026 public:
00027   // Class type string
00028   static const std::string _CT;
00029   // Parameter names
00030   static const std::string _INCR;
00031   static const std::string _ATTR;
00032   // DM 12 Jun 2002 - score thresholds used for counting attractive and
00033   // repulsive interactions
00034   static const std::string _THRESHOLD_POS;
00035   static const std::string _THRESHOLD_NEG;
00036
00037   RBTDLL_EXPORT static const std::string &GetIncr();
00038
00039   RBTDLL_EXPORT PolarIdxSF(const std::string &strName = "polar");
00040   virtual ~PolarIdxSF();
00041
00042   // Override BaseSF::ScoreMap to provide additional raw descriptors
00043   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00044
00045 protected:
00046   virtual void SetupReceptor();
00047   virtual void SetupLigand();
00048   virtual void SetupSolvent();
00049   virtual void SetupScore();
00050   virtual double RawScore() const;
00051
00052   // Clear the receptor and ligand grids and lists respectively
00053   // As we are not using smart pointers, there is some memory management to do
00054   void ClearReceptor();
00055   void ClearLigand();
00056   void ClearSolvent();
00057   // Helper function for above
```

```
00058   void DeleteList(InteractionCenterList &icList);
00059
00060   // DM 25 Oct 2000 - track changes to parameter values in local data members
00061   // ParameterUpdated is invoked by ParamHandler::SetParameter
00062   void ParameterUpdated(const std::string &strName);
00063
00064 private:
00065   double ReceptorScore() const;
00066   double SolventScore() const;
00067   double InterScore() const;
00068   double ReceptorSolventScore() const;
00069   double LigandSolventScore() const;
00070
00071   double InterScore(const InteractionCenterList &posList,
00072                     const InteractionCenterList &negList, bool bCount) const;
00073   InteractionGridPtr m_spPosGrid;
00074   InteractionGridPtr m_spNegGrid;
00075   InteractionCenterList m_recepPosList;
00076   InteractionCenterList m_recepNegList;
00077   InteractionCenterList m_flexRecPosList;
00078   InteractionCenterList m_flexRecNegList;
00079   InteractionListMap m_flexRecIntns;
00080   InteractionListMap m_flexRecPrtIntns; // partitioned interactions
00081
00082   InteractionCenterList m_ligPosList;
00083   InteractionCenterList m_ligNegList;
00084
00085   InteractionCenterList m_solventPosList;
00086   InteractionCenterList m_solventNegList;
00087   InteractionListMap m_solventIntns;
00088
00089   bool m_bAttr;
00090   bool m_bFlexRec;
00091   bool m_bSolvent;
00092   // DM 12 Jun 2002 - keep track of number of ligand atoms involved in non-zero
00093   // polar interactions
00094   mutable int m_nPos; //#positive centers with non-zero scores
00095   mutable int m_nNeg; //#negative centers with non-zero scores
00096   double m_posThreshold;
00097   double m_negThreshold;
00098 };
00099
00100 } // namespace rxdock
00101
00102 #endif //_RBTPOLARIDXSF_H_
```

6.123 PolarIntraSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Ligand intramolecular scoring function for all attractive polar
00014 // interactions (HBD,HBA,metal,guanidinium carbon)
00015
00016 #ifndef _RBTPOLARINTRASF_H_
00017 #define _RBTPOLARINTRASF_H_
00018
00019 #include "rxdock/BaseIntraSF.h"
00020 #include "rxdock/InteractionGrid.h"
00021 #include "rxdock/PolarSF.h"
00022
00023 #include <nlohmann/json.hpp>
00024
00025 using json = nlohmann::json;
00026
00027 namespace rxdock {
00028
00029 class PolarIntraSF : public BaseIntraSF, public PolarSF {
00030 public:
00031   // Class type string
00032   static const std::string _CT;
00033   // Parameter names
00034   static const std::string _ATTR;
00035
```

```
00036   PolarIntraSF(const std::string &strName = "polar");
00037   virtual ~PolarIntraSF();
00038
00039   friend void to_json(json &j, const PolarIntraSF &polarIntraSF);
00040   friend void from_json(const json &j, PolarIntraSF &polarIntraSF);
00041
00042 protected:
00043   virtual void SetupScore();
00044   virtual double RawScore() const;
00045
00046   // Request Handling method
00047   // Handles the Partition request
00048   virtual void HandleRequest(RequestPtr spRequest);
00049
00050   // Clear the model lists
00051   // As we are not using smart pointers, there is some memory management to do
00052   void ClearModel();
00053
00054   // DM 25 Oct 2000 - track changes to parameter values in local data members
00055   // ParameterUpdated is invoked by ParamHandler::SetParameter
00056   void ParameterUpdated(const std::string &strName);
00057
00058 private:
00059   InteractionCenterList m_posList;
00060   InteractionCenterList m_negList;
00061   InteractionListMap m_intns;
00062   InteractionListMap m_prtIntns;
00063   bool m_bAttr;
00064 };
00065
00066 void to_json(json &j, const PolarIntraSF &polarIntraSF);
00067 void from_json(const json &j, PolarIntraSF &polarIntraSF);
00068
00069 } // namespace rxdock
00070
00071 #endif //_RBTPOLARINTRASF_H_
```

6.124 PolarSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base implementation class for all attractive and repulsive polar scoring fns
00014 // Provides methods for: creating interaction centers, calculating interaction
00015 // scores
00016
00017 #ifndef _RBTPOLARSF_H_
00018 #define _RBTPOLARSF_H_
00019
00020 #include "rxdock/AnnotationHandler.h"
00021 #include "rxdock/BaseSF.h"
00022 #include "rxdock/InteractionGrid.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
00030 class PolarSF : public virtual BaseSF, public virtual AnnotationHandler {
00031 public:
00032   // Class type string
00033   static const std::string _CT;
00034   // Parameter names
00035   static const std::string _INCR;
00036   static const std::string _R12FACTOR;
00037   static const std::string _R12INCR;
00038   static const std::string _DR12MIN;
00039   static const std::string _DR12MAX;
00040   static const std::string _A1;
00041   static const std::string _DA1MIN;
00042   static const std::string _DA1MAX;
00043   static const std::string _A2;
00044   static const std::string _DA2MIN;
```

```
00045   static const std::string _DA2MAX;
00046   static const std::string _INCMETAL;
00047   static const std::string _INCHBD;
00048   static const std::string _INCHBA;
00049   static const std::string _INCGUAN;
00050   static const std::string _GUAN_PLANE;
00051   static const std::string _ABS_DR12;
00052   static const std::string _LP_OSP2;
00053   static const std::string _LP_PHI;
00054   static const std::string _LP_DPHIMIN;
00055   static const std::string _LP_DPHIMAX;
00056   static const std::string _LP_DTHETAMIN;
00057   static const std::string _LP_DTHETAMAX;
00058
00059   virtual ~PolarSF();
00060
00061   friend void to_json(json &j, const PolarSF &polarSF);
00062   friend void from_json(const json &j, PolarSF &polarSF);
00063
00064 protected:
00065   PolarSF();
00066
00067   InteractionCenterList
00068   CreateAcceptorInteractionCenters(const AtomList &atomList) const;
00069   InteractionCenterList
00070   CreateDonorInteractionCenters(const AtomList &atomList) const;
00071
00072   // Index the intramolecular interactions between two lists
00073   void BuildIntraMap(const InteractionCenterList &ICList1,
00074                      const InteractionCenterList &ICList2,
00075                      InteractionListMap &intns) const;
00076   // Index the intramolecular interactions within a single list
00077   void BuildIntraMap(const InteractionCenterList &ICList,
00078                      InteractionListMap &intns) const;
00079
00080   double IntraScore(const InteractionCenterList &posList,
00081                     const InteractionCenterList &negList,
00082                     const InteractionListMap &prtIntns, bool attr) const;
00083   void Partition(const InteractionCenterList &posList,
00084                  const InteractionCenterList &negList,
00085                  const InteractionListMap &intns, InteractionListMap &prtIntns,
00086                  double dist = 0.0) const;
00087
00088   // Generic scoring function params
00089   struct f1prms {
00090     double R0, DRMin, DRMax, slope;
00091     f1prms(double R, double DMin, double DMax)
00092         : R0(R), DRMin(DMin), DRMax(DMax), slope(1.0 / (DMax - DMin)) {}
00093   };
00094
00095   inline f1prms GetRprms() const { return f1prms(0.0, m_DR12Min, m_DR12Max); }
00096   inline f1prms GetA1prms() const { return f1prms(m_A1, m_DA1Min, m_DA1Max); }
00097   inline f1prms GetA2prms() const { return f1prms(m_A2, m_DA2Min, m_DA2Max); }
00098
00099   double PolarScore(const InteractionCenter *intn,
00100                     const InteractionCenterList &intnList, const f1prms &Rprms,
00101                     const f1prms &A1prms, const f1prms &A2prms) const;
00102
00103   // As this has a virtual base class we need a separate OwnParameterUpdated
00104   // which can be called by concrete subclass ParameterUpdated methods
00105   // See Stroustrup C++ 3rd edition, p395, on programming virtual base classes
00106   void OwnParameterUpdated(const std::string &strName);
00107
00108 private:
00109   // Generic scoring function primitive
00110   inline double f1(double DR, const f1prms &prms) const {
00111     return (DR > prms.DRMax)   ? 0.0
00112            : (DR > prms.DRMin) ? 1.0 - prms.slope * (DR - prms.DRMin)
00113                                : 1.0;
00114   }
00115
00116   void UpdateLPprms();
00117
00118   // DM 25 Oct 2000 - heavily used params
00119   double m_R12Factor;
00120   double m_R12Incr;
00121   double m_DR12Min;
00122   double m_DR12Max;
00123   double m_A1;
00124   double m_DA1Min;
00125   double m_DA1Max;
00126   double m_A2;
00127   double m_DA2Min;
00128   double m_DA2Max;
00129   bool m_bAbsDR12;
00130   double m_LP_PHI;
00131   double m_LP_DPHIMin;
```

```
00132   double m_LP_DPHIMax;
00133   double m_LP_DTHETAMin;
00134   double m_LP_DTHETAMax;
00135   flprms m_PHI_lp_prms;
00136   flprms m_PHI_plane_prms;
00137   flprms m_THETAprms;
00138 };
00139
00140 void to_json(json &j, const PolarSF &polarSF);
00141 void from_json(const json &j, PolarSF &polarSF);
00142
00143 } // namespace rxdock
00144
00145 #endif //_RBTPOLARSF_H_
```

6.125 Population.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages a population of genomes.
00014 // Main public method (GAstep) performs one iteration of a GA.
00015 #ifndef _RBTPOPULATION_H_
00016 #define _RBTPOPULATION_H_
00017
00018 #include "rxdock/Error.h"
00019 #include "rxdock/Genome.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class BaseSF; // forward definition
00028
00029 class Population {
00030 public:
00031   static const std::string _CT;
00032   // Sole constructor to create a randomised genome population of a fixed size.
00033   // pChr is the seed chromosome to clone to create each genome.
00034   // size is the population size to create.
00035   // pSF is the scoring function used to rank the genomes.
00036   // Notes:
00037   // 1) Each genome is randomised by calling the Randomise() method
00038   // of the underlying chromosome.
00039   // 2) Scoring function scores are calculated.
00040   // 3) Genomes are sorted by score.
00041   // 4) Roulette wheel fitness values are calculated.
00042   // 5) Model coords are updated to match the fittest chromosome
00043   // An BadArgument error is thrown if size is <=0, or if pChr or pSF is
00044   // null.
00045   RBTDLL_EXPORT Population(ChromElement *pChr, int size, BaseSF *pSF);
00046   virtual ~Population();
00047
00048   friend void to_json(json &j, const Population &population);
00049   friend void from_json(const json &j, Population &population);
00050
00051   // Gets the maximum size of the population as defined in the constructor.
00052   int GetMaxSize() const { return m_size; }
00053   // Gets the actual size of the population (may be < GetMaxSize())
00054   int GetActualSize() const { return m_pop.size(); }
00055   // Gets the best genome in the sorted population (element zero).
00056   RBTDLL_EXPORT GenomePtr Best() const;
00057   // Gets the average raw score across entire population
00058   double GetScoreMean() const { return m_scoreMean; }
00059   // Gets the raw score variance across entire population
00060   double GetScoreVariance() const { return m_scoreVariance; }
00061   // Gets the vector of genomes in the population.
00062   const GenomeList &GetGenomeList() const { return m_pop; }
00063
00064   // Gets the scoring function used for ranking genomes
00065   BaseSF *GetSF() const { return m_pSF; }
00066   // Sets the scoring function used for ranking genomes
```

```
00067   // Forces a recalculation of the genome scores, and the roulette wheel fitness
00068   // values SetSF should be called whenever the scoring function parameters have
00069   // changed e.g. in between GA stages. An BadArgument error is thrown if pSF
00070   // is null. Model coords are updated to match the fittest chromosome
00071   void SetSF(BaseSF *pSF);
00072
00073   // Main method for performing a GA iteration
00074   RBTDLL_EXPORT void
00075   GAstep(int nReplicates,    // Number of new genomes to create in the iteration
00076          double relStepSize, // Relative step size for chromosome mutations
00077          double equalityThreshold, // Equality threshold for genomes
00078          double pcross,            // Probability of crossover
00079          bool xovermut, // if true, perform cauchy mutation following crossover
00080          bool cmutate   // true=cauchy mutations, false=regular mutations
00081   );
00082   RBTDLL_EXPORT GenomePtr RouletteWheelSelect() const;
00083
00084   void Print(std::ostream &) const;
00085   friend std::ostream &operator«(std::ostream &, const Population &);
00086
00087 private:
00088   // Merges the new individuals created into the main population
00089   // Duplicate genomes are removed (based on equality of chromosome elements,
00090   // not scores)
00091   void MergeNewPop(GenomeList &newPop, double equalityThreshold);
00092   void EvaluateRWFitness();
00093   Population(const Population &);            // Disable
00094   Population &operator=(const Population &); // Disable
00095
00096   GenomeList m_pop;       // The population of genomes
00097   unsigned int m_size;    // The maximum size of the population
00098   double m_c;             // Sigma Truncation Multiplier
00099   BaseSF *m_pSF;          // The scoring function
00100   Rand &m_rand;           // reference to the singleton random number generator
00101   double m_scoreMean;     // the average raw score across all genomes
00102   double m_scoreVariance; // the variance of raw scores across all genomes
00103 };
00104
00105 void to_json(json &j, const Population &population);
00106 void from_json(const json &j, Population &population);
00107
00108 typedef SmartPtr<Population> PopulationPtr;
00109
00110 } // namespace rxdock
00111
00112 #endif //_RBTPOPULATION_H_
```

6.126 PrincipalAxes.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Principal axes calculation routines (in rxdock namespace)
00014
00015 #ifndef _RBTPRINCIPALAXES_H_
00016 #define _RBTPRINCIPALAXES_H_
00017
00018 #include "rxdock/Atom.h"
00019 #include "rxdock/Config.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // Struct for holding principal axes info
00028 // Center of mass, three principal axis vectors, three principal moments
00029 class PrincipalAxes {
00030 public:
00031   // Default is X,Y,Z cartesian axes centered at origin
00032   PrincipalAxes()
00033       : com(0.0, 0.0, 0.0), axis1(1.0, 0.0, 0.0), axis2(0.0, 1.0, 0.0),
00034         axis3(0.0, 0.0, 1.0), moment1(1.0), moment2(1.0), moment3(1.0) {}
```

```
00035   Coord com;
00036   Vector axis1;
00037   Vector axis2;
00038   Vector axis3;
00039   double moment1;
00040   double moment2;
00041   double moment3;
00042 };
00043
00044 void to_json(json &j, const PrincipalAxes &principalAxes);
00045 void from_json(const json &j, PrincipalAxes &principalAxes);
00046
00047 typedef std::vector<PrincipalAxes> PrincipalAxesList;
00048 typedef PrincipalAxesList::iterator PrincipalAxesListIter;
00049 typedef PrincipalAxesList::const_iterator PrincipalAxesListConstIter;
00050
00051 // Calculates principal axes and center of mass for the atoms in the atom list
00052 RBTDLL_EXPORT PrincipalAxes GetPrincipalAxesOfAtoms(const AtomList &atomList);
00053 // Calculates principal axes and center of mass for the coords in the coord list
00054 // (assumes all masses=1)
00055 RBTDLL_EXPORT PrincipalAxes GetPrincipalAxesOfAtoms(const CoordList &coordList);
00056 // Special case for water
00057 PrincipalAxes GetSolventPrincipalAxes(const AtomPtr &oAtom,
00058                                       const AtomPtr &h1Atom,
00059                                       const AtomPtr &h2Atom);
00060 // DM 17 Jul 2001 - returns the quaternion used to effect the transformation
00061 Quat AlignPrincipalAxesOfAtoms(AtomList &atomList,
00062                                const PrincipalAxes &alignAxes = PrincipalAxes(),
00063                                bool bAlignCOM = true);
00064 // Returns the quaternion required to align principal axes with reference axes
00065 RBTDLL_EXPORT Quat GetQuatFromAlignAxes(const PrincipalAxes &prAxes,
00066                                         const PrincipalAxes &refAxes);
00067 // Returns the quaternion required to effect an alignment of v onto ref vector.
00068 // Vectors do not have to be unit length.
00069 // BadArgument exception is thrown if either v or ref is zero length.
00070 Quat GetQuatFromAlignVectors(const Vector &v, const Vector &ref);
00071
00072 } // namespace rxdock
00073
00074 #endif //_RBTPRINCIPALAXES_H_
```

6.127 PRMFactory.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Factory class for generating workspace objects from <receptor>.prm file
00014 // Responsible for generation of:
00015 // Receptor, Ligand, Solvent,
00016 #ifndef RBTPRMFACTORY_H_
00017 #define RBTPRMFACTORY_H_
00018
00019 #include "rxdock/BaseMolecularFileSource.h"
00020 #include "rxdock/Error.h"
00021 #include "rxdock/Model.h"
00022
00023 namespace rxdock {
00024
00025 class ParameterFileSource;
00026 class DockingSite;
00027
00028 class PRMFactory {
00029 public:
00030   static const std::string &_CT;
00031   // Receptor parameters
00032   static const std::string &_REC_SECTION;
00033   static const std::string &_REC_FILE;
00034   static const std::string &_REC_TOPOL_FILE;
00035   static const std::string &_REC_COORD_FILE;
00036   static const std::string &_REC_NUM_COORD_FILES;
00037   static const std::string &_REC_FLEX_DISTANCE;
00038   static const std::string &_REC_DIHEDRAL_STEP;
00039   static const std::string &_REC_ALL_H;
00040   static const std::string &_REC_MASSES_FILE;
```

```
00041   static const std::string &_REC_SEGMENT_NAME;
00042
00043   // Ligand parameters
00044   static const std::string &_LIG_SECTION;
00045   // Ligand file is not read here, so no need for _LIG_FILE
00046   // No need to specify ligand flexibility parameter names explicitly
00047   // as we assume that all parameters in SECTION LIGAND
00048   // are flexibility params.
00049
00050   // Solvent parameters
00051   static const std::string &_SOLV_SECTION;
00052   static const std::string &_SOLV_FILE;
00053   // No need to specify solvent flexibility parameter names explicitly
00054   // as we assume that all parameters in SECTION SOLVENT
00055   // are flexibility params (except for _SOLV_FILE)
00056
00057   RBTDLL_EXPORT PRMFactory(ParameterFileSource *pParamSource);
00058   RBTDLL_EXPORT PRMFactory(ParameterFileSource *pParamSource, DockingSite *pDS);
00059
00060   DockingSite *GetDockingSite() const { return m_pDS; }
00061   void SetDockingSite(DockingSite *pDS) { m_pDS = pDS; }
00062
00063   RBTDLL_EXPORT ModelPtr CreateReceptor();
00064   RBTDLL_EXPORT ModelPtr CreateLigand(BaseMolecularFileSource *pSource);
00065   RBTDLL_EXPORT ModelList CreateSolvent();
00066
00067 private:
00068   // Creates the appropriate source according to the file extension
00069   MolecularFileSourcePtr CreateMolFileSource(const std::string &fileName);
00070   void AttachReceptorFlexData(Model *pReceptor);
00071   void AttachLigandFlexData(Model *pLigand);
00072   void AttachSolventFlexData(Model *pSolvent);
00073   ParameterFileSource *m_pParamSource;
00074   DockingSite *m_pDS;
00075 };
00076
00077 } // namespace rxdock
00078
00079 #endif /*RBTPRMFACTORY_H_*/
```

6.128 PseudoAtom.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Derived class for pseudo-atoms.
00014
00015 #ifndef _RBTPSEUDOATOM_H_
00016 #define _RBTPSEUDOATOM_H_
00017
00018 #include "rxdock/Atom.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 class PseudoAtom : public Atom {
00027 public:
00029   // Constructors / destructors
00030
00031   // Constructor supplying atom list of constituent atoms
00032   PseudoAtom(const AtomList &atomList, int nAtomId = -1,
00033              std::string strAtomName = "PSEUDO");
00034
00035   friend void to_json(json &j, const PseudoAtom &pseudoAtom);
00036   friend void from_json(const json &j, PseudoAtom &pseudoAtom);
00037
00038   // Default destructor
00039   virtual ~PseudoAtom();
00040
00042   // Public accessor functions
00043   // Virtual function for dumping pseudoatom details to an output stream
```

```
00044   virtual std::ostream &Print(std::ostream &s) const;
00045
00047   // Public accessor functions
00048   // 2-D attributes
00049
00050   // Derived class methods for returning the constituent atom list
00051   unsigned int GetNumAtoms() const { return m_atomList.size(); }
00052   AtomList GetAtomList() const { return m_atomList; }
00053
00055   // Public accessor functions
00056   // 3-D attributes
00057
00058   // Coords
00059   // DM 8 Dec 1998 - override base Atom methods
00060   // virtual Coord GetCoords() const;
00061   // virtual Double GetX() const;
00062   // virtual Double GetY() const;
00063   // virtual Double GetZ() const;
00064   //
00065   // DM 11 Jul 2000 - base GetCoords method is now non-virtual again
00066   // Pseudoatoms must update their mean coords by calling UpdateCoords
00067   // each time the underlying atoms are moved
00068   void UpdateCoords();
00069
00071
00072   // Other public methods
00073
00074 protected:
00075 private:
00076   // Private methods
00077   PseudoAtom();                              // Disable default constructor
00078   PseudoAtom(const PseudoAtom &pseudoAtom); // Disable copy constructor
00079   PseudoAtom &
00080   operator=(const PseudoAtom &pseudoAtom); // Disable copy assignment
00081
00082 private:
00083   // Private data
00084   AtomList m_atomList; // List of smart pointers to constituent atoms
00085 };
00086
00087 void to_json(json &j, const PseudoAtom &pseudoAtom);
00088 void from_json(const json &j, PseudoAtom &pseudoAtom);
00089
00090 // Useful typedefs
00091 typedef SmartPtr<PseudoAtom> PseudoAtomPtr;        // Smart pointer
00092 typedef std::vector<PseudoAtomPtr> PseudoAtomList; // Vector of smart pointers
00093 typedef PseudoAtomList::iterator PseudoAtomListIter;
00094 typedef PseudoAtomList::const_iterator PseudoAtomListConstIter;
00095
00096 } // namespace rxdock
00097
00098 #endif //_RBTPSEUDOATOM_H_
```

6.129 PsfFileSink.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Class for writing Models to PSF files
00014
00015 #ifndef _RBTPSFFILESINK_H_
00016 #define _RBTPSFFILESINK_H_
00017
00018 #include "rxdock/BaseMolecularFileSink.h"
00019
00020 namespace rxdock {
00021
00022 class PsfFileSink : public BaseMolecularFileSink {
00023 public:
00025   // Constructors/destructors
00026   // PsfFileSink(const char* fileName, ModelPtr spModel);
00027   RBTDLL_EXPORT PsfFileSink(const std::string &fileName, ModelPtr spModel);
00028
00029   virtual ~PsfFileSink(); // Default destructor
```

```
00030
00032   // Public methods
00034   // Override public methods from BaseFileSink
00035   virtual void Render();
00036
00037 protected:
00039   // Protected methods
00041
00042 private:
00044   // Private methods
00046
00047   PsfFileSink();                    // Disable default constructor
00048   PsfFileSink(const PsfFileSink &); // Copy constructor disabled by
00049                                     // default
00050   PsfFileSink &
00051   operator=(const PsfFileSink &); // Copy assignment disabled by default
00052
00053 protected:
00055   // Protected data
00057
00058 private:
00060   // Private data
00062 };
00063
00064 // Useful typedefs
00065 typedef SmartPtr<PsfFileSink> PsfFileSinkPtr; // Smart pointer
00066
00067 } // namespace rxdock
00068
00069 #endif //_RBTPSFFILESINK_H_
```

6.130 PsfFileSource.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Handles retrieval of molecular info from Charmm PSF files.
00014
00015 #ifndef _RBTPSFFILESOURCE_H_
00016 #define _RBTPSFFILESOURCE_H_
00017
00018 #include "rxdock/BaseMolecularFileSource.h"
00019 #include "rxdock/CharmmDataSource.h"
00020 #include "rxdock/ElementFileSource.h"
00021 #include "rxdock/ParameterFileSource.h"
00022
00023 namespace rxdock {
00024
00025 class PsfFileSource : public BaseMolecularFileSource {
00026 public:
00027   // Constructors
00028   PsfFileSource(
00029       const std::string &fileName,
00030       const std::string &strMassesFile = GetDataFileName("data", "masses.rtf"),
00031       bool bImplHydrogens = true);
00032
00033   // Default destructor
00034   virtual ~PsfFileSource();
00035
00037   // Override public methods from BaseMolecularDataSource
00038   virtual bool isTitleListSupported() { return true; }
00039   virtual bool isAtomListSupported() { return true; }
00040   virtual bool isCoordinatesSupported() { return false; }
00041   virtual bool isBondListSupported() { return true; }
00042
00043 protected:
00044   // Pure virtual in BaseFileSource - needs to be defined here
00045   virtual void Parse();
00046
00047 private:
00048   // Private methods
00049   PsfFileSource();                      // Disable default constructor
00050   PsfFileSource(const PsfFileSource &); // Copy constructor disabled by default
00051   PsfFileSource &
```

