

nMOLDYN

API Documentation

October 8, 2009

Contents

Contents	1
1 Package nMOLDYN	9
1.1 Modules	9
1.2 Variables	11
2 Package nMOLDYN.Analysis	12
2.1 Modules	12
3 Module nMOLDYN.Analysis.Analysis	13
3.1 Functions	13
3.2 Variables	13
3.3 Class Analysis	13
3.3.1 Methods	13
3.4 Class QVectors	16
3.4.1 Methods	17
4 Module nMOLDYN.Analysis.Dynamics	18
4.1 Class MeanSquareDisplacement	18
4.1.1 Methods	19
4.1.2 Class Variables	20
4.2 Class RootMeanSquareDeviation	20
4.2.1 Methods	21
4.2.2 Class Variables	22
4.3 Class CartesianVelocityAutoCorrelationFunction	22
4.3.1 Methods	24
4.3.2 Class Variables	24
4.4 Class CartesianDensityOfStates	25
4.4.1 Methods	26
4.4.2 Class Variables	27
4.5 Class AutoRegressiveAnalysis	27
4.5.1 Methods	28
4.5.2 Class Variables	29
4.6 Class PassBandFilteredTrajectory	29
4.6.1 Methods	30
4.6.2 Class Variables	31
4.7 Class RadiusOfGyration	31

4.7.1	Methods	32
4.7.2	Class Variables	32
4.8	Class GlobalMotionFilteredTrajectory	33
4.8.1	Methods	34
4.8.2	Class Variables	34
4.9	Class CenterOfMassTrajectory	35
4.9.1	Methods	36
4.9.2	Class Variables	36
4.10	Class QuasiHarmonicAnalysis	37
4.10.1	Methods	38
4.10.2	Class Variables	38
4.11	Class AngularCorrelation	38
4.11.1	Methods	39
4.11.2	Class Variables	40
4.12	Class RigidBodyTrajectory	40
4.12.1	Methods	41
4.12.2	Class Variables	42
4.13	Class ReorientationalCorrelationFunction	42
4.13.1	Methods	43
4.13.2	Class Variables	44
4.14	Class AngularVelocity	44
4.14.1	Methods	45
4.15	Class AngularVelocityAutoCorrelationFunction	45
4.15.1	Methods	46
4.15.2	Class Variables	47
4.16	Class AngularDensityOfStates	47
4.16.1	Methods	48
4.16.2	Class Variables	49
5	Module nMOLDYN.Analysis.NMR	50
5.1	Class OrderParameter	50
5.1.1	Methods	51
5.1.2	Class Variables	52
5.2	Class OrderParameterContactModel	52
5.2.1	Methods	53
5.2.2	Class Variables	54
6	Module nMOLDYN.Analysis.Scattering	55
6.1	Functions	55
6.2	Class DynamicCoherentStructureFactor	55
6.2.1	Methods	57
6.2.2	Class Variables	57
6.3	Class StaticCoherentStructureFactor	58
6.3.1	Methods	59
6.3.2	Class Variables	60
6.4	Class DynamicCoherentStructureFactorAR	60
6.4.1	Methods	61
6.4.2	Class Variables	62
6.5	Class DynamicIncoherentStructureFactor	62
6.5.1	Methods	64
6.5.2	Class Variables	64
6.6	Class DynamicIncoherentStructureFactorAR	65

6.6.1	Methods	66
6.6.2	Class Variables	67
6.7	Class DynamicIncoherentStructureFactorGaussian	67
6.7.1	Methods	68
6.7.2	Class Variables	69
6.8	Class ElasticIncoherentStructureFactor	69
6.8.1	Methods	70
6.8.2	Class Variables	71
6.9	Class SmoothedStaticCoherentStructureFactor	71
6.9.1	Methods	73
6.9.2	Class Variables	73
7	Module nMOLDYN.Analysis.Slave	75
7.1	Functions	75
8	Module nMOLDYN.Analysis.Structure	76
8.1	Class PairDistributionFunction	76
8.1.1	Methods	77
8.1.2	Class Variables	78
8.2	Class CoordinationNumber	78
8.2.1	Methods	79
8.2.2	Class Variables	80
8.3	Class ScrewFitAnalysis	80
8.3.1	Methods	81
8.3.2	Class Variables	82
8.4	Class SpatialDensity	83
8.4.1	Methods	84
8.4.2	Class Variables	85
9	Module nMOLDYN.Analysis.Template	86
9.1	Functions	86
9.2	Variables	86
9.3	Class SerialPerAtom	86
9.3.1	Methods	86
9.4	Class ParallelPerAtom	86
9.4.1	Methods	87
9.5	Class SerialPerFrame	87
9.5.1	Methods	87
9.6	Class ParallelPerFrame	87
9.6.1	Methods	87
9.7	Class SerialPerGroup	87
9.7.1	Methods	87
9.8	Class ParallelPerGroup	87
9.8.1	Methods	88
9.9	Class SerialPerQShell	88
9.9.1	Methods	88
9.10	Class ParallelPerQShell	88
9.10.1	Methods	88
9.11	Class MeanSquareDisplacement_serial	88
9.11.1	Methods	89
9.11.2	Class Variables	89
9.12	Class MeanSquareDisplacement_parallel	89

9.12.1	Methods	89
9.12.2	Class Variables	90
9.13	Class RootMeanSquareDeviation_serial	90
9.13.1	Methods	90
9.13.2	Class Variables	91
9.14	Class RootMeanSquareDeviation_parallel	91
9.14.1	Methods	91
9.14.2	Class Variables	91
9.15	Class CartesianVelocityAutoCorrelationFunction_serial	92
9.15.1	Methods	92
9.15.2	Class Variables	92
9.16	Class CartesianVelocityAutoCorrelationFunction_parallel	93
9.16.1	Methods	93
9.16.2	Class Variables	93
9.17	Class CartesianDensityOfStates_serial	94
9.17.1	Methods	94
9.17.2	Class Variables	94
9.18	Class CartesianDensityOfStates_parallel	95
9.18.1	Methods	95
9.18.2	Class Variables	95
9.19	Class AutoRegressiveAnalysis_serial	96
9.19.1	Methods	96
9.19.2	Class Variables	96
9.20	Class AutoRegressiveAnalysis_parallel	97
9.20.1	Methods	97
9.20.2	Class Variables	97
9.21	Class PassBandFilteredTrajectory_serial	98
9.21.1	Methods	98
9.21.2	Class Variables	98
9.22	Class PassBandFilteredTrajectory_parallel	99
9.22.1	Methods	99
9.22.2	Class Variables	99
9.23	Class DynamicIncoherentStructureFactor_serial	100
9.23.1	Methods	100
9.23.2	Class Variables	100
9.24	Class DynamicIncoherentStructureFactor_parallel	101
9.24.1	Methods	101
9.24.2	Class Variables	101
9.25	Class DynamicIncoherentStructureFactorGaussian_serial	102
9.25.1	Methods	102
9.25.2	Class Variables	102
9.26	Class DynamicIncoherentStructureFactorGaussian_parallel	103
9.26.1	Methods	103
9.26.2	Class Variables	103
9.27	Class ElasticIncoherentStructureFactor_serial	104
9.27.1	Methods	104
9.27.2	Class Variables	104
9.28	Class ElasticIncoherentStructureFactor_parallel	105
9.28.1	Methods	105
9.28.2	Class Variables	105
9.29	Class RadiusOfGyration_serial	106

9.29.1	Methods	106
9.29.2	Class Variables	106
9.30	Class RadiusOfGyration_parallel	107
9.30.1	Methods	107
9.30.2	Class Variables	107
9.31	Class DynamicIncoherentStructureFactorAR_serial	108
9.31.1	Methods	108
9.31.2	Class Variables	108
9.32	Class DynamicIncoherentStructureFactorAR_parallel	109
9.32.1	Methods	109
9.32.2	Class Variables	109
9.33	Class SmoothedStaticCoherentStructureFactor_serial	110
9.33.1	Methods	110
9.33.2	Class Variables	110
9.34	Class SmoothedStaticCoherentStructureFactor_parallel	111
9.34.1	Methods	111
9.34.2	Class Variables	111
9.35	Class PairDistributionFunction_serial	112
9.35.1	Methods	112
9.35.2	Class Variables	112
9.36	Class PairDistributionFunction_parallel	113
9.36.1	Methods	113
9.36.2	Class Variables	113
9.37	Class CoordinationNumber_serial	114
9.37.1	Methods	114
9.37.2	Class Variables	114
9.38	Class CoordinationNumber_parallel	115
9.38.1	Methods	115
9.38.2	Class Variables	115
9.39	Class ScrewFitAnalysis_serial	116
9.39.1	Methods	116
9.39.2	Class Variables	116
9.40	Class ScrewFitAnalysis_parallel	117
9.40.1	Methods	117
9.40.2	Class Variables	117
9.41	Class SpatialDensity_serial	118
9.41.1	Methods	118
9.41.2	Class Variables	118
9.42	Class SpatialDensity_parallel	119
9.42.1	Methods	119
9.42.2	Class Variables	119
9.43	Class GlobalMotionFilteredTrajectory_serial	120
9.43.1	Methods	120
9.43.2	Class Variables	120
9.44	Class GlobalMotionFilteredTrajectory_parallel	121
9.44.1	Methods	121
9.44.2	Class Variables	121
9.45	Class CenterOfMassTrajectory_serial	122
9.45.1	Methods	122
9.45.2	Class Variables	122
9.46	Class CenterOfMassTrajectory_parallel	123

9.46.1	Methods	123
9.46.2	Class Variables	123
9.47	Class OrderParameterContactModel_serial	124
9.47.1	Methods	124
9.47.2	Class Variables	124
9.48	Class OrderParameterContactModel_parallel	125
9.48.1	Methods	125
9.48.2	Class Variables	125
9.49	Class AngularCorrelation_serial	126
9.49.1	Methods	126
9.49.2	Class Variables	126
9.50	Class AngularCorrelation_parallel	127
9.50.1	Methods	127
9.50.2	Class Variables	127
9.51	Class RigidBodyTrajectory_serial	128
9.51.1	Methods	128
9.51.2	Class Variables	128
9.52	Class RigidBodyTrajectory_parallel	129
9.52.1	Methods	129
9.52.2	Class Variables	129
9.53	Class ReorientationalCorrelationFunction_serial	130
9.53.1	Methods	130
9.53.2	Class Variables	130
9.54	Class ReorientationalCorrelationFunction_parallel	131
9.54.1	Methods	131
9.54.2	Class Variables	131
9.55	Class AngularVelocityAutoCorrelationFunction_serial	132
9.55.1	Methods	132
9.55.2	Class Variables	132
9.56	Class AngularVelocityAutoCorrelationFunction_parallel	133
9.56.1	Methods	133
9.56.2	Class Variables	133
9.57	Class AngularDensityOfStates_serial	134
9.57.1	Methods	134
9.57.2	Class Variables	134
9.58	Class AngularDensityOfStates_parallel	135
9.58.1	Methods	135
9.58.2	Class Variables	135
9.59	Class OrderParameter_serial	136
9.59.1	Methods	136
9.59.2	Class Variables	136
9.60	Class OrderParameter_parallel	137
9.60.1	Methods	137
9.60.2	Class Variables	137
9.61	Class QuasiHarmonicAnalysis_serial	137
9.61.1	Methods	138
9.61.2	Class Variables	138
9.62	Class DynamicCoherentStructureFactor_serial	138
9.62.1	Methods	138
9.62.2	Class Variables	139
9.63	Class DynamicCoherentStructureFactor_parallel	139

9.63.1	Methods	139
9.63.2	Class Variables	139
9.64	Class StaticCoherentStructureFactor_serial	140
9.64.1	Methods	140
9.64.2	Class Variables	140
9.65	Class StaticCoherentStructureFactor_parallel	141
9.65.1	Methods	141
9.65.2	Class Variables	141
9.66	Class DynamicCoherentStructureFactorAR_serial	142
9.66.1	Methods	142
9.66.2	Class Variables	142
9.67	Class DynamicCoherentStructureFactorAR_parallel	143
9.67.1	Methods	143
9.67.2	Class Variables	143
10	Package nMOLDYN.Core	144
10.1	Modules	144
11	Module nMOLDYN.Core.Chemistry	145
11.1	Functions	145
12	Module nMOLDYN.Core.Config	146
12.1	Functions	146
13	Module nMOLDYN.Core.Error	147
13.1	Class Error	147
13.1.1	Methods	147
13.1.2	Properties	147
14	Module nMOLDYN.Core.IOFiles	148
14.1	Functions	148
14.2	Class TemporaryFile	149
14.2.1	Methods	149
14.3	Class EndOfFile	150
14.3.1	Methods	150
14.3.2	Properties	150
14.4	Class FortranBinaryFile	150
14.4.1	Methods	151
14.4.2	Properties	151
14.5	Class DCDFFile	151
14.5.1	Methods	152
14.5.2	Properties	152
14.6	Class AmberNetCDFConverter	152
14.6.1	Methods	153
14.7	Class CHARMMConverter	153
14.7.1	Methods	154
14.8	Class DL_POLYConverter	154
14.8.1	Methods	154
14.9	Class MaterialsStudioConverter	155
14.9.1	Methods	155
14.9.2	Class Variables	156
14.10	Class NAMDCConverter	156

14.10.1 Methods	156
14.11 Class VASPCConverter	156
14.11.1 Methods	157
15 Module nMOLDYN.Core.Logger	158
15.1 Functions	158
15.2 Variables	158
15.3 Class LogToGUI	159
15.3.1 Methods	159
15.4 Class LogToFile	160
15.4.1 Methods	160
15.5 Class LogToConsole	161
15.5.1 Methods	161
16 Module nMOLDYN.Core.Mathematics	163
16.1 Functions	163
16.2 Variables	168
17 Module nMOLDYN.Core.Misc	170
17.1 Functions	170
17.2 Variables	171
18 Module nMOLDYN.Core.Preferences	172
18.1 Variables	172
18.2 Class nMOLDYNPreferences	172
18.2.1 Methods	172
18.2.2 Properties	173
18.2.3 Class Variables	173
19 Package nMOLDYN.GUI	174
19.1 Modules	174
20 Module nMOLDYN.GUI.ASCIIToNetCDFConversionDialog	176
20.1 Class ASCIIToNetCDFConversionDialog	176
20.1.1 Methods	176
21 Module nMOLDYN.GUI.AnalysisBenchmarkDialog	178
21.1 Class AnalysisBenchmarkDialog	178
21.1.1 Methods	178
22 Module nMOLDYN.GUI.AnalysisDialog	180
22.1 Class AnalysisDialog	180
22.1.1 Methods	180
23 Module nMOLDYN.GUI.AnimationDialog	182
23.1 Class AnimationDialog	182
23.1.1 Methods	182
23.1.2 Class Variables	184
24 Module nMOLDYN.GUI.CheckJobsStatusDialog	185
24.1 Variables	185
24.2 Class CheckJobsStatusDialog	185
24.2.1 Methods	185

24.2.2 Class Variables	187
25 Module nMOLDYN.GUI.GeneralInformationsDialog	188
25.1 Variables	188
25.2 Class GeneralInformationsDialog	188
25.2.1 Methods	189
25.2.2 Class Variables	190
26 Package nMOLDYN.GUI.HTMLReader	191
26.1 Modules	191
27 Module nMOLDYN.GUI.HTMLReader.mfxtools	192
27.1 Functions	192
27.2 Variables	192
28 Module nMOLDYN.GUI.HTMLReader.mfxutil	193
28.1 Functions	193
28.2 Variables	194
28.3 Class SubclassResponsibility	194
28.3.1 Methods	194
28.3.2 Properties	195
28.4 Class Struct	195
28.4.1 Methods	195
28.5 Class KwStruct	195
28.5.1 Methods	195
29 Module nMOLDYN.GUI.HTMLReader.tkconst	197
29.1 Variables	197
30 Module nMOLDYN.GUI.HTMLReader.tkfont	198
30.1 Functions	198
30.2 Variables	198
31 Module nMOLDYN.GUI.HTMLReader.tkhtml	199
31.1 Functions	199
31.2 Class MfxScrolledText	199
31.2.1 Methods	200
31.2.2 Class Variables	202
31.3 Class MfxReadOnlyScrolledText	203
31.3.1 Methods	204
31.3.2 Class Variables	205
31.4 Class tkHTMLWriter	206
31.4.1 Methods	206
31.5 Class tkHTMLParser	207
31.5.1 Methods	207
31.5.2 Class Variables	208
31.6 Class tkHTMLViewer	209
31.6.1 Methods	209
32 Module nMOLDYN.GUI.HTMLReader.tkinit	211
32.1 Functions	211
33 Module nMOLDYN.GUI.HTMLReader.tkutil	213

33.1 Functions	213
34 Module nMOLDYN.GUI.HTMLReader.tkwidget	214
34.1 Functions	214
34.2 Class MfxDialog	214
34.2.1 Methods	214
34.2.2 Class Variables	214
34.3 Class MfxExceptionDialog	215
34.3.1 Methods	215
34.3.2 Class Variables	215
34.4 Class MfxSimpleSlider	215
34.4.1 Methods	215
34.4.2 Class Variables	216
34.5 Class MfxSimpleEntry	216
34.5.1 Methods	216
34.5.2 Class Variables	217
34.6 Class MfxTooltip	217
34.6.1 Methods	217
34.7 Class MfxScrolledCanvas	217
34.7.1 Methods	217
35 Module nMOLDYN.GUI.HTMLReader.util	219
35.1 Functions	219
35.2 Variables	219
35.3 Class Timer	219
35.3.1 Methods	219
35.4 Class DataLoader	220
35.4.1 Methods	220
36 Module nMOLDYN.GUI.HTMLReader.version	221
36.1 Variables	221
37 Module nMOLDYN.GUI.MainDialog	222
37.1 Variables	222
37.2 Class MainDialog	222
37.2.1 Methods	222
37.2.2 Class Variables	225
38 Module nMOLDYN.GUI.NetCDFToASCIIConversionDialog	226
38.1 Class NetCDFToASCIIConversionDialog	226
38.1.1 Methods	226
39 Module nMOLDYN.GUI.PDBSnapshotGeneratorDialog	228
39.1 Class PDBSnapshotGeneratorDialog	228
39.1.1 Methods	228
40 Module nMOLDYN.GUI.PlotNetCDFVariableDialog	230
40.1 Variables	230
40.2 Class SettingsDialog	230
40.2.1 Methods	230
40.3 Class PlotNetCDFVariableDialog	231
40.3.1 Methods	232

41 Module nMOLDYN.GUI.PreferencesDialog	234
41.1 Class dummy	234
41.1.1 Methods	234
41.2 Class PreferencesDialog	234
41.2.1 Methods	234
42 Module nMOLDYN.GUI.PyroServerDialog	236
42.1 Class PyroServerDialog	236
42.1.1 Methods	236
42.1.2 Class Variables	238
43 Module nMOLDYN.GUI.SelectionDialog	239
43.1 Class SelectionDialog	239
43.1.1 Methods	239
43.1.2 Class Variables	243
44 Module nMOLDYN.GUI.Tags	244
44.1 Variables	244
45 Module nMOLDYN.GUI.TrajectoryConversionDialog	245
45.1 Class AmberNetCDFConverterDialog	245
45.1.1 Methods	245
45.2 Class CHARMMConverterDialog	246
45.2.1 Methods	246
45.3 Class DL_POLYConverterDialog	247
45.3.1 Methods	247
45.4 Class MaterialsStudioConverterDialog	248
45.4.1 Methods	248
45.5 Class NAMDCConverterDialog	249
45.5.1 Methods	249
45.6 Class VASPCConverterDialog	250
45.6.1 Methods	250
46 Module nMOLDYN.GUI.ViewEffectiveModeDialog	251
46.1 Class ViewEffectiveModeDialog	251
46.1.1 Methods	251
46.1.2 Class Variables	253
47 Module nMOLDYN.GUI.Widgets	254
47.1 Functions	255
47.2 Variables	255
47.3 Class ValidatingEntry	256
47.3.1 Methods	256
47.3.2 Class Variables	258
47.4 Class IntegerEntry	258
47.4.1 Methods	259
47.4.2 Class Variables	260
47.5 Class FloatEntry	261
47.5.1 Methods	261
47.5.2 Class Variables	263
47.6 Class StringEntry	263
47.6.1 Methods	264