```
00052   operator=(const PsfFileSource &); // Copy assignment disabled by default
00053
00054   // Sets up all the atomic attributes that are not explicitly stored in the PSF
00055   // file
00056   void SetupAtomParams();
00057   void SetupVdWRadii();
00058   void SetupPartialIonicGroups();
00059   void RemoveNonPolarHydrogens();
00060   // Is atom in same substructure
00061   class isSS_eq : public std::function<bool(Atom *)> {
00062     const Atom *a1;
00063
00064   public:
00065     explicit isSS_eq(const Atom *aa) : a1(aa) {}
00066     bool operator()(const Atom *a2) const {
00067       return (a1->GetSubunitName() == a2->GetSubunitName()) &&
00068              (a1->GetSubunitId() == a2->GetSubunitId()) &&
00069              (a1->GetSegmentName() == a2->GetSegmentName());
00070     }
00071   };
00072
00073   // Private data
00074   ElementFileSourcePtr
00075       m_spElementData;               // Elemental data source for vdW radii etc
00076   CharmmDataSourcePtr m_spCharmmData; // Charmm data source for fftype lookup
00077   ParameterFileSourcePtr
00078       m_spParamSource;   // Parameter file source for partial charges etc
00079   bool m_bImplHydrogens; // If true, non-hydrogen bonding hydrogens are removed
00080 };
00081
00082 // useful typedefs
00083 typedef SmartPtr<PsfFileSource> PsfFileSourcePtr; // Smart pointer
00084
00085 } // namespace rxdock
00086
00087 #endif //_RBTPSFFILESOURCE_H_
```

6.131 Quat.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple class for handling quaternions
00014
00015 #ifndef _RBTQUAT_H_
00016 #define _RBTQUAT_H_
00017
00018 #include <cmath> //for sqrt
00019
00020 #include "rxdock/Coord.h"
00021
00022 namespace rxdock {
00023
00024 class Quat {
00026   // Data members
00028   //(leave public so we don't need
00029   // to write accessor functions)
00030 public:
00031   double s; // Scalar component
00032   Vector v; // Vector component
00033
00034   friend void to_json(json &j, const Quat &quat) {
00035     j = json{{"scalar", quat.s}, {"vector", quat.v}};
00036   };
00037   friend void from_json(const json &j, Quat &quat) {
00038     j.at("scalar").get_to(quat.s);
00039     j.at("vector").get_to(quat.v);
00040   };
00041
00043   // Constructors/destructors:
00045 public:
00046   // Default constructor (corresponds to no rotation (null transformation))
00047   inline Quat() : s(1.0), v(0.0, 0.0, 0.0) {}
00048
```

```
00049   // Constructor with initial values (vector component passed as 3 Double's)
00050   inline Quat(double s1, double vx, double vy, double vz)
00051       : s(s1), v(vx, vy, vz) {}
00052
00053   // Constructor with initial values (vector component passed as Vector)
00054   // CAUTION: Argument types are the reverse of below
00055   inline Quat(double s1, const Vector &v1) : s(s1), v(v1) {}
00056
00057   // Constructor accepting a rotation axis and rotation angle (radians).
00058   // CAUTION: Argument types are the reverse of above
00059   inline Quat(const Vector &axis, double phi) {
00060     double halfPhi(0.5 * phi);
00061     s = std::cos(halfPhi);
00062     v = std::sin(halfPhi) * axis.Unit();
00063   }
00064
00065   // Destructor
00066   virtual ~Quat() {}
00067
00068   // Copy constructor
00069   inline Quat(const Quat &quat) {
00070     s = quat.s;
00071     v = quat.v;
00072   }
00073
00075   // Operator functions:
00077
00078   // Copy assignment (*this = coord)
00079   inline Quat &operator=(const Quat &quat) {
00080     if (this != &quat) { // beware of self-assignment
00081       s = quat.s;
00082       v = quat.v;
00083     }
00084     return *this;
00085   }
00086
00087   //*this += quat
00088   inline void operator+=(const Quat &quat) {
00089     s += quat.s;
00090     v += quat.v;
00091   }
00092
00093   //*this -= quat
00094   inline void operator-=(const Quat &quat) {
00095     s -= quat.s;
00096     v -= quat.v;
00097   }
00098
00099   //*this *= const
00100   inline void operator*=(const double &d) {
00101     s *= d;
00102     v *= d;
00103   }
00104
00105   //*this /= const
00106   inline void operator/=(const double &d) {
00107     s /= d;
00108     v /= d;
00109   }
00110
00112   // Friend functions:
00114
00115   // Insertion operator
00116   friend std::ostream &operator«(std::ostream &s, const Quat &quat) {
00117     return s « quat.s « "," « quat.v;
00118   }
00119
00120   // Equality
00121   inline friend bool operator==(const Quat &quat1, const Quat &quat2) {
00122     return quat1.s == quat2.s && quat1.v == quat2.v;
00123   }
00124
00125   // Non-equality
00126   inline friend bool operator!=(const Quat &quat1, const Quat &quat2) {
00127     return quat1.s != quat2.s || quat1.v != quat2.v;
00128   }
00129
00130   // Addition
00131   inline friend Quat operator+(const Quat &quat1, const Quat &quat2) {
00132     return Quat(quat1.s + quat2.s, quat1.v + quat2.v);
00133   }
00134
00135   // Subtraction
00136   inline friend Quat operator-(const Quat &quat1, const Quat &quat2) {
00137     return Quat(quat1.s - quat2.s, quat1.v - quat2.v);
00138   }
00139
```

```
00140   // Negation
00141   inline friend Quat operator-(const Quat &quat) {
00142     return Quat(-quat.s, -quat.v);
00143   }
00144
00145   // Multiplication (non-commutative)
00146   inline friend Quat operator*(const Quat &quat1, const Quat &quat2) {
00147     return Quat(quat1.s * quat2.s - (quat1.v).Dot(quat2.v),
00148                 quat1.s * quat2.v + quat1.v * quat2.s +
00149                     (quat1.v).Cross(quat2.v));
00150   }
00151
00152   // Scalar product
00153   inline friend Quat operator*(const Quat &quat, const double &d) {
00154     return Quat(quat.s * d, quat.v * d);
00155   }
00156
00157   // Scalar product
00158   inline friend Quat operator*(const double &d, const Quat &quat) {
00159     return Quat(quat.s * d, quat.v * d);
00160   }
00161
00162   // Scalar division
00163   inline friend Quat operator/(const Quat &quat, const double &d) {
00164     return Quat(quat.s / d, quat.v / d);
00165   }
00166
00168   // Public methods
00170
00171   // Returns magnitude of quat
00172   // Member function (L = Q1.Length())
00173   inline double Length() const { return std::sqrt(s * s + v.Length2()); }
00174
00175   // Returns unit quat
00176   // Member function (U1 = Q1.Unit())
00177   inline Quat Unit() const {
00178     double l = Length();
00179     return (l > 0) ? *this / l : *this;
00180   }
00181
00182   // Dot product member function (D = Q1.Dot(Q2))
00183   inline double Dot(const Quat &quat) const {
00184     return s * quat.s + v.Dot(quat.v);
00185   }
00186
00187   // Returns conjugate
00188   // Member function (Q1* = Q1.Conj())
00189   inline Quat Conj() const { return Quat(s, -v); }
00190
00191   // Returns Q W Q* where Q=S+V is a quaternion and W is a 3-D vector
00192   // Q W Q* = (S^2 - V.V)W + 2S(VxW) + 2V(V.W)
00193   // where . = Dot product and x = Cross product
00194   // The end result is a 3-D rotation, assuming q has been obtained from
00195   // RotationQuat
00196   inline Coord Rotate(const Coord &w) const {
00197     return (s * s - v.Dot(v)) * w + 2 * s * v.Cross(w) + 2 * v * v.Dot(w);
00198   }
00199 };
00200
00201 // Useful typedefs
00202 typedef std::vector<Quat> QuatList; // Vector of quats
00203 typedef QuatList::iterator QuatListIter;
00204 typedef QuatList::const_iterator QuatListConstIter;
00205
00207 // Non-member functions (in rxdock namespace)
00209
00210 // Returns magnitude of quat (L = Length(Q1))
00211 inline double Length(const Quat &quat) { return quat.Length(); }
00212
00213 // Returns unit quat (U1 = Unit(Q1))
00214 inline Quat Unit(const Quat &quat) { return quat.Unit(); }
00215
00216 // Dot product (D = Dot(Q1,Q2))
00217 inline double Dot(const Quat &quat1, const Quat &quat2) {
00218   return quat1.Dot(quat2);
00219 }
00220
00221 // Returns conjugate (Q1* = Conj(Q1))
00222 inline Quat Conj(const Quat &quat) { return quat.Conj(); }
00223
00224 } // namespace rxdock
00225
00226 #endif //_RBTQUAT_H_
```

6.132 Rand.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Wrapper around Randint class
00014 // Function provided to return reference to single instance (singleton) of
00015 // Rand
00016
00017 #ifndef _RBTRAND_H_
00018 #define _RBTRAND_H_
00019
00020 // Use standard C++11 RNG on Solaris and Windows/MSVC due to PCG build failure
00021 // Solaris issue: https://github.com/imneme/pcg-cpp/issues/42
00022 // Windows/MSVC issue: https://github.com/imneme/pcg-cpp/issues/11
00023 #if !defined(__sun) && !(defined(_WIN32) && defined(_MSC_VER))
00024 #include <pcg_random.hpp>
00025 #endif
00026 #include <random>
00027
00028 #include "rxdock/Coord.h"
00029
00030 namespace rxdock {
00031
00032 class Rand {
00034   // Constructor
00035 public:
00036   Rand();
00038   // Destructor
00039   ~Rand();
00040
00042   // Public methods
00043
00044   // Seed the random number generator
00045   RBTDLL_EXPORT void Seed(int seed = 0);
00046   // Seed the random number generator from the random device
00047   void SeedFromRandomDevice();
00048   // Returns current seed
00049   RBTDLL_EXPORT int GetSeed();
00050   // Get a random double between 0 and 1
00051   RBTDLL_EXPORT double GetRandom01();
00052   // Get a random integer between 0 and nMax-1
00053   int GetRandomInt(int nMax);
00054   // Get a random unit vector distributed evenly over the surface of a sphere
00055   Vector GetRandomUnitVector();
00056   double GetGaussianRandom(double, double);
00057   double GetCauchyRandom(double, double);
00058
00059 private:
00060 #if defined(__sun) || (defined(_WIN32) && defined(_MSC_VER))
00061   std::default_random_engine m_rng;
00062 #else
00063   pcg32 m_rng; // Random number generator
00064 #endif
00065 };
00066
00068 // Non-member functions in rxdock namespace
00069
00070 // Returns reference to single instance of Rand class (singleton)
00071 RBTDLL_EXPORT Rand &GetRandInstance();
00072
00073 } // namespace rxdock
00074
00075 #endif //_RBTRAND_H_
```

6.133 RandLigTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
```

```
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Randomises all rotable bonds of the ligand, within specified +/- range
00014 // No bump check on final conformation
00015
00016 #ifndef _RBTRANDLIGTRANSFORM_H_
00017 #define _RBTRANDLIGTRANSFORM_H_
00018
00019 #include "rxdock/BaseUniMolTransform.h"
00020 #include "rxdock/Rand.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class RandLigTransform : public BaseUniMolTransform {
00029 public:
00030   // Static data member for class type
00031   static const std::string _CT;
00032   // Parameter names
00033   static const std::string _TORS_STEP;
00034
00036   // Constructors/destructors
00037   RandLigTransform(const std::string &strName = "RANDLIG");
00038   virtual ~RandLigTransform();
00039
00040   friend void to_json(json &j, const RandLigTransform &randLigTrans);
00041   friend void from_json(const json &j, RandLigTransform &randLigTrans);
00042
00044   // Public methods
00046
00047 protected:
00049   // Protected methods
00051   virtual void SetupTransform(); // Called by Update when model has changed
00052   virtual void Execute();
00053
00054 private:
00056   // Private methods
00058   RandLigTransform(
00059       const RandLigTransform &); // Copy constructor disabled by default
00060   RandLigTransform &
00061   operator=(const RandLigTransform &); // Copy assignment disabled by default
00062
00063 protected:
00065   // Protected data
00067
00068 private:
00070   // Private data
00072   Rand &m_rand; // keep a reference to the singleton random number generator
00073   BondList m_rotableBonds;
00074 };
00075
00076 void to_json(json &j, const RandLigTransform &randLigTrans);
00077 void from_json(const json &j, RandLigTransform &randLigTrans);
00078
00079 // Useful typedefs
00080 typedef SmartPtr<RandLigTransform> RandLigTransformPtr; // Smart pointer
00081
00082 } // namespace rxdock
00083
00084 #endif //_RBTRANDLIGTRANSFORM_H_
```

6.134 RandPopTransform.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Generates a randomised population for the GA
00014 #ifndef _RBTRANDPOPTRANSFORM_H_
```

```
00015 #define _RBTRANDPOPTRANSFORM_H_
00016
00017 #include "rxdock/BaseBiMolTransform.h"
00018 #include "rxdock/ChromElement.h"
00019
00020 namespace rxdock {
00021
00022 class RandPopTransform : public BaseBiMolTransform {
00023 public:
00024   static const std::string _CT;
00025   static const std::string _POP_SIZE;
00026   static const std::string _SCALE_CHROM_LENGTH;
00027
00029   // Constructors/destructors
00030   RBTDLL_EXPORT RandPopTransform(const std::string &strName = "RANDPOP");
00031   virtual ~RandPopTransform();
00032
00034   // Public methods
00036
00037 protected:
00039   // Protected methods
00041   virtual void
00042   SetupTransform(); // Called by Update when either model has changed
00043   virtual void SetupReceptor(); // Called by Update when receptor is changed
00044   virtual void SetupLigand();   // Called by Update when ligand is changed
00045   virtual void SetupSolvent();  // Called by Update when solvent is changed
00046   virtual void Execute();
00047
00048 private:
00050   // Private methods
00052   RandPopTransform(
00053       const RandPopTransform &); // Copy constructor disabled by default
00054   RandPopTransform &
00055   operator=(const RandPopTransform &); // Copy assignment disabled by default
00056
00057 protected:
00059   // Protected data
00061
00062 private:
00064   // Private data
00066   ChromElementPtr m_chrom;
00067 };
00068
00069 // Useful typedefs
00070 typedef SmartPtr<RandPopTransform> RandPopTransformPtr; // Smart pointer
00071
00072 } // namespace rxdock
00073
00074 #endif //_RBTRANDPOPTRANSFORM_H_
```

6.135 Rbt.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Miscellaneous non-member functions in rxdock namespace
00014 #ifndef _RBT_H_
00015 #define _RBT_H_
00016
00017 #include "rxdock/support/Export.h"
00018
00019 #include <istream>
00020 #include <map>
00021 #include <ostream>
00022 #include <vector>
00023
00024 namespace rxdock {
00025
00026 // Segment is a named part of an Model (usually an intact molecule)
00027 // For now, a segment is defined as just an String
00028 // SegmentMap holds a map of (key=unique segment name, value=number of atoms
00029 // in segment)
00030 typedef std::string Segment;
00031 typedef std::map<Segment, unsigned int> SegmentMap;
```

```
00032 typedef SegmentMap::iterator SegmentMapIter;
00033 typedef SegmentMap::const_iterator SegmentMapConstIter;
00034
00036 // RESOURCE HANDLING FUNCTIONS
00037 //
00038 // GetRoot - returns value of RBT_ROOT env variable
00039 RBTDLL_EXPORT std::string GetRoot();
00040 // GetHome - returns value of RBT_HOME env variable
00041 //(or HOME if RBT_HOME is undefined)
00042 std::string GetHome();
00043 // GetProgramName - returns program name
00044 RBTDLL_EXPORT std::string GetProgramName();
00045 // GetMetaDataPrefix - returns meta data prefix
00046 RBTDLL_EXPORT std::string GetMetaDataPrefix();
00047 // GetCopyright - returns legalese statement
00048 RBTDLL_EXPORT std::string GetCopyright();
00049 // GetProgramVersion - returns current library version
00050 RBTDLL_EXPORT std::string GetProgramVersion();
00051 // GetProduct - returns library product name
00052 RBTDLL_EXPORT std::string GetProduct();
00053 // GetTime - returns current time and date as an String
00054 std::string GetTime();
00055 // GetCurrentWorkingDirectory - returns current working directory
00056 RBTDLL_EXPORT std::string GetCurrentWorkingDirectory();
00057 //
00059
00061 // FILE/DIRECTORY HANDLING FUNCTIONS
00062 //
00063 // GetDirName
00064 // Returns the full path to a subdirectory in the rDock directory structure
00065 //
00066 // For example, if RBT_ROOT environment variable is ~dave/ribodev/molmod/ribodev
00067 // then GetDirName("data") would return ~dave/ribodev/molmod/ribodev/data/
00068 //
00069 // If RBT_ROOT is not set, then GetDirName returns ./ irrespective of the
00070 // subdirectory asked for. Thus the fall-back position is that parameter files
00071 // are read from the current directory if RBT_ROOT is not defined.
00072 std::string GetDirName(const std::string &strSubdir = "");
00073
00074 // GetDataFileName
00075 // As GetDirName but returns the full path to a file in the rDock directory
00076 // structure
00077 RBTDLL_EXPORT std::string GetDataFileName(const std::string &strSubdir,
00078                                           const std::string &strFile);
00079
00080 // GetFileType
00081 // Returns the string following the last "." in the file name.
00082 // e.g. GetFileType("receptor.psf") would return "psf"
00083 std::string GetFileType(const std::string &strFile);
00084
00085 // GetDirList
00086 // Returns a list of files in a directory (strDir) whose names begin with
00087 // strFilePrefix (optional) and whose type is strFileType (optional, as returned
00088 // by GetFileType)
00089 std::vector<std::string> GetDirList(const std::string &strDir,
00090                                     const std::string &strFilePrefix = "",
00091                                     const std::string &strFileType = "");
00092 //
00094
00096 // CONVERSION ROUTINES
00097 //
00098 // Converts (comma)-delimited string of segment names to segment map
00099 RBTDLL_EXPORT SegmentMap ConvertStringToSegmentMap(
00100     const std::string &strSegments, const std::string &strDelimiter = ",");
00101 // Converts segment map to (comma)-delimited string of segment names
00102 std::string ConvertSegmentMapToString(const SegmentMap &segmentMap,
00103                                       const std::string &strDelimiter = ",");
00104
00105 // Returns a segment map containing the members of map1 which are not in map2
00106 // I know, should really be a template so as to be more universal...one day
00107 // maybe. Or maybe there is already an STL algorithm for doing this.
00108 SegmentMap SegmentDiffMap(const SegmentMap &map1, const SegmentMap &map2);
00109
00110 // DM 30 Mar 1999
00111 // Converts (comma)-delimited string to string list (similar to
00112 // ConvertStringToSegmentMap, but returns list not map)
00113 RBTDLL_EXPORT std::vector<std::string>
00114 ConvertDelimitedStringToList(const std::string &strValues,
00115                              const std::string &strDelimiter = ",");
00116 // Converts string list to (comma)-delimited string (inverse of above)
00117 std::string
00118 ConvertListToDelimitedString(const std::vector<std::string> &listOfValues,
00119                              const std::string &strDelimiter = ",");
00120
00121 // Detect terminal width and wrap text to that width
00122 std::string WrapTextToTerminalWidth(const std::string &text);
00123
```

```
00124 //
00126
00128 // I/O ROUTINES
00129 //
00130 // PrintStdHeader
00131 // Print a standard header to an output stream (for log files etc)
00132 // Contains copyright info, library version, date, time etc
00133 // DM 19 Feb 1999 - include executable information
00134 RBTDLL_EXPORT std::ostream &
00135 PrintStdHeader(std::ostream &s, const std::string &strExecutable = "");
00136
00137 RBTDLL_EXPORT std::ostream &
00138 PrintBibliographyItem(std::ostream &s, const std::string &publicationKey = "");
00139
00140 // Helper functions to read/write chars from iostreams
00141 // Throws error if stream state is not Good() before and after the read/write
00142 // It appears the STL std::ios_base exception throwing is not yet implemented
00143 // at least on RedHat 6.1, so this is a temporary workaround (yeah right)
00144 //
00145 // DM 25 Sep 2000 - with MIPSPro CC compiler, streamsize is not defined,
00146 // at least with the iostreams library we are using, so typedef it here
00147 // it is not in gcc 3.2.1 either SJ
00148 //#ifdef __sgi
00149 typedef int streamsize;
00150 //#endif
00151
00152 RBTDLL_EXPORT void WriteWithThrow(std::ostream &ostr, const char *p,
00153                                   streamsize n);
00154 RBTDLL_EXPORT void ReadWithThrow(std::istream &istr, char *p, streamsize n);
00155
00156 } // namespace rxdock
00157
00158 #endif //_RBT_H_
```

6.136 RealGrid.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages an array of floats representing a 3-D lattice of grid points
00014
00015 #ifndef _RBTREALGRID_H_
00016 #define _RBTREALGRID_H_
00017
00018 #ifdef __PGI
00019 #define EIGEN_DONT_VECTORIZE
00020 #endif
00021 #ifdef _WIN32
00022 #define NOMINMAX
00023 #endif
00024 #include <unsupported/Eigen/CXX11/Tensor>
00025
00026 #include "rxdock/BaseGrid.h"
00027
00028 namespace rxdock {
00029
00030 class RealGrid : public BaseGrid {
00031 public:
00032   // Class type string
00033   static const std::string _CT;
00034
00036   // Constructors/destructors
00037   // Construct a NXxNYxNZ grid running from gridMin at gridStep resolution
00038   RBTDLL_EXPORT RealGrid(const Coord &gridMin, const Coord &gridStep,
00039                          unsigned int NX, unsigned int NY, unsigned int NZ,
00040                          unsigned int NPad = 0);
00041
00042   // Constructor reading all params from binary stream
00043   RBTDLL_EXPORT RealGrid(json j);
00044
00045   ~RealGrid(); // Default destructor
00046
00047   // Copy constructor
00048   RealGrid(const RealGrid &);
```

```
00049   // Copy constructor taking a base class argument
00050   // Sets up the grid dimensions, then creates an empty data array
00051   RealGrid(const BaseGrid &);
00052   // Copy assignment
00053   RealGrid &operator=(const RealGrid &);
00054   // Copy assignment taking a base class argument
00055   // Sets up the grid dimensions, then creates an empty data array
00056   RealGrid &operator=(const BaseGrid &);
00057
00059   // Virtual functions for reading/writing grid data to streams in
00060   // text and binary format
00061   // Subclasses should provide their own private OwnPrint
00062   // method to handle subclass data members, and override the public
00063   // Print method
00064   virtual void Print(std::ostream &ostr) const; // Text output
00065
00067   // Public methods
00069
00071   // Get attribute functions
00073   float *GetGridData() {
00074     float *data = m_grid.data();
00075     return data;
00076   }
00077
00079   // Get/Set value functions
00081
00082   // Get/Set tolerance used for assessing value equality
00083   double GetTolerance() const { return m_tol; }
00084   void SetTolerance(double tol) { m_tol = tol; }
00085
00086   // Get/Set single grid point value with bounds checking
00087   double GetValue(const Coord &c) const {
00088     return isValid(c) ? m_grid(GetIX(c), GetIY(c), GetIZ(c)) : 0.0;
00089   }
00090
00091   double GetValue(unsigned int iX, unsigned int iY, unsigned int iZ) const {
00092     return isValid(iX, iY, iZ) ? m_grid(iX, iY, iZ) : 0.0;
00093   }
00094
00095   double GetValue(unsigned int iXYZ) const {
00096     const float *data = m_grid.data();
00097     return isValid(iXYZ) ? data[iXYZ] : 0.0;
00098   }
00099
00100   // DM 20 Jul 2000 - get values smoothed by trilinear interpolation
00101   // D. Oberlin and H.A. Scheraga, J. Comp. Chem. (1998) 19, 71.
00102   RBTDLL_EXPORT double GetSmoothedValue(const Coord &c) const;
00103
00104   void SetValue(const Coord &c, double val) {
00105     if (isValid(c))
00106       m_grid(GetIX(c), GetIY(c), GetIZ(c)) = val;
00107   }
00108
00109   void SetValue(unsigned int iX, unsigned int iY, unsigned int iZ, double val) {
00110     if (isValid(iX, iY, iZ))
00111       m_grid(iX, iY, iZ) = val;
00112   }
00113
00114   void SetValue(unsigned int iXYZ, double val) {
00115     float *data = m_grid.data();
00116     if (isValid(iXYZ))
00117       data[iXYZ] = val;
00118   }
00119
00120   // Set all grid points to the given value
00121   RBTDLL_EXPORT void SetAllValues(double val);
00122   // Replaces all grid points with a given value (+/- tolerance) with newVal
00123   void ReplaceValue(double oldVal, double newVal) {
00124     ReplaceValueRange(oldVal - m_tol, oldVal + m_tol, newVal);
00125   }
00126   // Replaces all grid points between oldValMin and oldValMax with newVal
00127   void ReplaceValueRange(double oldValMin, double oldValMax, double newVal);
00128
00129   // Set all grid points within radius of coord to the given value
00130   // If bOverwrite is false, does not replace non-zero values
00131   // If bOverwrite is true, all grid points are set the new value
00132   void SetSphere(const Coord &c, double radius, double val,
00133                  bool bOverwrite = true);
00134
00135   // Set all grid points with radii between rad1 and rad2 from coord to the
00136   // given value If bOverwrite is false, does not replace non-zero values If
00137   // bOverwrite is true, all grid points are set the new value
00138   void SetSurface(const Coord &c, double innerRad, double outerRad, double val,
00139                   bool bOverwrite = true);
00140
00141   // Sets all grid points with value=oldValue, which are adjacent to those with
00142   // value=adjacentValue, to value=newValue
```

```
00143   //+/- tolerance is applied to oldValue and adjacentValue
00144   void CreateSurface(double oldVal, double adjVal, double newVal);
00145
00146   // DM 16 Apr 1999 - helper function for determining solvent accessible regions
00147   // Returns true if any of the grid points within a sphere around the central
00148   // coord have the specified value
00149   bool isValueWithinSphere(const Coord &c, double radius, double val);
00150   // Sets all grid points with value=oldValue, which have no grid points with
00151   // value=adjacentValue within a sphere of given radius, to value=newValue
00152   //+/- tolerance is applied to oldValue and adjacentValue
00153   // If bCenterOnly is true, just the center of the sphere is set the newValue
00154   // If bCenterOnly is false, all grid points in the sphere are set to the
00155   // newValue
00156   void SetAccessible(double radius, double oldVal, double adjVal, double newVal,
00157                      bool bCenterOnly = true);
00158
00160   // Statistical functions
00162
00163   // Returns number of occurrences of a given value (+/- tolerance)
00164   unsigned int Count(double val) const {
00165     return CountRange(val - m_tol, val + m_tol);
00166   }
00167   // Returns number of occurrences of a given value range
00168   unsigned int CountRange(double valMin, double valMax) const;
00169
00170   // Min/max values
00171   double MinValue() const;
00172   double MaxValue() const;
00173
00174   unsigned int
00175   FindMinValue() const; // iXYZ index of grid point with minimum value
00176   unsigned int
00177   FindMaxValue() const; // iXYZ index of grid point with maximum value
00178
00180   // I/O functions
00182
00183   // Dump grid in a format readable by Insight
00184   RBTDLL_EXPORT void PrintInsightGrid(std::ostream &s) const;
00185
00186   RBTDLL_EXPORT friend void to_json(json &j, const RealGrid &grid);
00187   friend void from_json(const json &j, RealGrid &grid);
00188
00189 protected:
00191   // Protected methods
00193   // Protected method for writing data members for this class to text stream
00194   void OwnPrint(std::ostream &ostr) const;
00195
00196 private:
00198   // Private methods
00200   RealGrid(); // Disable default constructor
00201
00202   // DM 17 Jul 2000 - analogous to isValueWithinSphere but iterates over
00203   // arbitrary set of IXYZ indices. Private method as there is no error checking
00204   // on iXYZ values out of bounds
00205   bool isValueWithinList(const std::vector<unsigned int> &iXYZList, double val);
00206   // Analogous to SetSphere but iterates over arbitrary set
00207   // of IXYZ indices. Private method as there is no error checking on iXYZ
00208   // values out of bounds If bOverwrite is false, does not replace non-zero
00209   // values If bOverwrite is true, all grid points are set the new value
00210   void SetValues(const std::vector<unsigned int> &iXYZList, double val,
00211                  bool bOverwrite = true);
00212
00213   void CreateArrays();
00214
00215   // Helper function called by copy constructor and assignment operator
00216   void CopyGrid(const RealGrid &);
00217
00218 protected:
00220   // Protected data
00222
00223 private:
00225   // Private data
00227   Eigen::Tensor<float, 3, Eigen::RowMajor>
00228       m_grid;   // 3-D array, accessed as m_grid(i, j, k), indicies from 0
00229   double m_tol; // Tolerance for comparing grid values;
00230 };
00231
00232 // Useful typedefs
00233 typedef SmartPtr<RealGrid> RealGridPtr; // Smart pointer
00234 typedef std::vector<RealGridPtr> RealGridList;
00235 typedef RealGridList::iterator RealGridListIter;
00236 typedef RealGridList::const_iterator RealGridListConstIter;
00237
00238 RBTDLL_EXPORT void to_json(json &j, const RealGrid &grid);
00239 void from_json(const json &j, RealGrid &grid);
00240
00241 } // namespace rxdock
```

```
00242
00243 #endif //_RBTREALGRID_H_
```

6.137 ReceptorFlexData.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages flexibility data for macromolecules
00014 #ifndef _RBTRECEPTORFLEXDATA_
00015 #define _RBTRECEPTORFLEXDATA_
00016
00017 #include "rxdock/FlexData.h"
00018
00019 namespace rxdock {
00020
00021 class ReceptorFlexData : public FlexData {
00022 public:
00023   // Distance from docking site within which OH/NH3+ groups are flexible (A)
00024   static const std::string &_FLEX_DISTANCE;
00025   // Dihedral mutation step length (deg)
00026   static const std::string &_DIHEDRAL_STEP;
00027   RBTDLL_EXPORT ReceptorFlexData(DockingSite *pDockSite);
00028   virtual void Accept(FlexDataVisitor &v) { v.VisitReceptorFlexData(this); }
00029
00030 private:
00031   ReceptorFlexData();
00032 };
00033
00034 } // namespace rxdock
00035
00036 #endif //_RBTRECEPTORFLEXDATA_
```

6.138 Request.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple struct to represent a request message
00014 // Stores a request ID (int) and an arbitrary length list of variant params
00015 // Base class cannot be instantiated.
00016 // Derived class constructors should set the ID and the param list accordingly
00017
00018 #ifndef _RBTREQUEST_H_
00019 #define _RBTREQUEST_H_
00020
00021 #include "rxdock/Variant.h"
00022
00023 #include <nlohmann/json.hpp>
00024
00025 using json = nlohmann::json;
00026
00027 namespace rxdock {
00028
00029 typedef int RequestID;
00030
00031 // Only check request assertions in debug build
00032 #ifdef _NDEBUG
00033 const Bool REQ_CHECK = false;
00034 #else
00035 const bool REQ_CHECK = true;
```

```
00036 #endif //_NDEBUG
00037
00038 class Request {
00039 public:
00041   // Constructors/destructors
00042   virtual ~Request() {}
00043
00044   Request(json j) {
00045     j.get_to(*this);
00046     _RBTOBJECTCOUNTER_CONSTR_("Request");
00047   }
00048
00049   friend void to_json(json &j, const Request &request) {
00050     /*json variantList;
00051     for (const auto &aIter : request.m_parameters) {
00052       json request = *aIter;
00053       variantList.push_back(request);
00054     }*/
00055     j = json{{"request-id", request.m_id},
00056              {"variant-list", request.m_parameters}};
00057   }
00058
00059   friend void from_json(const json &j, Request &request) {
00060     j.at("request-id").get_to(request.m_id);
00061     j.at("variant-list").get_to(request.m_parameters);
00062   }
00063
00065   // Public methods
00067   RequestID GetID() const { return m_id; }
00068   unsigned int GetNumParameters() const { return m_parameters.size(); }
00069   VariantList GetParameters() const { return m_parameters; }
00070
00072   // Protected methods
00074   // Protected constructor to be used by derived classes to set request ID
00075   Request(RequestID id) : m_id(id) {}
00076   // AddParameter can be used by derived classes to add parameters to list
00077   void AddParameter(const Variant &v) { m_parameters.push_back(v); }
00078
00079 private:
00081   // Private data
00083   RequestID m_id;
00084   VariantList m_parameters;
00085 };
00086
00087 // Useful typedefs
00088 typedef SmartPtr<Request> RequestPtr;
00089 typedef std::vector<RequestPtr> RequestList;
00090 typedef RequestList::iterator RequestListIter;
00091 typedef RequestList::const_iterator RequestListConstIter;
00092
00093 } // namespace rxdock
00094
00095 #endif //_RBTREQUEST_H_
```

6.139 RequestHandler.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract class for handling requests. Classes who want to handle requests
00014 // should derive from RequestHandler and override HandleRequest.
00015 // Useful for implementing a Chain of Responsibility in aggregate containers
00016
00017 #ifndef _RBTREQUESTHANDLER_H_
00018 #define _RBTREQUESTHANDLER_H_
00019
00020 #include "rxdock/Request.h"
00021
00022 namespace rxdock {
00023
00024 class RequestHandler {
00025 public:
00027   // Constructors/destructors
00028   RequestHandler() {}
```

```
00029   virtual ~RequestHandler() {}
00030
00032   // Public methods
00034   virtual void HandleRequest(RequestPtr spRequest) {}
00035 };
00036
00037 } // namespace rxdock
00038
00039 #endif //_RBTREQUESTHANDLER_H_
```

6.140 Resources.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Miscellaneous string constants
00014
00015 #ifndef _RBTRESOURCES_H_
00016 #define _RBTRESOURCES_H_
00017
00018 #include <string>
00019
00020 namespace rxdock {
00021
00022 const std::string IDS_NAME = "RxDock";
00023
00024 const std::string IDS_METADATA_PREFIX = "rxdock.";
00025
00026 const std::string IDS_COPYRIGHT =
00027     "The " + IDS_NAME +
00028     " molecular docking program is licensed under GNU LGPL version 3.\n" +
00029     IDS_NAME +
00030     " is maintained by Vedran Miletić, Patrik Nikolić, and Luka Vretenar.\n"
00031     "Visit https://rxdock.gitlab.io/ for more information.";
00032
00033 const std::string IDS_PRODUCT = "librxdock.so";
00034
00035 const std::string IDS_VERSION = "0.1.0";
00036
00037 } // namespace rxdock
00038
00039 #endif //_RBTRESOURCES_H_
```

6.141 RotSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple scoring function which estimates the loss of rotational entropy of the
00014 // ligand from the number of rotable bonds.
00015
00016 #ifndef _RBTROTSF_H_
00017 #define _RBTROTSF_H_
00018
00019 #include "rxdock/BaseInterSF.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
```

```
00026
00027 class RotSF : public BaseInterSF {
00028 public:
00029   // Static data member for class type
00030   static const std::string _CT;
00031   // Parameter names
00032   // Boolean controlling whether to include bonds to NH3+ in rotable bond count
00033   static const std::string _INCNH3;
00034   // Boolean controlling whether to include bonds to OH in  rotable bond count
00035   static const std::string _INCOH;
00036
00037   RotSF(const std::string &strName = "rot");
00038   virtual ~RotSF();
00039
00040   friend void to_json(json &j, const RotSF &rotsf);
00041   friend void from_json(const json &j, RotSF &rotsf);
00042
00043 protected:
00044   virtual void SetupReceptor();
00045   virtual void SetupLigand();
00046   virtual void SetupScore();
00047   virtual double RawScore() const;
00048   void ParameterUpdated(const std::string &strName);
00049
00050 private:
00051   int nRot;
00052   bool bIncNH3;
00053   bool bIncOH;
00054 };
00055
00056 void to_json(json &j, const RotSF &rotsf);
00057 void from_json(const json &j, RotSF &rotsf);
00058
00059 } // namespace rxdock
00060
00061 #endif //_RBTROTSF_H_
```

6.142 SAIdxSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Desolvation scoring function based on Weighted Solvent Accessible Surface
00014 // Area
00015
00016 #ifndef _RBTSAIDXSF_H_
00017 #define _RBTSAIDXSF_H_
00018
00019 #include "rxdock/AnnotationHandler.h"
00020 #include "rxdock/BaseIdxSF.h"
00021 #include "rxdock/BaseInterSF.h"
00022 #include "rxdock/NonBondedHHSGrid.h"
00023 #include "rxdock/ParameterFileSource.h"
00024 #include "rxdock/SATypes.h"
00025
00026 namespace rxdock {
00027
00028 class SAIdxSF : public BaseInterSF, public BaseIdxSF, public AnnotationHandler {
00029 public:
00030   RBTDLL_EXPORT SAIdxSF(const std::string &strName = "solv");
00031   virtual ~SAIdxSF();
00032   // write score components
00033   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00034
00035   static const std::string _CT;
00036   static const std::string _INCR;
00037
00038   // Request Handling method
00039   // Handles the Partition request
00040   virtual void HandleRequest(RequestPtr spRequest);
00041
00042 protected:
00043   virtual void SetupReceptor();
00044   virtual void SetupLigand();
```

```
00045   virtual void SetupSolvent();
00046   virtual void SetupScore();
00047   virtual double RawScore(void) const;
00048
00049   // clear indexed grids
00050   void ClearReceptor(void);
00051   void ClearLigand(void);
00052   void ClearSolvent(void);
00053
00054   double GetASP(HHSType::eType, double) const;
00055   double GetP_i(HHSType::eType) const;
00056   double GetR_i(HHSType::eType) const;
00057   void PrintWeightMatrix() const;
00058
00059 private:
00060   struct solvprms {
00061     solvprms(double r0, double p0, double asp0, bool b)
00062         : r(r0), p(p0), asp(asp0), chg_scaling(b) {}
00063     double r;
00064     double p;
00065     double asp;
00066     bool chg_scaling;
00067   };
00068
00069   typedef std::vector<SAIdxSF::solvprms> SolvTable;
00070   void Setup();
00071   HHS_SolvationRList CreateInteractionCenters(const AtomList &atomList) const;
00072   void BuildIntraMap(HHS_SolvationRList &ICList) const;
00073   void BuildIntraMap(HHS_SolvationRList &ICList1,
00074                      HHS_SolvationRList &ICList2) const;
00075
00076   // Sum the surface energies (ASP*area) for the list of solvation interaction
00077   // centers
00078   double TotalEnergy(const HHS_SolvationRList &intnCenters) const;
00079   void Partition(HHS_SolvationRList &intnCenters, double dist = 0.0);
00080
00081   HHS_SolvationRList theLSPList; // All ligand solvation interaction centers
00082   HHS_SolvationRList
00083       theRSPList; // All rigid receptor solvation interaction centers
00084   HHS_SolvationRList
00085       theFlexList; // All flexible receptor solvation interaction centers
00086   HHS_SolvationRList
00087       theCavList; // Subset of rigid receptor list within range of docking site
00088   // Subset of rigid receptor list NOT within range of docking site
00089   // but within range of the flexible atoms
00090   // Must remember to restore the invariant areas for these atoms
00091   // before updating the overlap of the flexible atoms
00092   HHS_SolvationRList thePeriphList;
00093   HHS_SolvationRList
00094       theSolventList; // DM 21 Dec 2005 - explicit solvent interaction centers
00095   NonBondedHHSGridPtr theIdxGrid;
00096   SolvTable m_solvTable;
00097   ParameterFileSourcePtr m_spSolvSource; // File source for solvation params
00098   double m_maxR;   // Maximum radius of any atom type, used to adjust Range()
00099                    // dynamically
00100   bool m_bFlexRec; // Is receptor flexible?
00101   mutable double
00102       m_lig_0; // Solvation energy of the free ligand (initial conformation)
00103   mutable double
00104       m_lig_free; // Solvation energy of the free ligand (current conformation)
00105   mutable double m_lig_bound;     // Solvation energy of the bound ligand
00106                                   // (current conformation)
00107   mutable double m_site_0;        // Solvation energy of the free docking site
00108                                   // (initial conformation)
00109   mutable double m_site_free;     // Solvation energy of the free docking site
00110                                   // (current conformation)
00111   mutable double m_site_bound;    // Solvation energy of the bound docking site
00112                                   // (current conformation)
00113   mutable double m_solvent_0;     // Solvation energy of the free explicit
00114                                   // solvent (initial conformation)
00115   mutable double m_solvent_free;  // Solvation energy of the free explicit
00116                                   // solvent (current conformation)
00117   mutable double m_solvent_bound; // Solvation energy of the bound explicit
00118                                   // solvent (current conformation)
00119 };
00120
00121 } // namespace rxdock
00122
00123 #endif // _RBTSAIDXSF_H_
```

6.143 SATypes.h

```
00001 /**********************************************************************
```

```
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 #ifndef _RBTSATYPES_H_
00014 #define _RBTSATYPES_H_
00015
00016 #include "rxdock/Config.h"
00017
00018 #include <functional>
00019
00020 namespace rxdock {
00021
00022 class Atom;
00023
00024 // Atom typing class for the solvation atom types
00025 // HHS = Hasel, Hendrickson and Still, Tetrahedron Computer Methodology, 1
00026 // (1988) 103-116 This paper describes the fast approximation to
00027 // solvent-accessible surface area used here Solvation atom types are inspired
00028 // by Wang, Wang, Huo, Lee and Kollman, J. Phys. Chem. B 105 (2001) 5055-5067
00029 class HHSType {
00030 public:
00031   enum eType {
00032     UNDEFINED = 0,
00033     C_sp3,
00034     CH_sp3,
00035     CH2_sp3,
00036     CH3_sp3,
00037     C_sp2,
00038     CH_sp2,
00039     CH2_sp2,
00040     C_sp2p,
00041     C_ar,
00042     CH_ar,
00043     C_sp,
00044     C_sp3_P,
00045     CH_sp3_P,
00046     CH2_sp3_P,
00047     CH3_sp3_P,
00048     C_sp2_P,
00049     CH_sp2_P,
00050     CH2_sp2_P,
00051     C_ar_P,
00052     CH_ar_P,
00053     H,
00054     HO,
00055     HN,
00056     HNp,
00057     HS,
00058     O_sp3,
00059     OH_sp3,
00060     OW,
00061     O_tri,
00062     OH_tri,
00063     O_sp2,
00064     ON,
00065     Om,
00066     N_sp3,
00067     NH_sp3,
00068     NH2_sp3,
00069     N_sp3p,
00070     N_tri,
00071     NH_tri,
00072     NH2_tri,
00073     N_sp2,
00074     N_sp2p,
00075     N_ar,
00076     N_sp,
00077     S_sp3,
00078     S_sp2,
00079     P,
00080     F,
00081     Cl,
00082     Br,
00083     I,
00084     Metal,
00085     MAXTYPES // KEEP AS LAST TYPE: used to size the atom name string list
00086   };
00087
00088   HHSType();
```

```
00089   virtual ~HHSType();
00090
00091   eType operator()(Atom *anAtom) const;
00092   std::string Type2Str(eType) const;
00093   eType Str2Type(const std::string &) const;
00094
00095 private:
00096   void SetupTypeNames();
00097   std::vector<std::string> m_typeNames;
00098 };
00099
00100 typedef SmartPtr<HHSType *> HHSTypePtr; // Smart pointer
00101 typedef std::vector<HHSType::eType> HHSTypeList;
00102 typedef HHSTypeList::iterator HHSTypeListIter;
00103 typedef HHSTypeList::const_iterator HHSTypeListConstIter;
00104
00105 // Interaction center class for solvation calculations
00106 // A "decorated" Atom in effect
00107 // Designed as a "create-once, read-many" object
00108 // Attributes (radius, ASP etc) cannot be changed once constructed
00109 // The Overlap method accumulates the surface areas for each atom
00110 // based on pair-wise distances to surrounding atoms
00111 class HHS_Solvation {
00112 public:
00113   // Constants from the HHS paper
00114   static const double r_s;    // solvent probe radius
00115   static const double d_s;    // solvent probe diameter
00116   static const double Pij_12; // correction factor for 1-2 connections
00117   static const double Pij_13; // ditto for 1-3
00118   static const double Pij_14; // and for 1-4+
00119
00120   // Default p_i, r_i, and sigma values
00121   HHS_Solvation(HHSType::eType t, Atom *a, double p = 1.0, double r = 1.0,
00122                 double s = 0.0);
00123
00124   // Get properties
00125   double GetA_i(void) const { return A_i; }
00126   double GetS_i(void) const { return S_i; }
00127   double GetP_i(void) const { return p_i; }
00128   double GetR_i(void) const { return r_i; }
00129   double GetSigma(void) const { return sigma; }
00130   HHSType::eType GetHHSType() const { return hhsType; }
00131   Atom *GetAtom() const { return atom; }
00132   inline double GetArea() const { return S_i * A_i; }   // Exposed area
00133   inline double GetEnergy() const { return E_i * A_i; } // Surface energy
00134
00135   // Actions
00136   // Initialise all calculated parameters to an isolated atom (no overlap)
00137   void Init();
00138   // Saves current value of A_i to A_inv
00139   // Use to memorise the partially calculated area due to invariant interaction
00140   // distances
00141   inline void Save() { A_inv = A_i; };
00142   // Restores A_inv to A_i
00143   // Use prior to continuing the calculation due to variable interaction
00144   // distances
00145   inline void Restore() { A_i = A_inv; };
00146   // Calculate overlap between this center and another (h)
00147   // Updates the exposed fractions (A_i) for both centers
00148   // p_ij is the correction factor for 1-2, 1-3, and 1-4+ connected atoms
00149   void Overlap(HHS_Solvation *h, double p_ij);
00150   // Get a list of all the current (partitioned) variable-distance interactions
00151   // to this center
00152   const std::vector<HHS_Solvation *> &GetVariable() const { return m_prt; }
00153   int GetNumVariable() const { return m_prt.size(); }
00154   // Add a variable-distance interaction
00155   void AddVariable(HHS_Solvation *anAtom);
00156   // Helper function to calculate the overlap for all the variable interactions
00157   // to this center All are 1-4+ by definition
00158   void OverlapVariable();
00159   // As above, but ignores disabled interaction centers
00160   void OverlapVariableEnabledOnly();
00161   // Partition the list of variable interactions
00162   // Copy those that are closer than distance d to the partitioned list
00163   // d=0 => clear the partition
00164   void Partition(double d = 0.0);
00165
00166 private:
00167   double p_i;   // atom-type dependent correction
00168   double r_i;   // quasi vdW radius
00169   double S_i;   // Surface area of isolated atom (constant)
00170   double A_i;   // Fraction of atom surface currently exposed
00171   double A_inv; // invariant part of atom surface (memory, used by Save/Restore)
00172   double sigma; // atomic solvation parameter
00173   double E_i;   // energy of isolated atom (S_i * sigma)
00174   double PI_r_i_plus_r_s; // PI*(r_i+r_s)
00175   double p_i_over_S_i;    // p_i / S_i
```

```
00176
00177   HHSType::eType hhsType;             // solvation atom type
00178   Atom *atom;                         // the Atom itself
00179   std::vector<HHS_Solvation *> m_var; // Vector of all variable distances
00180   std::vector<HHS_Solvation *>
00181       m_prt; // Vector of current partioned variable distances
00182 };
00183
00184 typedef std::vector<HHS_Solvation *> HHS_SolvationRList;
00185 typedef HHS_SolvationRList::iterator HHS_SolvationRListIter;
00186 typedef HHS_SolvationRList::const_iterator HHS_SolvationRListConstIter;
00187
00188 typedef std::vector<HHS_SolvationRList>
00189     HHS_SolvationListMap; // vector of regular pointers
00190 typedef HHS_SolvationListMap::iterator HHS_SolvationListMapIter;
00191 typedef HHS_SolvationListMap::const_iterator HHS_SolvationListMapConstIter;
00192
00193 typedef SmartPtr<HHS_Solvation> HHS_SolvationPtr; // Smart pointer
00194 typedef std::vector<HHS_SolvationPtr>
00195     HHS_SolvationList; // Vector of smart pointers
00196 typedef HHS_SolvationList::iterator HHS_SolvationListIter;
00197 typedef HHS_SolvationList::const_iterator HHS_SolvationListConstIter;
00198
00199 // Useful STL predicates and function objects
00200 class InitHHS {
00201 public:
00202   explicit InitHHS() {}
00203   void operator()(HHS_Solvation *pHHS) { pHHS->Init(); }
00204 };
00205 class SaveHHS {
00206 public:
00207   explicit SaveHHS() {}
00208   void operator()(HHS_Solvation *pHHS) { pHHS->Save(); }
00209 };
00210 class RestoreHHS {
00211 public:
00212   explicit RestoreHHS() {}
00213   void operator()(HHS_Solvation *pHHS) { pHHS->Restore(); }
00214 };
00215 class PartitionHHS {
00216   double d;
00217
00218 public:
00219   explicit PartitionHHS(double dd) : d(dd) {}
00220   void operator()(HHS_Solvation *pHHS) { pHHS->Partition(d); }
00221 };
00222 class OverlapVariableHHS {
00223 public:
00224   explicit OverlapVariableHHS() {}
00225   void operator()(HHS_Solvation *pHHS) { pHHS->OverlapVariable(); }
00226 };
00227 class OverlapHHS {
00228   HHS_Solvation *pHHSi;
00229   double p_ij;
00230
00231 public:
00232   explicit OverlapHHS(HHS_Solvation *pHHS, double p) : pHHSi(pHHS), p_ij(p) {}
00233   void operator()(HHS_Solvation *pHHSj) { pHHSi->Overlap(pHHSj, p_ij); }
00234 };
00235
00236 // Is HHS type equal to t ?
00237 class isHHSType_eq : public std::function<bool(HHS_Solvation *)> {
00238   HHSType::eType t;
00239
00240 public:
00241   explicit isHHSType_eq(HHSType::eType tt) : t(tt) {}
00242   bool operator()(const HHS_Solvation *pHHS) const {
00243     return pHHS->GetHHSType() == t;
00244   }
00245 };
00246
00247 // Is HHS_Solvation selected ?
00248 class isHHSSelected : public std::function<bool(HHS_Solvation *)> {
00249 public:
00250   explicit isHHSSelected() {}
00251   bool operator()(const HHS_Solvation *pHHS) const;
00252 };
00253
00254 inline double AccumHHSEnergy(double val, const HHS_Solvation *pHHS) {
00255   return val + pHHS->GetEnergy();
00256 }
00257
00258 } // namespace rxdock
00259
00260 #endif //_RBTSATYPES_H_
```

6.144 SetupPMFSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTSETUPPMFSF_H_
00014 #define _RBTSETUPPMFSF_H_
00015
00016 #include "rxdock/BaseInterSF.h"
00017
00018 namespace rxdock {
00019
00020 class SetupPMFSF : public BaseInterSF {
00021   AtomList theLigandList;   // ligand typing
00022   AtomList theReceptorList; // receptor typing
00023
00024 public:
00025   static const std::string _CT;
00026
00027   SetupPMFSF(const std::string &strName = "setup-pmf");
00028   ~SetupPMFSF();
00029
00030 protected:
00031   virtual void SetupReceptor();
00032   virtual void SetupLigand();
00033   virtual void SetupScore();
00034   virtual double RawScore() const;
00035
00036   void SetupReceptorPMFTypes(void);
00037   void SetupLigandPMFTypes(void);
00041   PMFType GetPMFfor_lC(AtomPtr); // for ligands
00042   PMFType GetPMFfor_lN(AtomPtr);
00043   PMFType GetPMFfor_lO(AtomPtr);
00044   PMFType GetPMFfor_lS(AtomPtr);
00045
00046   PMFType GetPMFfor_rC(AtomPtr); // for receptor
00047   PMFType GetPMFfor_rN(AtomPtr);
00048   PMFType GetPMFfor_rO(AtomPtr);
00049   PMFType GetPMFfor_rS(AtomPtr);
00050
00051   bool IsChargedNitrogen(AtomPtr); // true for guanidino or other charged
00052
00053 private:
00054   friend void to_json(json &j, const SetupPMFSF &pmfsf) {
00055     json atomList;
00056     for (const auto &cIter : pmfsf.theLigandList) {
00057       json atom = *cIter;
00058       atomList.push_back(atom);
00059     }
00060     json atomList2;
00061     for (const auto &cIter : pmfsf.theReceptorList) {
00062       json atom = *cIter;
00063       atomList2.push_back(atom);
00064     }
00065
00066     j = json{{"lig-list", atomList}, {"recep-list", atomList2}};
00067   };
00068   friend void from_json(const json &j, SetupPMFSF &pmfsf) {
00069     pmfsf.theLigandList.clear();
00070     pmfsf.theLigandList.reserve(j.at("lig-list").size());
00071     for (auto &atom : j.at("lig-list")) {
00072       AtomPtr spAtom = AtomPtr(new Atom(atom));
00073       pmfsf.theLigandList.push_back(spAtom);
00074     }
00075     pmfsf.theReceptorList.clear();
00076     pmfsf.theReceptorList.reserve(j.at("recep-list").size());
00077     for (auto &atom : j.at("recep-list")) {
00078       AtomPtr spAtom = AtomPtr(new Atom(atom));
00079       pmfsf.theReceptorList.push_back(spAtom);
00080     }
00081   };
00082 };
00083
00084 } // namespace rxdock
00085
00086 #endif //_RBTSETUPPMFSF_H_
```

6.145 SetupPolarSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Pseudo-scoring function who's only role is to setup the receptor local
00014 // neighbour density property and store in each atom's User1Value
00015 // This SF should be registered with the workspace BEFORE any SF which needs
00016 // this property (e.g. HBOND, IONIC) to ensure that User1Value is calculated
00017 // before being used. Scoring function is disabled by default.
00018
00019 #ifndef _RBTSETUPPOLARSF_H_
00020 #define _RBTSETUPPOLARSF_H_
00021
00022 #include "rxdock/BaseInterSF.h"
00023
00024 namespace rxdock {
00025
00026 class SetupPolarSF : public BaseInterSF {
00027 public:
00028   // Class type string
00029   static const std::string _CT;
00030   // Parameter names
00031   static const std::string _RADIUS;
00032   static const std::string _NORM;
00033   static const std::string _POWER;
00034   static const std::string _CHGFACTOR;
00035   // DM 14 Nov 2001 - relative strength of guanidinium intns
00036   static const std::string _GUANFACTOR;
00037
00038   RBTDLL_EXPORT SetupPolarSF(const std::string &strName = "setup-polar");
00039   virtual ~SetupPolarSF();
00040
00041 protected:
00042   virtual void SetupReceptor();
00043   virtual void SetupLigand();
00044   virtual void SetupSolvent();
00045   virtual void SetupScore();
00046   virtual double RawScore() const;
00047
00048 private:
00049   void SetupAtomList(AtomList &atomList, const AtomList &neighbourList);
00050 };
00051
00052 } // namespace rxdock
00053
00054 #endif //_RBTSETUPPOLARSF_H_
```

6.146 SetupSASF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTSETUPSASF_H_
00014 #define _RBTSETUPSASF_H_
00015
00016 #include "rxdock/BaseInterSF.h"
00017
00018 namespace rxdock {
00019
00020 // after Hasel, Hendrickson and Still
00021 struct SimpleSolvationParameters {
00022   double p;
00023   double r;
00024 };
```

```
00025
00026 class SetupSASF : public BaseInterSF {
00027   AtomList theLigandList;   // ligand typing
00028   AtomList theReceptorList; // receptor typing
00029
00030 public:
00031   static const std::string _CT;
00032
00033   SetupSASF(const std::string &strName = "setup-sa");
00034   ~SetupSASF();
00035
00036 protected:
00037   virtual void SetupReceptor();
00038   virtual void SetupLigand();
00039   virtual void SetupScore();
00040   virtual double RawScore() const;
00041
00042   void SetupReceptorSATypes(void);
00043   void SetupLigandSATypes(void);
00044
00045 private:
00046   friend void to_json(json &j, const SetupSASF &sasf) {
00047     json atomList;
00048     for (const auto &cIter : sasf.theLigandList) {
00049       json atom = *cIter;
00050       atomList.push_back(atom);
00051     }
00052     json atomList2;
00053     for (const auto &cIter : sasf.theReceptorList) {
00054       json atom = *cIter;
00055       atomList2.push_back(atom);
00056     }
00057
00058     j = json{{"ligand-list", atomList}, {"receptor-list", atomList2}};
00059   };
00060
00061   friend void from_json(const json &j, SetupSASF &sasf) {
00062     sasf.theLigandList.clear();
00063     sasf.theLigandList.reserve(j.at("atoms").size());
00064     for (auto &atom : j.at("atoms")) {
00065       AtomPtr spAtom = AtomPtr(new Atom(atom));
00066       sasf.theLigandList.push_back(spAtom);
00067     }
00068
00069     sasf.theReceptorList.clear();
00070     sasf.theReceptorList.reserve(j.at("atoms").size());
00071     for (auto &atom : j.at("atoms")) {
00072       AtomPtr spAtom = AtomPtr(new Atom(atom));
00073       sasf.theReceptorList.push_back(spAtom);
00074     }
00075   };
00076 };
00077
00078 } // namespace rxdock
00079
00080 #endif //_RBTSETUPSASF_H_
```

6.147 SFAgg.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Scoring Function aggregate class. Manages collection of child scoring
00014 // functions.
00015
00016 #ifndef _RBTSFAGG_H_
00017 #define _RBTSFAGG_H_
00018
00019 #include "rxdock/BaseSF.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
```

```
00025 namespace rxdock {
00026
00027 // Only check SF aggregate assertions in debug build
00028 #ifdef _NDEBUG
00029 const Bool SFAGG_CHECK = false;
00030 #else
00031 const bool SFAGG_CHECK = true;
00032 #endif //_NDEBUG
00033
00034 // Useful typedefs
00035
00036 class SFAgg : public BaseSF {
00037 public:
00038   // Static data member for class type (i.e. "SFAgg")
00039   static const std::string _CT;
00040
00042   // Constructors/destructors
00043   RBTDLL_EXPORT SFAgg(const std::string &strName = GetMetaDataPrefix() +
00044                                                    "score");
00045   virtual ~SFAgg();
00046
00047   friend void to_json(json &j, const SFAgg &sfAgg);
00048   friend void from_json(const json &j, SFAgg &sfAgg);
00049
00051   // Public methods
00053
00054   // Returns all child component scores as a string-variant map
00055   // Key = fully qualified component name, value = weighted score
00056   //(for saving in a Model's data fields)
00057   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00058
00059   // Aggregate handling methods
00060   virtual void Add(BaseSF *);
00061   virtual void Remove(BaseSF *);
00062   virtual bool isAgg() const;
00063   virtual unsigned int GetNumSF() const;
00064   virtual BaseSF *GetSF(unsigned int iSF) const;
00065
00066   // WorkSpace handling methods
00067   // Register scoring function with a workspace
00068   // Aggregate version registers all children, but NOT itself
00069   //(Aggregates are just containers, and have no need for model information
00070   virtual void Register(WorkSpace *);
00071   // Unregister with a workspace
00072   // Aggregate version unregisters all children, but NOT itself
00073   virtual void Unregister();
00074
00075   // Override Observer pure virtual
00076   // Notify observer that subject has changed
00077   // Does nothing in SFAgg as aggregates do not require updating
00078   virtual void Update(Subject *theChangedSubject);
00079
00080   // Request Handling method
00081   virtual void HandleRequest(RequestPtr spRequest);
00082
00083   // Virtual function for dumping scoring function details to an output stream
00084   // Called by operator «
00085   virtual void Print(std::ostream &s) const;
00086
00087 protected:
00089   // Protected methods
00091   virtual double RawScore() const;
00092
00093 private:
00095   // Private methods
00097   SFAgg(const SFAgg &);            // Copy constructor disabled by default
00098   SFAgg &operator=(const SFAgg &); // Copy assignment disabled by default
00099
00100 protected:
00102   // Protected data
00104
00105 private:
00107   // Private data
00109   BaseSFList m_sf;
00110   int m_nNonHLigandAtoms; // for normalised scores (score / non-H ligand atoms)
00111 };
00112
00113 void to_json(json &j, const SFAgg &sfAgg);
00114 void from_json(const json &j, SFAgg &sfAgg);
00115
00116 // Useful typedefs
00117 typedef SmartPtr<SFAgg> SFAggPtr;        // Smart pointer
00118 typedef std::vector<SFAggPtr> SFAggList; // Vector of smart pointers
00119 typedef SFAggList::iterator SFAggListIter;
00120 typedef SFAggList::const_iterator SFAggListConstIter;
00121
00122 } // namespace rxdock
```