47.6.2 Class Variables	265
47.7 Class ComboFrame	266
47.7.1 Methods	266
47.7.2 Class Variables	267
47.8 Class ComboLabel	268
47.8.1 Methods	268
47.8.2 Class Variables	270
47.9 Class ComboIntegerEntry	270
47.9.1 Methods	271
47.9.2 Class Variables	272
47.10 Class ComboFloatEntry	273
47.10.1 Methods	273
47.10.2 Class Variables	275
47.11 Class ComboStringEntry	275
47.11.1 Methods	276
47.11.2 Class Variables	277
47.12 Class ComboRadiobutton	278
47.12.1 Methods	279
47.12.2 Class Variables	280
47.13 Class ComboCheckbutton	281
47.13.1 Methods	282
47.13.2 Class Variables	284
47.14 Class ComboOptionMenu	284
47.14.1 Methods	285
47.14.2 Class Variables	286
47.15 Class ComboSpinbox	287
47.15.1 Methods	288
47.15.2 Class Variables	289
47.16 Class ComboScale	290
47.16.1 Methods	290
47.16.2 Class Variables	292
47.17 Class ComboButton	292
47.17.1 Methods	293
47.17.2 Class Variables	294
47.18 Class ComboFileBrowser	295
47.18.1 Methods	296
47.18.2 Class Variables	297
47.19 Class ComboText	298
47.19.1 Methods	298
47.19.2 Class Variables	300
47.20 Class ComboListbox	301
47.20.1 Methods	301
47.20.2 Class Variables	303
47.21 Class StatusBar	304
47.21.1 Methods	304
47.21.2 Class Variables	305
48 Package nMOLDYN.Tests	306
48.1 Modules	306
49 Package nMOLDYN.Tests.ARA	307
49.1 Modules	307

50 Module nMOLDYN.Tests.ARA.TestsContents	308
50.1 Variables	308
51 Module nMOLDYN.Tests.ARA.runTests	309
51.1 Variables	309
52 Package nMOLDYN.Tests.AVACF	310
52.1 Modules	310
53 Module nMOLDYN.Tests.AVACF.TestsContents	311
53.1 Variables	311
54 Module nMOLDYN.Tests.AVACF.runTests	312
54.1 Variables	312
55 Module nMOLDYN.Tests.AnalysisTests	313
55.1 Class AnalysisTest	313
55.1.1 Methods	313
56 Module nMOLDYN.Tests.BuildTestCases	314
56.1 Variables	314
57 Package nMOLDYN.Tests.DCSF	315
57.1 Modules	315
58 Module nMOLDYN.Tests.DCSF.TestsContents	316
58.1 Variables	316
59 Module nMOLDYN.Tests.DCSF.runTests	317
59.1 Variables	317
60 Package nMOLDYN.Tests.DISF	318
60.1 Modules	318
61 Module nMOLDYN.Tests.DISF.TestsContents	319
61.1 Variables	319
62 Module nMOLDYN.Tests.DISF.runTests	320
62.1 Variables	320
63 Package nMOLDYN.Tests.DISFG	321
63.1 Modules	321
64 Module nMOLDYN.Tests.DISFG.TestsContents	322
64.1 Variables	322
65 Module nMOLDYN.Tests.DISFG.runTests	323
65.1 Variables	323
66 Package nMOLDYN.Tests.DOS	324
66.1 Modules	324
67 Module nMOLDYN.Tests.DOS.TestsContents	325
67.1 Variables	325

68 Module nMOLDYN.Tests.DOS.runTests	326
68.1 Variables	326
69 Package nMOLDYN.Tests.EISF	327
69.1 Modules	327
70 Module nMOLDYN.Tests.EISF.TestsContents	328
70.1 Variables	328
71 Module nMOLDYN.Tests.EISF.runTests	329
71.1 Variables	329
72 Package nMOLDYN.Tests.MSD	330
72.1 Modules	330
73 Module nMOLDYN.Tests.MSD.TestsContents	331
73.1 Variables	331
74 Module nMOLDYN.Tests.MSD.runTests	332
74.1 Variables	332
75 Module nMOLDYN.Tests.StabilityTests	333
75.1 Variables	333
75.2 Class AnalysisTest	333
75.2.1 Methods	333
75.3 Class MSDTests	334
75.3.1 Methods	334
75.3.2 Class Variables	334
75.4 Class VACFTests	335
75.4.1 Methods	335
75.4.2 Class Variables	335
75.5 Class ARATests	336
75.5.1 Methods	336
75.5.2 Class Variables	336
75.6 Class AVACFTests	337
75.6.1 Methods	337
75.6.2 Class Variables	337
75.7 Class DOSTests	338
75.7.1 Methods	338
75.7.2 Class Variables	338
75.8 Class DCSFTests	339
75.8.1 Methods	339
75.8.2 Class Variables	339
75.9 Class DISFTests	340
75.9.1 Methods	340
75.9.2 Class Variables	340
75.10Class DISFGTests	341
75.10.1 Methods	341
75.10.2Class Variables	341
75.11Class EISFTests	342
75.11.1 Methods	342
75.11.2Class Variables	342
75.12Class ALLTests	343

75.12.1 Methods	343
75.12.2 Class Variables	343
76 Package nMOLDYN.Tests.VACF	344
76.1 Modules	344
77 Module nMOLDYN.Tests.VACF.TestsContents	345
77.1 Variables	345
78 Module nMOLDYN.Tests.VACF.runTests	346
78.1 Variables	346
79 Module nMOLDYN.__pkginfo__	347

1 Package nMOLDYN

1.1 Modules

- **Analysis** (*Section 2, p. 12*)
 - **Analysis**: This module implements the base class for all the analysis available in nMOLDYN. (*Section 3, p. 13*)
 - **Dynamics**: Collections of classes for the determination of dynamics-related properties. (*Section 4, p. 18*)
 - **NMR**: Collections of classes for the determination of NMR-related properties. (*Section 5, p. 50*)
 - **Scattering**: Collections of classes for the determination of scattering-related properties. (*Section 6, p. 55*)
 - **Slave**: This module contains the functions used by Pyro slave to perform analysis remotely. (*Section 7, p. 75*)
 - **Structure**: Collections of classes for the determination of structure-related properties. (*Section 8, p. 76*)
 - **Template**: This module implements the estimate, serial and parallel templates for all analysis. (*Section 9, p. 86*)
- **Core** (*Section 10, p. 144*)
 - **Chemistry**: This module implements the functions and procedures that are related to chemistry. (*Section 11, p. 145*)
 - **Config**: This module implements the procedures that handles nMOLDYN PREFERENCES. (*Section 12, p. 146*)
 - **Error** (*Section 13, p. 147*)
 - **IOFiles**: This module implements IO-related classes, functions and procedures. (*Section 14, p. 148*)
 - **Logger**: This module implements the classes used to handle the nMOLDYN logger. (*Section 15, p. 158*)
 - **Mathematics**: This module implements the mathematics-related classes, functions and procedures. (*Section 16, p. 163*)
 - **Misc**: This module implements the functions and procedures that can not be classified anywhere else in the library. (*Section 17, p. 170*)
 - **Preferences**: This module stores some the nMOLDYN PREFERENCES variables that will be used throughout all nMOLDYN code. (*Section 18, p. 172*)
- **GUI** (*Section 19, p. 174*)
 - **ASCIIToNetCDFConversionDialog**: This module implements I{File->Convert ASCII to NetCDF} dialog. (*Section 20, p. 176*)
 - **AnalysisBenchmarkDialog**: This module implements I{Help->nMOLDYN benchmark} dialog. (*Section 21, p. 178*)
 - **AnalysisDialog**: This module implements I{Analysis->selected analysis} dialog. (*Section 22, p. 180*)
 - **AnimationDialog**: This module implements I{View->Animation} dialog. (*Section 23, p. 182*)
 - **CheckJobsStatusDialog**: This module implements I{Help->Check job status} dialog. (*Section 24, p. 185*)

- **GeneralInformationsDialog**: This module implements I{Help->About nMOLDYN} dialog.
(Section 25, p. 188)
- **HTMLReader** (Section 26, p. 191)
 - * **mfxtools** (Section 27, p. 192)
 - * **mfxutil** (Section 28, p. 193)
 - * **tkconst** (Section 29, p. 197)
 - * **tkfont** (Section 30, p. 198)
 - * **tkhtml** (Section 31, p. 199)
 - * **tkinit** (Section 32, p. 211)
 - * **tkutil** (Section 33, p. 213)
 - * **tkwidget** (Section 34, p. 214)
 - * **util** (Section 35, p. 219)
 - * **version** (Section 36, p. 221)
- **MainDialog**: This is where the main window of nMOLDYN is defined.
(Section 37, p. 222)
- **NetCDFToASCIIConversionDialog**: This module implements I{File->Convert NetCDF to ASCII} dialog.
(Section 38, p. 226)
- **PDBSnapshotGeneratorDialog**: This module implements I{File->Frame snapshot} dialog.
(Section 39, p. 228)
- **PlotNetCDFVariableDialog**: This module implements I{View->Plot} dialog.
(Section 40, p. 230)
- **PreferencesDialog**: This module implements I{File->Preferences} dialog.
(Section 41, p. 234)
- **PyroServerDialog**: This module implements I{View->Animation} dialog.
(Section 42, p. 236)
- **SelectionDialog**: This module implements the atom selection dialog used in almost all nMOLDYN analysis.
(Section 43, p. 239)
- **Tags** (Section 44, p. 244)
- **TrajectoryConversionDialog**: This module implements I{File -> Trajectory conversion -> converter} dialog.
(Section 45, p. 245)
- **ViewEffectiveModeDialog**: This module implements I{View -> Effective Mode} dialog.
(Section 46, p. 251)
- **Widgets**: This module implements all classes used for the generation of combo widgets.
(Section 47, p. 254)
- **Tests** (Section 48, p. 306)
 - **ARA** (Section 49, p. 307)
 - * **TestsContents** (Section 50, p. 308)
 - * **runTests** (Section 51, p. 309)
 - **AVACF** (Section 52, p. 310)
 - * **TestsContents** (Section 53, p. 311)
 - * **runTests** (Section 54, p. 312)
 - **AnalysisTests**: Test cases for analysis modules.
(Section 55, p. 313)
 - **BuildTestCases** (Section 56, p. 314)
 - **DCSF** (Section 57, p. 315)
 - * **TestsContents** (Section 58, p. 316)
 - * **runTests** (Section 59, p. 317)
 - **DISF** (Section 60, p. 318)

- * **TestsContents** (Section 61, p. 319)
 - * **runTests** (Section 62, p. 320)
- **DISFG** (Section 63, p. 321)
 - * **TestsContents** (Section 64, p. 322)
 - * **runTests** (Section 65, p. 323)
- **DOS** (Section 66, p. 324)
 - * **TestsContents** (Section 67, p. 325)
 - * **runTests** (Section 68, p. 326)
- **EISF** (Section 69, p. 327)
 - * **TestsContents** (Section 70, p. 328)
 - * **runTests** (Section 71, p. 329)
- **MSD** (Section 72, p. 330)
 - * **TestsContents** (Section 73, p. 331)
 - * **runTests** (Section 74, p. 332)
- **StabilityTests**: Test cases for stability of the current version of nMOLDYN versus nMoldyn v2.1.0, the last stable release of nMoldyn. (Section 75, p. 333)
- **VACF** (Section 76, p. 344)
 - * **TestsContents** (Section 77, p. 345)
 - * **runTests** (Section 78, p. 346)
- **__pkginfo__** (Section 79, p. 347)

1.2 Variables

Name	Description
<code>nmoldyn_package_path</code>	Value: <code>os.path.split(__file__)[0]</code>
<code>fileHandler</code>	Value: <code>LogToFile(os.path.join(PREFERENCES.logfile_path, 'nMOLDYN...'</code>
<code>consoleHandler</code>	Value: <code>LogToConsole()</code>
<code>dialogHandler</code>	Value: <code>LogToGUI()</code>

2 Package nMOLDYN.Analysis

2.1 Modules

- **Analysis:** This module implements the base class for all the analysis available in nMOLDYN.
(Section 3, p. 13)
- **Dynamics:** Collections of classes for the determination of dynamics-related properties.
(Section 4, p. 18)
- **NMR:** Collections of classes for the determination of NMR-related properties.
(Section 5, p. 50)
- **Scattering:** Collections of classes for the determination of scattering-related properties.
(Section 6, p. 55)
- **Slave:** This module contains the functions used by Pyro slave to perform analysis remotely.
(Section 7, p. 75)
- **Structure:** Collections of classes for the determination of structure-related properties.
(Section 8, p. 76)
- **Template:** This module implements the estimate, serial and parallel templates for all analysis.
(Section 9, p. 86)

3 Module nMOLDYN.Analysis.Analysis

This modules implements the base class for all the analysis available in nMOLDYN.

3.1 Functions

setUniverseContents (<i>universe</i>)
Sets the contents of each object found in the universe.
Parameters
universe: the MMTK universe to look in. (<i>type=a instance of MMTK.Universe.</i>)

3.2 Variables

Name	Description
residusChemFamily	Value: {'acidic': ('Asp', 'Glu'), 'aliphatic': ('Ile', 'Leu', 'Val...')}
nmoldyn.package_path	Value: os.path.dirname(os.path.split(__file__)[0])

3.3 Class Analysis

Base class for all analysis defined in nMOLDYN.

The class Analysis is an abstract-base-class that defines attributes and methods common to all the analysis available in nMOLDYN. To set up an analysis object, use one of its subclass.

3.3.1 Methods

__init__ (<i>self, parameters=None, statusBar=None</i>)
The constructor.
Parameters
parameters: a dictionary that contains parameters of the selected analysis. (<i>type=dict</i>)
statusBar: if not None, an instance of nMOLDYN.GUI.Widgets.StatusBar. Will attach a status bar to the selected analysis. (<i>type=instance of nMOLDYN.GUI.Widgets.StatusBar</i>)

setInputParameters (<i>self, parameters</i>)
Sets the input parameters dictionary.

parseInputParameters(*self*)

Parses the input parameters stored in |parameters| dictionnary.

Return Value

a dictionnary of the parsed parameters.

(*type=dict*)

buildTimeInfo(*self*)

Builds some attributes related to the frame selection string. They will be used to define at which times a given analysis should be run.

preLoadTrajectory(*self*, *structure*, *differentiation=1*)**saveAnalysis(*self*, *filename*)**

Saves the settings of an analysis to an output file.

Parameters

filename: the name of the output file. If the extension is '.nmi' the output file will be a nMOLDYN input script otherwise the output file will be a nMOLDYN autostart script.

runAnalysis(*self*)

Runs an analysis.

Return Value

a dictionnary of the form {'days' : d, 'hours' : h, 'minutes' : m, 'seconds' : s} specifying the time the analysis took in dayx, hours, minutes and seconds.

(*type=dict*)

updateJobProgress(*self*, *norm*)

Check the progress of the running analysis and displays periodically on the console and the logfile how far is the analysis. Called each time a step of an analysis loop is achieved.

Parameters

norm: the maximum number of steps of the analysis.

buildJobInfo(*self*)

Display on the console and in the log file the main information about the analysis to run.

analysisTime(*self*, *time*)

Converts a time in second in days, hours, minutes and seconds.

Parameters

time: the time (in seconds) to convert.

(*type=integer.*)

Return Value

a dictionary of the form {'days' : d, 'hours' : h, 'minutes' : m, 'seconds' : s}
where d, h, m and s are integers resulting respectively from the conversion of
|time| in days, hours, minutes and seconds.

(*type=dict*)

weightingScheme(*self*, *universe*, *atoms*, *deuter*, *scheme*='equal')

Returns the weights of |atoms| MMTK collection of |universe| MMTK universe using the weighting scheme |scheme|.

Parameters

universe: the MMTK universe.

(*type=instance of MMTK.Universe*)

atoms: the atoms to take into account when defining the weights.

(*type=instance of MMTK.Collections.Collection*)

deuter: the hydrogen atoms that will be parametrized as deuterium atoms.

(*type=instance of MMTK.Collections.Collection*)

scheme: a string equal to 'equal', 'mass', 'coherent', 'incoherent' or
'atomicNumber' that specifies the weighting scheme to use.

(*type=string*)

Return Value

the weights of the selected atoms.

(*type=an instance of MMTK.ParticleProperties.ParticledScalar*)

subsetSelection(*self*, *universe*, *selection*)

Returns a MMTK collection of atoms that matches |selection| selection string. Used to apply an analysis to a subset of atoms.

Parameters

universe: the universe on which the selection will be performed.

(*type=instance of MMTK.Universe*)

selection: the selection string that will define the atoms to select.

(*type=string*)

Return Value

a MMTK Collection of the atoms that matches |selection| selection string.

(*type=instance of MMTK.Collections.Collection*)

deuterationSelection(*self*, *universe*, *selection*)

Returns a MMTK collection of atoms that matches |selection| selection string. Used to switch the parameters of a subset (or all) of hydrogen atoms to the parameters of deuterium in order to simulate deuterated system.

Parameters

universe: the universe on which the selection will be performed.

(*type=instance of MMTK.Universe*)

selection: the selection string that will define the atoms to select.

(*type=string*)

Return Value

a MMTK Collection of the atoms that matches |selection| selection string.

(*type=instance of MMTK.Collections.Collection*)

groupSelection(*self*, *universe*, *selection*)

Returns a list of MMTK collections where each collection defines a group on which will be applied collectively an analysis.

Parameters

universe: the universe on which the selection will be performed.

(*type=instance of MMTK.Universe*)

selection: the selection string that will define the contents of each group.

(*type=string*)

Return Value

a list of MMTK Collection where each collection defines a group..

(*type=list*)

3.4 Class QVectors

Generates a set of QVectors within a given shell.

3.4.1 Methods

`__init__(self, universe, generator, qRadii, dq, qVectorsPerShell, qVectorsDirection=None)`

The constructor.

Parameters

universe:	the MMTK universe used to define the reciprocal space. (<i>type=a MMTK.Universe subclass object</i>)
generator:	a string being one of '3d isotropic', '2d isotropic' or 'anisotropic' the way the q-vectors should be generated. (<i>type=string</i>)
qRadii:	a list of floats specifying the radii of the shell in which the q vectors have to be generated. (<i>type=list</i>)
dq:	a float specifying the width of a qshell defined as $[qRadius - dq/2, qRadius + dq/2]$. (<i>type=float</i>)
qVectorsPerShell:	an integer specifying the number of q-vectors to generate for each shell. (<i>type=integer</i>)
qVectorsDirection:	a list of Scientific.Geometry.Vector objects specifying the directions along which the q-vectors should be generated. If None, the q-vectors generation will be isotropic. (<i>type=list</i>)

4 Module *nMOLDYN.Analysis.Dynamics*

Collections of classes for the determination of dynamics-related properties.