```
00123
00124 #endif //_RBTSFAGG_H_
```

6.148 SFFactory.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Factory class for creating scoring function objects. Scoring functions are
00014 // specified by string names (as defined in each class's static _CT string)
00015
00016 #ifndef _RBTSFFACTORY_H_
00017 #define _RBTSFFACTORY_H_
00018
00019 #include "rxdock/Config.h"
00020 #include "rxdock/ParameterFileSource.h"
00021 #include "rxdock/SFAgg.h"
00022
00023 namespace rxdock {
00024
00025 class SFFactory {
00026   // Parameter name which identifies a scoring function definition
00027   static const std::string _SF;
00028
00029 public:
00031   // Constructors/destructors
00032
00033   RBTDLL_EXPORT SFFactory(); // Default constructor
00034
00035   virtual ~SFFactory(); // Default destructor
00036
00038   // Public methods
00040
00041   // Creates a single scoring function object of type strSFClass, and name
00042   // strName e.g. strSFClass = HBondIntnSF
00043   virtual BaseSF *Create(const std::string &_strSFClass,
00044                          const std::string &strName);
00045
00046   // Creates an aggregate scoring function from a parameter file source
00047   // Each component SF is in a named section, which should minimally contain a
00048   // SCORING_FUNCTION parameter whose value is the class name to instantiate
00049   // strSFClasses contains a comma-delimited list of SF class names to
00050   // instantiate If strSFClasses is empty, all named sections in spPrmSource are
00051   // scanned for valid scoring function definitions SF parameters are set from
00052   // the list of parameters in each named section
00053   virtual SFAgg *
00054   CreateAggFromFile(ParameterFileSourcePtr spPrmSource,
00055                     const std::string &strName,
00056                     const std::string &strSFClasses = std::string());
00057
00058 protected:
00060   // Protected methods
00062
00063 private:
00065   // Private methods
00067
00068   SFFactory(const SFFactory &); // Copy constructor disabled by default
00069
00070   SFFactory &
00071   operator=(const SFFactory &); // Copy assignment disabled by default
00072
00073 protected:
00075   // Protected data
00077
00078 private:
00080   // Private data
00082 };
00083
00084 // Useful typedefs
00085 typedef SmartPtr<SFFactory> SFFactoryPtr; // Smart pointer
00086
00087 } // namespace rxdock
00088
00089 #endif //_RBTSFFACTORY_H_
```

6.149 SFRequest.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Scoring function specific requests
00014
00015 #ifndef _RBTSFREQUEST_H_
00016 #define _RBTSFREQUEST_H_
00017
00018 #include "rxdock/Request.h"
00019
00020 namespace rxdock {
00021
00022 // Request IDs
00023 const RequestID ID_REQ_SF_ENABLE = 1;
00024 const RequestID ID_REQ_SF_DISABLE = 2;
00025 const RequestID ID_REQ_SF_PARTITION = 3;
00026 const RequestID ID_REQ_SF_SETPARAM = 4;
00027
00028 class SFEnableRequest : public Request {
00029 public:
00030   // Request to enable a particular named scoring function
00031   SFEnableRequest(const std::string &sfName) : Request(ID_REQ_SF_ENABLE) {
00032     AddParameter(sfName);
00033   }
00034 };
00035
00036 class SFDisableRequest : public Request {
00037 public:
00038   // Request to disable a particular named scoring function
00039   SFDisableRequest(const std::string &sfName) : Request(ID_REQ_SF_DISABLE) {
00040     AddParameter(sfName);
00041   }
00042 };
00043
00044 class SFPartitionRequest : public Request {
00045 public:
00046   // Request to partition all scoring functions to a particular distance
00047   SFPartitionRequest(double dist) : Request(ID_REQ_SF_PARTITION) {
00048     AddParameter(dist);
00049   }
00050   // Request to partition a particular named scoring function to a particular
00051   // distance
00052   SFPartitionRequest(const std::string &sfName, double dist)
00053       : Request(ID_REQ_SF_PARTITION) {
00054     AddParameter(sfName);
00055     AddParameter(dist);
00056   }
00057 };
00058
00059 class SFSetParamRequest : public Request {
00060 public:
00061   // Request to set a named parameter of all scoring functions to a new value
00062   SFSetParamRequest(const std::string &paramName, const Variant &paramValue)
00063       : Request(ID_REQ_SF_SETPARAM) {
00064     AddParameter(paramName);
00065     AddParameter(paramValue);
00066   }
00067   // Request to set a named parameter of a named scoring function to a new value
00068   SFSetParamRequest(const std::string &sfName, const std::string &paramName,
00069                     const Variant &paramValue)
00070       : Request(ID_REQ_SF_SETPARAM) {
00071     AddParameter(sfName);
00072     AddParameter(paramName);
00073     AddParameter(paramValue);
00074   }
00075 };
00076
00077 } // namespace rxdock
00078
00079 #endif //_RBTSFREQUEST_H_
```

6.150 SimAnnTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simple, non-adaptive simulated annealing protocol
00014 #ifndef _RBTSIMANNTRANSFORM_H_
00015 #define _RBTSIMANNTRANSFORM_H_
00016
00017 #include "rxdock/BaseBiMolTransform.h"
00018 #include "rxdock/ChromElement.h"
00019 #include "rxdock/Rand.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 // Simple class to keep track of Monte Carlo sampling statistics
00028 class MCStats {
00029 public:
00030   MCStats();
00031   void Init(double score);
00032   void InitBlock(double score);
00033   void Accumulate(double score, bool bAccepted);
00034   double Mean() const;
00035   double Variance() const;
00036   double AccRate() const;
00037   double _total;
00038   double _total2;
00039   double _blockInitial;
00040   double _blockFinal;
00041   double _blockMin;
00042   double _blockMax;
00043   double _initial;
00044   double _final;
00045   double _min;
00046   double _max;
00047   int _steps;
00048   int _accepted;
00049 };
00050 typedef SmartPtr<MCStats> MCStatsPtr; // Smart pointer
00051
00052 class SimAnnTransform : public BaseBiMolTransform {
00053 public:
00054   // Static data member for class type
00055   static const std::string _CT;
00056   // Parameter names
00057   static const std::string _START_T;
00058   static const std::string _FINAL_T;
00059   static const std::string _BLOCK_LENGTH;
00060   static const std::string _SCALE_CHROM_LENGTH;
00061   static const std::string _NUM_BLOCKS;
00062   static const std::string _STEP_SIZE;
00063   static const std::string _MIN_ACC_RATE;
00064   static const std::string _PARTITION_DIST;
00065   static const std::string _PARTITION_FREQ;
00066   static const std::string _HISTORY_FREQ;
00067
00068   RBTDLL_EXPORT static const std::string &GetStartT();
00069   RBTDLL_EXPORT static const std::string &GetFinalT();
00070   RBTDLL_EXPORT static const std::string &GetBlockLength();
00071   RBTDLL_EXPORT static const std::string &GetNumBlocks();
00072   RBTDLL_EXPORT static const std::string &GetStepSize();
00073   RBTDLL_EXPORT static const std::string &GetPartitionDist();
00074   RBTDLL_EXPORT static const std::string &GetPartitionFreq();
00075
00077   // Constructors/destructors
00078   RBTDLL_EXPORT SimAnnTransform(const std::string &strName = "SIMANN");
00079   virtual ~SimAnnTransform();
00080
00081   friend void to_json(json &j, const SimAnnTransform &simAnnTransform);
00082   friend void from_json(const json &j, SimAnnTransform &simAnnTransform);
00083
00085   // Public methods
00087
00088 protected:
```

```
00090   // Protected methods
00092   virtual void SetupReceptor(); // Called by Update when receptor is changed
00093   virtual void SetupLigand();   // Called by Update when ligand is changed
00094   virtual void
00095   SetupTransform(); // Called by Update when either model has changed
00096   void MC(double t, int blockLen, double stepSize);
00097   virtual void Execute();
00098
00099 private:
00101   // Private methods
00103   SimAnnTransform(
00104       const SimAnnTransform &); // Copy constructor disabled by default
00105   SimAnnTransform &
00106   operator=(const SimAnnTransform &); // Copy assignment disabled by default
00107
00108 protected:
00110   // Protected data
00112
00113 private:
00115   // Private data
00117   MCStatsPtr m_spStats;
00118   Rand &m_rand; // keep a reference to the singleton random number generator
00119   RequestPtr m_spPartReq;  // Partitioning request
00120   ChromElementPtr m_chrom; // Current chromosome
00121   std::vector<double>
00122       m_minVector; // Chromosome vector corresponding to overall minimum score
00123   std::vector<double> m_lastGoodVector; // Saved chromosome before each MC
00124                                         // mutation (to allow revert)
00125 };
00126
00127 void to_json(json &j, const SimAnnTransform &simAnnTransform);
00128 void from_json(const json &j, SimAnnTransform &simAnnTransform);
00129
00130 // Useful typedefs
00131 typedef SmartPtr<SimAnnTransform> SimAnnTransformPtr; // Smart pointer
00132
00133 } // namespace rxdock
00134
00135 #endif //_RBTSIMANNTRANSFORM_H_
```

6.151 SimplexCostFunction.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 #ifndef _RBTSIMPLEXCOSTFUNCTION_H
00014 #define _RBTSIMPLEXCOSTFUNCTION_H
00015
00016 #ifdef __PGI
00017 #define EIGEN_DONT_VECTORIZE
00018 #endif
00019 #include <Eigen/Core>
00020
00021 #include "rxdock/BaseSF.h"
00022 #include "rxdock/ChromElement.h"
00023
00024 namespace rxdock {
00025
00026 // Rosenbrock cost function
00027 class SimplexCostFunction {
00028 public:
00029   SimplexCostFunction(BaseSF *pSF, ChromElementPtr chrom)
00030       : m_pSF(pSF), m_chrom(chrom) {}
00031   typedef double DataType;
00032   typedef Eigen::VectorXd ParameterType;
00033   double operator()(const ParameterType &parameters) { // const
00034     nCalls++;
00035     std::vector<double> vec(parameters.data(),
00036                             parameters.data() + parameters.size());
00037     m_chrom->SetVector(vec);
00038     m_chrom->SyncToModel();
00039     return m_pSF->Score();
00040   }
00041   unsigned long int nCalls = 0;
```

```
00042
00043 private:
00044   BaseSF *m_pSF;
00045   ChromElementPtr m_chrom;
00046 };
00047
00048 } // namespace rxdock
00049
00050 #endif // _RBTSIMPLEXCOSTFUNCTION_H
```

6.152 SimplexTransform.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Simplex Minimiser
00014 #ifndef _RBTSIMPLEXTRANSFORM_H_
00015 #define _RBTSIMPLEXTRANSFORM_H_
00016
00017 #include "rxdock/BaseBiMolTransform.h"
00018 #include "rxdock/ChromElement.h"
00019
00020 namespace rxdock {
00021
00022 class SimplexTransform : public BaseBiMolTransform {
00023 public:
00024   // Static data member for class type
00025   static const std::string _CT;
00026   // Parameter names
00027   static const std::string _MAX_CALLS;
00028   static const std::string _NCYCLES;
00029   static const std::string _STOPPING_STEP_LENGTH;
00030   static const std::string _PARTITION_DIST;
00031   static const std::string _STEP_SIZE;
00032   // Stop once score improves by less than convergence value
00033   // between cycles
00034   static const std::string _CONVERGENCE;
00035
00036   RBTDLL_EXPORT static const std::string &GetMaxCalls();
00037   RBTDLL_EXPORT static const std::string &GetNCycles();
00038   RBTDLL_EXPORT static const std::string &GetStepSize();
00039
00041   // Constructors/destructors
00042   RBTDLL_EXPORT SimplexTransform(const std::string &strName = "SIMPLEX");
00043   virtual ~SimplexTransform();
00044
00046   // Public methods
00048
00049 protected:
00051   // Protected methods
00053   virtual void
00054   SetupTransform(); // Called by Update when either model has changed
00055   virtual void SetupReceptor(); // Called by Update when receptor is changed
00056   virtual void SetupLigand();   // Called by Update when ligand is changed
00057   virtual void SetupSolvent();  // Called by Update when solvent is changed
00058   virtual void Execute();
00059
00060 private:
00062   // Private methods
00064   SimplexTransform(
00065       const SimplexTransform &); // Copy constructor disabled by default
00066   SimplexTransform &
00067   operator=(const SimplexTransform &); // Copy assignment disabled by default
00068 protected:
00070   // Protected data
00072
00073 private:
00075   // Private data
00077   ChromElementPtr m_chrom;
00078 };
00079
00080 // Useful typedefs
00081 typedef SmartPtr<SimplexTransform> SimplexTransformPtr; // Smart pointer
00082
```

```
00083 } // namespace rxdock
00084
00085 #endif //_RBTSIMPLEXTRANSFORM_H_
```

6.153 Singleton.h

```
00001 #ifndef SINGLETON_H
00002 #define SINGLETON_H
00003
00004 namespace rxdock {
00005
00006 template <class singleton_type> class Singleton {
00007 protected:
00008   enum action { DESTROY, CREATE, GET };
00009   static singleton_type *object(const action &a) {
00010     static singleton_type *obj = nullptr;
00011     switch (a) {
00012     case DESTROY:
00013       delete obj;
00014       break;
00015     case GET:
00016       return obj;
00017     case CREATE:
00018       return (obj = new singleton_type);
00019     }
00020     return nullptr;
00021   }
00022
00023 public:
00024   inline static singleton_type &instance() {
00025     if (object(GET) != nullptr) {
00026       return *object(GET);
00027     } else {
00028       return *object(CREATE);
00029     }
00030   }
00031
00032   virtual ~Singleton() { object(DESTROY); }
00033
00034 protected:
00035   Singleton() {}
00036 };
00037
00038 } // namespace rxdock
00039
00040 #endif
```

6.154 SiteMapper.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Base class for docking site mapper function classes
00014 // Designed in a similar fashion to scoring functions, in that mapper objects
00015 // are attached to a workspace and can respond to changes in the defined
00016 // receptor and ligand models
00017 // Sub classes should implement operator() which returns a list of cavity smart
00018 // pointers
00019
00020 #ifndef _RBTSITEMAPPER_H_
00021 #define _RBTSITEMAPPER_H_
00022
00023 #include "rxdock/BaseObject.h"
00024 #include "rxdock/Cavity.h"
00025 #include "rxdock/Config.h"
00026 #include "rxdock/Model.h"
00027
00028 #include <nlohmann/json.hpp>
00029
00030 using json = nlohmann::json;
```

```
00031
00032 namespace rxdock {
00033
00034 class SiteMapper : public BaseObject {
00035 public:
00036   // Class type string
00037   static const std::string _CT;
00038
00040   // Constructors/destructors
00041   virtual ~SiteMapper();
00042
00043   friend void to_json(json &j, const SiteMapper &siteMapper);
00044   friend void from_json(const json &j, SiteMapper &siteMapper);
00045
00047   // Public methods
00049   ModelPtr GetReceptor() const { return m_spReceptor; }
00050
00051   // PURE VIRTUAL - subclasses must override
00052   // NB - subclasses should also override Observer::Update pure virtual
00053   virtual CavityList operator()() = 0;
00054
00055   // Override Observer pure virtual
00056   // Notify observer that subject has changed
00057   virtual void Update(Subject *theChangedSubject);
00058
00059 protected:
00061   // Protected methods
00063   SiteMapper(const std::string &strClass, const std::string &strName);
00064
00065 private:
00067   // Private methods
00069   SiteMapper();                   // default constructor disabled
00070   SiteMapper(const SiteMapper &); // Copy constructor disabled by default
00071   SiteMapper &
00072   operator=(const SiteMapper &); // Copy assignment disabled by default
00073
00074 protected:
00076   // Protected data
00078
00079 private:
00081   // Private data
00083   ModelPtr m_spReceptor;
00084 };
00085
00086 void to_json(json &j, const SiteMapper &siteMapper);
00087 void from_json(const json &j, SiteMapper &siteMapper);
00088
00089 // Useful typedefs
00090 typedef SmartPtr<SiteMapper> SiteMapperPtr; // Smart pointer
00091
00092 } // namespace rxdock
00093
00094 #endif //_RBTSITEMAPPER_H_
```

6.155 SiteMapperFactory.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Factory class for creating docking site mapper objects.
00014
00015 #ifndef _RBTSITEMAPPERFACTORY_H_
00016 #define _RBTSITEMAPPERFACTORY_H_
00017
00018 #include "rxdock/Config.h"
00019 #include "rxdock/ParameterFileSource.h"
00020 #include "rxdock/SiteMapper.h"
00021
00022 namespace rxdock {
00023
00024 class SiteMapperFactory {
00025   // Parameter name which identifies a site mapper definition
00026   static const std::string _MAPPER;
00027
```

```
00028 public:
00030   // Constructors/destructors
00031
00032   RBTDLL_EXPORT SiteMapperFactory(); // Default constructor
00033   virtual ~SiteMapperFactory();      // Default destructor
00034
00036   // Public methods
00038
00039   // Creates a single site mapper object of named class
00040   virtual SiteMapper *Create(const std::string &_strMapperClass,
00041                              const std::string &strName);
00042
00043   // Creates a single site mapper object whose class is defined
00044   // as the value of the SITE_MAPPER parameter in the strName section of the
00045   // parameter file source argument. Also sets the site mapper parameters from
00046   // the remaining parameter values in the current section Note: the current
00047   // section is restored to its original value upon exit
00048   virtual SiteMapper *CreateFromFile(ParameterFileSourcePtr spPrmSource,
00049                                      const std::string &strName);
00050
00051 protected:
00053   // Protected methods
00055
00056 private:
00058   // Private methods
00060
00061   SiteMapperFactory(
00062       const SiteMapperFactory &); // Copy constructor disabled by default
00063   SiteMapperFactory &
00064   operator=(const SiteMapperFactory &); // Copy assignment disabled by default
00065
00066 protected:
00068   // Protected data
00070
00071 private:
00073   // Private data
00075 };
00076
00077 // Useful typedefs
00078 typedef SmartPtr<SiteMapperFactory> SiteMapperFactoryPtr; // Smart pointer
00079
00080 } // namespace rxdock
00081
00082 #endif //_RBTSITEMAPPERFACTORY_H_
```

6.156 SmartPointer.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Smart pointer template which can handle normal class inheritance
00014 // relationships. Inspired by Omar Loggiodice's ookit SharedPtr and
00015 // David Harvey's "Smart pointer templates in C++" article
00016 //(http://web.ftech.net/~honeyg/articles/smartp.htm)
00017 //
00018 // We don't use a separate reference counting object, but rather store two
00019 // pointers in each smart pointer:
00020 // 1) Pointer to underlying object
00021 // 2) Pointer to unsigned int (reference counter)
00022 //
00023 // Pros: the only way I could think of to be able to implement assignment of
00024 //       subclass smart pointer to base class smart pointer
00025 //       Possibly faster access to underlying object (only one dereference
00026 //       rather than two)
00027 // Cons: Less memory efficient (store two pointers per smart pointer, not one)
00028 //
00029 // To do: Should probably throw some other exception than std::exception if
00030 //        assert fails
00031 // NOTE: it's a BAD idea to pass a nullptr pointer to the SmartPtr<T>
00032 // constructor
00033
00034 #ifndef _RBTSMARTPOINTER_H_
00035 #define _RBTSMARTPOINTER_H_
00036
```

```
00037 //#include "rxdock/Error.h"
00038
00039 namespace rxdock {
00040
00041 // Only check smart pointer assertions in debug build
00042 //#ifdef _NDEBUG
00043 // const Bool SMART_CHECK = false;
00044 //#else
00045 // const Bool SMART_CHECK = true;
00046 //#endif //_NDEBUG
00047
00048 template <class T> class SmartPtr {
00049 public:
00051   // CONSTRUCTORS, DESTRUCTORS, ASSIGNMENT
00052
00053   // Default constructor, initialise both pointers to 0
00054   SmartPtr() : m_pT(nullptr), m_pCount(nullptr) {}
00055
00056   // Parameterised constructor
00057   // Create new counter, initialise to 1
00058   SmartPtr(T *pT) : m_pT(pT), m_pCount(new unsigned int(1)) {}
00059
00060   // Copy constructor - copy both pointers, increment counter
00061   SmartPtr(const SmartPtr<T> &sp) : m_pT(sp.m_pT), m_pCount(sp.m_pCount) {
00062     if (!Null())
00063       GetRef();
00064   }
00065
00066   // Copy constructor - template version for creating a base class smart pointer
00067   // from a subclass smart pointer. Need to use public accessor methods for
00068   // copying the pointers
00069   template <class T2> SmartPtr(const SmartPtr<T2> &sp) {
00070     m_pCount = sp.GetCountPtr();
00071     T2 *pT2 = const_cast<T2 *>(sp.Ptr());
00072     m_pT = dynamic_cast<T *>(pT2);
00073     if (!Null())
00074       GetRef();
00075   }
00076
00077   // Destructor - decrement counter, delete underlying object
00078   // if no refs remaining
00079   ~SmartPtr() { UnBind(); }
00080
00081   // Assignment operator - check for self assignment
00082   const SmartPtr<T> &operator=(const SmartPtr<T> &sp) {
00083     if (this != &sp) {
00084       UnBind();
00085       m_pCount = sp.m_pCount;
00086       m_pT = sp.m_pT;
00087       if (!sp.Null())
00088         sp.GetRef();
00089     }
00090     return *this;
00091   }
00092
00093   // Assignment operator - template version for assigning a subclass smart
00094   // pointer to a base class smart pointer. Will only compile if T2 is a
00095   // subclass of T No need to worry about self assignment here so can unbind the
00096   // existing ref first. Need to use public accessor methods to access the
00097   // pointers on rhs
00098   template <class T2> const SmartPtr<T> &operator=(const SmartPtr<T2> &sp) {
00099     UnBind();
00100     m_pCount = sp.GetCountPtr();
00101     T2 *pT2 = const_cast<T2 *>(sp.Ptr());
00102     m_pT = dynamic_cast<T *>(pT2);
00103     if (!Null())
00104       GetRef();
00105     return *this;
00106   }
00107
00109   // FRIEND FUNCTIONS - declaration
00110   // operator== checks for equivalence of underlying pointers
00111   // friend bool operator==(const SmartPtr<T>& lhs, const SmartPtr<T>& rhs);
00112
00114   // PUBLIC ACCESSOR METHODS
00115
00116   // Tests if smart pointer is empty (i.e. does it have a counter?)
00117   // Doesn't actually tell you if the underlying pointer is null
00118   bool Null() const { return m_pCount == nullptr; }
00119
00120   // Returns pointer to counter
00121   unsigned *GetCountPtr() const { return m_pCount; }
00122
00123   // Returns underlying pointer
00124   T *Ptr() { return m_pT; }
00125   const T *Ptr() const { return m_pT; }
00126
```

```
00127   // Manually remove reference to underlying object
00128   void SetNull() { UnBind(); }
00129
00130   // Various means of dereferencing the underlying object
00131   // Const and non-const versions
00132   // Throw assert error if smart pointer is empty
00133   T *operator->() {
00134     // Assert<Assertion>(!SMART_CHECK||!Null());
00135     return m_pT;
00136   }
00137   const T *operator->() const {
00138     // Assert<Assertion>(!SMART_CHECK||!Null());
00139     return m_pT;
00140   }
00141   T &operator*() {
00142     // Assert<Assertion>(!SMART_CHECK||!Null());
00143     return *m_pT;
00144   }
00145   const T &operator*() const {
00146     // Assert<Assertion>(!SMART_CHECK||!Null());
00147     return *m_pT;
00148   }
00149
00150   // DM 12 Jun 2000 - yet another way to dereference the smart pointer
00151   // Allows a smart pointer to be passed as a regular pointer in a function call
00152   // for example.
00153   operator T *() const { return m_pT; }
00154
00156   // PRIVATE METHODS AND DATA
00157 private:
00158   // Increments counter and returns new value
00159   unsigned GetRef() const { return ++(*m_pCount); }
00160   // Decrements counter and returns new value
00161   // ASSERT: counter should be non-zero before decrementing
00162   unsigned FreeRef() const {
00163     // Assert<Assertion>(!SMART_CHECK||(*m_pCount)!=0);
00164     return --(*m_pCount);
00165   }
00166   // Decrements counter and deletes underlying object and counter
00167   // if count is zero
00168   void UnBind() {
00169     if (!Null() && FreeRef() == 0) {
00170       delete m_pT;
00171       delete m_pCount;
00172     }
00173     m_pT = nullptr;
00174     m_pCount = nullptr;
00175   }
00176
00177   T *m_pT;            // Pointer to the underlying object
00178   unsigned *m_pCount; // Pointer to counter
00179 };
00180
00182 // FRIEND FUNCTIONS - implementation
00183 // operator== checks for equivalence of underlying pointers
00184 // DM 22 May 2000 - use .Ptr() method so we don't have to declare as friend
00185 template <class T>
00186 bool operator==(const SmartPtr<T> &lhs, const SmartPtr<T> &rhs) {
00187   return lhs.Ptr() == rhs.Ptr();
00188 }
00189
00190 // operator!= checks for non-equivalence of underlying pointers
00191 template <class T>
00192 bool operator!=(const SmartPtr<T> &lhs, const SmartPtr<T> &rhs) {
00193   return !(lhs == rhs);
00194 }
00195
00196 // DM 13 Jul 2000 - operator< for sorting containers of smart pointers
00197 template <class T>
00198 bool operator<(const SmartPtr<T> &lhs, const SmartPtr<T> &rhs) {
00199   return lhs.Ptr() < rhs.Ptr();
00200 }
00201
00202 // DM 13 Jul 2000 - operator> for sorting containers of smart pointers
00203 template <class T>
00204 bool operator>(const SmartPtr<T> &lhs, const SmartPtr<T> &rhs) {
00205   return lhs.Ptr() > rhs.Ptr();
00206 }
00207
00208 } // namespace rxdock
00209
00210 #endif //_RBTSMARTPOINTER_H_
```

6.157 SolventFlexData.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages flexibility data for solvent
00014 #ifndef _RBTSOLVENTFLEXDATA_
00015 #define _RBTSOLVENTFLEXDATA_
00016
00017 #include "rxdock/LigandFlexData.h"
00018
00019 namespace rxdock {
00020
00021 class SolventFlexData : public LigandFlexData {
00022 public:
00023   // Probability of solvent being visible
00024   static const std::string &_OCCUPANCY;
00025   // Nominal step size for mutating the occupancy variable
00026   static const std::string &_OCCUPANCY_STEP;
00027   SolventFlexData(DockingSite *pDockSite);
00028   virtual void Accept(FlexDataVisitor &v) { v.VisitSolventFlexData(this); }
00029
00030 private:
00031   SolventFlexData();
00032 };
00033
00034 } // namespace rxdock
00035
00036 #endif //_RBTSOLVENTFLEXDATA_
```

6.158 SphereSiteMapper.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // 2-sphere site mapper function class
00014
00015 #ifndef _RBTSPHERESITEMAPPER_H_
00016 #define _RBTSPHERESITEMAPPER_H_
00017
00018 #include "rxdock/SiteMapper.h"
00019
00020 namespace rxdock {
00021
00022 class SphereSiteMapper : public SiteMapper {
00023 public:
00024   // Static data member for class type
00025   static const std::string _CT;
00026   // Parameter names
00027   static const std::string _VOL_INCR;
00028   static const std::string _SMALL_SPHERE;
00029   static const std::string _LARGE_SPHERE;
00030   static const std::string _GRIDSTEP;
00031   static const std::string _CENTER;
00032   static const std::string _RADIUS;
00033   static const std::string _MIN_VOLUME;
00034   static const std::string _MAX_CAVITIES;
00035
00036   SphereSiteMapper(const std::string &strName = "SPHERE_MAPPER");
00037   virtual ~SphereSiteMapper();
00038
00039   // Override SiteMapper pure virtual
00040   // This is the function which actually does the mapping
00041   virtual CavityList operator()();
00042 };
```

```
00043
00044 // Useful typedefs
00045 typedef SmartPtr<SphereSiteMapper> SphereSiteMapperPtr; // Smart pointer
00046
00047 } // namespace rxdock
00048
00049 #endif //_RBTSPHERESITEMAPPER_H_
```

6.159 StringTokenIter.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract StringTokenIter class. Defines the interface for a token iterator
00014 // to be used by the parser.
00015
00016 #ifndef _RBT_STRINGTOKENITER_H_
00017 #define _RBT_STRINGTOKENITER_H_
00018
00019 #include "rxdock/Config.h"
00020 #include "rxdock/Context.h"
00021 #include "rxdock/TokenIter.h"
00022 #include "rxdock/geneticprogram/GPChromosome.h"
00023 #include <fstream>
00024 #include <stack>
00025
00026 namespace rxdock {
00027
00028 typedef SmartPtr<std::istream> istreamPtr;
00029 class StringTokenIter : public TokenIter {
00030 public:
00031   static const std::string _CT;
00033   // Constructors
00035   StringTokenIter(const StringTokenIter &);
00036   StringTokenIter(const istreamPtr, ContextPtr);
00037
00039   // Destructor
00041   virtual ~StringTokenIter();
00042
00043   void copy(const StringTokenIter &);
00044   // void First();
00045   void Next(ContextPtr);
00046   // Bool IsDone() const;
00047   TokenPtr Current();
00048
00049   //  Bool operator==(const StringTokenIter&) const;
00050   //  std::ostream& Print(std::ostream&) const;
00051   //  friend std::ostream& operator«(std::ostream& s, const StringTokenIter
00052   //  &p);
00053
00055   // Private methods
00057
00058 private:
00059   StringTokenIter(); // Default constructor disabled
00060   TokenPtr translate(std::string);
00061
00062   istreamPtr filep;
00063   TokenPtr current;
00064   std::string strtok;
00065   ContextPtr contextp;
00066 };
00067
00068 // Useful typedefs
00069 typedef SmartPtr<StringTokenIter> StringTokenIterPtr; // Smart pointer
00070 typedef std::vector<StringTokenIterPtr>
00071     StringTokenIterList; // Vector of smart pointers
00072 typedef StringTokenIterList::iterator StringTokenIterListIter;
00073 typedef StringTokenIterList::const_iterator StringTokenIterListConstIter;
00074
00075 } // namespace rxdock
00076
00077 #endif //_StringTokenIter_H_
```

6.160 Subject.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract Subject class for Subject-Observer design pattern. Subject will
00014 // notify registered observers of any change of state of the Subject, by
00015 // invoking Update method on each observer. It is up to the observer to
00016 // interrogate subject as to what has changed.
00017 // Main use is for Workspace which manages the model list and notifies
00018 // scoring functions and transforms of any changes.
00019 //
00020 // Design considerations:
00021 // Constructor is protected to prevent instantiation of base class.
00022 // No smart pointers are used for storing observers.
00023 // Subject is NOT responsible for deleting observers in destructor, but will
00024 // notify registered observers of impending destruction by invoking Deleted
00025 // method on each observer.
00026
00027 #ifndef _RBTSUBJECT_H_
00028 #define _RBTSUBJECT_H_
00029
00030 #include "rxdock/Config.h"
00031 #include "rxdock/Observer.h"
00032
00033 namespace rxdock {
00034
00035 class Subject {
00036 public:
00038   // Constructors/destructors
00039   virtual ~Subject();
00040
00042   // Public methods
00044   virtual void Attach(Observer *);
00045   virtual void Detach(Observer *);
00046   virtual void Notify();
00047
00048 protected:
00050   // Protected methods
00052   Subject(); // Disable constructor
00053
00054 private:
00056   // Private methods
00058
00059 protected:
00061   // Protected data
00063
00064 private:
00066   // Private data
00068   ObserverList m_observers;
00069 };
00070
00071 } // namespace rxdock
00072
00073 #endif //_RBTSUBJECT_H_
```

6.161 Export.h File Reference

Macros

- #define **RBTDLL_EXPORT**
- #define **RBTDLL_IMPORT**
- #define **RBTDLL_LOCAL**
- #define **LOGURU_EXPORT** RBTDLL_EXPORT

6.161.1 Detailed Description

Symbol export macros.

6.162 Export.h

Go to the documentation of this file.

```
00001 //===-- Export.h - Symbol export macros ------------------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_SUPPORT_EXPORT_H
00019 #define RXDOCK_SUPPORT_EXPORT_H
00020
00021 #if defined _WIN32 || defined __CYGWIN__
00022 #ifdef __GNUC__
00023 #define RBTDLL_EXPORT __attribute__((dllexport))
00024 #define RBTDLL_IMPORT __attribute__((dllimport))
00025 #else
00026 #define RBTDLL_EXPORT __declspec(dllexport)
00027 #define RBTDLL_IMPORT __declspec(dllimport)
00028 #endif
00029 #define RBTDLL_LOCAL
00030 #else
00031 #if __GNUC__ >= 4
00032 #define RBTDLL_EXPORT __attribute__((visibility("default")))
00033 #define RBTDLL_IMPORT __attribute__((visibility("default")))
00034 #define RBTDLL_LOCAL __attribute__((visibility("hidden")))
00035 #else
00036 #define RBTDLL_EXPORT
00037 #define RBTDLL_IMPORT
00038 #define RBTDLL_LOCAL
00039 #endif
00040 #endif
00041
00042 // Same rules apply to Loguru logging library
00043 #define LOGURU_EXPORT RBTDLL_EXPORT
00044
00045 #endif // RXDOCK_SUPPORT_EXPORT_H
```

6.163 Number.h File Reference

```
#include ¨rxdock/support/Export.h¨
#include <string>
```

Functions

- RBTDLL_EXPORT std::string rxdock::support::getRomanNumeral (std::size_t value)

  Converts a numeric value to the corresponding Roman numeral.

6.163.1 Detailed Description

Number conversion.

6.163.2 Function Documentation

6.163.2.1 getRomanNumeral()

```
std::string rxdock::support::getRomanNumeral (
            std::size_t value )
```

Converts a numeric value to the corresponding Roman numeral.

Parameters

value	a numeric value.

Returns

the corresponding Roman numeral

6.164 Number.h

Go to the documentation of this file.

```
00001 //===-- Number.h - Number conversion ---------------------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_SUPPORT_NUMBER_H
00019 #define RXDOCK_SUPPORT_NUMBER_H
00020
00021 #include "rxdock/support/Export.h"
00022
00023 #include <string>
00024
00025 namespace rxdock {
00026 namespace support {
00027
00033 RBTDLL_EXPORT std::string getRomanNumeral(std::size_t value);
00034
00035 } // namespace support
00036 } // namespace rxdock
00037
00038 #endif // RXDOCK_SUPPORT_NUMBER_H
```

6.165 Quote.h File Reference

```
#include ¨rxdock/support/Export.h¨
#include <string>
```

Functions

- RBTDLL_EXPORT std::string rxdock::support::getRandomQuote ()

  Selects a random quote from the database and formats it for printing.

6.165.1 Detailed Description

Quote reading, selection, and formatting.

6.165.2 Function Documentation

6.165.2.1 getRandomQuote()

```
std::string rxdock::support::getRandomQuote ( )
```

Selects a random quote from the database and formats it for printing.

Will return an empty string if the quotes database can not be opened for reading.

Returns

a random quote formatted for printing

6.166 Quote.h

Go to the documentation of this file.

```
00001 //===-- Quote.h - Quote reading, selection, and formatting ------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_SUPPORT_QUOTE_H
00019 #define RXDOCK_SUPPORT_QUOTE_H
00020
00021 #include "rxdock/support/Export.h"
00022
00023 #include <string>
00024
00025 namespace rxdock {
00026 namespace support {
00027
00035 RBTDLL_EXPORT std::string getRandomQuote();
00036
00037 } // namespace support
00038 } // namespace rxdock
00039
00040 #endif // RXDOCK_SUPPORT_QUOTE_H
```

6.167 TetherSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Tethered atoms distance restraint scoring function
00014
00015 #ifndef _RBTTETHERSF_H_
00016 #define _RBTTETHERSF_H_
00017
00018 #include "rxdock/BaseInterSF.h"
00019
00020 #include <nlohmann/json.hpp>
00021
```

```
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 class TetherSF : public BaseInterSF {
00027 public:
00028   // Class type string
00029   static const std::string _CT;
00030   // Parameter names
00031   static const std::string _REFERENCE_FILE;
00032
00033   TetherSF(const std::string &strName = "tether");
00034   virtual ~TetherSF();
00035
00036   friend void to_json(json &j, const TetherSF &tetsf);
00037   friend void from_json(const json &j, TetherSF &tetsf);
00038
00039 protected:
00040   virtual void SetupReceptor();
00041   virtual void SetupLigand();
00042   virtual void SetupScore();
00043   virtual double RawScore() const;
00044   // DM 25 Oct 2000 - track changes to parameter values in local data members
00045   // ParameterUpdated is invoked by ParamHandler::SetParameter
00046   void ParameterUpdated(const std::string &strName);
00047
00048 private:
00049   std::vector<int> ReadTetherAtoms(std::vector<std::string> &);
00050   AtomList m_ligAtomList;
00051   std::vector<int> m_tetherAtomList;
00052   CoordList m_tetherCoords;
00053 };
00054
00055 void to_json(json &j, const TetherSF &tetsf);
00056 void from_json(const json &j, TetherSF &tetsf);
00057
00058 } // namespace rxdock
00059
00060 #endif //_RBTTETHERSF_H_
```

6.168 Token.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Token class. Defines a token of the grammar to be used by the parser.
00014
00015 #ifndef _RBT_TOKEN_H_
00016 #define _RBT_TOKEN_H_
00017
00018 #include "rxdock/Commands.h"
00019 #include "rxdock/Config.h"
00020 #include "rxdock/Vble.h"
00021
00022 #include <nlohmann/json.hpp>
00023
00024 using json = nlohmann::json;
00025
00026 namespace rxdock {
00027
00028 class Token {
00029 public:
00030   static const std::string _CT;
00032   // Constructors
00034
00035   Token(const Vble &);
00036   Token(Commands);
00037   Token(const Token &);
00038
00040   // Destructor
00042   virtual ~Token();
00043
00044   friend void to_json(json &j, const Token &token);
00045   friend void from_json(const json &j, Token &token);
```

```
00046
00047   const Vble &GetVble() const;
00048   //  void SetVbleNumber(Int);
00049
00050   //  void copy(const Token &);
00051   bool IsVble();
00052   bool IsLog();
00053   bool IsExp();
00054   bool IsAdd();
00055   bool IsSub();
00056   bool IsMul();
00057   bool IsDiv();
00058   bool IsAnd();
00059   bool IsIf();
00060
00061   //  Bool operator==(const Token&) const;
00062   //  std::ostream& Print(std::ostream&) const;
00063   //  friend std::ostream& operator«(std::ostream& s, const Token &p);
00064
00066   // Private methods
00068
00069 private:
00070   Token(); // Default constructor disabled
00071   Commands comm;
00072   const Vble &vble;
00073   bool isvble;
00074 };
00075
00076 void to_json(json &j, const Token &token);
00077 void from_json(const json &j, Token &token);
00078
00079 // Useful typedefs
00080 typedef SmartPtr<Token> TokenPtr;        // Smart pointer
00081 typedef std::vector<TokenPtr> TokenList; // Vector of smart pointers
00082 typedef TokenList::iterator TokenListIter;
00083 typedef TokenList::const_iterator TokenListConstIter;
00084
00085 // Compare class to compare different genomes inside a population
00086 // This is used to sort the genomes in a population depending
00087 // of the value of their scoring function.
00088
00089 } // namespace rxdock
00090
00091 #endif //_Token_H_
```

6.169 TokenIter.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Abstract TokenIter class. Defines the interface for a token iterator
00014 // to be used by the parser.
00015
00016 #ifndef _RBT_TOKENITER_H_
00017 #define _RBT_TOKENITER_H_
00018
00019 #include "rxdock/Context.h"
00020 #include "rxdock/Token.h"
00021
00022 namespace rxdock {
00023
00024 class TokenIter {
00025 public:
00026   //  virtual void copy(const TokenIter &) = 0;
00027   virtual void Next(ContextPtr) = 0;
00028   virtual TokenPtr Current() = 0;
00029   virtual ~TokenIter() {}
00030
00031   //  virtual TokenPtr IsDone() const = 0;
00032   //  virtual void First() = 0;
00033   //  Bool operator==(const TokenIter&) const;
00034   //  std::ostream& Print(std::ostream&) const;
00035   //  friend std::ostream& operator«(std::ostream& s, const TokenIter &p);
00036 };
```

```
00037
00038 // Useful typedefs
00039 typedef SmartPtr<TokenIter> TokenIterPtr;        // Smart pointer
00040 typedef std::vector<TokenIterPtr> TokenIterList; // Vector of smart pointers
00041 typedef TokenIterList::iterator TokenIterListIter;
00042 typedef TokenIterList::const_iterator TokenIterListConstIter;
00043
00044 } // namespace rxdock
00045
00046 #endif //_TokenIter_H_
```

6.170 TransformAgg.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Transform aggregate class. Manages collection of child transforms
00014
00015 #ifndef _RBTTRANSFORMAGG_H_
00016 #define _RBTTRANSFORMAGG_H_
00017
00018 #include "rxdock/BaseTransform.h"
00019
00020 namespace rxdock {
00021
00022 // Only check transform aggregate assertions in debug build
00023 #ifdef _NDEBUG
00024 const Bool TRANSFORMAGG_CHECK = false;
00025 #else
00026 const bool TRANSFORMAGG_CHECK = true;
00027 #endif //_NDEBUG
00028
00029 // Useful typedefs
00030
00031 class TransformAgg : public BaseTransform {
00032 public:
00033   // Static data member for class type (i.e. "TransformAgg")
00034   static const std::string _CT;
00035
00037   // Constructors/destructors
00038   RBTDLL_EXPORT TransformAgg(const std::string &strName = "dock");
00039   virtual ~TransformAgg();
00040
00042   // Public methods
00044
00045   // Aggregate handling methods
00046   virtual void Add(BaseTransform *);
00047   virtual void Remove(BaseTransform *);
00048   virtual bool isAgg() const;
00049   virtual unsigned int GetNumTransforms() const;
00050   virtual BaseTransform *GetTransform(unsigned int iTransform) const;
00051
00052   // WorkSpace handling methods
00053   // Register scoring function with a workspace
00054   // Aggregate version registers all children, but NOT itself
00055   //(Aggregates are just containers, and have no need for model information
00056   virtual void Register(WorkSpace *);
00057   // Unregister with a workspace
00058   // Aggregate version unregisters all children, but NOT itself
00059   virtual void Unregister();
00060
00061   // Override Observer pure virtual
00062   // Notify observer that subject has changed
00063   // Does nothing in TransformAgg as aggregates do not require updating
00064   virtual void Update(Subject *theChangedSubject);
00065
00066   // Request Handling method
00067   virtual void HandleRequest(RequestPtr spRequest);
00068
00069   // Virtual function for dumping transform details to an output stream
00070   // Called by operator «
00071   virtual void Print(std::ostream &s) const;
00072
00073 protected:
```

```
00075   // Protected methods
00077   // Actually apply the transform
00078   // Aggregate version loops over all child transforms
00079   virtual void Execute();
00080
00081 private:
00083   // Private methods
00085   TransformAgg(const TransformAgg &); // Copy constructor disabled by default
00086   TransformAgg &
00087   operator=(const TransformAgg &); // Copy assignment disabled by default
00088
00089 protected:
00091   // Protected data
00093
00094 private:
00096   // Private data
00098   BaseTransformList m_transforms;
00099 };
00100
00101 // Useful typedefs
00102 typedef SmartPtr<TransformAgg> TransformAggPtr; // Smart pointer
00103 typedef std::vector<TransformAggPtr>
00104     TransformAggList; // Vector of smart pointers
00105 typedef TransformAggList::iterator TransformAggListIter;
00106 typedef TransformAggList::const_iterator TransformAggListConstIter;
00107
00108 } // namespace rxdock
00109
00110 #endif //_RBTTRANSFORMAGG_H_
```

6.171 TransformFactory.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Factory class for creating transform objects. Transforms are specified by
00014 // string names (as defined in each class's static _CT string)
00015 // Directly analogous to SFFactory for creating scoring function objects
00016
00017 #ifndef _RBTTRANSFORMFACTORY_H_
00018 #define _RBTTRANSFORMFACTORY_H_
00019
00020 #include "rxdock/Config.h"
00021 #include "rxdock/ParameterFileSource.h"
00022 #include "rxdock/TransformAgg.h"
00023
00024 namespace rxdock {
00025
00026 class TransformFactory {
00027   // Parameter name which identifies a scoring function definition
00028   static const std::string _TRANSFORM;
00029
00030 public:
00032   // Constructors/destructors
00033
00034   RBTDLL_EXPORT TransformFactory(); // Default constructor
00035
00036   virtual ~TransformFactory(); // Default destructor
00037
00039   // Public methods
00041
00042   // Creates a single transform object of type strTransformClass, and name
00043   // strName e.g. strTransformClass = SimAnnTransform
00044   virtual BaseTransform *Create(const std::string &strTransformClass,
00045                                 const std::string &strName);
00046
00047   // Creates an aggregate transform from a parameter file source
00048   // Each component transform is in a named section, which should minimally
00049   // contain a TRANSFORM parameter whose value is the class name to instantiate
00050   // strTransformClasses contains a comma-delimited list of transform class
00051   // names to instantiate If strTransformClasses is empty, all named sections in
00052   // spPrmSource are scanned for valid transform definitions Transform
00053   // parameters and scoring function requests are set from the list of
00054   // parameters in each named section
```

```
00055   virtual TransformAgg *
00056   CreateAggFromFile(ParameterFileSourcePtr spPrmSource,
00057                     const std::string &strName,
00058                     const std::string &strTransformClasses = std::string());
00059
00060 protected:
00062   // Protected methods
00064
00065 private:
00067   // Private methods
00069
00070   TransformFactory(
00071       const TransformFactory &); // Copy constructor disabled by default
00072
00073   TransformFactory &
00074   operator=(const TransformFactory &); // Copy assignment disabled by default
00075
00076 protected:
00078   // Protected data
00080
00081 private:
00083   // Private data
00085 };
00086
00087 // Useful typedefs
00088 typedef SmartPtr<TransformFactory> TransformFactoryPtr; // Smart pointer
00089
00090 } // namespace rxdock
00091
00092 #endif //_RBTTRANSFORMFACTORY_H_
```

6.172 TriposAtomType.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Atom typing class for Tripos 5.2 forcefield
00014
00015 #ifndef _RBTTRIPOSATOMTYPE_H_
00016 #define _RBTTRIPOSATOMTYPE_H_
00017
00018 #include "rxdock/Config.h"
00019
00020 #include <nlohmann/json.hpp>
00021
00022 using json = nlohmann::json;
00023
00024 namespace rxdock {
00025
00026 class Atom;
00027
00028 class TriposAtomType {
00029 public:
00031   // Enums
00033
00034   // Atom types
00035   enum eType {
00036     UNDEFINED = 0,
00037     Al,
00038     Br,
00039     C_cat,
00040     C_1,
00041     C_1_H1,
00042     C_2,
00043     C_2_H1,
00044     C_2_H2,
00045     C_3,
00046     C_3_H1,
00047     C_3_H2,
00048     C_3_H3,
00049     C_ar,
00050     C_ar_H1,
00051     Ca,
00052     Cl,
```

```
00053     Du,
00054     F,
00055     H,
00056     H_P,
00057     I,
00058     K,
00059     Li,
00060     LP,
00061     N_1,
00062     N_2,
00063     N_3,
00064     N_4,
00065     N_am,
00066     N_ar,
00067     N_pl3,
00068     Na,
00069     O_2,
00070     O_3,
00071     O_co2,
00072     P_3,
00073     S_2,
00074     S_3,
00075     S_o,
00076     S_o2,
00077     Si,
00078     MAXTYPES // KEEP AS LAST TYPE: used to size the atom name string list
00079   };
00080
00081   struct info {
00082     std::string name;
00083     int atomicNo;
00084     int hybrid;
00085     info() : atomicNo(0), hybrid(0) {}
00086     info(const std::string &n, int a, int h)
00087         : name(n), atomicNo(a), hybrid(h) {}
00088   };
00089
00091   // Constructors/destructors
00092
00093   RBTDLL_EXPORT TriposAtomType();
00094   RBTDLL_EXPORT virtual ~TriposAtomType();
00095
00096   friend void to_json(json &j, const TriposAtomType &triposAtomType);
00097   friend void from_json(const json &j, TriposAtomType &triposAtomType);
00098
00100   // Public methods
00102
00103   // Returns Tripos atom type for given atom
00104   // If useExtendedTypes is true, pseudo types for extended carbons and polar Hs
00105   // are used
00106   eType operator()(Atom *pAtom, bool useExtendedTypes = false) const;
00107   // Converts Tripos type to string
00108   RBTDLL_EXPORT std::string Type2Str(eType) const;
00109   // Get hybridisation from Tripos type
00110   int Type2Hybrid(eType) const;
00111   // Get atomic number from Tripos type
00112   int Type2AtomicNo(eType) const;
00113   // Converts string to Tripos type
00114   RBTDLL_EXPORT eType Str2Type(const std::string &) const;
00115
00116 protected:
00118   // Protected methods
00120
00121 private:
00123   // Private methods
00125   void SetupTypeInfo();
00126
00127 protected:
00129   // Protected data
00131
00132 private:
00134   // Private data
00136   std::vector<info> m_typeInfo;
00137 };
00138
00139 void to_json(json &j, const TriposAtomType &triposAtomType);
00140 void from_json(const json &j, TriposAtomType &triposAtomType);
00141
00142 // Useful typedefs
00143 typedef std::vector<TriposAtomType::info> TriposInfoList;
00144 typedef TriposInfoList::iterator TriposInfoListIter;
00145 typedef TriposInfoList::const_iterator TriposInfoListConstIter;
00146
00147 typedef std::vector<TriposAtomType::eType> TriposAtomTypeList;
00148 typedef TriposAtomTypeList::iterator TriposAtomTypeListIter;
00149 typedef TriposAtomTypeList::const_iterator TriposAtomTypeListConstIter;
00150
```

```
00151 } // namespace rxdock
00152
00153 #endif //_RBTTRIPOSATOMTYPE_H_
```

6.173 Variant.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Variant class along the lines of the Visual Basic variant data type.
00014 // Handles int, double, const char*, string and std::vector<string> types
00015 // Underlying value is stored as double and std::vector<string>
00016
00017 #ifndef _RBTVARIANT_H_
00018 #define _RBTVARIANT_H_
00019
00020 #include <sstream>
00021
00022 #include "rxdock/Coord.h"
00023
00024 namespace rxdock {
00025
00026 // Strings representing TRUE and FALSE
00027 const std::string _TRUE("TRUE");
00028 const std::string _FALSE("FALSE");
00029
00030 class Variant {
00031 public:
00033   // Constructors/destructors
00034   Variant() : m_d(0.0) {}
00035   Variant(int i) { SetDouble(i); }
00036   Variant(double d) { SetDouble(d); }
00037   Variant(const std::string &s) { SetString(s); }
00038   Variant(const char *c) { SetString(std::string(c)); }
00039   Variant(const std::vector<std::string> &sl) { SetStringList(sl); }
00040   Variant(bool b) { SetBool(b); }
00041   Variant(const Coord &c) { SetCoord(c); }
00042   // Renders a vector of doubles into comma-separated strings of maxCols in
00043   // length
00044   Variant(const std::vector<double> &dl, int maxCols, int precision) {
00045     SetDoubleList(dl, maxCols, precision);
00046   }
00047   virtual ~Variant() { m_sl.clear(); }
00048
00050   // Stream functions
00052
00053   // Insertion operator
00054   friend std::ostream &operator<<(std::ostream &s, const Variant &v) {
00055     // Try and guess the most appropriate output format
00056     // Empty variant
00057     if (v.isEmpty())
00058       s << "Undefined variant";
00059     // String/double
00060     else if (v.Size() == 1)
00061       s << v.m_sl.front();
00062     // String list
00063     else {
00064       s << std::endl;
00065       for (std::vector<std::string>::const_iterator iter = v.m_sl.begin();
00066            iter != v.m_sl.end(); iter++)
00067         s << *iter << std::endl;
00068     }
00069     return s;
00070   }
00071
00073   // Operator functions:
00075
00076   // Set methods
00077   Variant &operator=(int i) {
00078     SetDouble(i);
00079     return *this;
00080   }
00081   Variant &operator=(double d) {
00082     SetDouble(d);
```

```
00083     return *this;
00084   }
00085   Variant &operator=(const std::string &s) {
00086     SetString(s);
00087     return *this;
00088   }
00089   Variant &operator=(const char *c) {
00090     SetString(std::string(c));
00091     return *this;
00092   }
00093   Variant &operator=(const std::vector<std::string> &sl) {
00094     SetStringList(sl);
00095     return *this;
00096   }
00097   Variant &operator=(bool b) {
00098     SetBool(b);
00099     return *this;
00100   }
00101   Variant &operator=(const Coord &c) {
00102     SetCoord(c);
00103     return *this;
00104   }
00105
00106   void operator+=(const Variant &v) {
00107     std::copy(v.m_sl.begin(), v.m_sl.end(), std::back_inserter(m_sl));
00108     m_d = std::atof(GetString().c_str());
00109   }
00110
00111   // Conversion operators to convert back to basic types
00112   operator int() const { return int(GetDouble()); }
00113   operator unsigned int() const { return (unsigned int)(GetDouble()); }
00114   operator double() const { return GetDouble(); }
00115   operator std::string() const { return GetString(); }
00116   operator std::vector<std::string>() const { return GetStringList(); }
00117   operator bool() const { return GetBool(); }
00118   operator Coord() const { return GetCoord(); }
00119
00120   friend void to_json(json &j, const Variant &v) {
00121     j = json{{"double", v.m_d}, {"string-list", v.m_sl}};
00122   }
00123
00124   friend void from_json(const json &j, Variant &v) {
00125     j.at("double").get_to(v.m_d);
00126     j.at("string-list").get_to(v.m_sl);
00127   }
00128
00130   // Public methods
00132
00133   // Get methods
00134   double GetDouble() const { return m_d; }
00135   std::string GetString() const {
00136     return m_sl.empty() ? std::string() : m_sl.front();
00137   }
00138   std::vector<std::string> GetStringList() const { return m_sl; }
00139   bool GetBool() const {
00140     return m_d != 0.0 || GetString() == _TRUE || GetString() == "true";
00141   }
00142   Coord GetCoord() const {
00143     Coord c;
00144     if (m_sl.empty())
00145       return c;
00146     else {
00147       std::istringstream(m_sl.front()) » c;
00148       return c;
00149     }
00150   }
00151
00152   unsigned int Size() const { return m_sl.size(); }
00153   bool isEmpty() const { return m_sl.empty(); }
00154
00155 protected:
00157   // Protected methods
00159
00160 private:
00162   // Private methods
00164   void SetDouble(double d) {
00165     m_d = d;
00166     m_sl.clear();
00167     std::ostringstream ostr;
00168     // Don't need "ends" with std::ostringstream apparently
00169     //(was introducing a non-ASCII \0 char into log files
00170     // ostr « d « ends;
00171     ostr « d;
00172     std::string s(ostr.str());
00173     // delete ostr.str();
00174     m_sl.push_back(s);
00175   }
```

```
00176
00177   void SetString(const std::string &s) {
00178     m_sl.clear();
00179     if (!s.empty())
00180       m_sl.push_back(s);
00181     m_d = std::atof(GetString().c_str());
00182   }
00183
00184   void SetStringList(const std::vector<std::string> &sl) {
00185     m_sl = sl;
00186     m_d = std::atof(GetString().c_str());
00187   }
00188
00189   void SetBool(bool b) {
00190     if (b) {
00191       SetString(_TRUE);
00192     } else {
00193       SetString(_FALSE);
00194     }
00195   }
00196
00197   void SetCoord(const Coord &c) {
00198     m_d = c.Length();
00199     m_sl.clear();
00200     std::ostringstream ostr;
00201     // Don't need "ends" with std::ostringstream apparently
00202     //(was introducing a non-ASCII \0 char into log files
00203     // ostr << c << ends;
00204     ostr << c;
00205     std::string s(ostr.str());
00206     // delete ostr.str();
00207     m_sl.push_back(s);
00208   }
00209
00210   void SetDoubleList(const std::vector<double> &dl, int maxColumns,
00211                      int precision) {
00212     int nValues = dl.size();
00213     m_d = (nValues > 0) ? dl[0] : 0.0;
00214     m_sl.clear();
00215     std::ostringstream ostr;
00216     ostr.precision(precision);
00217     ostr.setf(std::ios_base::fixed, std::ios_base::floatfield);
00218     int lastIndex = nValues - 1;
00219     for (int i = 0; i < nValues; ++i) {
00220       ostr << dl[i];
00221       if ((ostr.tellp() >= maxColumns) || (i == lastIndex)) {
00222         m_sl.push_back(ostr.str());
00223         ostr.str("");
00224       } else {
00225         ostr << ",";
00226       }
00227     }
00228   }
00229
00230 protected:
00232   // Protected data
00234
00235 private:
00237   // Private data
00239   double m_d;
00240   std::vector<std::string> m_sl;
00241 };
00242
00243 // Useful typedefs
00244 // typedef SharedPtr<Variant> VariantPtr;//Smart pointer
00245 typedef std::vector<Variant> VariantList; // Vector of variants
00246 typedef VariantList::iterator VariantListIter;
00247 typedef VariantList::const_iterator VariantListConstIter;
00248
00249 // Map of (key=string, value=variant)
00250 typedef std::map<std::string, Variant> StringVariantMap;
00251 typedef StringVariantMap::iterator StringVariantMapIter;
00252 typedef StringVariantMap::const_iterator StringVariantMapConstIter;
00253
00254 } // namespace rxdock
00255
00256 #endif //_RBTVARIANT_H_
```