Classes:

* MeanSquareDisplacement	: sets up a Mean-Square-Displacement analysis.
* RootMeanSquareDeviation	: sets up a Root Mean-Square-Deviation analysis.
* GyrationRadius	: sets up a Gyration Radius analysis.
* AngularCorrelation	: sets up an Angular Correlation analysis.
* CartesianVelocityAutoCorrelationFunction	: sets up a Cartesian Velocity AutoCorrelation analysis.
* DensityOfStates	: sets up a Density Of States analysis.
* AutoRegressiveAnalysis	: sets up an Auto-Regressive analysis.
* QuasiHarmonicAnalysis	: sets up a Quasi-Harmonic analysis.
* PassBandTrajectoryFilter	: sets up a Pass-Band Trajectory Filter.
* GlobalMotionTrajectoryFilter	: sets up a Global Motion Trajectory Filter.
* CenterOfMassTrajectory	: sets up a Center Of Mass Trajectory.
* RigidBodyTrajectory	: sets up a Rigid-Body Trajectory.
* AngularVelocityAutoCorrelationFunction	: sets up an Angular Velocity AutoCorrelation Function.
* AngularDensityOfStates	: sets up an Angular Density Of States.

4.1 Class MeanSquareDisplacement

```
nMOLDYN.Analysis.Analysis.Analysis └─ nMOLDYN.Analysis.Dynamics.MeanSquareDisplacement
```

Sets up a Mean Square Displacement analysis.

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *MeanSquareDisplacement*(*|parameters|* = None)

Arguments:

- *|parameters|* -- a dictionary of the input parameters, or 'None' to set up the analysis without parameters.
- * *trajectory* -- a trajectory file name or an instance of *MMTK.Trajectory.Trajectory* class.
- * *timeinfo* -- a string of the form 'first:last:step' where 'first' is an integer specifying the first frame number to consider, 'last' is an integer specifying the last frame number to consider, and 'step' is an integer specifying the step number between two frames.
- * *projection* -- a string of the form 'vx,vy,vz' specifying the vector along which the analysis will be computed. 'vx', 'vy', and 'vz' are floats specifying respectively the components of that vector.
- * *subset* -- a selection string specifying the atoms to consider for the analysis.
- * *deuteration* -- a selection string specifying the hydrogen atoms whose atomic parameters will be used.
- * *weights* -- a string equal to 'equal', 'mass', 'coherent', 'incoherent' or 'atomicNumber' specifying the scheme to use.
- * *msd* -- the output NetCDF file name. A CDL version of this file will also be generated instead of the '.nc' extension.
- * *pyroserver* -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

Comments:

- The algorithm is based on the Fast Correlation Algorithm (FCA) algorithm

4.1.1 Methods

`__init__(self)`

The constructor. Insures that the class can not be instantiated directly from here.

Parameters

- parameters:** a dictionary that contains parameters of the selected analysis.
- statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`.
Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

`initialize(self)`

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

`calc(self, atom, trajname)`

Calculates the atomic term.

Parameters

- atom:** the atom on which the atomic term has been calculated.
(*type=an instance of `MMTK.Atom` class.*)
- trajname:** the name of the trajectory file name.
(*type=string*)

`combine(self, atom, x)`

`finalize(self)`

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

atomicMSD (<i>self</i> , <i>atom</i> , <i>series</i>)
Returns the atomic Mean-Square-Displacement.
Parameters
atom: the atom on which the atomic MSD has been calculated. (<i>type=an instance of MMTK.Atom class.</i>)
series: a array of dimension (self.nFrames,3) specifying the coordinates of atom atom for the selected frames. (<i>type=NumPy array</i>)
Return Value
the MSD computed for atom atom with trajectory series . (<i>type=Numpy array</i>)

Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

4.1.2 Class Variables

Name	Description
inputParametersNames	Value: 'trajectory', 'timeinfo', 'projection', 'subset', 'deuter...
shortName	Value: 'MSD'
canBeEstimated	Value: True

4.2 Class *RootMeanSquareDeviation*

nMOLDYN.Analysis.Analysis.Analysis

nMOLDYN.Analysis.Dynamics.RootMeanSquareDevia

Sets up a Root Mean Square Deviation analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: RootMeanSquareDeviation(|parameters| = None)

Arguments:

- |parameters| -- a dictionnary of the input parameters, or 'None' to set up the anal
- * trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Tra

```

* timeinfo      -- a string of the form 'first:last:step' where 'first' is an i
                  number to consider, 'last' is an integer specifying the last
                  'step' is an integer specifying the step number between two
* referenceframe -- an integer in [1,len(trajecotory)] specifying which frame sho
* subset        -- a selection string specifying the atoms to consider for the
* deuteration   -- a selection string specifying the hydrogen atoms whose atomi
* weights       -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent'
                  scheme to use.
* rmsd          -- the output NetCDF file name. A CDL version of this file will
                  instead of the '.nc' extension.
* pyroserver    -- a string specifying if Pyro will be used and how to run the

```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

4.2.1 Methods

`__init__(self)`

The constructor. Insures that the class can not be instanciated directly from here.

Parameters

parameters: a dictionnary that contains parameters of the selected analysis.

statusBar: if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

`initialize(self)`

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

calc(*self*, *atom*, *trajname*)

Calculates the atomic term.

Parameters

atom: the atom on which the atomic term has been calculated.
(type=an instance of MMTK.Atom class.)

trajname: the name of the trajectory file name.
(type=string)

Note: an atom-by-atom implementation was preferred than a frame-by-frame implementation of the type: `msd = t.configuration[frame] - t.configuration[self.referenceFrame]` `msd = self.weights * msd * msd` `self.RMSD[frameIndex] = N.sqrt(N.add.reduce(msd))`

combine(*self*, *atom*, *x*)

finalize(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...)

Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

4.2.2 Class Variables

Name	Description
<code>inputParametersNames</code>	Value: 'trajectory', 'timeinfo', 'referenceframe', 'subset', 'de...
<code>shortName</code>	Value: 'RMSD'
<code>canBeEstimated</code>	Value: True

4.3 Class CartesianVelocityAutoCorrelationFunction

nMOLDYN.Analysis.Analysis.Analysis

nMOLDYN.Analysis.Dynamics.CartesianVelocityAutoCorrelationFunction

Sets up a Cartesian Velocity AutoCorrelation analysis.

A Subclass of `nMOLDYN.Analysis.Analysis`.

Constructor: `CartesianVelocityAutoCorrelationFunction(|parameters| = None)`

Arguments:

- `|parameters|` -- a dictionary of the input parameters, or 'None' to set up the analysis.
- * `trajectory` -- a trajectory file name or an instance of `MMTK.Trajectory.Trajectory`.
- * `timeinfo` -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two consecutive steps.
- * `differentiation` -- an integer in [0,5] specifying the order of the differentiation of the coordinates. 0 means that the velocities are already computed for analysis.
- * `projection` -- a string of the form 'vx,vy,vz' specifying the vector along which the projection will be computed. 'vx', 'vy', and 'vz' are floats specifying the components of that vector.
- * `normalize` -- a string being one of 'Yes' or 'No' specifying whether the velocities are normalized at $t = 0$ ('Yes') or not ('No').
- * `subset` -- a selection string specifying the atoms to consider for the analysis.
- * `deuteration` -- a selection string specifying the hydrogen atoms whose atom weights are to be modified.
- * `weights` -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' specifying the scheme to use.
- * `vacf` -- the output NetCDF file name. A CDL version of this file will be created instead of the '.nc' extension.
- * `pyroserver` -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

4.3.1 Methods

__init__(*self*)

The constructor. Insures that the class can not be instanciated directly from here.

Parameters

parameters: a dictionnary that contains parameters of the selected analysis.

statusBar: if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

initialize(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

calc(*self*, *atom*, *trajname*)

Calculates the atomic term.

Parameters

atom: the atom on which the atomic term has been calculated.
(*type=an instance of `MMTK.Atom` class.*)

trajname: the name of the trajectory file name.
(*type=string*)

combine(*self*, *atom*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...)

Inherited from `nMOLDYN.Analysis.Analysis.Analysis`(Section 3.3)

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

4.3.2 Class Variables

Name	Description
inputParametersNames	Value: 'trajectory', 'timeinfo', 'differentiation', 'projection'...
shortName	Value: 'VACF'
canBeEstimated	Value: True

4.4 Class CartesianDensityOfStates

nMOLDYN.Analysis.Analysis.Analysis └
nMOLDYN.Analysis.Dynamics.CartesianDensityOfStates

Sets up a Cartesian Density Of States analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: CartesianDensityOfStates(|parameters| = None)

Arguments:

- |parameters| -- a dictionnary of the input parameters, or 'None' to set up the analysis.
 - * trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory
 - * timeinfo -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers.
 - * differentiation -- an integer in [0,5] specifying the order of the differentiation of the coordinates. 0 means that the velocities are already computed for analysis.
 - * projection -- a string of the form 'vx,vy,vz' specifying the vector along which the projection will be computed. 'vx', 'vy', and 'vz' are floats specifying the components of that vector.
 - * fftwindow -- a float in]0.0,100.0[specifying the width of the gaussian window that will be used in the smoothing procedure.
 - * subset -- a selection string specifying the atoms to consider for the analysis.
 - * deuteration -- a selection string specifying the hydrogen atoms whose atom weights are to be considered.
 - * weights -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' specifying the scheme to use.
 - * dos -- the output NetCDF file name. A CDL version of this file will be generated instead of the '.nc' extension.
 - * pyroserver -- a string specifying if Pyro will be used and how to run the server.

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

4.4.1 Methods

`__init__(self)`

The constructor. Insures that the class can not be instanciated directly from here.

Parameters

parameters: a dictionnary that contains parameters of the selected analysis.

statusBar: if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

`initialize(self)`

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

`calc(self, atom, trajname)`

Calculates the atomic term.

Parameters

atom: the atom on which the atomic term has been calculated.
(*type=an instance of `MMTK.Atom` class.*)

trajname: the name of the trajectory file name.
(*type=string*)

`combine(self, atom, x)`

`finalize(self)`

Finalizes the calculations (e.g. averaging the total term, output files creations ...)

Inherited from `nMOLDYN.Analysis.Analysis.Analysis`(Section 3.3)

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSe-`

lection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

4.4.2 Class Variables

Name	Description
inputParametersNames	Value: 'trajectory', 'timeinfo', 'differentiation', 'projection'...
shortName	Value: 'DOS'
canBeEstimated	Value: True

4.5 Class *AutoRegressiveAnalysis*

nMOLDYN.Analysis.Analysis.Analysis —
nMOLDYN.Analysis.Dynamics.AutoRegressiveAnalysis

Sets up an AutoRegressive Analysis analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: *AutoRegressiveAnalysis*(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- * trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory.
- * timeinfo -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers.
- * differentiation -- an integer in [0,5] specifying the order of the differentiation of the coordinates. 0 means that the velocities are already computed for analysis.
- * projection -- a string of the form 'vx,vy,vz' specifying the vector along which the projection will be computed. 'vx', 'vy', and 'vz' are floats specifying the components of that vector.
- * armodelorder -- an integer in [1, len(trajectory)] specifying the order of the autoregressive model.
- * subset -- a selection string specifying the atoms to consider for the analysis.
- * deuteration -- a selection string specifying the hydrogen atoms whose atom weights are to be scaled.
- * weights -- a string equal to 'equal', 'mass', 'coherent', 'incoherent', or 'none' specifying the weighting scheme to use.
- * ara -- the output NetCDF file name. A CDL version of this file will be generated.

instead of the '.nc' extension.
 * pyroserver -- a string specifying if Pyro will be used and how to run the

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

4.5.1 Methods

`__init__(self)`

The constructor. Insures that the class can not be instanciated directly from here.

Parameters

parameters: a dictionnary that contains parameters of the selected analysis.

statusBar: if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

`initialize(self)`

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

`calc(self, atom, trajname)`

Calculates the atomic term.

Parameters

atom: the atom on which the atomic term has been calculated.
(type=an instance of `MMTK.Atom` class.)

trajname: the name of the trajectory file name.
(type=string)

`combine(self, atom, x)`

finalize (<i>self</i>)

Finalizes the calculations (e.g. averaging the total term, output files creations ...)
--

Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

4.5.2 Class Variables

Name	Description
inputParametersNames	Value: 'trajectory', 'timeinfo', 'differentiation', 'projection'...
shortName	Value: 'ARA'
canBeEstimated	Value: True

4.6 Class *PassBandFilteredTrajectory*

nMOLDYN.Analysis.Analysis.Analysis

nMOLDYN.Analysis.Dynamics.PassBandFilteredTrajectory

Sets up a Pass-Band Trajectory Filter analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: *PassBandFilteredTrajectory*(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- * trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory
- * timeinfo -- a string of the form 'first:last:step' where 'first' is an integer specifying the first frame to consider, 'last' is an integer specifying the last frame to consider, 'step' is an integer specifying the step number between two frames.
- * filter -- a string of the form 'low:high' where 'low' and 'high' are floats specifying the lower and the upper bounds of the pass-band filter.
- * subset -- a selection string specifying the atoms to consider for the analysis.
- * pbft -- the output NetCDF file name.
- * pyroserver -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

4.6.1 Methods

`__init__(self)`

The constructor. Insures that the class can not be instanciated directly from here.

Parameters

parameters: a dictionnary that contains parameters of the selected analysis.

statusBar: if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

`initialize(self)`

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

`calc(self, atom, trajname)`

Calculates the atomic term.

Parameters

atom: the atom on which the atomic term has been calculated.
(*type=an instance of `MMTK.Atom` class.*)

trajname: the name of the trajectory file name.
(*type=string*)

`combine(self, atom, x)`

`finalize(self)`

Finalizes the calculations (e.g. averaging the total term, output files creations ...)

Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

4.6.2 Class Variables

Name	Description
inputParametersNames	Value: 'trajectory', 'timeinfo', 'filter', 'subset', 'pbft', 'py...
shortName	Value: 'PBFT'
canBeEstimated	Value: True

4.7 Class RadiusOfGyration

nMOLDYN.Analysis.Analysis.Analysis —
nMOLDYN.Analysis.Dynamics.RadiusOfGyration

Sets up a Radius Of Gyration analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: RadiusOfGyration(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis
 - * trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory
 - * timeinfo -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers
 - * subset -- a selection string specifying the atoms to consider for the analysis
 - * rog -- the output NetCDF file name. A CDL version of this file will be generated instead of the '.nc' extension.
 - * pyroserver -- a string specifying if Pyro will be used and how to run the analysis

Running modes:

- To run the analysis do: a.runAnalysis() where a is the analysis object.
- To estimate the analysis do: a.estimateAnalysis() where a is the analysis object.
- To save the analysis to 'file' file name do: a.saveAnalysis(file) where a is the analysis object.

4.7.1 Methods

__init__(*self*)

The constructor. Insures that the class can not be instanciated directly from here.

Parameters

parameters: a dictionnary that contains parameters of the selected analysis.

statusBar: if not None, an instance of nMOLDYN.GUI.Widgets.StatusBar. Will attach a status bar to the selected analysis.

Overrides: nMOLDYN.Analysis.Analysis.Analysis.__init__

initialize(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

calc(*self*, *atom*, *trajname*)

Calculates the atomic term.

Parameters

atom: the atom on which the atomic term has been calculated.
(*type=an instance of MMTK.Atom class.*)

trajname: the name of the trajectory file name.
(*type=string*)

combine(*self*, *atom*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

4.7.2 Class Variables

Name	Description
inputParametersNames	Value: 'trajectory', 'timeinfo', 'subset', 'rog', 'pyroserver',
shortName	Value: 'ROG'
canBeEstimated	Value: True

4.8 Class GlobalMotionFilteredTrajectory

```
nMOLDYN.Analysis.Analysis.Analysis └─
                                     nMOLDYN.Analysis.Dynamics.GlobalMotionFilteredT
```

Sets up a Global Motion Trajectory Filter analysis.

A Subclass of `nMOLDYN.Analysis.Analysis`.

Constructor: GlobalMotionFilteredTrajectory(|parameters| = None)

Arguments:

- ```
- |parameters| -- a dictionary of the input parameters, or 'None' to set up the anal
 * trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Traject
 * timeinfo -- a string of the form 'first:last:step' where 'first' is an integ
 number to consider, 'last' is an integer specifying the last fra
 'step' is an integer specifying the step number between two fram
 * subset -- a selection string specifying the atoms to consider for the anal
 * gmft -- the output NetCDF file name.
 * pyroserver -- a string specifying if Pyro will be used and how to run the anal
```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

## 4.8.1 Methods

**\_\_init\_\_**(*self*)

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**initialize**(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *frameIndex*, *trajname*)

Calculates the contribution for one frame.

**Parameters**

**frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**combine**(*self*, *frameIndex*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

***Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

## 4.8.2 Class Variables

| Name                 | Description                                                             |
|----------------------|-------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'subset', 'gmft', 'pyroserver', |
| shortName            | <b>Value:</b> 'GMFT'                                                    |
| canBeEstimated       | <b>Value:</b> True                                                      |

## 4.9 Class *CenterOfMassTrajectory*

nMOLDYN.Analysis.Analysis.Analysis —  
nMOLDYN.Analysis.Dynamics.CenterOfMassTrajectory

Sets up a Center Of Mass Trajectory analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: *CenterOfMassTrajectory*(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is an integer number to consider, 'last' is an integer specifying the last frame, 'step' is an integer specifying the step number between two frames.
- \* group -- a selection string specifying the groups of atoms on which the center of mass is calculated (one center of mass per group).
- \* comt -- the output NetCDF file name.
- \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: *a.runAnalysis()* where *a* is the analysis object.
- To estimate the analysis do: *a.estimateAnalysis()* where *a* is the analysis object.
- To save the analysis to 'file' file name do: *a.saveAnalysis(file)* where *a* is the analysis object.

## 4.9.1 Methods

**\_\_init\_\_**(*self*)

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**initialize**(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *frameIndex*, *trajname*)

Calculates the contribution for one frame.

**Parameters**

**frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**combine**(*frameIndex*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

## 4.9.2 Class Variables

| Name                 | Description                                                            |
|----------------------|------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'group', 'comt', 'pyroserver', |
| shortName            | <b>Value:</b> 'COMT'                                                   |
| canBeEstimated       | <b>Value:</b> True                                                     |

#### 4.10 Class *QuasiHarmonicAnalysis*

nMOLDYN.Analysis.Analysis.Analysis —  
nMOLDYN.Analysis.Dynamics.QuasiHarmonicAnalysis

Sets up a Quasi Harmonic Analysis analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: *QuasiHarmonicAnalysis*(|parameters| = None)

Arguments:

- |parameters| -- a dictionnary of the input parameters, or 'None' to set up the anal
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajec
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is an inte
- number to consider, 'last' is an integer specifying the last fr
- 'step' is an integer specifying the step number between two fra
- \* temperature -- the temperature at which the MD was performed.
- \* subset -- a selection string specifying the atoms to consider for the ana
- \* qha -- the output NetCDF file name.

Running modes:

- To run the analysis do: a.runAnalysis() where a is the analysis object.
- To estimate the analysis do: a.estimateAnalysis() where a is the analysis object.
- To save the analysis to 'file' file name do: a.saveAnalysis(file) where a is the a

Comments:

- This analysis is used to get effective modes of vibration from fluctuations calcul
- The results of such an analysis can be seen by generating pseudo-trajectories repr
- a vibration mode.
- For more details: Brooks et al., J. Comp. Chem. 1995, 16, 1522-1542.

## 4.10.1 Methods

**\_\_init\_\_**(*self*)

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**initialize**(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**internalRun**(*self*)

Runs the analysis.

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

## 4.10.2 Class Variables

| Name                              | Description                                                                         |
|-----------------------------------|-------------------------------------------------------------------------------------|
| <code>inputParametersNames</code> | <b>Value:</b> <code>'trajectory', 'timeinfo', 'temperature', 'subset', 'qha'</code> |
| <code>shortName</code>            | <b>Value:</b> <code>'QHA'</code>                                                    |
| <code>canBeEstimated</code>       | <b>Value:</b> <code>False</code>                                                    |

4.11 Class *AngularCorrelation*

`nMOLDYN.Analysis.Analysis.Analysis`

`nMOLDYN.Analysis.Dynamics.AngularCorrelation`

Sets up an Angular Correlation analysis.