6.174 Vble.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
```

```
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Vble class. Stores information about the variables used by the filters.
00014
00015 #ifndef _RBT_VBLE_H_
00016 #define _RBT_VBLE_H_
00017
00018 #include "rxdock/geneticprogram/GPTypes.h"
00019
00020 namespace rxdock {
00021
00022 class Vble {
00023 public:
00024   static const std::string _CT;
00025   enum VbleType { CTE, LIG, SCORE, SITE };
00026   Vble() : vt(CTE), value(0.0), name("") {
00027     _RBTOBJECTCOUNTER_CONSTR_(_CT);
00028   } // default const.
00029   Vble(std::string s, ReturnType val = 0.0) : value(val), name(s) {
00030     if (s.find("LIG_") == 0)
00031       vt = LIG;
00032     else if (s.find("SITE_") == 0)
00033       vt = SITE;
00034     else if (s.find(GetMetaDataPrefix() + "score") == 0)
00035       vt = SCORE;
00036     else
00037       vt = CTE;
00038     _RBTOBJECTCOUNTER_CONSTR_(_CT);
00039   }
00040   Vble(const Vble &v) : vt(v.vt), value(v.value), name(v.name) {
00041     _RBTOBJECTCOUNTER_COPYCONSTR_(_CT);
00042   }
00043   void SetValue(ReturnType val) {
00044     value = val;
00045     if (name == "") {
00046       name = std::to_string(value);
00047     }
00048   }
00049   ReturnType GetValue() const { return value; }
00050   virtual ~Vble() { _RBTOBJECTCOUNTER_DESTR_(_CT); }
00051
00052   Vble &operator=(const Vble &v) {
00053     vt = v.vt;
00054     value = v.value;
00055     name = v.name;
00056     return *this;
00057   }
00058
00059   void SetName(std::string nm) { name = nm; }
00060   std::string GetName() const { return name; }
00061
00062   bool IsLig() { return (vt == LIG); }
00063   bool IsScore() { return (vt == SCORE); }
00064   bool IsSite() { return (vt == SITE); }
00065
00066 private:
00067   VbleType vt;
00068   ReturnType value;
00069   std::string name;
00070 };
00071
00072 // Useful typedefs
00073 typedef SmartPtr<Vble> VblePtr;        // Smart pointer
00074 typedef std::vector<VblePtr> VbleList; // Vector of smart pointers
00075 typedef VbleList::iterator VbleListIter;
00076 typedef VbleList::const_iterator VbleListConstIter;
00077
00078 } // namespace rxdock
00079
00080 #endif //_Vble_H_
```

6.175 VdwGridSF.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
```

```
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Precalculated-grid-based intermolecular vdw scoring function
00014
00015 #ifndef _RBTVDWGRIDSF_H_
00016 #define _RBTVDWGRIDSF_H_
00017
00018 #include "rxdock/BaseInterSF.h"
00019 #include "rxdock/RealGrid.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class VdwGridSF : public BaseInterSF {
00028 public:
00029   // Class type string
00030   static const std::string _CT;
00031
00032   RBTDLL_EXPORT static const std::string &GetCt();
00033
00034   // Parameter names
00035   static const std::string _GRID; // Suffix for grid filename
00036   static const std::string
00037       _SMOOTHED; // Controls whether to smooth the grid values
00038
00039   VdwGridSF(const std::string &strName = "vdw");
00040   virtual ~VdwGridSF();
00041
00042   friend void to_json(json &j, const VdwGridSF &vdwGridSF);
00043   friend void from_json(const json &j, VdwGridSF &vdwGridSF);
00044
00045 protected:
00046   virtual void SetupReceptor();
00047   virtual void SetupLigand();
00048   virtual void SetupSolvent();
00049   virtual void SetupScore();
00050   virtual double RawScore() const;
00051   // DM 25 Oct 2000 - track changes to parameter values in local data members
00052   // ParameterUpdated is invoked by ParamHandler::SetParameter
00053   void ParameterUpdated(const std::string &strName);
00054
00055 private:
00056   // Read grids from input stream
00057   void ReadGrids(json vdwGrids);
00058
00059   RealGridList m_grids;
00060   AtomRList m_ligAtomList;
00061   TriposAtomTypeList m_ligAtomTypes;
00062   bool m_bSmoothed;
00063 };
00064
00065 void to_json(json &j, const VdwGridSF &vdwGridSF);
00066 void from_json(const json &j, VdwGridSF &vdwGridSF);
00067
00068 } // namespace rxdock
00069
00070 #endif //_RBTVDWGRIDSF_H_
```

6.176 VdwIdxSF.h

```
00001 /***********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Indexed-grid-based intermolecular scoring function for vdw interactions
```

```
00014 #ifndef _RBTVDWIDXSF_H_
00015 #define _RBTVDWIDXSF_H_
00016
00017 #include "rxdock/BaseIdxSF.h"
00018 #include "rxdock/BaseInterSF.h"
00019 #include "rxdock/VdwSF.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class VdwIdxSF : public BaseInterSF, public BaseIdxSF, public VdwSF {
00028 public:
00029   // Class type string
00030   static const std::string _CT;
00031   // Parameter names
00032   // DM 12 Jun 2002 - score thresholds used for counting attractive and
00033   // repulsive interactions (thresholds represent partial vdW scores summed per
00034   // ligand atom, used for outputting ligand atom counts)
00035   static const std::string _THRESHOLD_ATTR;
00036   static const std::string _THRESHOLD_REP;
00037   // DM 03 Feb 2003 - score threshold for outputting individual (atom-atom)
00038   // lipophilic vdW interactions i.e between two non-polar carbons/hydrogens
00039   static const std::string _ANNOTATION_LIPO;
00040   // DM 10 Apr 2003 - vdw annotation can be quite slow so provide boolean option
00041   // to disable
00042   static const std::string _ANNOTATE;
00043   // DM 14 Jun 2006 - option to disable solvent performance enhancements, mainly
00044   // for testing
00045   static const std::string _FAST_SOLVENT;
00046
00047   RBTDLL_EXPORT VdwIdxSF(const std::string &strName = "vdw");
00048   virtual ~VdwIdxSF();
00049
00050   friend void to_json(json &j, const VdwIdxSF &vdwIdxSF);
00051   friend void from_json(const json &j, VdwIdxSF &vdwIdxSF);
00052
00053   // Override BaseSF::ScoreMap to provide additional raw descriptors
00054   virtual void ScoreMap(StringVariantMap &scoreMap) const;
00055
00056 protected:
00057   virtual void SetupReceptor();
00058   virtual void SetupLigand();
00059   virtual void SetupSolvent();
00060   virtual void SetupScore();
00061   virtual double RawScore() const;
00062   double InterScore() const;
00063   double ReceptorScore() const;
00064   double SolventScore() const;
00065   double ReceptorSolventScore() const;
00066   double LigandSolventScore() const;
00067
00068   // DM 25 Oct 2000 - track changes to parameter values in local data members
00069   // ParameterUpdated is invoked by ParamHandler::SetParameter
00070   void ParameterUpdated(const std::string &strName);
00071
00072 private:
00073   void RenderAnnotationsByResidue(std::vector<std::string> &retVal) const;
00074
00075   NonBondedGridPtr m_spGrid;        // Indexing grid for receptor
00076   NonBondedGridPtr m_spSolventGrid; // Indexing grid for fixed/tethered solvent
00077   AtomList m_recAtomList;
00078   AtomRList m_recRigidAtomList;
00079   AtomRList m_recFlexAtomList;
00080   AtomRList m_ligAtomList;
00081   AtomRList m_solventAtomList;        // All solvent atoms
00082   AtomRList m_solventFixTethAtomList; // Fixed or tethered solvent atoms
00083   AtomRList m_solventFreeAtomList;    // Free solvent atoms
00084   AtomRListList m_recFlexIntns;       // Intra-target flexible interactions
00085   AtomRListList
00086       m_recFlexPrtIntns; // Intra-target flexible partitioned interactions
00087   AtomRListList
00088       m_solventFixTethIntns; // Intra-solvent intns between fixed/tethered atoms
00089   AtomRListList m_solventFixTethPrtIntns; // Partitioned intns between
00090                                           // fixed/tethered solvent
00091   AtomRListList
00092       m_solventFreeIntns; // Intra-solvent intns between free solvent atoms
00093   // DM 12 Jun 2002 - keep track of number of ligand atoms involved in non-zero
00094   // vdW interactions
00095   mutable int m_nAttr; //#atoms with net attractive (-ve) vdw scores
00096   mutable int m_nRep;  //#atoms with net repulsive (+ve) vdw scores
00097   double m_attrThreshold;
00098   double m_repThreshold;
00099   double m_lipoAnnot;
00100   bool m_bAnnotate;
```

```
00101   bool m_bFlexRec;
00102   bool m_bFastSolvent;
00103 };
00104
00105 void to_json(json &j, const VdwIdxSF &vdwIdxSF);
00106 void from_json(const json &j, VdwIdxSF &vdwIdxSF);
00107
00108 } // namespace rxdock
00109
00110 #endif //_RBTVDWIDXSF_H_
```

6.177 VdwIntraSF.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Ligand intramolecular scoring function for vdw potential
00014
00015 #ifndef _RBTVDWINTRASF_H_
00016 #define _RBTVDWINTRASF_H_
00017
00018 #include "rxdock/BaseIntraSF.h"
00019 #include "rxdock/VdwSF.h"
00020
00021 #include <nlohmann/json.hpp>
00022
00023 using json = nlohmann::json;
00024
00025 namespace rxdock {
00026
00027 class VdwIntraSF : public BaseIntraSF, public VdwSF {
00028 public:
00029   // Class type string
00030   static const std::string _CT;
00031
00032   RBTDLL_EXPORT VdwIntraSF(const std::string &strName = "vdw");
00033   virtual ~VdwIntraSF();
00034
00035   friend void to_json(json &j, const VdwIntraSF &vdwIntraSF);
00036   friend void from_json(const json &j, VdwIntraSF &vdwIntraSF);
00037
00038   // Request Handling method
00039   // Handles the Partition request
00040   virtual void HandleRequest(RequestPtr spRequest);
00041
00042 protected:
00043   virtual void SetupScore();
00044   virtual double RawScore() const;
00045
00046   // DM 25 Oct 2000 - track changes to parameter values in local data members
00047   // ParameterUpdated is invoked by ParamHandler::SetParameter
00048   void ParameterUpdated(const std::string &strName);
00049
00050 private:
00051   AtomRListList m_vdwIntns; // The full list of vdW interactions
00052   AtomRListList
00053       m_prtIntns; // The partitioned interactions (within partition distance)
00054   AtomRList m_ligAtomList;
00055 };
00056
00057 void to_json(json &j, const VdwIntraSF &vdwIntraSF);
00058 void from_json(const json &j, VdwIntraSF &vdwIntraSF);
00059
00060 } // namespace rxdock
00061
00062 #endif //_RBTVDWINTRASF_H_
```

6.178 VdwSF.h

```
00001 /**********************************************************************
```

```
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  ***********************************************************************/
00012
00013 // Base implementation class for all vdW scoring functions
00014
00015 #ifndef _RBTVDWSF_H_
00016 #define _RBTVDWSF_H_
00017
00018 #include "rxdock/AnnotationHandler.h"
00019 #include "rxdock/Atom.h"
00020 #include "rxdock/BaseSF.h"
00021 #include "rxdock/ParameterFileSource.h"
00022 #include "rxdock/TriposAtomType.h"
00023
00024 #include <nlohmann/json.hpp>
00025
00026 using json = nlohmann::json;
00027
00028 namespace rxdock {
00029
00030 class VdwSF : public virtual BaseSF, public virtual AnnotationHandler {
00031 public:
00032   // Class type string
00033   static const std::string _CT;
00034   // Parameter names
00035   static const std::string _USE_4_8; // TRUE = 4-8; FALSE = 6-12
00036   static const std::string
00037       _USE_TRIPOS; // TRUE = Tripos 5.2 well depths; FALSE = GOLD well depths
00038   static const std::string _RMAX; // Maximum range as a multiple of rmin
00039   static const std::string _ECUT; // Energy cutoff for transition to short-range
00040                                   // quadratic, as a multiple of well depth
00041   static const std::string
00042       _E0; // Energy at zero distance, as a multiple of ECUT
00043
00044   RBTDLL_EXPORT static const std::string &GetEcut();
00045
00046   virtual ~VdwSF();
00047
00048   friend void to_json(json &j, const VdwSF &vswSF);
00049   friend void from_json(const json &j, VdwSF &vswSF);
00050
00051 protected:
00052   VdwSF();
00053
00054   // As this has a virtual base class we need a separate OwnParameterUpdated
00055   // which can be called by concrete subclass ParameterUpdated methods
00056   // See Stroustrup C++ 3rd edition, p395, on programming virtual base classes
00057   void OwnParameterUpdated(const std::string &strName);
00058
00059   // Used by subclasses to calculate vdW potential between pAtom and all atoms
00060   // in atomList
00061   double VdwScore(const Atom *pAtom, const AtomRList &atomList) const;
00062   // As above, but with additional checks for enabled state of each atom
00063   double VdwScoreEnabledOnly(const Atom *pAtom,
00064                              const AtomRList &atomList) const;
00065   // XB Same as above, used to calcutate intra terms without the reweighting
00066   // factors Double VdwScoreIntra(const Atom* pAtom, const AtomRList&
00067   // atomList) const; Looks up the maximum range (rmax_sq) for any interaction
00068   // i.e. from across a row of m_vdwTable
00069   double MaxVdwRange(const Atom *pAtom) const;
00070   double MaxVdwRange(TriposAtomType::eType t) const;
00071
00072   // Index the intramolecular interactions
00073   void BuildIntraMap(const AtomRList &atomList, AtomRListList &intns) const;
00074   void BuildIntraMap(const AtomRList &atomList1, const AtomRList &atomList2,
00075                      AtomRListList &intns) const;
00076   void Partition(const AtomRList &atomList, const AtomRListList &intns,
00077                  AtomRListList &prtIntns, double dist = 0.0) const;
00078
00079 private:
00080   // vdW scoring function params
00081   struct vdwprms {
00082     double A, B;       // 6-12 or 4-8 params
00083     double rmin;       // Sum of vdw radii
00084     double kij;        // Well depth
00085     double rmax_sq;    // Max distance**2 over which the vdw score should be
00086                        // calculated
00087     double rcutoff_sq; // Distance**2 at which the short-range quadratic
00088                        // potential kicks in
```

```
00089     double ecutoff;    // Energy at the cutoff point
00090     double slope;      // Slope of the short-range quadratic potential
00091     double e0;         // Energy at zero distance
00092   };
00093
00094   typedef std::vector<VdwSF::vdwprms> VdwRow;
00095   typedef VdwRow::iterator VdwRowIter;
00096   typedef VdwRow::const_iterator VdwRowConstIter;
00097   typedef std::vector<VdwRow> VdwTable;
00098   typedef VdwTable::iterator VdwTableIter;
00099   typedef VdwTable::const_iterator VdwTableConstIter;
00100
00101   // Generic scoring function primitive for 6-12
00102   inline double f6_12(double R_sq, const vdwprms &prms) const {
00103     // Zero well depth or long range: return zero
00104     if ((prms.kij == 0.0) || (R_sq > prms.rmax_sq)) {
00105       return 0.0;
00106     }
00107     // Short range: use quadratic potential
00108     else if (R_sq < prms.rcutoff_sq) {
00109       return prms.e0 - (prms.slope * R_sq);
00110     }
00111     // Everywhere else is 6-12
00112     else {
00113       double rr6 = 1.0 / (R_sq * R_sq * R_sq);
00114       return rr6 * (rr6 * prms.A - prms.B);
00115     }
00116   }
00117
00118   // Generic scoring function primitive for 4-8
00119   inline double f4_8(double R_sq, const vdwprms &prms) const {
00120     // Zero well depth or long range: return zero
00121     if ((prms.kij == 0.0) || (R_sq > prms.rmax_sq)) {
00122       return 0.0;
00123     }
00124     // Short range: use quadratic potential
00125     else if (R_sq < prms.rcutoff_sq) {
00126       return prms.e0 - (prms.slope * R_sq);
00127     }
00128     // Everywhere else is 4-8
00129     else {
00130       double rr4 = 1.0 / (R_sq * R_sq);
00131       return rr4 * (rr4 * prms.A - prms.B);
00132     }
00133   }
00134
00135   void Setup(); // Initialise m_vdwTable with appropriate params for each atom
00136                 // type pair
00137   void SetupCloseRange(); // Regenerate the short-range params only (called more
00138                           // frequently)
00139
00140   // Private predicate
00141   // Is the distance between atoms less than a given value ?
00142   // Function checks d**2 to save performing a sqrt
00143   class isD_lt : public std::function<bool(Atom *)> {
00144     double d_sq;
00145     Atom *a;
00146
00147   public:
00148     explicit isD_lt(Atom *aa, double dd) : d_sq(dd * dd), a(aa) {}
00149     bool operator()(Atom *aa) const {
00150       return Length2(aa->GetCoords(), a->GetCoords()) < d_sq;
00151     }
00152   };
00153
00154   // Private data members
00155   bool m_use_4_8;
00156   bool m_use_tripos;
00157   double m_rmax;
00158   double m_ecut;
00159   double m_e0;
00160   ParameterFileSourcePtr m_spVdwSource; // File source for vdw params
00161   VdwTable m_vdwTable; // Lookup table for all vdW params (indexed by Tripos
00162                        // atom type)
00163   std::vector<double>
00164       m_maxRange; // Vector of max ranges for each Tripos atom type
00165 };
00166
00167 void to_json(json &j, const VdwSF &vswSF);
00168 void from_json(const json &j, VdwSF &vswSF);
00169
00170 } // namespace rxdock
00171
00172 #endif //_RBTVDWSF_H_
```

6.179 WorkSpace.h

```
00001 /**********************************************************************
00002  * The rDock program was developed from 1998 - 2006 by the software team
00003  * at RiboTargets (subsequently Vernalis (R&D) Ltd).
00004  * In 2006, the software was licensed to the University of York for
00005  * maintenance and distribution.
00006  * In 2012, Vernalis and the University of York agreed to release the
00007  * program as Open Source software.
00008  * This version is licensed under GNU-LGPL version 3.0 with support from
00009  * the University of Barcelona.
00010  * http://rdock.sourceforge.net/
00011  **********************************************************************/
00012
00013 // Manages all models used in simulation, coordinates updates of all registered
00014 // Observer objects (scoring functions and transforms)
00015 // In this version, the number of models in the workspace is fixed in the
00016 // constructor, and can't be changed. Models can still be null however.
00017 //
00018 // Design Pattern: Workspace objects are the Subject in the Observer pattern
00019 //                 (Design Patterns, Gamma et al, Addison Wesley, p293).
00020
00021 #ifndef _RBTWORKSPACE_H_
00022 #define _RBTWORKSPACE_H_
00023
00024 #include "rxdock/BaseMolecularFileSink.h"
00025 #include "rxdock/Config.h"
00026 #include "rxdock/DockingSite.h"
00027 #include "rxdock/Filter.h"
00028 #include "rxdock/Model.h"
00029 #include "rxdock/ParamHandler.h"
00030 #include "rxdock/Population.h"
00031 #include "rxdock/Subject.h"
00032
00033 namespace rxdock {
00034
00035 class BaseSF;        // Forward definition
00036 class BaseTransform; // Forward definition
00037
00038 class WorkSpace : public Subject, public ParamHandler {
00039 public:
00040   // Class type string
00041   static const std::string _CT;
00042   // Parameter names
00043   static const std::string _NAME;
00044
00046   // Constructors/destructors
00047
00048   WorkSpace(unsigned int nModels = 2);
00049   virtual ~WorkSpace();
00050
00052   // Public methods
00054
00055   // Get/set workspace name
00056   RBTDLL_EXPORT std::string GetName() const;
00057   RBTDLL_EXPORT void SetName(const std::string &);
00058
00059   // Model handling
00060
00061   // Returns number of models in workspace
00062   RBTDLL_EXPORT unsigned int GetNumModels() const;
00063   // Returns vector of all models in workspace
00064   RBTDLL_EXPORT ModelList GetModels() const;
00065   // Returns a specific (numbered) model
00066   // Throws BadArgument if iModel out of range
00067   RBTDLL_EXPORT ModelPtr GetModel(unsigned int iModel) const;
00068
00069   // Replace an existing model
00070   // Throws BadArgument if iModel out of range
00071   void SetModel(unsigned int iModel, ModelPtr spModel);
00072
00073   // Returns vector of models, starting from index iModel
00074   ModelList GetModels(unsigned int iModel) const;
00075   // Append a number of models to the workspace, increasing the total number of
00076   // models
00077   void AddModels(ModelList modelList);
00078   // Replace a number of existing models
00079   // iModel is the index of the first model to replace
00080   // Throws BadArgument if iModel out of range or modelList too large
00081   void SetModels(unsigned int iModel, ModelList modelList);
00082   // Removes a number of models from the workspace
00083   // Removes from index iModel to end of model list
00084   void RemoveModels(unsigned int iModel);
00085
00086   // Model I/O
00087   // Get/set the molecular file sink (for outputting ligands)
00088   MolecularFileSinkPtr GetSink() const;
```

```
00089   RBTDLL_EXPORT void SetSink(MolecularFileSinkPtr);
00090   // Saves models to file sink
00091   // If bSaveScores is true, write component scores also
00092   // Base workspace does nothing
00093   // It is up to subclasses to decide what action to take
00094   virtual void Save(bool bSaveScores = true);
00095   // Get/set the history file sink
00096   MolecularFileSinkPtr GetHistorySink() const;
00097   RBTDLL_EXPORT void SetHistorySink(MolecularFileSinkPtr);
00098   // Saves models to file sink
00099   // If bSaveScores is true, write component scores also
00100   // Base workspace does nothing
00101   // It is up to subclasses to decide what action to take
00102   virtual void SaveHistory(bool bSaveScores = true);
00103
00104   // Get/Set the scoring function
00105   // SetSF automatically registers the scoring function with the workspace
00106   RBTDLL_EXPORT BaseSF *GetSF() const;
00107   RBTDLL_EXPORT void SetSF(BaseSF *);
00108
00109   // Get/Set the transform
00110   // SetTransform automatically registers the transform with the workspace
00111   BaseTransform *GetTransform() const;
00112   RBTDLL_EXPORT void SetTransform(BaseTransform *);
00113
00114   // Run the simulation!
00115   virtual void Run();
00116
00117   // Population handling
00118   // DM 30 Oct 2001 - provide a home for persistent populations
00119   void SetPopulation(PopulationPtr population);
00120   PopulationPtr GetPopulation() const;
00121   void ClearPopulation();
00122
00123   // Docking site handling
00124   // DM 09 Apr 2002 - workspace now manages the docking site
00125   RBTDLL_EXPORT DockingSitePtr GetDockingSite() const;
00126   RBTDLL_EXPORT void SetDockingSite(DockingSitePtr spDockSite);
00127
00128   // Filter handling
00129   // BGD 27 Jan 2003 - SetFilter automotically registers the Filter
00130   // with the workspace
00131   FilterPtr GetFilter() const;
00132   RBTDLL_EXPORT void SetFilter(FilterPtr spFilter);
00133
00134 protected:
00136   // Protected methods
00138
00139 private:
00141   // Private methods
00143
00144   WorkSpace(const WorkSpace &); // Copy constructor disabled by default
00145
00146   WorkSpace &
00147   operator=(const WorkSpace &); // Copy assignment disabled by default
00148
00149 protected:
00151   // Protected data
00153
00154 private:
00156   // Private data
00158   ModelList m_models;
00159   MolecularFileSinkPtr m_spSink;
00160   MolecularFileSinkPtr m_spHisSink;
00161   BaseSF *m_SF;
00162   BaseTransform *m_transform;
00163   PopulationPtr m_population;
00164   DockingSitePtr m_spDockSite;
00165   FilterPtr m_spFilter;
00166 };
00167
00168 // Useful typedefs
00169 typedef SmartPtr<WorkSpace> WorkSpacePtr; // Smart pointer
00170
00171 } // namespace rxdock
00172
00173 #endif //_RBTWORKSPACE_H_
```

6.180 CSVFileSink.cxx File Reference

```
#include ¨rxdock/CSVFileSink.h¨
#include <iomanip>
```

6.180.1 Detailed Description

Outputs Model(s) to a CSV file.

6.181 CavitySearch.cxx File Reference

```
#include ¨rxdock/operation/CavitySearch.h¨
#include ¨rxdock/BiMolWorkSpace.h¨
#include ¨rxdock/CrdFileSink.h¨
#include ¨rxdock/Error.h¨
#include ¨rxdock/PRMFactory.h¨
#include ¨rxdock/PsfFileSink.h¨
#include ¨rxdock/SiteMapperFactory.h¨
#include <fmt/format.h>
#include <fmt/ostream.h>
```

6.181.1 Detailed Description

CavitySearch operation.

6.182 Dock.cxx File Reference

```
#include ¨rxdock/operation/Dock.h¨
#include ¨rxdock/BiMolWorkSpace.h¨
#include ¨rxdock/CrdFileSink.h¨
#include ¨rxdock/DockingError.h¨
#include ¨rxdock/Error.h¨
#include ¨rxdock/LigandError.h¨
#include ¨rxdock/MdlFileSink.h¨
#include ¨rxdock/MdlFileSource.h¨
#include ¨rxdock/PRMFactory.h¨
#include ¨rxdock/ParameterFileSource.h¨
#include ¨rxdock/SFFactory.h¨
#include ¨rxdock/TransformFactory.h¨
#include <fmt/chrono.h>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <chrono>
```

6.182.1 Detailed Description

Dock operation.

6.183 Tabularize.cxx File Reference

```
#include ¨rxdock/operation/Tabularize.h¨
#include ¨rxdock/CSVFileSink.h¨
#include ¨rxdock/MdlFileSource.h¨
#include ¨rxdock/ModelError.h¨
#include <fmt/format.h>
```

6.183.1 Detailed Description

Tabularize operation.

6.184 Transform.cxx File Reference

```
#include ¨rxdock/operation/Transform.h¨
#include ¨rxdock/MdlFileSink.h¨
#include ¨rxdock/MdlFileSource.h¨
#include ¨rxdock/ModelError.h¨
#include <algorithm>
#include <chrono>
#include <functional>
#include <fmt/chrono.h>
#include <fmt/format.h>
```

6.184.1 Detailed Description

Transform operation.

6.185 Number.cxx File Reference

```
#include ¨rxdock/support/Number.h¨
```

6.185.1 Detailed Description

Number conversion.

6.186 Quote.cxx File Reference

```
#include ¨rxdock/support/Quote.h¨
#include ¨rxdock/support/Number.h¨
#include ¨rxdock/Rand.h¨
#include <nlohmann/json.hpp>
#include <fstream>
#include <sstream>
```

6.186.1 Detailed Description

Quote reading, selection, and formatting.

6.187 ChromTest.h

```
00001 // Unit tests for ChromElement and subclasses
00002 //
00003 // Required input files:
00004 // 1koc.json RxDock receptor file
00005 // 1koc.psf  Receptor topology file
00006 // 1koc.crd  Receptor coordinate file
00007 // 1koc_c.sd Ligand coordinate file
00008 // 1koc.as   Docking site
00009 //
00010 // Required environment:
00011 // Make sure the above files are colocated in a single directory
00012 // and define RBT_HOME env. variable to point at this directory
00013 #ifndef RBTCHROMTEST_H_
00014 #define RBTCHROMTEST_H_
00015
00016 #include <gtest/gtest.h>
00017
00018 #include "rxdock/BiMolWorkSpace.h"
00019 #include "rxdock/Chrom.h"
00020 #include "rxdock/DockingSite.h"
00021 #include "rxdock/Model.h"
00022 #include "rxdock/SFAgg.h"
00023
00024 namespace rxdock {
00025
00026 namespace unittest {
00027
00028 class ChromTest : public ::testing::Test {
00029 protected:
00030   static double TINY;
00031   // TextFixture methods
00032   void SetUp() override;
00033   void TearDown() override;
00034
00035   // extra setup for building a simple scoring function and workspace
00036   void setupWorkSpace();
00037
00038   // rdock helper methods
00039   // RMSD calculation between two coordinate lists
00040   double rmsd(const CoordList &rc, const CoordList &c);
00041   // Measures the variation in chromosome vector values for repeated
00042   // mutations or randomisations
00043   void measureRandOrMutateDiff(ChromElement *chrom, int nTrials,
00044                                bool bMutate, // true=mutate; false=randomise
00045                                double &meanDiff, double &minDiff,
00046                                double &maxDiff);
00047   // Measures the variation in chromosome vector values for repeated
00048   // crossovers
00049   void measureCrossoverDiff(ChromElement *chrom, int nTrials, double &meanDiff,
00050                             double &minDiff, double &maxDiff);
00051
00052   ModelPtr m_recep_1koc;
00053   ModelPtr m_lig_1koc;
00054   DockingSitePtr m_site_1koc;
00055   ChromElementPtr m_chrom_1koc;
00056   AtomList m_atomList;           // All atoms in receptor and ligand
00057   BiMolWorkSpacePtr m_workSpace; // simple workspace
00058   SFAggPtr m_SF;                 // simple scoring function
00059 };
00060
00061 } // namespace unittest
00062
00063 } // namespace rxdock
00064
00065 #endif /*RBTCHROMTEST_H_*/
```

6.188 OccupancyTest.h

```
00001 // Unit tests for solvent occupancy
00002 //
00003 // Required input files:
```

```
00004 // 1YET.json RxDock receptor file
00005 // 1YET.psf  Receptor topology file
00006 // 1YET.crd  Receptor coordinate file
00007 // 1YET_c.sd Ligand coordinate file
00008 // 1YET.as   Docking site
00009 //
00010 // Required environment:
00011 // Make sure the above files are colocated in a single directory
00012 // and define RBT_HOME env. variable to point at this directory
00013 #ifndef OCCUPANCY_H_
00014 #define OCCUPANCY_H_
00015
00016 #include <gtest/gtest.h>
00017
00018 #include "rxdock/BiMolWorkSpace.h"
00019 #include "rxdock/Model.h"
00020
00021 namespace rxdock {
00022
00023 namespace unittest {
00024
00025 class OccupancyTest : public ::testing::Test {
00026 protected:
00027   static double TINY;
00028   // TextFixture methods
00029   void SetUp() override;
00030   void TearDown() override;
00031
00032   // Helper functions
00033   double CompareScoresForDisabledAndNoSolvent();
00034   double CompareScoresForSolventModes();
00035   void CheckFlexAtomFactory(Model *pModel, int expected[9][3]);
00036   BiMolWorkSpacePtr m_workSpace;
00037   ModelList m_solventList;
00038 };
00039
00040 } // namespace unittest
00041
00042 } // namespace rxdock
00043
00044 #endif /*OCCUPANCY_TEST_H_*/
```

6.189 SearchTest.h

```
00001 // Unit tests for rDock search algorithms
00002 //
00003 // Required input files:
00004 // 1YET.json RxDock receptor file
00005 // 1YET.psf  Receptor topology file
00006 // 1YET.crd  Receptor coordinate file
00007 // 1YET_c.sd Ligand coordinate file
00008 // 1YET.as   Docking site
00009 //
00010 // Required environment:
00011 // Make sure the above files are colocated in a single directory
00012 // and define RBT_HOME env. variable to point at this directory
00013 #ifndef SEARCHTEST_H_
00014 #define SEARCHTEST_H_
00015
00016 #include <gtest/gtest.h>
00017
00018 #include "rxdock/BiMolWorkSpace.h"
00019 #include "rxdock/Chrom.h"
00020 #include "rxdock/DockingSite.h"
00021 #include "rxdock/Model.h"
00022 #include "rxdock/SFAgg.h"
00023
00024 namespace rxdock {
00025
00026 namespace unittest {
00027
00028 class SearchTest : public ::testing::Test {
00029 protected:
00030   // TextFixture methods
00031   void SetUp() override;
00032   void TearDown() override;
00033
00034   // rdock helper methods
00035   // RMSD calculation between two coordinate lists
00036   double rmsd(const CoordList &rc, const CoordList &c);
00037
00038   AtomList m_atomList;           // All atoms in receptor, ligand and solvent
00039   BiMolWorkSpacePtr m_workSpace; // simple workspace
```

```
00040   SFAggPtr m_SF;                // simple scoring function
00041 };
00042
00043 } // namespace unittest
00044
00045 } // namespace rxdock
00046
00047 #endif /*SEARCHTEST_H_*/
```

6.190 ParseCavitySearch.cxx File Reference

```
#include ¨ParseCavitySearch.h¨
#include ¨rxdock/operation/CavitySearch.h¨
#include <cxxopts.hpp>
#include <fmt/format.h>
#include <stdexcept>
```

6.190.1 Detailed Description

Parse command-line interface parameters for CavitySearch operation.

6.191 ParseCavitySearch.h File Reference

Functions

- int **rxdock::parseCavitySearch** (int argc, char ∗argv[ ])

6.191.1 Detailed Description

Parse command-line interface parameters for CavitySearch operation.

6.192 ParseCavitySearch.h

Go to the documentation of this file.