A Subclass of `nMOLDYN.Analysis.Analysis`.

Constructor: `AngularCorrelation(|parameters| = None)`

Arguments:

- `|parameters|` -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* `trajectory` -- a trajectory file name or an instance of `MMTK.Trajectory.Trajectory`.
- \* `timeinfo` -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two consecutive steps.
- \* `triplet` -- a selection string specifying the groups of three atoms that will be used to compute the angular correlation.
- \* `atomorder` -- a string of the form 'atom1,atom2,atom3' where 'atom1', 'atom2', 'atom3' are respectively the MMTK atom names of the atoms in the way they are used to compute the angular correlation.
- \* `ac` -- the output NetCDF file name. A CDL version of this file will be generated instead of the '.nc' extension.
- \* `pyroserver` -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

#### 4.11.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instantiated directly from here.

**Parameters**

- parameters:** a dictionary that contains parameters of the selected analysis.
- statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**initialize**(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *tripletIndex*, *trajname*)

Calculates the contribution for one group.

**Parameters**

**tripletIndex:** the index of the triplet in |self.triplet| list.

(*type=integer.*)

**trajname:** the name of the trajectory file name.

(*type=string*)

**combine**(*self*, *tripletIndex*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

**Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)**

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

**4.11.2 Class Variables**

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'triplet', 'atomorder', 'ac', '... |
| shortName            | <b>Value:</b> 'AC'                                                         |
| canBeEstimated       | <b>Value:</b> True                                                         |

**4.12 Class RigidBodyTrajectory**

nMOLDYN.Analysis.Analysis.Analysis

nMOLDYN.Analysis.Dynamics.RigidBodyTrajectory

Sets up a Rigid Body Trajectory analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: RigidBodyTrajectory(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory.
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two consecutive frames.
- \* referenceframe -- an integer in [1,len(trajectory)] specifying which frame should be used as the reference frame.
- \* stepwiserbt -- a string being one of 'Yes' or 'No' specifying whether the reference frame should be the frame i - 1 ('Yes') or should be a fixed frame defined with the referenceframe parameter.
- \* group -- a selection string specifying the groups of atoms on which the analysis should be performed (each group being a rigid body).
- \* rbt -- the output NetCDF file name.
- \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: a.runAnalysis() where a is the analysis object.
- To estimate the analysis do: a.estimateAnalysis() where a is the analysis object.
- To save the analysis to 'file' file name do: a.saveAnalysis(file) where a is the analysis object.

#### 4.12.1 Methods

**\_\_init\_\_(self)**

The constructor. Insures that the class can not be instantiated directly from here.

**Parameters**

**parameters:** a dictionary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of nMOLDYN.GUI.Widgets.StatusBar. Will attach a status bar to the selected analysis.

Overrides: nMOLDYN.Analysis.Analysis.Analysis.\_\_init\_\_

**initialize**(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *groupIndex*, *trajname*)

Calculates the contribution for one group.

**Parameters**

**groupIndex:** the index of the group in |self.group| list.

(*type=integer.*)

**trajname:** the name of the trajectory file name.

(*type=string*)

**combine**(*self*, *groupIndex*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

#### 4.12.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'referenceframe', 'removetransl... |
| shortName            | <b>Value:</b> 'RBT'                                                        |
| canBeEstimated       | <b>Value:</b> True                                                         |

### 4.13 Class *ReorientationalCorrelationFunction*

nMOLDYN.Analysis.Analysis.Analysis



nMOLDYN.Analysis.Dynamics.ReorientationalCorrelationFunction

Sets up a Reorientational Correlation Function analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: ReorientationalCorrelationFunction(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory.
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two consecutive frames.
- \* referenceframe -- an integer in [1,len(trajectory)] specifying which frame should be used as the reference frame.
- \* stepwiserbtf -- a string being one of 'Yes' or 'No' specifying whether the reference frame should be the frame i - 1 ('Yes') or should be a fixed frame defined with the parameter referenceframe.
- \* wignerindexes -- a string of the form 'j,m,n' where 'j', 'm' and 'n' are respectively the indices of the Wigner function  $D_{jmn}$ .
- \* group -- a selection string specifying the groups of atoms on which the analysis should be performed (each group being a rigid body).
- \* rcf -- the output NetCDF file name. A CDL version of this file will be generated instead of the '.nc' extension.
- \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: a.runAnalysis() where a is the analysis object.
- To estimate the analysis do: a.estimateAnalysis() where a is the analysis object.
- To save the analysis to 'file' file name do: a.saveAnalysis(file) where a is the analysis object.

#### 4.13.1 Methods

**\_\_init\_\_(self)**

The constructor. Insures that the class can not be instantiated directly from here.

##### Parameters

- parameters:** a dictionary that contains parameters of the selected analysis.
- statusBar:** if not None, an instance of nMOLDYN.GUI.Widgets.StatusBar. Will attach a status bar to the selected analysis.

Overrides: nMOLDYN.Analysis.Analysis.Analysis.\_\_init\_\_

**initialize(*self*)**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc(*self*, *groupIndex*, *trajname*)**

Calculates the contribution for one group.

**Parameters**

**groupIndex:** the index of the group in `|self.group|` list.

(*type=integer.*)

**trajname:** the name of the trajectory file name.

(*type=string*)

**combine(*self*, *groupIndex*, *x*)****finalize(*self*)**

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

**4.13.2 Class Variables**

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'referenceframe', 'stepwiserbt'... |
| shortName            | <b>Value:</b> 'RCF'                                                        |
| canBeEstimated       | <b>Value:</b> True                                                         |

**4.14 Class AngularVelocity**

An intermediate class used by `|AngularVelocityAutoCorrelationFunction|` and `|AngularDensityOfStates|` classes.

## 4.14.1 Methods

`__init__(self)``qMatrix(self, data)``getAngularVelocity(self, t, g)`Computes the Angular Velocity Function for a group `|g|` (a MMTK Collection).4.15 Class *AngularVelocityAutoCorrelationFunction*

```

nMOLDYN.Analysis.Analysis.Analysis
nMOLDYN.Analysis.Dynamics.AngularVelocity
nMOLDYN.Analysis.Dynamics.AngularVelocity

```

Sets up an Angular Velocity AutoCorrelation Function analysis.

A Subclass of `nMOLDYN.Analysis.Analysis`.Constructor: `AngularVelocityAutoCorrelationFunction(|parameters| = None)`

Arguments:

- `|parameters|` -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* `trajectory` -- a trajectory file name or an instance of `MMTK.Trajectory.Trajectory`.
- \* `timeinfo` -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two consecutive steps.
- \* `differentiation` -- an integer in [0,5] specifying the order of the differentiation out of the coordinates. 0 means that the velocities are already computed for analysis.
- \* `projection` -- a string of the form 'vx,vy,vz' specifying the vector along which the angular velocity will be computed. 'vx', 'vy', and 'vz' are floats specifying the components of that vector.
- \* `referenceframe` -- an integer in [1,len(trajectory)] specifying which frame should be used.
- \* `stepwiserbt` -- a string being one of 'Yes' or 'No' specifying whether the reference frame `i - 1` ('Yes') or should be a fixed frame defined by the user.
- \* `group` -- a selection string specifying the groups of atoms on which the analysis should be performed (each group being a rigid body).
- \* `avacf` -- the output NetCDF file name. A CDL version of this file will be generated.

instead of the '.nc' extension.  
 \* pyroserver      -- a string specifying if Pyro will be used and how to run the

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

#### 4.15.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Dynamics.AngularVelocity.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, groupIndex, trajname)`**

Calculates the contribution for one group.

**Parameters**

**groupIndex:** the index of the group in `|self.group|` list.  
*(type=integer.)*

**trajname:** the name of the trajectory file name.  
*(type=string)*

**`combine(self, groupIndex, x)`**

|                                 |
|---------------------------------|
| <b>finalize</b> ( <i>self</i> ) |
|---------------------------------|

|                                                                                         |
|-----------------------------------------------------------------------------------------|
| Finalizes the calculations (e.g. averaging the total term, output files creations ...). |
|-----------------------------------------------------------------------------------------|

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)**

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

**Inherited from *nMOLDYN.Analysis.Dynamics.AngularVelocity*(Section 4.14)**

getAngularVelocity(), qMatrix()

#### 4.15.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'differentiation', 'projection'... |
| shortName            | <b>Value:</b> 'AVACF'                                                      |
| canBeEstimated       | <b>Value:</b> True                                                         |

#### 4.16 Class *AngularDensityOfStates*

```

nMOLDYN.Analysis.Analysis.Analysis
nMOLDYN.Analysis.Dynamics.AngularVelocity
nMOLDYN.Analysis.Dynamics.AngularDensityOfStates

```

Sets up an Angular Density Of States analysis.

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *AngularDensityOfStates*(*|parameters|* = None)

Arguments:

- *|parameters|* -- a dictionary of the input parameters, or 'None' to set up the analysis
  - \* trajectory -- a trajectory file name or an instance of *MMTK.Trajectory.Trajectory*
  - \* timeinfo -- a string of the form 'first:last:step' where 'first' is an integer specifying the first number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers

```

* differentiation -- an integer in [0,5] specifying the order of the differentia
 out of the coordinates. 0 means that the velocities are alr
 for analysis.
* projection -- a string of the form 'vx,vy,vz' specifying the vector along
 will be computed. 'vx', 'vy', and 'vz' are floats specifying
 of that vector.
* referenceframe -- an integer in [1,len(trajectory)] specifying which frame sh
* stepwiserbt -- a string being one of 'Yes' or 'No' specifying whether the
 the frame i - 1 ('Yes') or should be a fixed frame defined
* fftwindow -- a float in]0.0,100.0[specifying the width of the gaussian
 that will be used in the smoothing procedure.
* group -- a selection string specifying the groups of atoms on which
 (each group being a rigid body).
* ados -- the output NetCDF file name. A CDL version of this file wil
 instead of the '.nc' extension.
* pyroserver -- a string specifying if Pyro will be used and how to run the

```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

#### 4.16.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Dynamics.AngularVelocity.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *groupIndex*, *trajname*)

Calculates the contribution for one group.

**Parameters**

**groupIndex:** the index of the group in |self.group| list.  
*(type=integer.)*

**trajname:** the name of the trajectory file name.  
*(type=string)*

**combine**(*self*, *groupIndex*, *x*)

**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)**

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

**Inherited from *nMOLDYN.Analysis.Dynamics.AngularVelocity*(Section 4.14)**

getAngularVelocity(), qMatrix()

#### 4.16.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeInfo', 'differentiation', 'projection'... |
| shortName            | <b>Value:</b> 'ADOS'                                                       |
| canBeEstimated       | <b>Value:</b> True                                                         |

## 5 Module *nMOLDYN.Analysis.NMR*

Collections of classes for the determination of NMR-related properties.

Classes:

- \* *OrderParameter* : sets up an order parameter analysis.
- \* *OrderParameterContactModel* : sets up an order parameter analysis using the contact

### 5.1 Class *OrderParameter*

*nMOLDYN.Analysis.Analysis.Analysis* └─ *nMOLDYN.Analysis.NMR.OrderParameter*

Sets up an order parameter analysis.

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *OrderParameter*(*|parameters|* = None)

Arguments:

- *|parameters|* -- a dictionary of the input parameters, or 'None' to set up the analysis
  - \* *trajectory* -- a trajectory file name or an instance of *MMTK.Trajectory*
  - \* *timeinfo* -- a string of the form 'first:last:step' where 'first' is number to consider, 'last' is an integer specifying the 'step' is an integer specifying the step number between
  - \* *group* -- a selection string specifying the groups of atoms that w analysis will be computed. Each group must contain two a
  - \* *atomorder* -- a string of the form 'atom1,atom2,atom3' where 'atom1', respectively the *MMTK* atom names of the atoms in the way
  - \* *op* -- the output *NetCDF* file name. A *CDL* version of this file instead of the '.nc' extension.
  - \* *pyroserver* -- a string specifying if *Pyro* will be used and how to run

Running modes:

- To run the analysis do: *a.runAnalysis()* where *a* is the analysis object.
- To estimate the analysis do: *a.estimateAnalysis()* where *a* is the analysis object.
- To save the analysis to 'file' file name do: *a.saveAnalysis(file)* where *a* is the a

## Comments:

- This code is based on a first implementation made by Vania Calandrini.

## 5.1.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, bondIndex, trajname)`**

Calculates the contribution for one group.

**Parameters**

**bondIndex:** the index of the group in `|self.bond|` list.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**`combine(self, bondIndex, x)`****`finalize(self)`**

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSe-`

lection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

### 5.1.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'bond', 'atomorder', 'op', 'pyr... |
| shortName            | <b>Value:</b> 'OP'                                                         |
| canBeEstimated       | <b>Value:</b> True                                                         |

## 5.2 Class *OrderParameterContactModel*

```
nMOLDYN.Analysis.Analysis.Analysis └─ nMOLDYN.Analysis.NMR.OrderParameterContactModel
```

Sets up an order parameter analysis using the contact model .

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *OrderParameterContactModel*(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis
  - \* trajectory -- a trajectory file name or an instance of *MMTK.Trajectory.Trajectory*
  - \* timeinfo -- a string of the form 'first:last:step' where 'first' is an integer number to consider, 'last' is an integer specifying the last frame, 'step' is an integer specifying the step number between two frames
  - \* opcm -- the output NetCDF file name. A CDL version of this file will also be generated instead of the '.nc' extension.
  - \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis

Running modes:

- To run the analysis do: *a.runAnalysis()* where *a* is the analysis object.
- To estimate the analysis do: *a.estimateAnalysis()* where *a* is the analysis object.
- To save the analysis to 'file' file name do: *a.saveAnalysis(file)* where *a* is the analysis object.

Comments:

- This code is adapted from the s2predict code developped by F. Zhang and R. Bruschweiler  
<http://nmr.clarku.edu/software/S2/s2predict.html>
- For more details about the method: Zhang, F., Bruschweiler, R. J. AM. Chem. Soc. 2001, 123, 12345

### 5.2.1 Methods

#### **`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

#### **Parameters**

- parameters:** a dictionary that contains parameters of the selected analysis.
- statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

#### **`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

#### **`calc(self, frameIndex, trajname)`**

Calculates the contribution for one group.

#### **Parameters**

- frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
(*type=integer.*)
- trajname:** the name of the trajectory file name.  
(*type=string*)

#### **`combine(self, frameIndex, x)`**

#### **`finalize(self)`**

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

*Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)*

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

### 5.2.2 Class Variables

| Name                 | Description                                                   |
|----------------------|---------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'opcm', 'pyroserver', |
| shortName            | <b>Value:</b> 'OPCM'                                          |
| canBeEstimated       | <b>Value:</b> True                                            |

## 6 Module `nMOLDYN.Analysis.Scattering`

Collections of classes for the determination of scattering-related properties.

Classes:

|                                                         |                                                           |
|---------------------------------------------------------|-----------------------------------------------------------|
| * <code>DynamicCoherentStructureFactor</code>           | : sets up a Dynamic Coherent Structure Factor             |
| * <code>DynamicCoherentStructureFactorARModel</code>    | : sets up a Dynamic Coherent Structure Factor AR Model    |
| * <code>DynamicIncoherentStructureFactor</code>         | : sets up an Dynamic Incoherent Structure Factor          |
| * <code>DynamicIncoherentStructureFactorGaussian</code> | : sets up an Dynamic Incoherent Structure Factor Gaussian |
| * <code>IncoherentStructureFactorARModel</code>         | : sets up an Dynamic Incoherent Structure Factor AR Model |
| * <code>ElasticIncoherentStructureFactor</code>         | : sets up an Elastic Incoherent Structure Factor          |
| * <code>StaticCoherentStructureFactor</code>            | : sets up a Static Coherent Structure Factor              |

Procedures:

\* `DynamicStructureFactor` : returns the Dynamic Structure Factor.

### 6.1 Functions

**DynamicStructureFactor**(*netcdf*, *alpha*)

Computes the dynamic structure factor from an intermediate scattering function.

**Parameters**

**netcdf**: the intermediate scattering function from which the dynamic structure factor will be computed..

(*type=string or instance of `_NetCDFFile`*)

**alpha**: the width, in percentage of the trajectory length, of the gaussian used in the smoothing procedure.

(*type=float*)

### 6.2 Class `DynamicCoherentStructureFactor`

`nMOLDYN.Analysis.Analysis.Analysis`

`nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactor`

Sets up a Dynamic Coherent Structure Factor analysis.

A Subclass of `nMOLDYN.Analysis.Analysis`.

Constructor: `DynamicCoherentStructureFactor(|parameters| = None)`

## Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is a number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers.
- \* qshellvalues -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...' where 'qmin1', 'qmax1', 'dq1', 'qmin2', 'qmax2', 'dq2' ... are floats that represent the q minimum, the q maximum and the q steps for q intervals.
- \* qshellwidth -- a float specifying the width of the q shells.
- \* qvectorspershell -- a float specifying the number of q vectors to generate per shell.
- \* qvectorsgenerator -- a string being one of 'isotropic', 'anisotropic' or 'explicit'. The q vectors will be generated.
- \* qvectorsdirection -- a string of the form 'v1x,v1y,v1z;v2x,v2y,v2z...' where 'v1x', 'v1y', 'v1z', 'v2x', 'v2y', 'v2z' ... are floats that represents respectively the direction of the q vectors which the q vectors should be generated.
- \* fftwindow -- a float in ]0.0,100.0[ specifying the width of the gaussian window that will be used in the smoothing procedure.
- \* subset -- a selection string specifying the atoms to consider for the analysis.
- \* deuteration -- a selection string specifying the hydrogen atoms whose atoms are deuterated.
- \* weights -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' specifying the weighting scheme to use.
- \* dcsf -- the output NetCDF file name for the intermediate scattering function.
- \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis.

## Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

### 6.2.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, qIndex, trajname)`**

Calculates the contribution for one Q-shell.

**Parameters**

**qIndex:** the index of the Q-shell in `|self.qRadii|` list.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**`combine(self, qIndex, x)`**

**`finalize(self)`**

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis`(Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

### 6.2.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'qshellvalues', 'qshellwidth', ... |
| default              | <b>Value:</b> {'weights': 'coherent'}                                      |
| shortName            | <b>Value:</b> 'DCSF'                                                       |
| canBeEstimated       | <b>Value:</b> False                                                        |

### 6.3 Class *StaticCoherentStructureFactor*

nMOLDYN.Analysis.Analysis.Analysis —  
nMOLDYN.Analysis.Scattering.StaticCoherentStructureFactor

Sets up a Coherent Structure Factor analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: *StaticCoherentStructureFactor*(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is a number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers.
- \* qshellvalues -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...' where 'qmin1', 'qmax1', 'dq1', 'qmin2', 'qmax2', 'dq2' ... are floats that represent the q minimum, the q maximum and the q steps for q intervals.
- \* qshellwidth -- a float specifying the width of the q shells.
- \* qvectorspershell -- a float specifying the number of q vectors to generate per shell.
- \* qvectorsgenerator -- a string being one of 'isotropic', 'anisotropic' or 'explicit'. The type of q vectors will be generated.
- \* qvectorsdirection -- a string of the form 'v1x,v1y,v1z;v2x,v2y,v2z...' where 'v1x', 'v1y', 'v1z', 'v2x', 'v2y', 'v2z' ... are floats that represents respectively the direction of the q vectors which the q vectors should be generated.
- \* fftwindow -- a float in ]0.0,100.0[ specifying the width of the gaussian window that will be used in the smoothing procedure.
- \* subset -- a selection string specifying the atoms to consider for the analysis.
- \* deuteration -- a selection string specifying the hydrogen atoms whose atoms are deuterated.
- \* weights -- a string equal to 'equal', 'mass', 'coherent', 'incoherent' ... specifying the weighting scheme to use.
- \* csf -- the output NetCDF file name for the intermediate scattering function.

```
* pyroserver -- a string specifying if Pyro will be used and how to run t
```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

### 6.3.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, qIndex, trajname)`**

Calculates the contribution for one Q-shell.