```
00001 //===-- ParseCavitySearch.h - Parse CLI params for CavSearch ----*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_TOOLS_RXCMD_PARSECAVITYSEARCH_H
00019 #define RXDOCK_TOOLS_RXCMD_PARSECAVITYSEARCH_H
00020
00021 namespace rxdock {
00022
00023 int parseCavitySearch(int argc, char *argv[]);
00024
00025 } // namespace rxdock
00026
00027 #endif // RXDOCK_TOOLS_RXCMD_PARSECAVITYSEARCH_H
```

6.193 ParseDock.cxx File Reference

```
#include ¨ParseDock.h¨
#include ¨rxdock/operation/Dock.h¨
#include <cxxopts.hpp>
#include <fmt/format.h>
#include <stdexcept>
```

6.193.1 Detailed Description

Parse command-line interface parameters for Dock operation.

6.194 ParseDock.h File Reference

Functions

- int **rxdock::parseDock** (int argc, char ∗argv[ ])

6.194.1 Detailed Description

Parse command-line interface parameters for Dock operation.

6.195 ParseDock.h

Go to the documentation of this file.

```
00001 //===-- ParseDock.h - Parse CLI params for Dock -----------------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_TOOLS_RXCMD_PARSEDOCK_H
00019 #define RXDOCK_TOOLS_RXCMD_PARSEDOCK_H
00020
00021 namespace rxdock {
00022
00023 int parseDock(int argc, char *argv[]);
00024
00025 } // namespace rxdock
00026
00027 #endif // RXDOCK_TOOLS_RXCMD_PARSEDOCK_H
```

6.196 ParseTabularize.cxx File Reference

```
#include ¨ParseTabularize.h¨
#include ¨rxdock/operation/Tabularize.h¨
#include <cxxopts.hpp>
#include <fmt/format.h>
#include <stdexcept>
```

6.196.1 Detailed Description

Parse command-line interface parameters for Tabularize operation.

6.197 ParseTabularize.h File Reference

Functions

- int **rxdock::parseTabularize** (int argc, char ∗argv[ ])

6.197.1 Detailed Description

Parse command-line interface parameters for Tabularize operation.

6.198 ParseTabularize.h

Go to the documentation of this file.

```
00001 //===-- ParseTabularize.h - Parse CLI params for Tabularize -----*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_TOOLS_RXCMD_PARSETABULARIZE_H
00019 #define RXDOCK_TOOLS_RXCMD_PARSETABULARIZE_H
00020
00021 namespace rxdock {
00022
00023 int parseTabularize(int argc, char *argv[]);
00024
00025 } // namespace rxdock
00026
00027 #endif // RXDOCK_TOOLS_RXCMD_PARSETRANSFORM_H
```

6.199 ParseTransform.cxx File Reference

```
#include ¨ParseTransform.h¨
#include ¨rxdock/operation/Transform.h¨
#include <cxxopts.hpp>
#include <fmt/format.h>
#include <stdexcept>
```

6.199.1 Detailed Description

Parse command-line interface parameters for Transform operation.

6.200 ParseTransform.h File Reference

Functions

- int **rxdock::parseTransform** (int argc, char ∗argv[ ])

6.200.1 Detailed Description

Parse command-line interface parameters for Transform operation.

6.201 ParseTransform.h

Go to the documentation of this file.

```
00001 //===-- ParseTransform.h - Parse CLI params for Transform -------*- C++ -*-===//
00002 //
00003 // Part of the RxDock project, under the GNU LGPL version 3.
00004 // Visit https://rxdock.gitlab.io/ for more information.
00005 // Copyright (c) 1998--2006 RiboTargets (subsequently Vernalis (R&D) Ltd)
00006 // Copyright (c) 2006--2012 University of York
00007 // Copyright (c) 2012--2014 University of Barcelona
00008 // Copyright (c) 2019--2020 RxTx
00009 // SPDX-License-Identifier: LGPL-3.0-only
00010 //
00011 //===----------------------------------------------------------------------===//
00016 //===----------------------------------------------------------------------===//
00017
00018 #ifndef RXDOCK_TOOLS_RXCMD_PARSETRANSFORM_H
00019 #define RXDOCK_TOOLS_RXCMD_PARSETRANSFORM_H
00020
00021 namespace rxdock {
00022
00023 int parseTransform(int argc, char *argv[]);
00024
00025 } // namespace rxdock
00026
00027 #endif // RXDOCK_TOOLS_RXCMD_PARSETRANSFORM_H
```

6.202 rxcmd.cxx File Reference