**Parameters**

**qIndex:** the index of the Q-shell in `|self.qRadii|` list.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**`combine(self, qIndex, x)`**

|                              |
|------------------------------|
| <b>finalize(<i>self</i>)</b> |
|------------------------------|

|                                                                                         |
|-----------------------------------------------------------------------------------------|
| Finalizes the calculations (e.g. averaging the total term, output files creations ...). |
|-----------------------------------------------------------------------------------------|

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)**

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

### 6.3.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'qshellvalues', 'qshellwidth', ... |
| default              | <b>Value:</b> {'weights': 'coherent'}                                      |
| shortName            | <b>Value:</b> 'SCSF'                                                       |
| canBeEstimated       | <b>Value:</b> False                                                        |

## 6.4 Class *DynamicCoherentStructureFactorAR*

*nMOLDYN.Analysis.Analysis.Analysis*

*nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactorAR*

Sets up a Dynamic Coherent Structure Factor analysis using an Auto Regressive model.

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *DynamicCoherentStructureFactorARModel*(|parameters| = None)

Arguments:

- |parameters| -- a dictionnary of the input parameters, or 'None' to set up the analysis
  - \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.
  - \* timeinfo -- a string of the form 'first:last:step' where 'first' is a number to consider, 'last' is an integer specifying the last time step, 'step' is an integer specifying the step number between two time steps.
  - \* armodelorder -- an integer in [1, len(trajectory)] specifying the order of the autoregressive model.
  - \* qshellvalues -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...' where 'qmin1', 'qmax1', 'dq1', 'qmin2', 'qmax2', 'dq2' ... are floats that define the q minimum, the q maximum and the q steps for q intervals.

```

* qshellwidth -- a float specifying the width of the q shells.
* qvectorspershell -- a float specifying the number of q vectors to generate per
* qvectorsgenerator -- a string being one of 'isotropic', 'anisotropic' or 'explicit'
 will be generated.
* qvectorsdirection -- a string of the form 'v1x,v1y,v1z;v2x,v2y,v2z...' where 'v1x',
 'v1y', 'v1z', 'v2x', 'v2y', 'v2z' ... are floats that represents respectively
 which the q vectors should be generated.
* subset -- a selection string specifying the atoms to consider for the analysis.
* deuteration -- a selection string specifying the hydrogen atoms whose atoms are
* weights -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent'
 scheme to use.
* dcsfar -- the output NetCDF file name.
* pyroserver -- a string specifying if Pyro will be used and how to run the server.

```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

#### 6.4.1 Methods

##### `__init__(self)`

The constructor. Insures that the class can not be instanciated directly from here.

##### Parameters

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

##### `initialize(self)`

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *qIndex*, *trajname*)

Calculates the contribution for one Q-shell.

**Parameters**

**qIndex:** the index of the Q-shell in `|self.qRadii|` list.  
*(type=integer.)*

**trajname:** the name of the trajectory file name.  
*(type=string)*

**combine**(*self*, *qIndex*, *x*)

**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

**6.4.2 Class Variables**

| Name                              | Description                                                                              |
|-----------------------------------|------------------------------------------------------------------------------------------|
| <code>inputParametersNames</code> | <b>Value:</b> <code>'trajectory', 'timeinfo', 'armodelorder', 'qshellvalues', ...</code> |
| <code>shortName</code>            | <b>Value:</b> <code>'DCSFAR'</code>                                                      |
| <code>canBeEstimated</code>       | <b>Value:</b> <code>False</code>                                                         |
| <code>default</code>              | <b>Value:</b> <code>{'weights': 'coherent'}</code>                                       |

**6.5 Class *DynamicIncoherentStructureFactor***

`nMOLDYN.Analysis.Analysis.Analysis`

`nMOLDYN.Analysis.Scattering.DynamicIncoherentStr`

Sets up an Dynamic Incoherent Structure Factor analysis.

A Subclass of `nMOLDYN.Analysis.Analysis`.

Constructor: `DynamicIncoherentStructureFactorARModel(|parameters| = None)`

## Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is a number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers.
- \* qshellvalues -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...' where 'qmin1', 'qmax1', 'dq1', 'qmin2', 'qmax2', 'dq2' ... are floats that represent the q minimum, the q maximum and the q steps for q intervals.
- \* qshellwidth -- a float specifying the width of the q shells.
- \* qvectorspershell -- a float specifying the number of q vectors to generate per shell.
- \* qvectorsgenerator -- a string being one of 'isotropic', 'anisotropic' or 'explicit'. The q vectors will be generated.
- \* qvectorsdirection -- a string of the form 'v1x,v1y,v1z;v2x,v2y,v2z...' where 'v1x', 'v1y', 'v1z', 'v2x', 'v2y', 'v2z' ... are floats that represents respectively the direction of the q vectors which the q vectors should be generated.
- \* fftwindow -- a float in ]0.0,100.0[ specifying the width of the gaussian window that will be used in the smoothing procedure.
- \* subset -- a selection string specifying the atoms to consider for the analysis.
- \* deuteration -- a selection string specifying the hydrogen atoms whose atoms are deuterated.
- \* weights -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' specifying the weighting scheme to use.
- \* disf -- the output NetCDF file name for the intermediate scattering function.
- \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis.

## Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

### 6.5.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, atom, trajname)`**

Calculates the atomic term.

**Parameters**

**atom:** the atom on which the atomic term has been calculated.  
(*type=an instance of `MMTK.Atom` class.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**`combine(self, atom, x)`**

**`finalize(self)`**

Finalizes the calculations (e.g. averaging the total term, output files creations ...)

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis`(Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

### 6.5.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'qshellvalues', 'qshellwidth', ... |
| default              | <b>Value:</b> {'weights': 'incoherent'}                                    |
| shortName            | <b>Value:</b> 'DISF'                                                       |
| canBeEstimated       | <b>Value:</b> True                                                         |

## 6.6 Class *DynamicIncoherentStructureFactorAR*

nMOLDYN.Analysis.Analysis.Analysis —  
nMOLDYN.Analysis.Scattering.DynamicIncoherentStr

Sets up an Dynamic Incoherent Structure Factor analysis using an Auto Regressive model.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: *DynamicIncoherentStructureFactorARModel*(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the anal
  - \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.
  - \* timeinfo -- a string of the form 'first:last:step' where 'first' is a number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two numbers.
  - \* armodelorder -- an integer in [1, len(trajectory)] specifying the order of the autoregressive model.
  - \* qshellvalues -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...' where 'qmin1', 'qmax1', 'dq1', 'qmin2', 'qmax2', 'dq2' ... are floats that represent the q minimum, the q maximum and the q steps for q intervals.
  - \* qshellwidth -- a float specifying the width of the q shells.
  - \* qvectorspershell -- a float specifying the number of q vectors to generate per shell.
  - \* qvectorsgenerator -- a string being one of 'isotropic', 'anisotropic' or 'explicit'. The q vectors will be generated.
  - \* qvectorsdirection -- a string of the form 'v1x,v1y,v1z;v2x,v2y,v2z...' where 'v1x', 'v1y', 'v1z', 'v2x', 'v2y', 'v2z' ... are floats that represents respectively the direction of the q vectors which the q vectors should be generated.
  - \* subset -- a selection string specifying the atoms to consider for the analysis.
  - \* deuteration -- a selection string specifying the hydrogen atoms whose scattering length is to be considered.
  - \* weights -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' specifying the weighting scheme to use.
  - \* disfar -- the output NetCDF file name for the intermediate scattering function.
  - \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

### 6.6.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, atom, trajname)`**

Calculates the atomic term.

**Parameters**

**atom:** the atom on which the atomic term has been calculated.  
(*type=an instance of `MMTK.Atom` class.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**`combine(self, atom, x)`**

**`finalize(self)`**

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

***Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)***

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

**6.6.2 Class Variables**

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeInfo', 'armodelorder', 'qshellvalues',... |
| shortName            | <b>Value:</b> 'DISFAR'                                                     |
| canBeEstimated       | <b>Value:</b> True                                                         |
| default              | <b>Value:</b> {'weights': 'incoherent'}                                    |

**6.7 Class DynamicIncoherentStructureFactorGaussian**

nMOLDYN.Analysis.Analysis.Analysis

nMOLDYN.Analysis.Scattering.DynamicIncoherentStr

Sets up an Dynamic Incoherent Structure Factor analysis within Gaussian approximation.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: DynamicIncoherentStructureFactorGaussian(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the anal
  - \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Traje
  - \* timeinfo -- a string of the form 'first:last:step' where 'first' is an int
  - \* qshellvalues -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...' wher
  - \* fftwindow -- a float in ]0.0,100.0[ specifying the width of the gaussian, i
  - \* subset -- a selection string specifying the atoms to consider for the an
  - \* deuteration -- a selection string specifying the hydrogen atoms whose atomic
  - \* weights -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' o
- scheme to use.

```
* disfg -- the output NetCDF file name for the intermediate scattering fu
* pyroserver -- a string specifying if Pyro will be used and how to run the an
```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

### 6.7.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, atom, trajname)`**

Calculates the atomic term.

**Parameters**

**atom:** the atom on which the atomic term has been calculated.  
(*type=an instance of `MMTK.Atom` class.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**`combine(self, atom, x)`**

**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...)

**getMSD**(*self*, *series*)

Computes the atomic component of the Mean-Square-Displacement. This is the exact copy of the version written in *nMOLDYN.Simulations.Dynamics* but rewritten here for to keep the module *Scattering* independant from module *Dynamics*.

### *Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

*analysisTime()*, *buildJobInfo()*, *buildTimeInfo()*, *deuterationSelection()*, *groupSelection()*, *parseInputParameters()*, *preLoadTrajectory()*, *runAnalysis()*, *saveAnalysis()*, *setInputParameters()*, *subsetSelection()*, *updateJobProgress()*, *weightingScheme()*

#### 6.7.2 Class Variables

| Name                              | Description                                                                |
|-----------------------------------|----------------------------------------------------------------------------|
| <code>inputParametersNames</code> | <b>Value:</b> 'trajectory', 'timeinfo', 'qshellvalues', 'fftwindow', 's... |
| <code>default</code>              | <b>Value:</b> {'weights': 'incoherent'}                                    |
| <code>shortName</code>            | <b>Value:</b> 'DISFG'                                                      |
| <code>canBeEstimated</code>       | <b>Value:</b> True                                                         |

## 6.8 Class *ElasticIncoherentStructureFactor*

*nMOLDYN.Analysis.Analysis.Analysis*

*nMOLDYN.Analysis.Scattering.ElasticIncoherentStruc*

Sets up an Elastic Incoherent Structure Factor.

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *ElasticIncoherentStructureFactor*(`|parameters|` = None)

Arguments:

- `|parameters|` -- a dictionnary of the input parameters, or 'None' to set up the anal
  - \* `trajectory` -- a trajectory file name or an instance of *MMTK.Trajectory*.
  - \* `timeinfo` -- a string of the form 'first:last:step' where 'first' is a

```

 number to consider, 'last' is an integer specifying the l
 'step' is an integer specifying the step number between t
* qshellvalues -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...'
 'qmax1', 'qmax2' ... and 'dq1', 'dq2' ... are floats that
 the q minimum, the q maximum and the q steps for q interv
* qshellwidth -- a float specifying the width of the q shells.
* qvectorspershell -- a float specifying the number of q vectors to generate pe
* qvectorsgenerator -- a string being one of 'isotropic', 'anisotropic' or 'expl
 will be generated.
* qvectorsdirection -- a string of the form 'v1x,v1y,v1z;v2x,v2y,v2z...' where '
 'v1z', 'v2z' ... are floats that represents respectively
 which the q vectors should be generated.
* subset -- a selection string specifying the atoms to consider for t
* deuteration -- a selection string specifying the hydrogen atoms whose at
* weights -- a string equal to 'equal', 'mass', 'coherent' , 'incohere
 scheme to use.
* eisf -- the output NetCDF file name. A CDL version of this file w
 instead of the '.nc' extension.
* pyroserver -- a string specifying if Pyro will be used and how to run t

```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

### 6.8.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**initialize**(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *atom*, *trajname*)

Calculates the atomic term.

**Parameters**

**atom:** the atom on which the atomic term has been calculated.  
*(type=an instance of MMTK.Atom class.)*

**trajname:** the name of the trajectory file name.  
*(type=string)*

**combine**(*self*, *atom*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...)

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

### 6.8.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'qshellvalues', 'qshellwidth', ... |
| default              | <b>Value:</b> {'weights': 'incoherent'}                                    |
| shortName            | <b>Value:</b> 'EISF'                                                       |
| canBeEstimated       | <b>Value:</b> True                                                         |

## 6.9 Class SmoothedStaticCoherentStructureFactor

nMOLDYN.Analysis.Analysis.Analysis

nMOLDYN.Analysis.Scattering.SmoothedStaticCoherentStructureFactor

Sets up an Smoothed Static Coherent Structure Factor.

A Subclass of `nMOLDYN.Analysis.Analysis`.

Constructor: `SmoothedStaticCoherentStructureFactor(|parameters| = None)`

Arguments:

- `|parameters|` -- a dictionnary of the input parameters, or 'None' to set up the analysis.
- \* `trajectory` -- a trajectory file name or an instance of `MMTK.Trajectory`.
- \* `timeinfo` -- a string of the form 'first:last:step' where 'first' is a number to consider, 'last' is an integer specifying the last number to consider, 'step' is an integer specifying the step number between two steps.
- \* `qshellvalues` -- a string of the form 'qmin1:qmax1:dq1;qmin2:qmax2:dq2...' where 'qmin1', 'qmax1', 'dq1', 'qmin2', 'qmax2', 'dq2' ... are floats that specify the q minimum, the q maximum and the q steps for q intervals.
- \* `subset` -- a selection string specifying the atoms to consider for the analysis.
- \* `deuteration` -- a selection string specifying the hydrogen atoms whose scattering length is to be used.
- \* `weights` -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' to specify the scheme to use.
- \* `scsf` -- the output NetCDF file name. A CDL version of this file will be generated instead of the '.nc' extension.
- \* `pyroserver` -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

Comments:

- The analysis is based on the angular averaged coherent static structure factor for which the summation over the q vectors is replaced by an integral over the q space. The form is given in equation 2.35 of Fischer et al. Rep. Prog. Phys. 69 (2006) 233-299.

**6.9.1 Methods****`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**`calc(self, frameIndex, trajname)`**

Calculates the contribution for one frame.

**Parameters**

**frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**`combine(self, frameIndex, x)`****`finalize(self)`**

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis`(Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

**6.9.2 Class Variables**

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'qshellvalues', 'subset', 'deut... |
| default              | <b>Value:</b> {'weights': 'coherent'}                                      |
| shortName            | <b>Value:</b> 'SSCSF'                                                      |
| canBeEstimated       | <b>Value:</b> True                                                         |

## 7 Module `nMOLDYN.Analysis.Slave`

This module contains the functions used by Pyro slave to perform analysis remotely.

Functions:

- \* `do_analysisPerElement`: performs an analysis element-by-element.

### 7.1 Functions

**`do_analysisPerElement`**(*analysis, element, trajname*)

Performs the analysis element-by-element, the element being either an atom (atom-by-atom analysis), a frame index (frame-by-frame analysis), a group of atom (group-by-group analysis) or a set of q vectors.

**Parameters**

**analysis**: the selected analysis.

(*type=a subclass of  
nMOLDYN.Analysis.Analysis.Analysis class*)

**element**: the element on which the analysis is based.

(*type=MMTK.Atom|integer|MMTK.Collections.Collection|nMOLDYN.Mathematics*)

**trajname**: a string specifying the name of the trajectory.

(*type=string*)

**Return Value**

the results of the analysis performed on one element.

(*type=depends on the analysis*)

## 8 Module *nMOLDYN.Analysis.Structure*

Collections of classes for the determination of structure-related properties.

Classes:

```
* PairDistributionFunction : sets up a Pair Distribution Function Analysis.
* CoordinationNumber : sets up a Coordination Number Analysis.
* SpatialDensity : sets up a Spatial Density Analysis.
* ScrewFit : sets up a Screw Fit Analysis.
```

### 8.1 Class *PairDistributionFunction*

```
nMOLDYN.Analysis.Analysis.Analysis └─ nMOLDYN.Analysis.Structure.PairDistributionFunction
```

Sets up a Pair Distribution Function analysis.

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *PairDistributionFunction*(|parameters| = None)

Arguments:

```
- |parameters| -- a dictionary of the input parameters, or 'None' to set up the anal
 * trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajec
 * timeinfo -- a string of the form 'first:last:step' where 'first' is an inte
 number to consider, 'last' is an integer specifying the last fr
 'step' is an integer specifying the step number between two fra
 * rvalues -- a string of the form 'rmin:rmax:dr' where 'rmin' is a float spe
 consider, 'rmax' is a float specifying the maximum distance val
 specifying the distance increment.
 * subset -- a selection string specifying the atoms to consider for the ana
 * deuteration -- a selection string specifying the hydrogen atoms whose atomic p
 * weights -- a string equal to 'equal', 'mass', 'coherent' , 'incoherent' or
 scheme to use.
 * pdf -- the output NetCDF file name. A CDL version of this file will al
 instead of the '.nc' extension.
 * pyroserver -- a string specifying if Pyro will be used and how to run the ana
```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

#### Comments:

- This code contains a pyrex function for the distance histogram calculation that is written by Miguel Gonzalez, Insitut Laue Langevin, Grenoble, France.

#### 8.1.1 Methods

##### **`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

##### **Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

##### **`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

##### **`calc(self, frameIndex, trajname)`**

Calculates the contribution for one frame.

##### **Parameters**

**frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

##### **`combine(self, frameIndex, x)`**

|                        |
|------------------------|
| <b>finalize</b> (self) |
|------------------------|

|                                                                                         |
|-----------------------------------------------------------------------------------------|
| Finalizes the calculations (e.g. averaging the total term, output files creations ...). |
|-----------------------------------------------------------------------------------------|

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)**

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

### 8.1.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'rvalues', 'subset', 'deuterati... |
| shortName            | <b>Value:</b> 'PDF'                                                        |
| canBeEstimated       | <b>Value:</b> True                                                         |

## 8.2 Class CoordinationNumber

nMOLDYN.Analysis.Analysis.Analysis —  
nMOLDYN.Analysis.Structure.CoordinationNumber

Sets up a Coordination Number analysis.

A Subclass of nMOLDYN.Analysis.Analysis.

Constructor: CoordinationNumber(|parameters| = None)

Arguments:

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is an integer number to consider, 'last' is an integer specifying the last frame, 'step' is an integer specifying the step number between two frames
- \* rvalues -- a string of the form 'rmin:rmax:dr' where 'rmin' is a float specifying the minimum distance to consider, 'rmax' is a float specifying the maximum distance value, 'dr' is a float specifying the distance increment.
- \* group -- a selection string specifying the groups of atoms that will be considered. The coordination number will be computed. For each group, there

```

 gravity of the group.
* subset -- a selection string specifying the atoms to consider for the ana
* deuteration -- a selection string specifying the hydrogen atoms whose atomic p
* cn -- the output NetCDF file name. A CDL version of this file will al
 instead of the '.nc' extension.
* pyroserver -- a string specifying if Pyro will be used and how to run the ana

```

Running modes:

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the a

Comments:

- This code contains a pyrex function for the distance histogram calculation than en  
performance.

### 8.2.1 Methods

**`__init__(self)`**

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**`initialize(self)`**

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *frameIndex*, *trajname*)

Calculates the contribution for one frame.

**Parameters**

**frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
*(type=integer.)*

**trajname:** the name of the trajectory file name.  
*(type=string)*

**combine**(*self*, *frameIndex*, *x*)

**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

### 8.2.2 Class Variables

| Name                              | Description                                                                              |
|-----------------------------------|------------------------------------------------------------------------------------------|
| <code>inputParametersNames</code> | <b>Value:</b> <code>'trajectory', 'timeinfo', 'rvalues', 'group', 'subset', '...'</code> |
| <code>shortName</code>            | <b>Value:</b> <code>'CN'</code>                                                          |
| <code>canBeEstimated</code>       | <b>Value:</b> <code>True</code>                                                          |

## 8.3 Class *ScrewFitAnalysis*

`nMOLDYN.Analysis.Analysis.Analysis`



`nMOLDYN.Analysis.Structure.ScrewFitAnalysis`

Set up a Screw Fit analysis.

A Subclass of `nMOLDYN.Analysis.Analysis`.

Constructor: `ScrewFit(|parameters| = None)`

**Arguments:**

- |parameters| -- a dictionary of the input parameters, or 'None' to set up the analysis
- \* trajectory -- a trajectory file name or an instance of MMTK.Trajectory.Trajectory
- \* timeinfo -- a string of the form 'first:last:step' where 'first' is an integer number to consider, 'last' is an integer specifying the last frame, 'step' is an integer specifying the step number between two frames
- \* sfa -- the output NetCDF file name. A CDL version of this file will all instead of the '.nc' extension.
- \* pyroserver -- a string specifying if Pyro will be used and how to run the analysis

**Running modes:**

- To run the analysis do: `a.runAnalysis()` where `a` is the analysis object.
- To estimate the analysis do: `a.estimateAnalysis()` where `a` is the analysis object.
- To save the analysis to 'file' file name do: `a.saveAnalysis(file)` where `a` is the analysis object.

**Comments:**

- This code is based on a first implementation made by Paolo Calligari.
- For more details: Kneller, G.R., Calligari, P. Acta Crystallographica , D62, 302-307

**8.3.1 Methods**

|                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b><code>__init__(self)</code></b>                                                                                                            |
| The constructor. Insures that the class can not be instanciated directly from here.                                                           |
| <b>Parameters</b>                                                                                                                             |
| <b>parameters:</b> a dictionary that contains parameters of the selected analysis.                                                            |
| <b>statusBar:</b> if not None, an instance of <code>nMOLDYN.GUI.Widgets.StatusBar</code> . Will attach a status bar to the selected analysis. |
| Overrides: <code>nMOLDYN.Analysis.Analysis.Analysis.__init__</code>                                                                           |

|                                                                                             |
|---------------------------------------------------------------------------------------------|
| <b><code>initialize(self)</code></b>                                                        |
| Initializes the analysis (e.g. parses and checks input parameters, set some variables ...). |

**calc**(*self*, *frameIndex*, *trajname*)

Calculates the contribution for one frame.

**Parameters**

**frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
*(type=integer.)*

**trajname:** the name of the trajectory file name.  
*(type=string)*

**combine**(*self*, *frameIndex*, *x*)

**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

**findQuaternionMatrix**(*self*, *peptide*, *point\_ref*, *conf1*, *conf2=None*,  
*matrix=True*)

Returns the complete matrix of quaternions compatibles with linear trasformation. `|conf1|` is the reference configuration. `|point_ref|` is the reference point about which the fit is calculated

**findGenericTransformation**(*self*, *peptide*, *point\_ref*, *conf1*, *conf2=None*)

**angularDistance**(*self*, *chain*)

**screwMotionAnalysis**(*self*, *chain*)

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

### 8.3.2 Class Variables

| Name                              | Description                                                  |
|-----------------------------------|--------------------------------------------------------------|
| <code>inputParametersNames</code> | <b>Value:</b> 'trajectory', 'timeinfo', 'sfa', 'pyroserver', |
| <code>shortName</code>            | <b>Value:</b> 'SFA'                                          |
| <code>canBeEstimated</code>       | <b>Value:</b> True                                           |

## 8.4 Class *SpatialDensity*

```
nMOLDYN.Analysis.Analysis.Analysis ┌
 │
 └─ nMOLDYN.Analysis.Structure.SpatialDensity
```

Sets up a Spatial Density analysis.

A Subclass of *nMOLDYN.Analysis.Analysis*.

Constructor: *SpatialDensity*(*|parameters|* = None)

Arguments:

- *|parameters|* -- a dictionary of the input parameters, or 'None' to set up the analysis.
- \* *trajectory* -- a trajectory file name or an instance of *MMTK.Trajectory.Trajectory*.
- \* *timeinfo* -- a string of the form 'first:last:step' where 'first' is an integer number to consider, 'last' is an integer specifying the last frame, 'step' is an integer specifying the step number between two frames.
- \* *rvalues* -- a string of the form 'rmin:rmax:dr' where 'rmin' is a float specifying the minimum distance to consider, 'rmax' is a float specifying the maximum distance value, 'dr' is a float specifying the distance increment.
- \* *group* -- a selection string specifying the groups of atoms that will be used to compute the coordination number. For each group, there will be a gravity of the group.
- \* *atomorder* -- a string of the form 'atom1,atom2,atom3' where 'atom1', 'atom2', 'atom3' respectively the MMTK atom names of the atoms in the way they should be considered.
- \* *target* -- a selection string specifying the groups of atoms that will be used to compute the coordination number. For each group, there will be a gravity of the group.
- \* *sd* -- the output NetCDF file name. A CDL version of this file will also be generated instead of the '.nc' extension.
- \* *pyroserver* -- a string specifying if Pyro will be used and how to run the analysis.

Running modes:

- To run the analysis do: *a.runAnalysis()* where *a* is the analysis object.
- To estimate the analysis do: *a.estimateAnalysis()* where *a* is the analysis object.
- To save the analysis to 'file' file name do: *a.saveAnalysis(file)* where *a* is the analysis object.

Comments:

- This code contains a pyrex function for the distance histogram calculation than improves performance.

## 8.4.1 Methods

**\_\_init\_\_**(*self*)

The constructor. Insures that the class can not be instanciated directly from here.

**Parameters**

**parameters:** a dictionnary that contains parameters of the selected analysis.

**statusBar:** if not None, an instance of `nMOLDYN.GUI.Widgets.StatusBar`. Will attach a status bar to the selected analysis.

Overrides: `nMOLDYN.Analysis.Analysis.Analysis.__init__`

**initialize**(*self*)

Initializes the analysis (e.g. parses and checks input parameters, set some variables ...).

**calc**(*self*, *frameIndex*, *trajname*)

Calculates the contribution for one frame.

**Parameters**

**frameIndex:** the index of the frame in `|self.frameIndexes|` array.  
(*type=integer.*)

**trajname:** the name of the trajectory file name.  
(*type=string*)

**combine**(*self*, *frameIndex*, *x*)**finalize**(*self*)

Finalizes the calculations (e.g. averaging the total term, output files creations ...).

**constructBasisFromAtoms**(*self*, *triplet*)

This method construct a set of three oriented orthonormal axes *i*, *j*, *k* from a triplet such as (*i*,*j*,*k*) forms a clockwise orthonormal basis.

If *a1*, *a2* and *a3* stand respectively for the three atoms of the triplet then:

vector1 = (vector(*a1*,*a2*)\_normalized + vector(*a1*,*a3*)\_normalized)\_normalized

vector3 = (vector1 ^ vector(*a1*,*a3*))\_normalized and correctly oriented

vector2 = (vector3 ^ vector1)\_normalized

@param triplet: the triplet of atoms.

@type triplet: a list of three MMTK Atoms

@return: the three axis.

@rtype: a list of three Scientific Vector

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

#### 8.4.2 Class Variables

| Name                 | Description                                                                |
|----------------------|----------------------------------------------------------------------------|
| inputParametersNames | <b>Value:</b> 'trajectory', 'timeinfo', 'rvalues', 'thetavalues', 'phiv... |
| shortName            | <b>Value:</b> 'SD'                                                         |
| canBeEstimated       | <b>Value:</b> True                                                         |

## 9 Module nMOLDYN.Analysis.Template

This modules implements the estimate, serial and parrallel templates for all analysis.

### 9.1 Functions

**startSlaves**(*taskName*, *pyroServer*, *pyroNodes*)

Starts the slaves.

**Parameters**

**pyroServer**: the type of pyro server. One of 'multiprocessor' or 'cluster'.

(*type=string.*)

**pyroNodes**: a dictionary whose keys are the name of the nodes and the value the number of cpus to allocate to this node.

(*type=dict.*)

### 9.2 Variables

| Name                 | Description                                                  |
|----------------------|--------------------------------------------------------------|
| nmoldyn_package_path | <b>Value:</b><br>os.path.dirname(os.path.split(__file__)[0]) |

### 9.3 Class SerialPerAtom

Template class for an analysis atom-by-atom ran in serial mode.

#### 9.3.1 Methods

**internalRun**(*self*)

Performs the analysis in serial mode.

### 9.4 Class ParallelPerAtom

Template class for an analysis atom-by-atom ran in parallel mode.

### 9.4.1 Methods

|                                         |
|-----------------------------------------|
| <b>internalRun</b> ( <i>self</i> )      |
| Performs the analysis in parallel mode. |

## 9.5 Class **SerialPerFrame**

Template class for an analysis atom-by-atom ran in serial mode.

### 9.5.1 Methods

|                                       |
|---------------------------------------|
| <b>internalRun</b> ( <i>self</i> )    |
| Performs the analysis in serial mode. |

## 9.6 Class **ParallelPerFrame**

Template class for an analysis frame-by-frame ran in parallel mode.

### 9.6.1 Methods

|                                         |
|-----------------------------------------|
| <b>internalRun</b> ( <i>self</i> )      |
| Performs the analysis in parallel mode. |

## 9.7 Class **SerialPerGroup**

Template class for an analysis group-by-group ran in serial mode.

### 9.7.1 Methods

|                                       |
|---------------------------------------|
| <b>internalRun</b> ( <i>self</i> )    |
| Performs the analysis in serial mode. |

## 9.8 Class **ParallelPerGroup**

Template class for an analysis group-by-group ran in parallel mode.

### 9.8.1 Methods

**internalRun**(*self*)

Performs the analysis in parallel mode.

## 9.9 Class **SerialPerQShell**

Template class for an analysis qshell-by-qshell ran in serial mode.

### 9.9.1 Methods

**internalRun**(*self*)

Performs the analysis in serial mode.

## 9.10 Class **ParallelPerQShell**

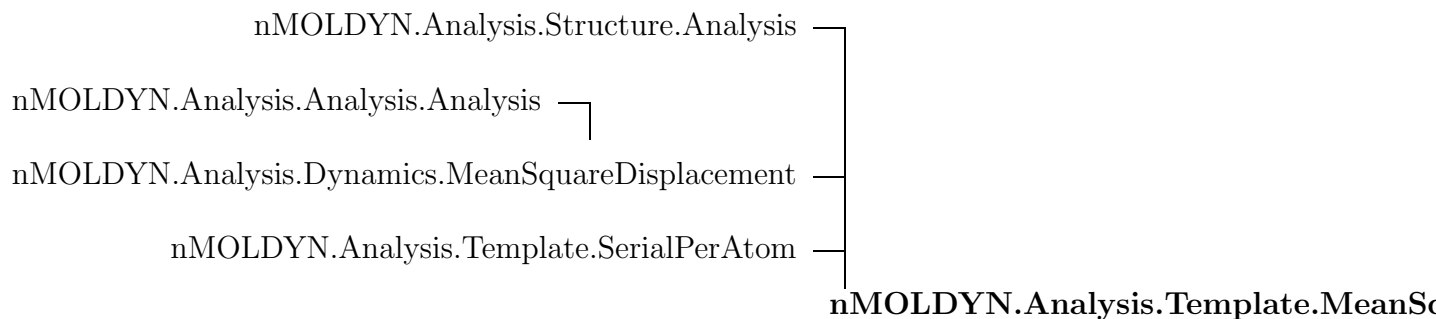
Template class for an analysis qshell-by-qshell ran in parallel mode.

### 9.10.1 Methods

**internalRun**(*self*)

Performs the analysis in parallel mode.

## 9.11 Class **MeanSquareDisplacement\_serial**



### 9.11.1 Methods

*Inherited from nMOLDYN.Analysis.Dynamics.MeanSquareDisplacement(Section 4.1)*

`__init__()`, `atomicMSD()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

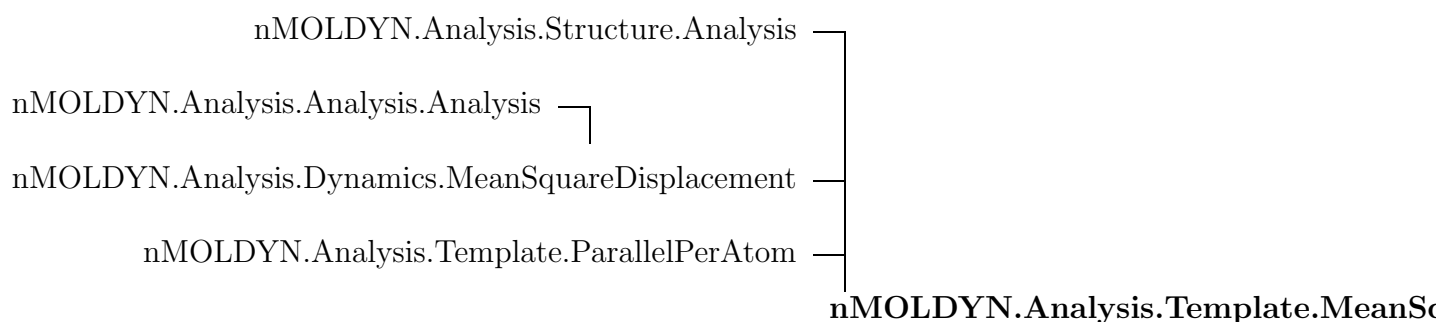
*Inherited from nMOLDYN.Analysis.Template.SerialPerAtom(Section 9.3)*

`internalRun()`

### 9.11.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.MeanSquareDisplacement (Section 4.1)</i>     |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.12 Class MeanSquareDisplacement\_parallel



### 9.12.1 Methods

*Inherited from nMOLDYN.Analysis.Dynamics.MeanSquareDisplacement(Section 4.1)*

`__init__()`, `atomicMSD()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

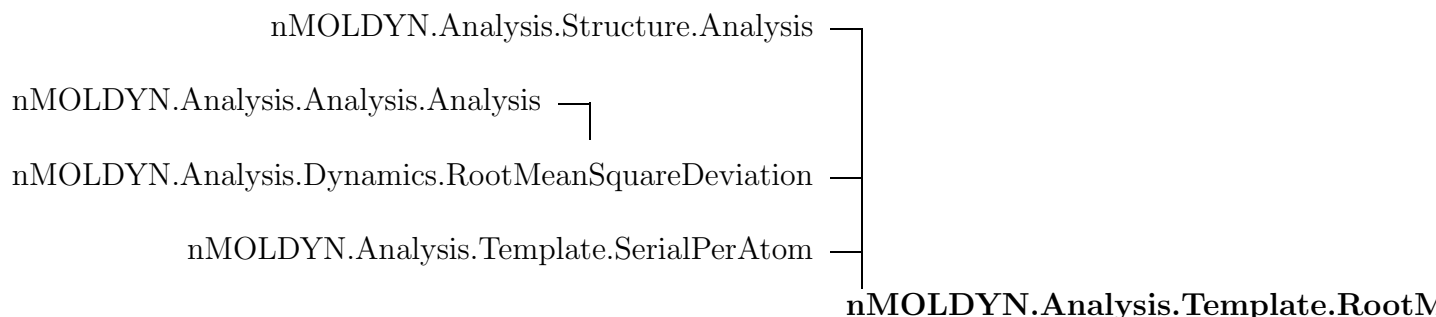
***Inherited from nMOLDYN.Analysis.Template.ParallelPerAtom(Section 9.4)***

internalRun()

### 9.12.2 Class Variables

| Name                                                                                 | Description |
|--------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.MeanSquareDisplacement (Section 4.1)</i> |             |
| canBeEstimated, inputParametersNames, shortName                                      |             |

## 9.13 Class *RootMeanSquareDeviation\_serial*



### 9.13.1 Methods

***Inherited from nMOLDYN.Analysis.Dynamics.RootMeanSquareDeviation(Section 4.2)***

\_\_init\_\_(), calc(), combine(), finalize(), initialize()

***Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)***

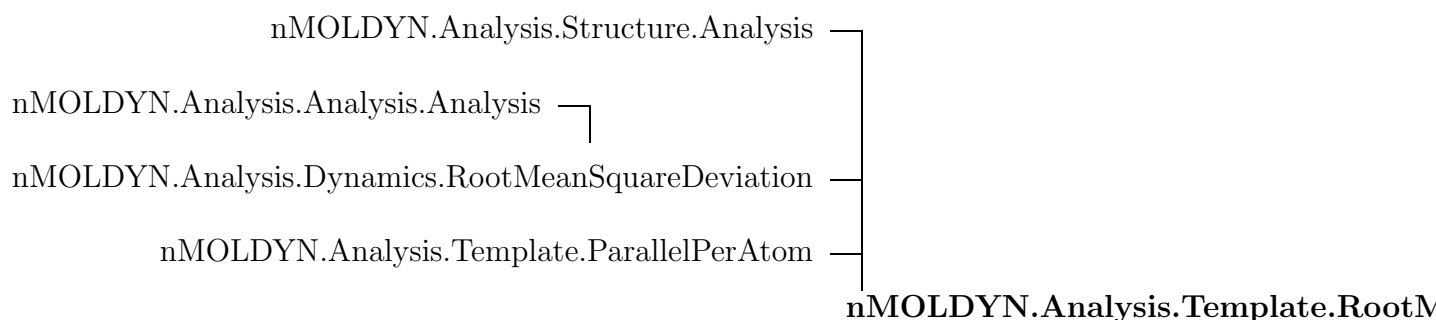
analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

***Inherited from nMOLDYN.Analysis.Template.SerialPerAtom(Section 9.3)***

internalRun()

**9.13.2 Class Variables**

| Name                                                                                                                                     | Description |
|------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.RootMeanSquareDeviation (Section 4.2)</i><br>canBeEstimated, inputParametersNames, shortName |             |

**9.14 Class RootMeanSquareDeviation\_parallel****9.14.1 Methods**

*Inherited from nMOLDYN.Analysis.Dynamics.RootMeanSquareDeviation(Section 4.2)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

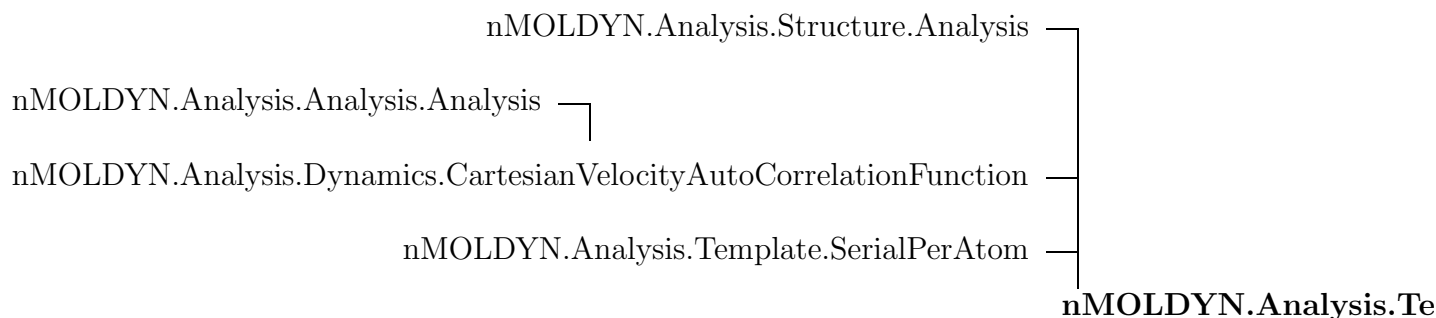
*Inherited from nMOLDYN.Analysis.Template.ParallelPerAtom(Section 9.4)*