```
#include ¨ParseCavitySearch.h¨
#include ¨ParseDock.h¨
#include ¨ParseTabularize.h¨
#include ¨ParseTransform.h¨
#include ¨rxdock/Rbt.h¨
#include ¨rxdock/support/Quote.h¨
#include <cxxopts.hpp>
#include <fmt/chrono.h>
#include <fmt/format.h>
#include <loguru.hpp>
#include <algorithm>
#include <chrono>
#include <functional>
#include <iomanip>
#include <random>
#include <stdexcept>
```

Functions

- int **rxdock::parseNull** (int argc, char ∗argv[ ])

  Null function for RxDock commands that aren't yet implemented.
- int **main** (int argc, char ∗argv[ ])

  Entry point for the RxDock command-line interface.

6.202.1 Detailed Description

The starting point of the RxDock command-line interface.

Index

_ANNOTATION_LIPO
  rxdock::VdwIdxSF, 413
_CT
  rxdock::PMFIdxSF, 294
_MIN_ACC_RATE
  rxdock::SimAnnTransform, 370
_PMFDIR
  rxdock::PMFIdxSF, 294
_REC_DIHEDRAL_STEP
  rxdock::PRMFactory, 316
_REC_NUM_COORD_FILES
  rxdock::PRMFactory, 316
_SCALE_CHROM_LENGTH
  rxdock::RandPopTransform, 331
  rxdock::SimAnnTransform, 370
_STOPPING_STEP_LENGTH
  rxdock::SimplexTransform, 376
_TRIPOS_DELIM_SIZE
  rxdock::MOL2FileSource, 253
_VOL_INCR
  rxdock::LigandSiteMapper, 238
  rxdock::SphereSiteMapper, 384
∼PMFIdxSF
  rxdock::PMFIdxSF, 292

Accept
  rxdock::FilterAddExp, 157
  rxdock::FilterAndExp, 158
  rxdock::FilterDivExp, 160
  rxdock::FilterExpExp, 161
  rxdock::FilterIfExp, 164
  rxdock::FilterLogExp, 166
  rxdock::FilterMulExp, 167
  rxdock::FilterSubExp, 169
  rxdock::FilterVbleExp, 170
  rxdock::LigandFlexData, 235
  rxdock::ReceptorFlexData, 335
  rxdock::SolventFlexData, 381
Add
  rxdock::Chrom, 88
  rxdock::SFAgg, 362
  rxdock::TransformAgg, 399
AddAtomList
  rxdock::HBAConstraint, 191
  rxdock::HBDConstraint, 193
  rxdock::HeavyConstraint, 195
  rxdock::HydroAliphaticConstraint, 198
  rxdock::HydroAromaticConstraint, 200
  rxdock::HydroConstraint, 201
  rxdock::NegChargeConstraint, 256
  rxdock::PosChargeConstraint, 310
  rxdock::RingAromaticConstraint, 338
AlignTransform.h, 423
Annotation.h, 424
AnnotationHandler.h, 425
AromIdxSF.h, 426
Assign
  rxdock::CellContext, 82
  rxdock::StringContext, 386
Atom.h, 428
AtomFuncs.h, 442

BaseBiMolTransform.h, 443
BaseFileSink.h, 444
BaseFileSource.h, 445
BaseGrid.h, 446
BaseIdxSF.h, 449
BaseInterSF.h, 450
BaseIntraSF.h, 451
BaseMolecularDataSource.h, 452
BaseMolecularFileSink.h, 453
BaseMolecularFileSource.h, 455
BaseObject.h, 456
BaseSF.h, 458
BaseTransform.h, 459
BaseUniMolTransform.h, 460
BiMolWorkSpace.h, 461
Bond.h, 462

CalculateFitness
  rxdock::geneticprogram::GPFFCHK1, 180
  rxdock::geneticprogram::GPFFCrossDock, 182
  rxdock::geneticprogram::GPFFGold, 183
  rxdock::geneticprogram::GPFFHSP90, 185
  rxdock::geneticprogram::GPFFSpike, 187
Cavity.h, 465
CavityFillSF.h, 467
CavityGridSF.h, 468
CavitySearch.cxx, 644
CavitySearch.h, 565, 566
Cell.h, 519
CellTokenIter.h, 469
CharmmDataSource.h, 470
CharmmTypesFileSource.h, 471
Chrom.h, 472
ChromDihedralElement.h, 473
ChromDihedralRefData.h, 474
ChromElement.h, 475
ChromFactory.h, 477
ChromOccupancyElement.h, 478

ChromOccupancyRefData.h, 479
ChromPositionElement.h, 479
ChromPositionRefData.h, 480
ChromTest.h, 646
clone
    rxdock::Chrom, 88
    rxdock::ChromDihedralElement, 92
    rxdock::ChromOccupancyElement, 99
    rxdock::ChromPositionElement, 103
Command.h, 520
Commands.h, 482
CompareVector
    rxdock::Chrom, 88
    rxdock::ChromDihedralElement, 92
    rxdock::ChromOccupancyElement, 99
    rxdock::ChromPositionElement, 103
Config.h, 483
Constraint.h, 483
ConstSF.h, 485
Context.h, 486
Coord.h, 488
counter
    rxdock::HeavyConstraint, 195
CrdFileSink.h, 493
CrdFileSource.h, 494
CSVFileSink.cxx, 643
CSVFileSink.h, 494, 495
Current
    rxdock::CellTokenIter, 84
    rxdock::StringTokenIter, 388

d_s
    rxdock::HHS_Solvation, 196
Debug.h, 496
Deleted
    rxdock::BaseObject, 60
DihedralIntraSF.h, 497
DihedralSF.h, 497
DihedralTargetSF.h, 499
DirectorySource
    rxdock::DirectorySource, 132
DirectorySource.h, 500
Dock.cxx, 644
Dock.h, 566, 567
DockingError.h, 500
DockingSite.h, 501

ElementFileSource.h, 503
Error.h, 504
Euler.h, 506
Execute
    rxdock::AddCommand, 22
    rxdock::AlignTransform, 25
    rxdock::DivCommand, 134
    rxdock::ExpCommand, 144
    rxdock::GATransform, 176
    rxdock::IfCommand, 202
    rxdock::LogCommand, 239
    rxdock::MulCommand, 255
    rxdock::NullTransform, 271
    rxdock::RandLigTransform, 328
    rxdock::RandPopTransform, 331
    rxdock::SimAnnTransform, 370
    rxdock::SimplexTransform, 375
    rxdock::SubCommand, 390
    rxdock::TransformAgg, 399
Export.h, 623, 624

FFTGrid.h, 507
FileError.h, 509
Filter.h, 510
FilterExpression.h, 511
FilterExpressionVisitor.h, 514
FlexAtomFactory.h, 516
FlexData.h, 517
FlexDataVisitor.h, 517

GATransform.h, 518
Genome.h, 531
GetAtomList
    rxdock::BaseMolecularFileSource, 57
GetBestParameters
    rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion >, 372
GetBestValue
    rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion >, 372
GetBondList
    rxdock::BaseMolecularFileSource, 57
GetDataFieldList
    rxdock::BaseMolecularFileSource, 57
GetDataMap
    rxdock::BaseMolecularFileSource, 57
GetDataValue
    rxdock::BaseMolecularFileSource, 57
GetFullName
    rxdock::BaseSF, 63
    rxdock::BaseTransform, 66
GetLength
    rxdock::Chrom, 88
    rxdock::ChromDihedralElement, 92
    rxdock::ChromOccupancyElement, 99
    rxdock::ChromPositionElement, 104
GetLinearCloseRangeValue
    rxdock::PMFIdxSF, 293
GetName
    rxdock::AddCommand, 22
    rxdock::DivCommand, 134
    rxdock::ExpCommand, 144
    rxdock::IfCommand, 202
    rxdock::LogCommand, 239
    rxdock::MulCommand, 255
    rxdock::SubCommand, 390
GetNArgs
    rxdock::AddCommand, 22
    rxdock::DivCommand, 135
    rxdock::ExpCommand, 145
    rxdock::IfCommand, 202

rxdock::LogCommand, 239
rxdock::MulCommand, 255
rxdock::SubCommand, 390
GetNOps
rxdock::FilterAddExp, 157
rxdock::FilterAndExp, 158
rxdock::FilterDivExp, 160
rxdock::FilterExpExp, 161
rxdock::FilterIfExp, 164
rxdock::FilterLogExp, 166
rxdock::FilterMulExp, 167
rxdock::FilterSubExp, 169
rxdock::FilterVbleExp, 170
GetNumAtoms
rxdock::BaseMolecularFileSource, 58
GetNumBonds
rxdock::BaseMolecularFileSource, 58
GetNumData
rxdock::BaseMolecularFileSource, 58
GetNumSegments
rxdock::BaseMolecularFileSource, 58
GetNumSF
rxdock::SFAgg, 362
GetNumTitles
rxdock::BaseMolecularFileSource, 58
GetNumTransforms
rxdock::TransformAgg, 399
GetOp
rxdock::FilterAddExp, 157
rxdock::FilterAndExp, 159
rxdock::FilterDivExp, 160
rxdock::FilterExpExp, 162
rxdock::FilterIfExp, 165
rxdock::FilterLogExp, 166
rxdock::FilterMulExp, 168
rxdock::FilterSubExp, 169
rxdock::FilterVbleExp, 171
GetPMFfor_IC
rxdock::SetupPMFSF, 352
getRandomQuote
Quote.h, 626
getRomanNumeral
Number.h, 624
GetSegmentMap
rxdock::BaseMolecularFileSource, 58
GetSF
rxdock::SFAgg, 363
GetStepVector
rxdock::Chrom, 88
rxdock::ChromDihedralElement, 92
rxdock::ChromOccupancyElement, 99
rxdock::ChromPositionElement, 104
GetTitleList
rxdock::BaseMolecularFileSource, 58
GetTransform
rxdock::TransformAgg, 399
GetVble
rxdock::CellContext, 83
rxdock::StringContext, 387
GetVector
rxdock::Chrom, 89
rxdock::ChromDihedralElement, 93
rxdock::ChromOccupancyElement, 99
rxdock::ChromPositionElement, 104
GetXOverLength
rxdock::Chrom, 89
rxdock::ChromDihedralElement, 93
rxdock::ChromOccupancyElement, 100
rxdock::ChromPositionElement, 104
GPChromosome.h, 522
GPFFCHK1.h, 523
GPFFCrossDock.h, 524
GPFFGold.h, 525
GPFFHSP90.h, 525
GPFFSpike.h, 526
GPFitnessFunction.h, 527
GPGenome.h, 528
GPParser.h, 529
GPPopulation.h, 530
GPTypes.h, 531

HandleRequest
rxdock::BaseObject, 60
rxdock::PolarIntraSF, 304
rxdock::SAIdxSF, 346
rxdock::SFAgg, 363
rxdock::TransformAgg, 399
rxdock::VdwIntraSF, 417

InteractionGrid.h, 533
InteractionTemplate.h, 535
isAgg
rxdock::SFAgg, 363
rxdock::TransformAgg, 400
isAtomListSupported
rxdock::CrdFileSource, 119
rxdock::MdlFileSource, 244
rxdock::MOL2FileSource, 252
rxdock::PdbFileSource, 279
rxdock::PsfFileSource, 323
isBondListSupported
rxdock::CrdFileSource, 119
rxdock::MdlFileSource, 244
rxdock::MOL2FileSource, 252
rxdock::PdbFileSource, 279
rxdock::PsfFileSource, 323
isCoordinatesSupported
rxdock::CrdFileSource, 120
rxdock::MdlFileSource, 244
rxdock::MOL2FileSource, 252
rxdock::PdbFileSource, 279
rxdock::PsfFileSource, 323
isDataFieldPresent
rxdock::BaseMolecularFileSource, 59
isDataSupported
rxdock::MdlFileSource, 245
rxdock::MOL2FileSource, 253

isMultiConfSupported
rxdock::CrdFileSink, 117
isTitleListSupported
rxdock::CrdFileSource, 120
rxdock::MdlFileSource, 245
rxdock::MOL2FileSource, 253
rxdock::PdbFileSource, 279
rxdock::PsfFileSource, 323

LigandError.h, 537
LigandFlexData.h, 538
LigandSiteMapper.h, 539

Main Page, 1
MdlFileSink.h, 539
MdlFileSource.h, 541
Model.h, 542
ModelError.h, 546
ModelMutator.h, 547
MOEGrid.h, 548
MOL2FileSource.h, 551
Mutate
rxdock::Chrom, 89
rxdock::ChromDihedralElement, 93
rxdock::ChromOccupancyElement, 100
rxdock::ChromPositionElement, 104

Next
rxdock::CellTokenIter, 84
rxdock::StringTokenIter, 388
NMCriteria.h, 552
NmrRestraintFileSource.h, 553
NmrSF.h, 555
NMSimplex.h, 556
NMState.h, 558
NoeRestraint.h, 559
NonBondedGrid.h, 561
NonBondedHHSGrid.h, 563
NullTransform.h, 564
Number.cxx, 645
Number.h, 624, 625
getRomanNumeral, 624

Observer.h, 564
OccupancyTest.h, 646
operator()
rxdock::LigandSiteMapper, 237
rxdock::SphereSiteMapper, 383

ParameterFileSource.h, 569
ParameterUpdated
rxdock::AromIdxSF, 32
rxdock::BaseObject, 61
rxdock::BaseSF, 63
rxdock::CavityFillSF, 76
rxdock::CavityGridSF, 80
rxdock::ConstSF, 113
rxdock::NmrSF, 261
rxdock::PharmaSF, 283
rxdock::PolarIdxSF, 299
rxdock::PolarIntraSF, 304
rxdock::RotSF, 342
rxdock::TetherSF, 395
rxdock::VdwGridSF, 407
rxdock::VdwIdxSF, 412
rxdock::VdwIntraSF, 417
ParamHandler.h, 570
Parse
rxdock::CrdFileSource, 120
rxdock::MdlFileSource, 245
rxdock::MOL2FileSource, 253
rxdock::PdbFileSource, 279
rxdock::PsfFileSource, 323
ParseCavitySearch.cxx, 648
ParseCavitySearch.h, 648
ParseDock.cxx, 649
ParseDock.h, 649
Parser.h, 572
ParseTabularize.cxx, 649
ParseTabularize.h, 650
ParseTransform.cxx, 650
ParseTransform.h, 651
PdbFileSource.h, 572
PharmaSF.h, 573
Pij_12
rxdock::HHS_Solvation, 196
Plane.h, 574
PMF.h, 575
PMFDirSource.h, 576
PMFGridSF.h, 577
PMFIdxSF
rxdock::PMFIdxSF, 292
PMFIdxSF.h, 578
PolarIdxSF.h, 579
PolarIntraSF.h, 580
PolarSF.h, 581
Population.h, 583
PrincipalAxes.h, 584
Print
rxdock::BaseObject, 61
rxdock::Chrom, 89
rxdock::ChromDihedralElement, 93
rxdock::ChromOccupancyElement, 100
rxdock::ChromPositionElement, 104
rxdock::FFTGrid, 148
rxdock::FilterAddExp, 157
rxdock::FilterAndExp, 159
rxdock::FilterDivExp, 160
rxdock::FilterExpExp, 162
rxdock::FilterIfExp, 165
rxdock::FilterLogExp, 166
rxdock::FilterMulExp, 168
rxdock::FilterSubExp, 169
rxdock::FilterVbleExp, 171
rxdock::InteractionGrid, 206
rxdock::InteractionGridTemplate< T >, 209
rxdock::NonBondedGrid, 267

rxdock::NonBondedHHSGrid, 269
rxdock::PseudoAtom, 319
rxdock::RealGrid, 334
rxdock::SFAgg, 363
rxdock::TransformAgg, 400
PRMFactory.h, 585
PseudoAtom.h, 586
PsfFileSink.h, 587
PsfFileSource.h, 588

Quat.h, 589
Quote.cxx, 645
Quote.h, 625, 626
getRandomQuote, 626

Rand.h, 592
RandLigTransform.h, 592
Randomise
rxdock::Chrom, 89
rxdock::ChromDihedralElement, 93
rxdock::ChromOccupancyElement, 100
rxdock::ChromPositionElement, 105
RandPopTransform.h, 593
RawScore
rxdock::AromIdxSF, 32
rxdock::BaseSF, 64
rxdock::CavityFillSF, 76
rxdock::CavityGridSF, 80
rxdock::ConstSF, 113
rxdock::DihedralIntraSF, 125
rxdock::DihedralTargetSF, 131
rxdock::NmrSF, 261
rxdock::PharmaSF, 283
rxdock::PMFGridSF, 288
rxdock::PMFIdxSF, 293
rxdock::PolarIdxSF, 299
rxdock::PolarIntraSF, 304
rxdock::RotSF, 342
rxdock::SAIdxSF, 346
rxdock::SetupPMFSF, 352
rxdock::SetupPolarSF, 355
rxdock::SetupSASF, 359
rxdock::SFAgg, 363
rxdock::TetherSF, 395
rxdock::VdwGridSF, 407
rxdock::VdwIdxSF, 412
rxdock::VdwIntraSF, 417
Rbt.h, 594
ReadTables
rxdock::geneticprogram::GPFFCHK1, 180
rxdock::geneticprogram::GPFFCrossDock, 182
rxdock::geneticprogram::GPFFGold, 184
rxdock::geneticprogram::GPFFHSP90, 185
rxdock::geneticprogram::GPFFSpike, 187
RealGrid.h, 596
ReceptorFlexData.h, 599
Register
rxdock::SFAgg, 363
rxdock::TransformAgg, 400
Remove
rxdock::SFAgg, 363
rxdock::TransformAgg, 400
Render
rxdock::CrdFileSink, 117
rxdock::CSVFileSink, 122
rxdock::MdlFileSink, 242
rxdock::PsfFileSink, 321
Request.h, 599
RequestHandler.h, 600
Reset
rxdock::BaseMolecularFileSource, 59
rxdock::Chrom, 89
rxdock::ChromDihedralElement, 93
rxdock::ChromOccupancyElement, 100
rxdock::ChromPositionElement, 105
Resources.h, 601
RotSF.h, 601
rxcmd.cxx, 651
rxdock::AddCommand, 21
Execute, 22
GetName, 22
GetNArgs, 22
SetArg, 22
SetNameArg, 22
rxdock::AlignTransform, 23
Execute, 25
SetupLigand, 25
SetupReceptor, 25
SetupTransform, 26
rxdock::Ann_Cmp_AtomId2, 26
rxdock::Annotation, 27
rxdock::AnnotationHandler, 27
rxdock::AromIdxSF, 28
ParameterUpdated, 32
RawScore, 32
SetupLigand, 32
SetupReceptor, 32
SetupScore, 32
rxdock::Assertion, 32
rxdock::Atom, 33
rxdock::AtomPtrCmp_AtomicNo, 35
rxdock::AtomPtrCmp_AtomId, 36
rxdock::AtomPtrCmp_AtomName, 36
rxdock::AtomPtrCmp_Ptr, 36
rxdock::BadArgument, 36
rxdock::BadReceptorFile, 37
rxdock::BaseBiMolTransform, 38
Update, 40
rxdock::BaseFileSink, 41
rxdock::BaseFileSource, 41
rxdock::BaseGrid, 42
rxdock::BaseIdxSF, 44
rxdock::BaseInterSF, 47
SetupLigand, 49
SetupReceptor, 49
SetupScore, 49
Update, 50

rxdock::BaseIntraSF, 50
ScoreMap, 52
Update, 52
rxdock::BaseMolecularDataSource, 53
rxdock::BaseMolecularFileSink, 54
rxdock::BaseMolecularFileSource, 55
GetAtomList, 57
GetBondList, 57
GetDataFieldList, 57
GetDataMap, 57
GetDataValue, 57
GetNumAtoms, 58
GetNumBonds, 58
GetNumData, 58
GetNumSegments, 58
GetNumTitles, 58
GetSegmentMap, 58
GetTitleList, 58
isDataFieldPresent, 59
Reset, 59
rxdock::BaseObject, 59
Deleted, 60
HandleRequest, 60
ParameterUpdated, 61
Print, 61
rxdock::BaseSF, 61
GetFullName, 63
ParameterUpdated, 63
RawScore, 64
rxdock::BaseTransform, 64
GetFullName, 66
rxdock::BaseUniMolTransform, 66
Update, 69
rxdock::BiMolWorkSpace, 69
Save, 71
SaveHistory, 71
rxdock::Bond, 71
rxdock::BondPCmp_BondId, 72
rxdock::Cavity, 72
rxdock::CavityFillSF, 73
ParameterUpdated, 76
RawScore, 76
SetupLigand, 76
SetupReceptor, 76
SetupScore, 76
rxdock::CavityGridSF, 77
ParameterUpdated, 80
RawScore, 80
SetupLigand, 80
SetupReceptor, 80
SetupScore, 80
SetupSolvent, 80
rxdock::CavityGridSF::HeavyAtomFactory, 193
VisitLigandFlexData, 193
VisitReceptorFlexData, 193
VisitSolventFlexData, 194
rxdock::CavityPtrCmp_Distance, 81
rxdock::CavityPtrCmp_Volume, 81
rxdock::Cell, 81
rxdock::CellContext, 82
Assign, 82
GetVble, 83
SetVble, 83
rxdock::CellTokenIter, 83
Current, 84
Next, 84
rxdock::CharmmDataSource, 84
rxdock::CharmmType, 85
rxdock::CharmmTypesFileSource, 85
rxdock::Chrom, 86
Add, 88
clone, 88
CompareVector, 88
GetLength, 88
GetStepVector, 88
GetVector, 89
GetXOverLength, 89
Mutate, 89
Print, 89
Randomise, 89
Reset, 89
SetVector, 90
SyncFromModel, 90
SyncToModel, 90
rxdock::ChromDihedralElement, 90
clone, 92
CompareVector, 92
GetLength, 92
GetStepVector, 92
GetVector, 93
GetXOverLength, 93
Mutate, 93
Print, 93
Randomise, 93
Reset, 93
SetVector, 94
SyncFromModel, 94
SyncToModel, 94
rxdock::ChromDihedralRefData, 94
rxdock::ChromElement, 95
rxdock::ChromFactory, 96
VisitLigandFlexData, 97
VisitReceptorFlexData, 97
VisitSolventFlexData, 97
rxdock::ChromOccupancyElement, 97
clone, 99
CompareVector, 99
GetLength, 99
GetStepVector, 99
GetVector, 99
GetXOverLength, 100
Mutate, 100
Print, 100
Randomise, 100
Reset, 100
SetVector, 100

SyncFromModel, 101
SyncToModel, 101
rxdock::ChromOccupancyRefData, 101
rxdock::ChromPositionElement, 102
clone, 103
CompareVector, 103
GetLength, 104
GetStepVector, 104
GetVector, 104
GetXOverLength, 104
Mutate, 104
Print, 104
Randomise, 105
Reset, 105
SetVector, 105
SyncFromModel, 105
SyncToModel, 105
rxdock::ChromPositionRefData, 106
rxdock::Command, 107
rxdock::Commands, 108
rxdock::Constraint, 109
rxdock::ConstSF, 110
ParameterUpdated, 113
RawScore, 113
ScoreMap, 113
SetupLigand, 113
SetupReceptor, 113
SetupScore, 113
rxdock::Context, 114
rxdock::Coord, 114
rxdock::CoordCmp, 115
rxdock::CrdFileSink, 115
isMultiConfSupported, 117
Render, 117
rxdock::CrdFileSource, 117
isAtomListSupported, 119
isBondListSupported, 119
isCoordinatesSupported, 120
isTitleListSupported, 120
Parse, 120
rxdock::CSVFileSink, 120
Render, 122
rxdock::CyclicAtom, 122
rxdock::DihedralElement, 122
rxdock::DihedralElement::prms, 317
rxdock::DihedralIntraSF, 123
RawScore, 125
SetupScore, 125
rxdock::DihedralSF, 126
rxdock::DihedralTargetSF, 128
RawScore, 131
SetupLigand, 131
SetupReceptor, 131
SetupScore, 131
rxdock::DirectorySource, 132
DirectorySource, 132
rxdock::DirIsNotAccessible, 133
rxdock::DivCommand, 134
Execute, 134
GetName, 134
GetNArgs, 135
SetArg, 135
SetNameArg, 135
rxdock::DockingError, 135
rxdock::DockingSite, 136
rxdock::DockingSite::isAtomInRange, 218
rxdock::ElementData, 137
rxdock::ElementFileSource, 137
rxdock::EnumerateSymCoords, 139
rxdock::EnvNotDefined, 139
rxdock::Error, 140
rxdock::Euler, 141
rxdock::EvaluateVisitor, 141
VisitAddExp, 142
VisitAndExp, 142
VisitDivExp, 142
VisitExpExp, 142
VisitIfExp, 143
VisitLogExp, 143
VisitMulExp, 143
VisitSubExp, 143
VisitVbleExp, 143
rxdock::ExpCommand, 144
Execute, 144
GetName, 144
GetNArgs, 145
SetArg, 145
SetNameArg, 145
rxdock::FFTGrid, 146
Print, 148
rxdock::FFTPeak, 149
rxdock::FileError, 149
rxdock::FileMissingParameter, 150
rxdock::FileParseError, 151
rxdock::FileReadError, 152
rxdock::FileWriteError, 153
rxdock::Filter, 154
Update, 156
rxdock::FilterAddExp, 156
Accept, 157
GetNOps, 157
GetOp, 157
Print, 157
rxdock::FilterAndExp, 158
Accept, 158
GetNOps, 158
GetOp, 159
Print, 159
rxdock::FilterDivExp, 159
Accept, 160
GetNOps, 160
GetOp, 160
Print, 160
rxdock::FilterExpExp, 161
Accept, 161
GetNOps, 161

GetOp, 162
Print, 162
rxdock::FilterExpression, 162
rxdock::FilterExpressionVisitor, 163
rxdock::FilterIfExp, 164
Accept, 164
GetNOps, 164
GetOp, 165
Print, 165
rxdock::FilterLogExp, 165
Accept, 166
GetNOps, 166
GetOp, 166
Print, 166
rxdock::FilterMulExp, 167
Accept, 167
GetNOps, 167
GetOp, 168
Print, 168
rxdock::FilterSubExp, 168
Accept, 169
GetNOps, 169
GetOp, 169
Print, 169
rxdock::FilterVbleExp, 170
Accept, 170
GetNOps, 170
GetOp, 171
Print, 171
rxdock::FlexAtomFactory, 171
VisitLigandFlexData, 172
VisitReceptorFlexData, 172
VisitSolventFlexData, 172
rxdock::FlexData, 172
rxdock::FlexDataVisitor, 173
rxdock::GATransform, 174
Execute, 176
SetupLigand, 176
SetupReceptor, 177
SetupTransform, 177
rxdock::geneticprogram::GPChromosome, 178
rxdock::geneticprogram::GPFFCHK1, 179
CalculateFitness, 180
ReadTables, 180
rxdock::geneticprogram::GPFFCrossDock, 181
CalculateFitness, 182
ReadTables, 182
rxdock::geneticprogram::GPFFGold, 182
CalculateFitness, 183
ReadTables, 184
rxdock::geneticprogram::GPFFHSP90, 184
CalculateFitness, 185
ReadTables, 185
rxdock::geneticprogram::GPFFSpike, 186
CalculateFitness, 187
ReadTables, 187
rxdock::geneticprogram::GPFitnessFunction, 187
rxdock::geneticprogram::GPGenome, 188
rxdock::geneticprogram::GPParser, 189
rxdock::geneticprogram::GPPopulation, 190
rxdock::geneticprogram::PopCmp, 308
rxdock::Genome, 177
rxdock::GenomeCmp_Score, 178
rxdock::HBAConstraint, 190
AddAtomList, 191
rxdock::HBDConstraint, 192
AddAtomList, 193
rxdock::HeavyConstraint, 194
AddAtomList, 195
counter, 195
rxdock::HHS_Solvation, 196
d_s, 196
Pij_12, 196
rxdock::HHSType, 197
rxdock::HydroAliphaticConstraint, 197
AddAtomList, 198
rxdock::HydroAromaticConstraint, 199
AddAtomList, 200
rxdock::HydroConstraint, 200
AddAtomList, 201
rxdock::IfCommand, 201
Execute, 202
GetName, 202
GetNArgs, 202
SetArg, 202
SetNameArg, 202
rxdock::InitHHS, 203
rxdock::InteractionCenter, 203
rxdock::InteractionCenterCmp, 204
rxdock::InteractionGrid, 204
Print, 206
rxdock::InteractionGridTemplate< T >, 207
Print, 209
rxdock::InteractionGridTemplatePtr< T >, 209
rxdock::InteractionNode< T >, 210
rxdock::InteractionNodeList< T >, 210
rxdock::InteractionNodeListMap< T >, 210
rxdock::InteractionNodeListMapConstIter< T >, 211
rxdock::InteractionNodeListMapIter< T >, 211
rxdock::InvalidRequest, 211
rxdock::InvertSelectAtom, 212
rxdock::isAtom_12Connected, 212
rxdock::isAtom_13Connected, 213
rxdock::isAtom_eq, 213
rxdock::isAtomAnionic, 214
rxdock::isAtomBridgehead, 214
rxdock::isAtomCationic, 214
rxdock::isAtomCharged, 215
rxdock::isAtomCyclic, 215
rxdock::isAtomEnabled, 216
rxdock::isAtomExtended, 216
rxdock::isAtomGuanidiniumCarbon, 216
rxdock::isAtomHBondAcceptor, 217
rxdock::isAtomHBondDonor, 217
rxdock::isAtomicNo_eq, 218
rxdock::isAtomId_eq, 218

rxdock::isAtomInsideCuboid, 219
rxdock::isAtomInsideSphere, 219
rxdock::isAtomIonic, 220
rxdock::isAtomLipophilic, 220
rxdock::isAtomMetal, 220
rxdock::isAtomName_eq, 221
rxdock::isAtomNearCoordList, 221
rxdock::isAtomNegCharged, 222
rxdock::isAtomNitroN, 222
rxdock::isAtomPlanar, 222
rxdock::isAtomPosCharged, 223
rxdock::isAtomPtr_eq, 223
rxdock::isAtomRNA, 224
rxdock::isAtomSelected, 224
rxdock::isBond_eq, 224
rxdock::isBondAmide, 225
rxdock::isBondCyclic, 225
rxdock::isBondPtr_eq, 226
rxdock::isBondRotatable, 226
rxdock::isBondSelected, 226
rxdock::isBondToNH3, 227
rxdock::isBondToOH, 227
rxdock::isCavityNearCoord, 228
rxdock::isCoordinationNumber_eq, 228
rxdock::isFFType_eq, 229
rxdock::isGenome_eq, 229
rxdock::isHHSSelected, 229
rxdock::isHHSType_eq, 230
rxdock::isHybridState_eq, 230
rxdock::isInteractionCenterSelected, 231
rxdock::isInteractionD_lt, 231
rxdock::isPiAtom, 231
rxdock::isSegmentName_eq, 232
rxdock::isSubunitId_eq, 232
rxdock::isSubunitName_eq, 233
rxdock::LigandError, 233
rxdock::LigandFlexData, 234
  Accept, 235
rxdock::LigandSiteMapper, 236
  _VOL_INCR, 238
  operator(), 237
rxdock::LogCommand, 238
  Execute, 239
  GetName, 239
  GetNArgs, 239
  SetArg, 239
  SetNameArg, 239
rxdock::MCStats, 240
rxdock::MdlFileSink, 240
  Render, 242
rxdock::MdlFileSource, 242
  isAtomListSupported, 244
  isBondListSupported, 244
  isCoordinatesSupported, 244
  isDataSupported, 245
  isTitleListSupported, 245
  Parse, 245
rxdock::Model, 245
rxdock::ModelError, 247
rxdock::ModelMutator, 248
rxdock::MOEGrid, 249
rxdock::MOEGridPoint, 249
rxdock::MOEGridShape, 249
rxdock::MOL2FileSource, 250
  _TRIPOS_DELIM_SIZE, 253
  isAtomListSupported, 252
  isBondListSupported, 252
  isCoordinatesSupported, 252
  isDataSupported, 253
  isTitleListSupported, 253
  Parse, 253
rxdock::MOL2Substructure, 253
rxdock::MulCommand, 254
  Execute, 255
  GetName, 255
  GetNArgs, 255
  SetArg, 255
  SetNameArg, 255
rxdock::NegChargeConstraint, 255
  AddAtomList, 256
rxdock::neldermead::AndCriteria< Criteria1, Criteria2 >, 26
rxdock::neldermead::IterationCriterion, 233
rxdock::neldermead::RelativeValueCriterion< DataType >, 336
rxdock::neldermead::Simplex< DataType, ParameterType, Function, Criterion >, 371
  GetBestParameters, 372
  GetBestValue, 372
rxdock::neldermead::State< DataType, ParameterType >, 384
rxdock::NmrRestraintFileSource, 257
rxdock::NmrSF, 258
  ParameterUpdated, 261
  RawScore, 261
  SetupLigand, 261
  SetupReceptor, 261
  SetupScore, 261
rxdock::NoeEndAtoms, 262
rxdock::NoeEndNames, 262
rxdock::NoeRestraintAtoms, 263
rxdock::NoeRestraintNames, 263
rxdock::NoFileInDir, 264
rxdock::NonBondedGrid, 265
  Print, 267
rxdock::NonBondedHHSGrid, 267
  Print, 269
rxdock::NullTransform, 269
  Execute, 271
  Update, 271
rxdock::Observer, 272
  Update, 272
rxdock::OverlapHHS, 273
rxdock::OverlapVariableHHS, 274
rxdock::ParameterFileSource, 274
rxdock::ParamHandler, 275

rxdock::Parser, 276
rxdock::PartitionHHS, 276
rxdock::PdbFileSource, 277
isAtomListSupported, 279
isBondListSupported, 279
isCoordinatesSupported, 279
isTitleListSupported, 279
Parse, 279
rxdock::PharmaSF, 280
ParameterUpdated, 283
RawScore, 283
ScoreMap, 283
SetupLigand, 283
SetupReceptor, 283
SetupScore, 283
rxdock::Plane, 284
rxdock::PMFDirSource, 284
rxdock::PMFGridSF, 285
RawScore, 288
SetupLigand, 288
SetupReceptor, 288
SetupScore, 288
rxdock::PMFIdxSF, 289
_CT, 294
_PMFDIR, 294
∼PMFIdxSF, 292
GetLinearCloseRangeValue, 293
PMFIdxSF, 292
RawScore, 293
SetupLigand, 293
SetupReceptor, 293
SetupScore, 293
Update, 293
rxdock::PMFValue, 294
rxdock::PolarIdxSF, 295
ParameterUpdated, 299
RawScore, 299
ScoreMap, 299
SetupLigand, 300
SetupReceptor, 300
SetupScore, 300
SetupSolvent, 300
rxdock::PolarIntraSF, 300
HandleRequest, 304
ParameterUpdated, 304
RawScore, 304
SetupScore, 304
rxdock::PolarSF, 305
rxdock::PolarSF::f1prms, 145
rxdock::Population, 309
rxdock::PosChargeConstraint, 309
AddAtomList, 310
rxdock::PrettyPrintVisitor, 311
VisitAddExp, 311
VisitAndExp, 311
VisitDivExp, 311
VisitExpExp, 312
VisitIfExp, 312
VisitLogExp, 312
VisitMulExp, 312
VisitSubExp, 312
VisitVbleExp, 312
rxdock::PrincipalAxes, 313
rxdock::PrintVisitor, 313
VisitAddExp, 314
VisitAndExp, 314
VisitDivExp, 314
VisitExpExp, 314
VisitIfExp, 315
VisitLogExp, 315
VisitMulExp, 315
VisitSubExp, 315
VisitVbleExp, 315
rxdock::PRMFactory, 316
_REC_DIHEDRAL_STEP, 316
_REC_NUM_COORD_FILES, 316
rxdock::PseudoAtom, 317
Print, 319
rxdock::PsfFileSink, 319
Render, 321
rxdock::PsfFileSource, 321
isAtomListSupported, 323
isBondListSupported, 323
isCoordinatesSupported, 323
isTitleListSupported, 323
Parse, 323
rxdock::Quat, 324
rxdock::Rand, 325
rxdock::RandLigTransform, 325
Execute, 328
SetupTransform, 328
rxdock::RandPopTransform, 328
_SCALE_CHROM_LENGTH, 331
Execute, 331
SetupLigand, 331
SetupReceptor, 331
SetupSolvent, 331
SetupTransform, 331
rxdock::RealGrid, 332
Print, 334
rxdock::ReceptorFlexData, 334
Accept, 335
rxdock::Request, 336
rxdock::RequestHandler, 337
rxdock::RestoreHHS, 337
rxdock::RingAromaticConstraint, 337
AddAtomList, 338
rxdock::RotateAtomUsingQuat, 339
rxdock::RotateAtomUsingQuatIfSelected, 339
rxdock::RotSF, 339
ParameterUpdated, 342
RawScore, 342
SetupLigand, 342
SetupReceptor, 342
SetupScore, 342
rxdock::SAIdxSF, 343

HandleRequest, 346
RawScore, 346
ScoreMap, 346
SetupLigand, 347
SetupReceptor, 347
SetupScore, 347
SetupSolvent, 347
rxdock::SaveHHS, 347
rxdock::SelectAtom, 348
rxdock::SelectFlexAtoms, 348
rxdock::SelectInteractionCenter, 349
rxdock::SetupPMFSF, 349
GetPMFfor_IC, 352
RawScore, 352
SetupLigand, 352
SetupReceptor, 352
SetupScore, 352
rxdock::SetupPolarSF, 353
RawScore, 355
SetupLigand, 355
SetupReceptor, 356
SetupScore, 356
SetupSolvent, 356
rxdock::SetupSASF, 356
RawScore, 359
SetupLigand, 359
SetupReceptor, 359
SetupScore, 360
rxdock::SFAgg, 360
Add, 362
GetNumSF, 362
GetSF, 363
HandleRequest, 363
isAgg, 363
Print, 363
RawScore, 363
Register, 363
Remove, 363
ScoreMap, 364
Unregister, 364
Update, 364
rxdock::SFDisableRequest, 364
rxdock::SFEnableRequest, 365
rxdock::SFFactory, 365
rxdock::SFPartitionRequest, 366
rxdock::SFSetParamRequest, 366
rxdock::SimAnnTransform, 367
_MIN_ACC_RATE, 370
_SCALE_CHROM_LENGTH, 370
Execute, 370
SetupLigand, 370
SetupReceptor, 370
SetupTransform, 370
rxdock::SimpleSolvationParameters, 371
rxdock::SimplexCostFunction, 372
rxdock::SimplexTransform, 373
_STOPPING_STEP_LENGTH, 376
Execute, 375
SetupLigand, 375
SetupReceptor, 376
SetupSolvent, 376
SetupTransform, 376
rxdock::Singleton< singleton_type >, 376
rxdock::SiteMapper, 377
Update, 379
rxdock::SiteMapperFactory, 379
rxdock::SmartPtr< T >, 379
rxdock::SolventFlexData, 380
Accept, 381
rxdock::SphereSiteMapper, 382
_VOL_INCR, 384
operator(), 383
rxdock::StdRestraintAtoms, 385
rxdock::StdRestraintNames, 385
rxdock::StringContext, 386
Assign, 386
GetVble, 387
SetVble, 387
rxdock::StringTokenIter, 387
Current, 388
Next, 388
rxdock::StringTooLong, 388
rxdock::SubCommand, 389
Execute, 390
GetName, 390
GetNArgs, 390
SetArg, 390
SetNameArg, 391
rxdock::Subject, 391
rxdock::SymBond, 391
rxdock::TetherSF, 392
ParameterUpdated, 395
RawScore, 395
SetupLigand, 395
SetupReceptor, 395
SetupScore, 395
rxdock::Token, 396
rxdock::TokenIter, 396
rxdock::TransformAgg, 397
Add, 399
Execute, 399
GetNumTransforms, 399
GetTransform, 399
HandleRequest, 399
isAgg, 400
Print, 400
Register, 400
Remove, 400
Unregister, 400
Update, 400
rxdock::TransformFactory, 401
rxdock::TranslateAtom, 401
rxdock::TranslateAtomIfSelected, 401
rxdock::TriposAtomType, 401
rxdock::TriposAtomType::info, 203
rxdock::unittest::ChromTest, 106

rxdock::unittest::OccupancyTest, 273
rxdock::unittest::SearchTest, 348
rxdock::Variant, 402
rxdock::Vble, 403
rxdock::VdwGridSF, 404
ParameterUpdated, 407
RawScore, 407
SetupLigand, 407
SetupReceptor, 407
SetupScore, 407
SetupSolvent, 407
rxdock::VdwIdxSF, 408
_ANNOTATION_LIPO, 413
ParameterUpdated, 412
RawScore, 412
ScoreMap, 412
SetupLigand, 412
SetupReceptor, 412
SetupScore, 412
SetupSolvent, 412
rxdock::VdwIntraSF, 413
HandleRequest, 417
ParameterUpdated, 417
RawScore, 417
SetupScore, 417
rxdock::VdwSF, 417
rxdock::WorkSpace, 421

SAIdxSF.h, 602
SATypes.h, 603
Save
rxdock::BiMolWorkSpace, 71
SaveHistory
rxdock::BiMolWorkSpace, 71
ScoreMap
rxdock::BaseIntraSF, 52
rxdock::ConstSF, 113
rxdock::PharmaSF, 283
rxdock::PolarIdxSF, 299
rxdock::SAIdxSF, 346
rxdock::SFAgg, 364
rxdock::VdwIdxSF, 412
SearchTest.h, 647
SetArg
rxdock::AddCommand, 22
rxdock::DivCommand, 135
rxdock::ExpCommand, 145
rxdock::IfCommand, 202
rxdock::LogCommand, 239
rxdock::MulCommand, 255
rxdock::SubCommand, 390
SetNameArg
rxdock::AddCommand, 22
rxdock::DivCommand, 135
rxdock::ExpCommand, 145
rxdock::IfCommand, 202
rxdock::LogCommand, 239
rxdock::MulCommand, 255
rxdock::SubCommand, 391
SetupLigand
rxdock::AlignTransform, 25
rxdock::AromIdxSF, 32
rxdock::BaseInterSF, 49
rxdock::CavityFillSF, 76
rxdock::CavityGridSF, 80
rxdock::ConstSF, 113
rxdock::DihedralTargetSF, 131
rxdock::GATransform, 176
rxdock::NmrSF, 261
rxdock::PharmaSF, 283
rxdock::PMFGridSF, 288
rxdock::PMFIdxSF, 293
rxdock::PolarIdxSF, 300
rxdock::RandPopTransform, 331
rxdock::RotSF, 342
rxdock::SAIdxSF, 347
rxdock::SetupPMFSF, 352
rxdock::SetupPolarSF, 355
rxdock::SetupSASF, 359
rxdock::SimAnnTransform, 370
rxdock::SimplexTransform, 375
rxdock::TetherSF, 395
rxdock::VdwGridSF, 407
rxdock::VdwIdxSF, 412
SetupPMFSF.h, 607
SetupPolarSF.h, 608
SetupReceptor
rxdock::AlignTransform, 25
rxdock::AromIdxSF, 32
rxdock::BaseInterSF, 49
rxdock::CavityFillSF, 76
rxdock::CavityGridSF, 80
rxdock::ConstSF, 113
rxdock::DihedralTargetSF, 131
rxdock::GATransform, 177
rxdock::NmrSF, 261
rxdock::PharmaSF, 283
rxdock::PMFGridSF, 288
rxdock::PMFIdxSF, 293
rxdock::PolarIdxSF, 300
rxdock::RandPopTransform, 331
rxdock::RotSF, 342
rxdock::SAIdxSF, 347
rxdock::SetupPMFSF, 352
rxdock::SetupPolarSF, 356
rxdock::SetupSASF, 359
rxdock::SimAnnTransform, 370
rxdock::SimplexTransform, 376
rxdock::TetherSF, 395
rxdock::VdwGridSF, 407
rxdock::VdwIdxSF, 412
SetupSASF.h, 608
SetupScore
rxdock::AromIdxSF, 32
rxdock::BaseInterSF, 49
rxdock::CavityFillSF, 76
rxdock::CavityGridSF, 80

rxdock::ConstSF, 113
rxdock::DihedralIntraSF, 125
rxdock::DihedralTargetSF, 131
rxdock::NmrSF, 261
rxdock::PharmaSF, 283
rxdock::PMFGridSF, 288
rxdock::PMFIdxSF, 293
rxdock::PolarIdxSF, 300
rxdock::PolarIntraSF, 304
rxdock::RotSF, 342
rxdock::SAIdxSF, 347
rxdock::SetupPMFSF, 352
rxdock::SetupPolarSF, 356
rxdock::SetupSASF, 360
rxdock::TetherSF, 395
rxdock::VdwGridSF, 407
rxdock::VdwIdxSF, 412
rxdock::VdwIntraSF, 417
SetupSolvent
rxdock::CavityGridSF, 80
rxdock::PolarIdxSF, 300
rxdock::RandPopTransform, 331
rxdock::SAIdxSF, 347
rxdock::SetupPolarSF, 356
rxdock::SimplexTransform, 376
rxdock::VdwGridSF, 407
rxdock::VdwIdxSF, 412
SetupTransform
rxdock::AlignTransform, 26
rxdock::GATransform, 177
rxdock::RandLigTransform, 328
rxdock::RandPopTransform, 331
rxdock::SimAnnTransform, 370
rxdock::SimplexTransform, 376
SetVble
rxdock::CellContext, 83
rxdock::StringContext, 387
SetVector
rxdock::Chrom, 90
rxdock::ChromDihedralElement, 94
rxdock::ChromOccupancyElement, 100
rxdock::ChromPositionElement, 105
SFAgg.h, 609
SFFactory.h, 611
SFRequest.h, 612
SimAnnTransform.h, 613
SimplexCostFunction.h, 614
SimplexTransform.h, 615
Singleton.h, 616
SiteMapper.h, 616
SiteMapperFactory.h, 617
SmartPointer.h, 618
Smarts.h, 495
SolventFlexData.h, 621
SphereSiteMapper.h, 621
StringTokenIter.h, 622
Subject.h, 623
SyncFromModel
rxdock::Chrom, 90
rxdock::ChromDihedralElement, 94
rxdock::ChromOccupancyElement, 101
rxdock::ChromPositionElement, 105
SyncToModel
rxdock::Chrom, 90
rxdock::ChromDihedralElement, 94
rxdock::ChromOccupancyElement, 101
rxdock::ChromPositionElement, 105

Tabularize.cxx, 645
Tabularize.h, 567, 568
TetherSF.h, 626
Token.h, 627
TokenIter.h, 628
Transform.cxx, 645
Transform.h, 568, 569
TransformAgg.h, 629
TransformFactory.h, 630
TriposAtomType.h, 631

Unregister
rxdock::SFAgg, 364
rxdock::TransformAgg, 400
Update
rxdock::BaseBiMolTransform, 40
rxdock::BaseInterSF, 50
rxdock::BaseIntraSF, 52
rxdock::BaseUniMolTransform, 69
rxdock::Filter, 156
rxdock::NullTransform, 271
rxdock::Observer, 272
rxdock::PMFIdxSF, 293
rxdock::SFAgg, 364
rxdock::SiteMapper, 379
rxdock::TransformAgg, 400

Variant.h, 633
Vble.h, 635
VdwGridSF.h, 636
VdwIdxSF.h, 637
VdwIntraSF.h, 639
VdwSF.h, 639
VisitAddExp
rxdock::EvaluateVisitor, 142
rxdock::PrettyPrintVisitor, 311
rxdock::PrintVisitor, 314
VisitAndExp
rxdock::EvaluateVisitor, 142
rxdock::PrettyPrintVisitor, 311
rxdock::PrintVisitor, 314
VisitDivExp
rxdock::EvaluateVisitor, 142
rxdock::PrettyPrintVisitor, 311
rxdock::PrintVisitor, 314
VisitExpExp
rxdock::EvaluateVisitor, 142
rxdock::PrettyPrintVisitor, 312
rxdock::PrintVisitor, 314

VisitIfExp
rxdock::EvaluateVisitor, 143
rxdock::PrettyPrintVisitor, 312
rxdock::PrintVisitor, 315
VisitLigandFlexData
rxdock::CavityGridSF::HeavyAtomFactory, 193
rxdock::ChromFactory, 97
rxdock::FlexAtomFactory, 172
VisitLogExp
rxdock::EvaluateVisitor, 143
rxdock::PrettyPrintVisitor, 312
rxdock::PrintVisitor, 315
VisitMulExp
rxdock::EvaluateVisitor, 143
rxdock::PrettyPrintVisitor, 312
rxdock::PrintVisitor, 315
VisitReceptorFlexData
rxdock::CavityGridSF::HeavyAtomFactory, 193
rxdock::ChromFactory, 97
rxdock::FlexAtomFactory, 172
VisitSolventFlexData
rxdock::CavityGridSF::HeavyAtomFactory, 194
rxdock::ChromFactory, 97
rxdock::FlexAtomFactory, 172
VisitSubExp
rxdock::EvaluateVisitor, 143
rxdock::PrettyPrintVisitor, 312
rxdock::PrintVisitor, 315
VisitVbleExp
rxdock::EvaluateVisitor, 143
rxdock::PrettyPrintVisitor, 312
rxdock::PrintVisitor, 315

WorkSpace.h, 642