`internalRun()`

**9.14.2 Class Variables**

| Name                                                                                                                                     | Description |
|------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.RootMeanSquareDeviation (Section 4.2)</i><br>canBeEstimated, inputParametersNames, shortName |             |

## 9.15 Class CartesianVelocityAutoCorrelationFunction\_serial



### 9.15.1 Methods

**Inherited from *nMOLDYN.Analysis.Dynamics.CartesianVelocityAutoCorrelationFunction* (Section 4.3)**

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

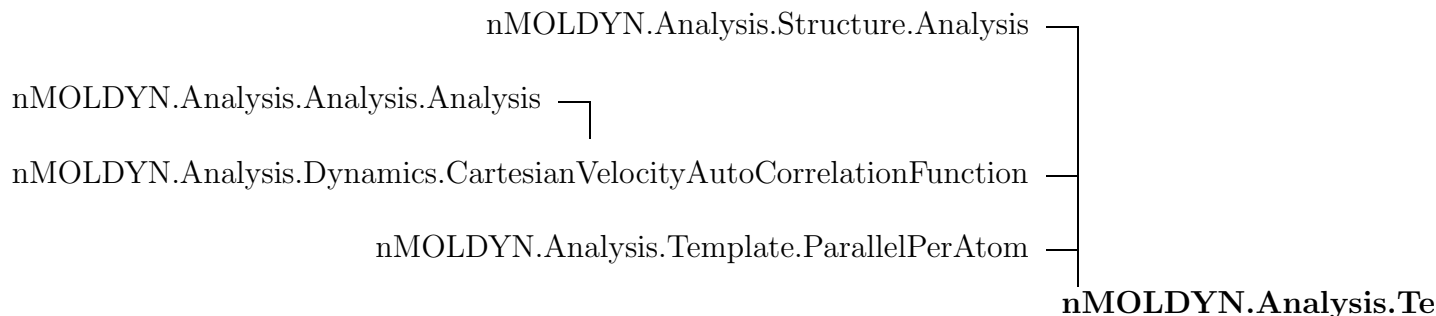
**Inherited from *nMOLDYN.Analysis.Template.SerialPerAtom* (Section 9.3)**

`internalRun()`

### 9.15.2 Class Variables

| Name                                                                                                                | Description |
|---------------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.Dynamics.CartesianVelocityAutoCorrelationFunction</code> (Section 4.3)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>                            |             |

## 9.16 Class CartesianVelocityAutoCorrelationFunction\_parallel



### 9.16.1 Methods

**Inherited from *nMOLDYN.Analysis.Dynamics.CartesianVelocityAutoCorrelationFunction* (Section 4.3)**

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

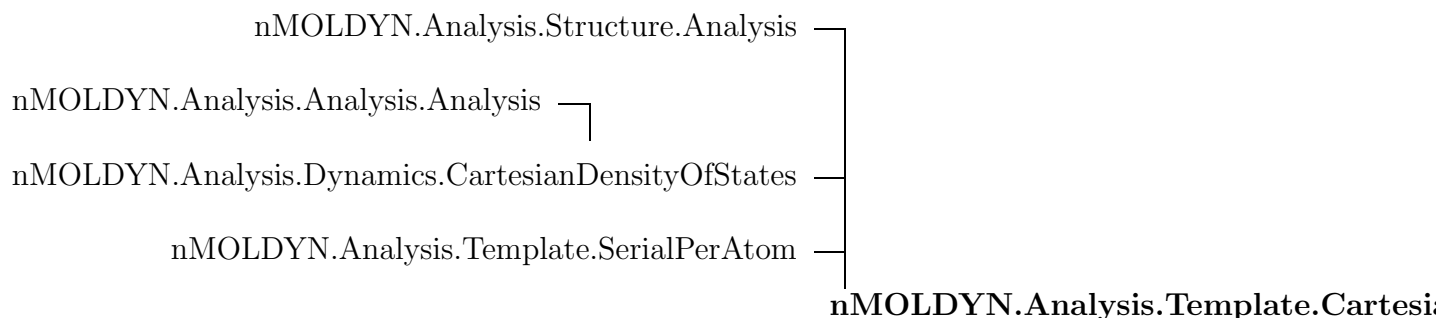
**Inherited from *nMOLDYN.Analysis.Template.ParallelPerAtom* (Section 9.4)**

`internalRun()`

### 9.16.2 Class Variables

| Name                                                                                                   | Description |
|--------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.CartesianVelocityAutoCorrelationFunction (Section 4.3)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>               |             |

## 9.17 Class *CartesianDensityOfStates\_serial*



### 9.17.1 Methods

***Inherited from `nMOLDYN.Analysis.Dynamics.CartesianDensityOfStates` (Section 4.4)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

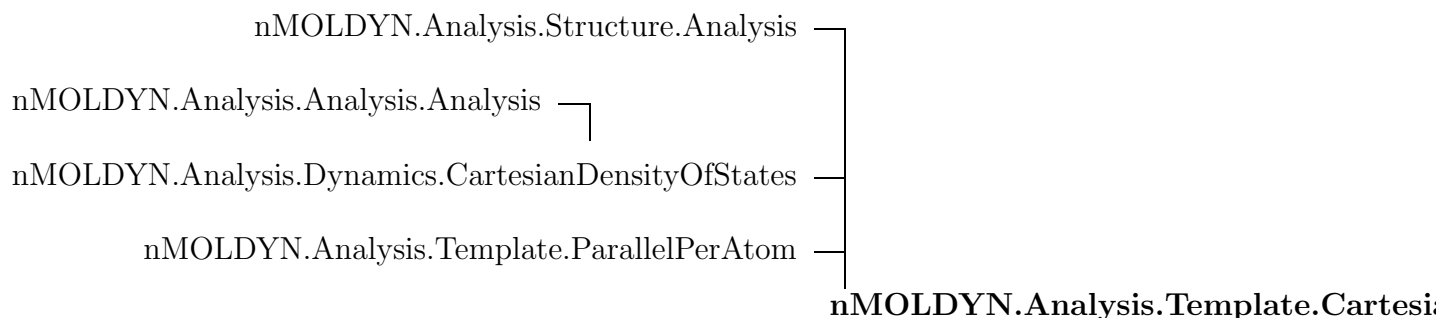
***Inherited from `nMOLDYN.Analysis.Template.SerialPerAtom` (Section 9.3)***

`internalRun()`

### 9.17.2 Class Variables

| Name                                                                                                | Description |
|-----------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.Dynamics.CartesianDensityOfStates</code> (Section 4.4)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>            |             |

## 9.18 Class *CartesianDensityOfStates\_parallel*



### 9.18.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.CartesianDensityOfStates* (Section 4.4)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

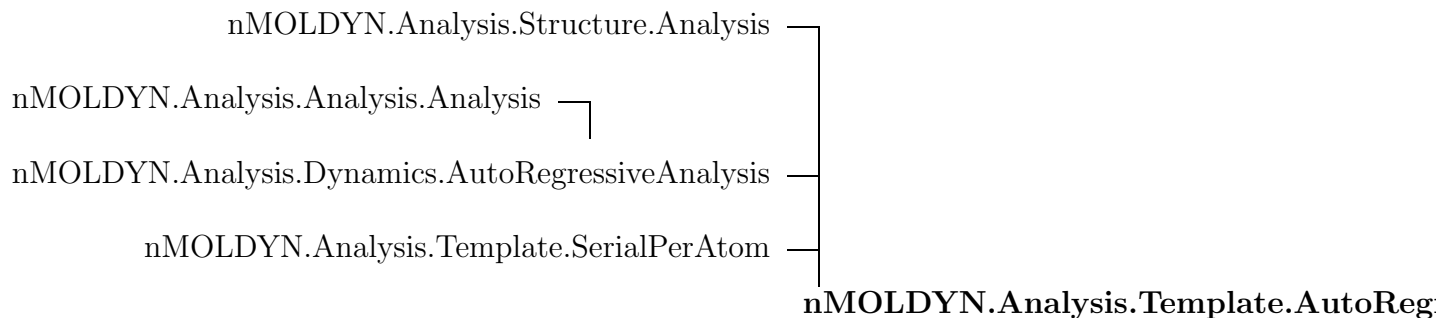
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerAtom* (Section 9.4)***

`internalRun()`

### 9.18.2 Class Variables

| Name                                                                                          | Description |
|-----------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.CartesianDensityOfStates</i> (Section 4.4)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>      |             |

## 9.19 Class *AutoRegressiveAnalysis\_serial*



### 9.19.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.AutoRegressiveAnalysis* (Section 4.5)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

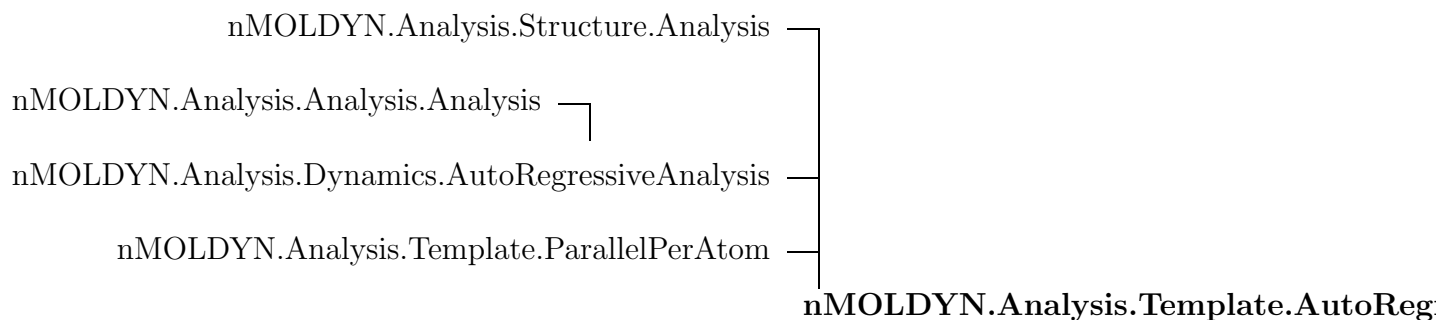
***Inherited from *nMOLDYN.Analysis.Template.SerialPerAtom* (Section 9.3)***

`internalRun()`

### 9.19.2 Class Variables

| Name                                                                                        | Description |
|---------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.AutoRegressiveAnalysis</i> (Section 4.5)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>    |             |

## 9.20 Class *AutoRegressiveAnalysis\_parallel*



### 9.20.1 Methods

***Inherited from nMOLDYN.Analysis.Dynamics.AutoRegressiveAnalysis (Section 4.5)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

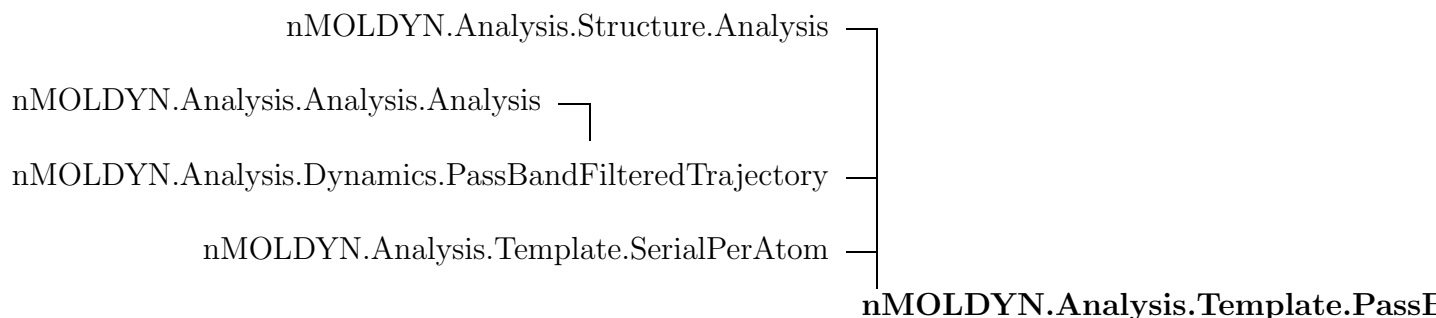
***Inherited from nMOLDYN.Analysis.Template.ParallelPerAtom (Section 9.4)***

`internalRun()`

### 9.20.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.AutoRegressiveAnalysis (Section 4.5)</i>     |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.21 Class *PassBandFilteredTrajectory\_serial*



### 9.21.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.PassBandFilteredTrajectory* (Section 4.6)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

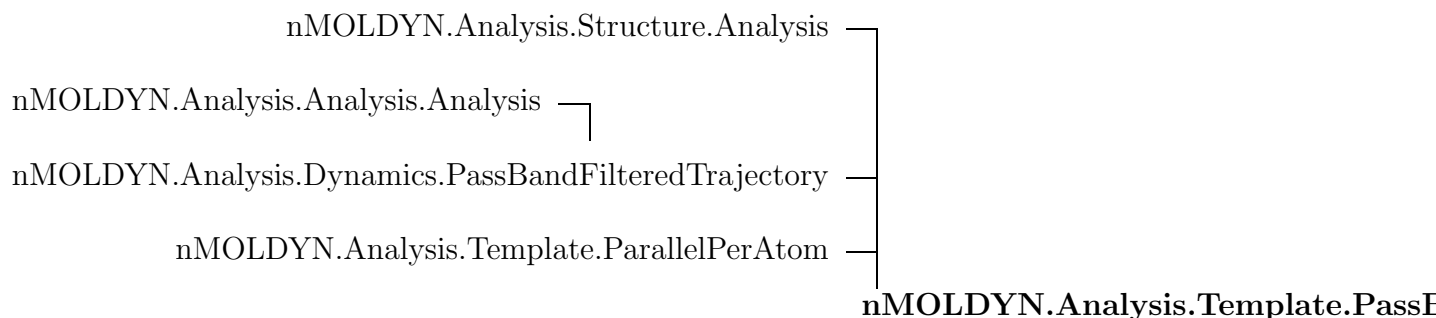
***Inherited from *nMOLDYN.Analysis.Template.SerialPerAtom* (Section 9.3)***

`internalRun()`

### 9.21.2 Class Variables

| Name                                                                                            | Description |
|-------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.PassBandFilteredTrajectory</i> (Section 4.6)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>        |             |

## 9.22 Class *PassBandFilteredTrajectory\_parallel*



### 9.22.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.PassBandFilteredTrajectory* (Section 4.6)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

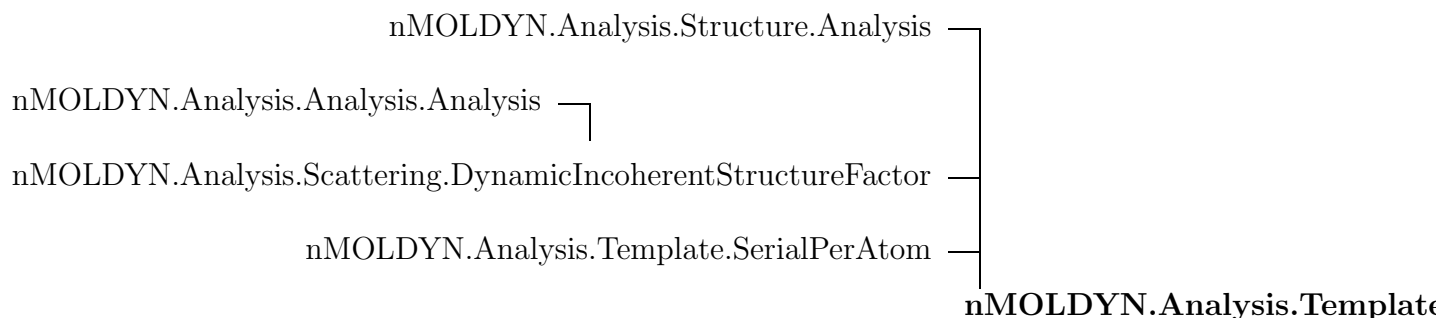
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerAtom* (Section 9.4)***

`internalRun()`

### 9.22.2 Class Variables

| Name                                                                                            | Description |
|-------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.PassBandFilteredTrajectory</i> (Section 4.6)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>        |             |

### 9.23 Class *DynamicIncoherentStructureFactor\_serial*



#### 9.23.1 Methods

***Inherited from *nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactor* (Section 6.5)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

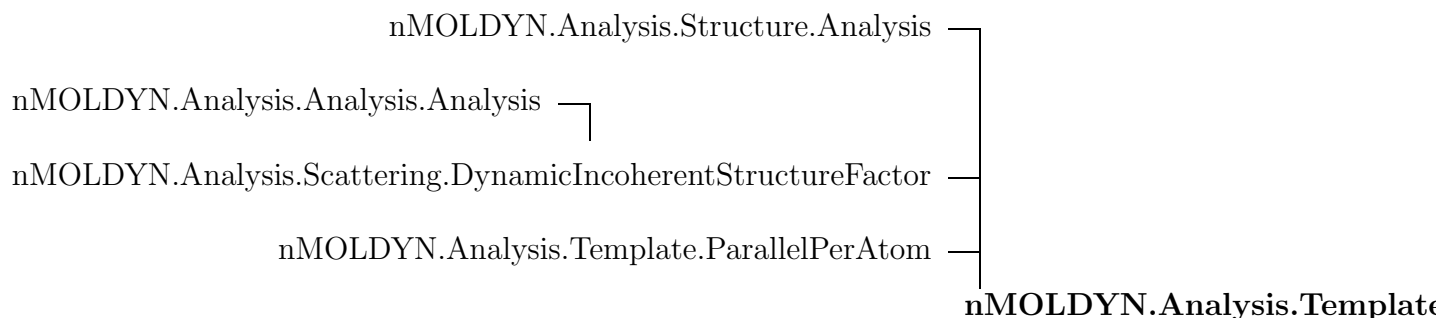
***Inherited from *nMOLDYN.Analysis.Template.SerialPerAtom* (Section 9.3)***

`internalRun()`

#### 9.23.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactor</i> (Section 6.5)</i>         |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.24 Class *DynamicIncoherentStructureFactor\_parallel*



### 9.24.1 Methods

***Inherited from *nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactor* (Section 6.5)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

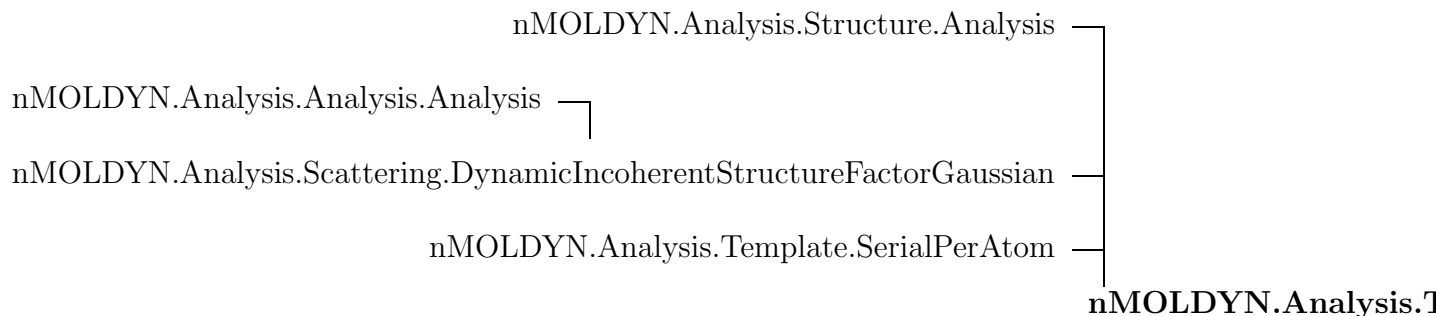
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerAtom* (Section 9.4)***

`internalRun()`

### 9.24.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactor</i> (Section 6.5)</i>         |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.25 Class *DynamicIncoherentStructureFactorGaussian\_serial*



### 9.25.1 Methods

***Inherited from nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorGaussian (Section 6.7)***

`_init_()`, `calc()`, `combine()`, `finalize()`, `getMSD()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

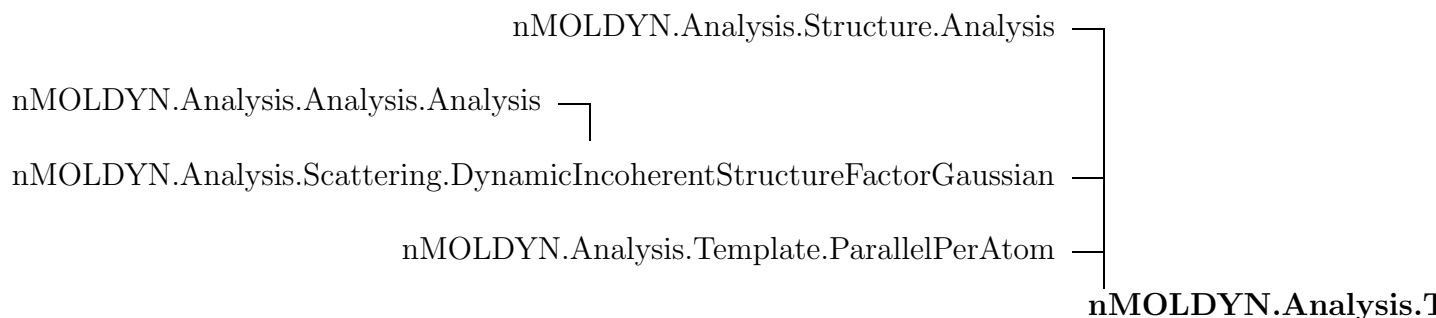
***Inherited from nMOLDYN.Analysis.Template.SerialPerAtom (Section 9.3)***

`internalRun()`

### 9.25.2 Class Variables

| Name                                                                                                     | Description                                                                                                     |
|----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <i>Inherited from nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorGaussian (Section 6.7)</i> | <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |

## 9.26 Class *DynamicIncoherentStructureFactorGaussian\_parallel*



### 9.26.1 Methods

***Inherited from nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorGaussian (Section 6.7)***

`_init_()`, `calc()`, `combine()`, `finalize()`, `getMSD()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

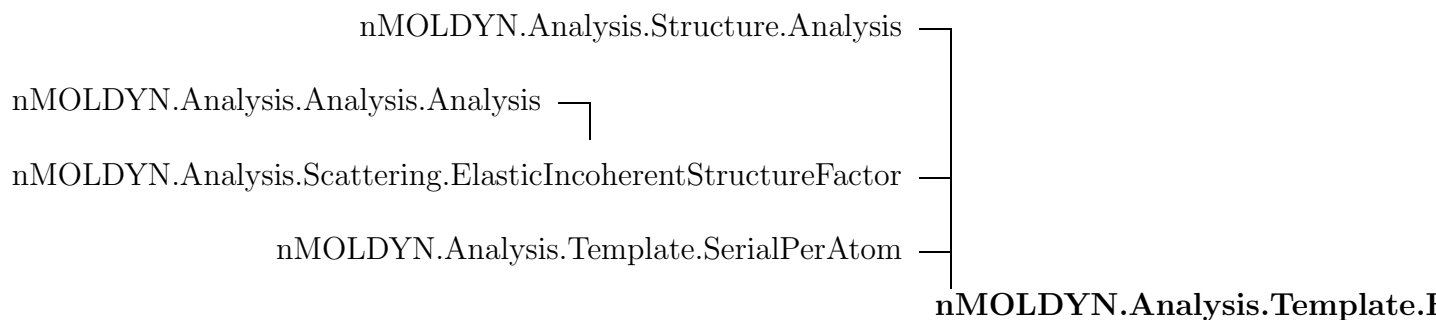
***Inherited from nMOLDYN.Analysis.Template.ParallelPerAtom (Section 9.4)***

`internalRun()`

### 9.26.2 Class Variables

| Name                                                                                                     | Description                                                                                                     |
|----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <i>Inherited from nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorGaussian (Section 6.7)</i> | <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |

## 9.27 Class *ElasticIncoherentStructureFactor\_serial*



### 9.27.1 Methods

*Inherited from nMOLDYN.Analysis.Scattering.ElasticIncoherentStructureFactor (Section 6.8)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

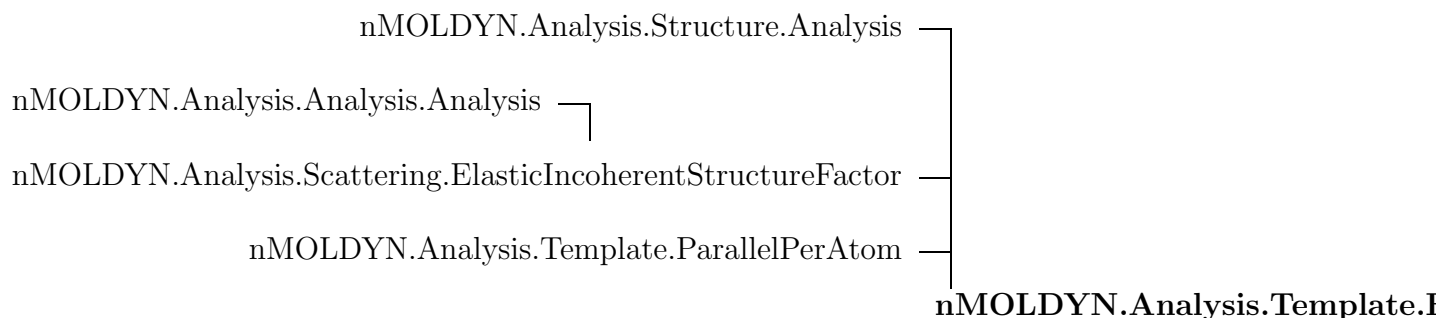
*Inherited from nMOLDYN.Analysis.Template.SerialPerAtom (Section 9.3)*

`internalRun()`

### 9.27.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Scattering.ElasticIncoherentStructureFactor (Section 6.8)</i>                |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.28 Class *ElasticIncoherentStructureFactor\_parallel*



### 9.28.1 Methods

***Inherited from `nMOLDYN.Analysis.Scattering.ElasticIncoherentStructureFactor` (Section 6.8)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

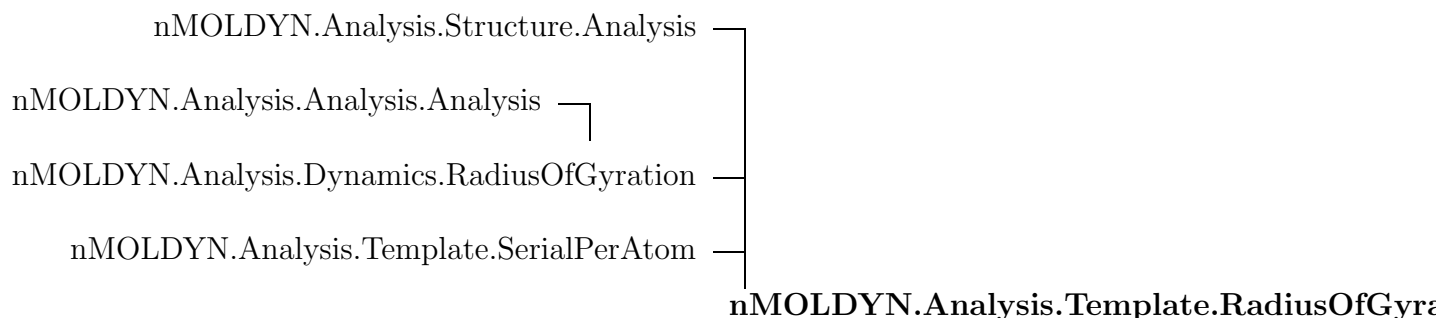
***Inherited from `nMOLDYN.Analysis.Template.ParallelPerAtom` (Section 9.4)***

`internalRun()`

### 9.28.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.Scattering.ElasticIncoherentStructureFactor</code> (Section 6.8)</i>   |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.29 Class RadiusOfGyration\_serial



### 9.29.1 Methods

**Inherited from *nMOLDYN.Analysis.Dynamics.RadiusOfGyration* (Section 4.7)**

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

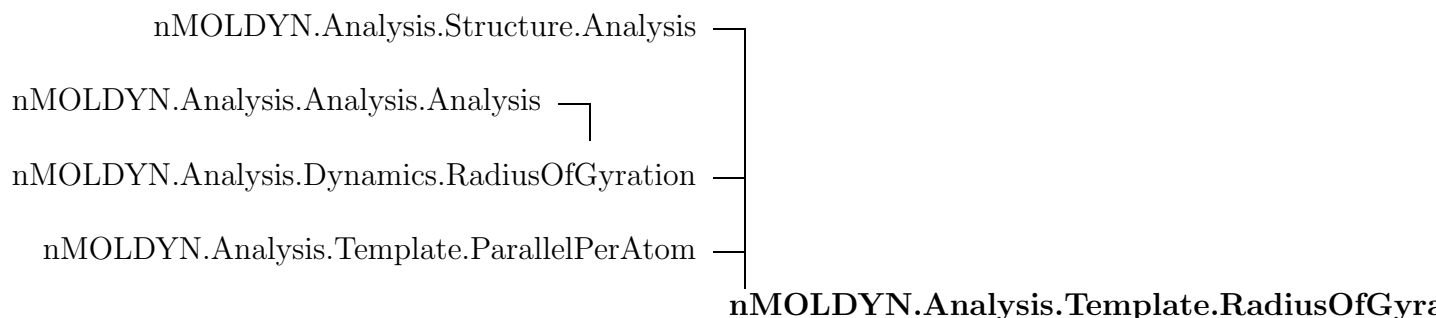
**Inherited from *nMOLDYN.Analysis.Template.SerialPerAtom* (Section 9.3)**

`internalRun()`

### 9.29.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.RadiusOfGyration (Section 4.7)</i>           |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.30 Class *RadiusOfGyration\_parallel*



#### 9.30.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.RadiusOfGyration* (Section 4.7)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

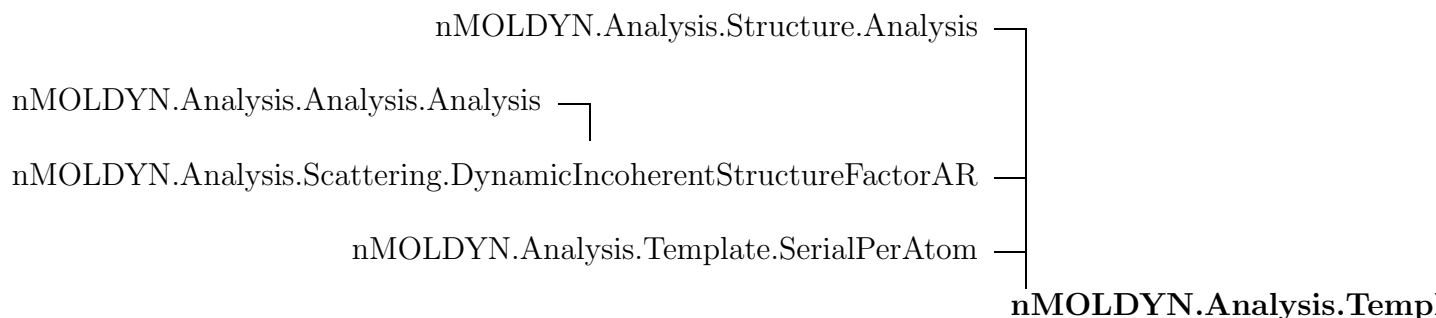
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerAtom* (Section 9.4)***

`internalRun()`

#### 9.30.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.RadiusOfGyration</i> (Section 4.7)</i>    |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.31 Class *DynamicIncoherentStructureFactorAR\_serial*



#### 9.31.1 Methods

***Inherited from nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorAR (Section 6.6)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

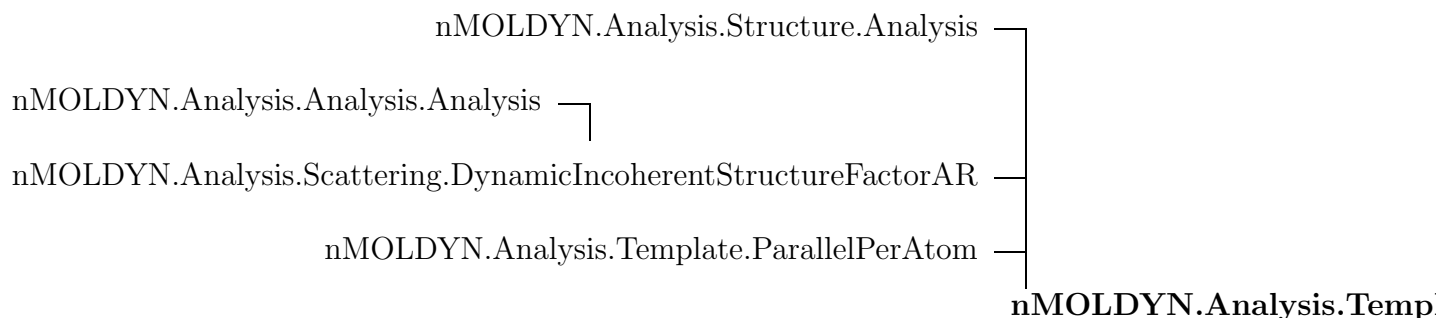
***Inherited from nMOLDYN.Analysis.Template.SerialPerAtom (Section 9.3)***

`internalRun()`

#### 9.31.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorAR (Section 6.6)</i>              |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.32 Class *DynamicIncoherentStructureFactorAR\_parallel*



#### 9.32.1 Methods

***Inherited from *nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorAR* (Section 6.6)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

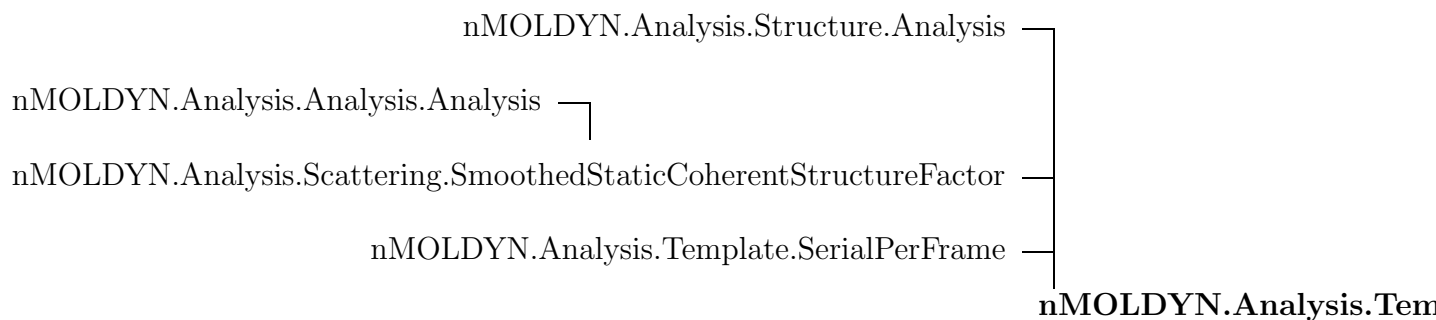
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerAtom* (Section 9.4)***

`internalRun()`

#### 9.32.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Scattering.DynamicIncoherentStructureFactorAR</i> (Section 6.6)</i>       |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.33 Class *SmoothedStaticCoherentStructureFactor\_serial*



#### 9.33.1 Methods

***Inherited from *nMOLDYN.Analysis.Scattering.SmoothedStaticCoherentStructureFactor* (Section 6.9)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

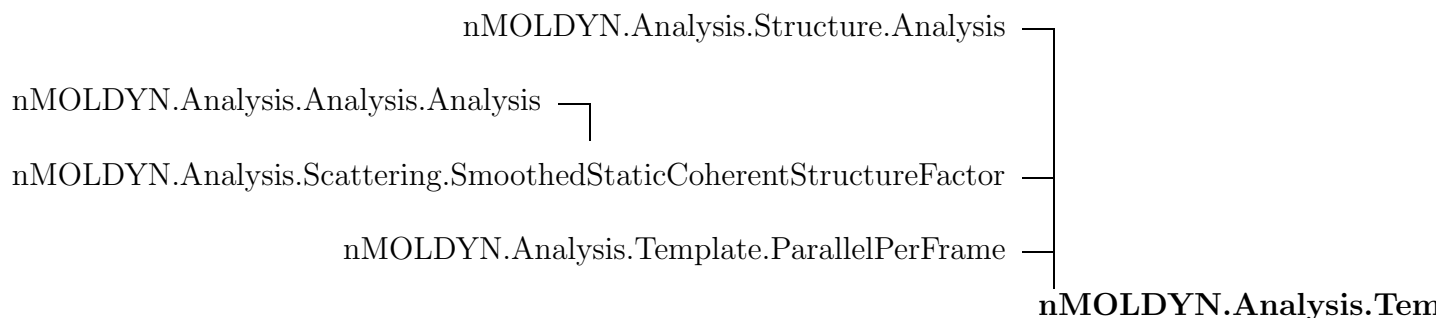
***Inherited from *nMOLDYN.Analysis.Template.SerialPerFrame* (Section 9.5)***

`internalRun()`

#### 9.33.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Scattering.SmoothedStaticCoherentStructureFactor</i> (Section 6.9)</i>    |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.34 Class *SmoothedStaticCoherentStructureFactor\_parallel*



#### 9.34.1 Methods

***Inherited from `nMOLDYN.Analysis.Scattering.SmoothedStaticCoherentStructureFactor` (Section 6.9)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

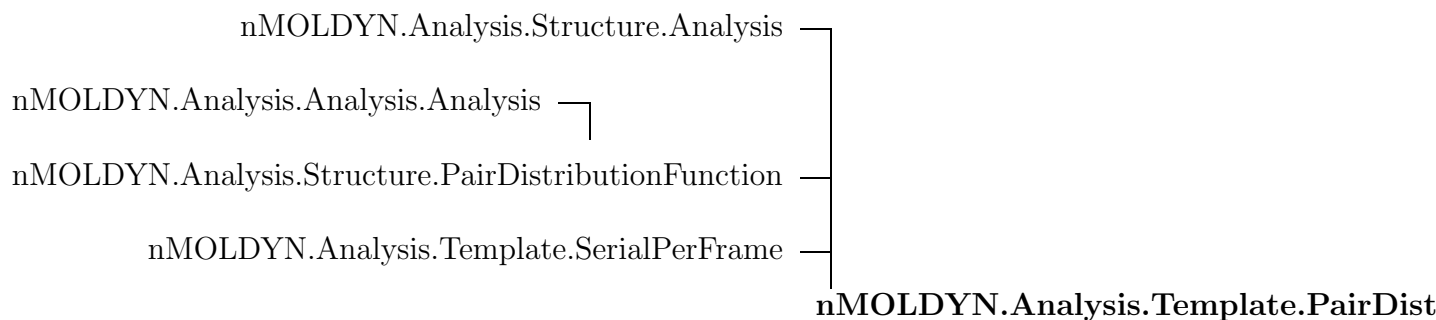
***Inherited from `nMOLDYN.Analysis.Template.ParallelPerFrame` (Section 9.6)***

`internalRun()`

#### 9.34.2 Class Variables

| Name                                                                                                               | Description |
|--------------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.Scattering.SmoothedStaticCoherentStructureFactor</code> (Section 6.9)</i> |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code>    |             |

### 9.35 Class *PairDistributionFunction\_serial*



#### 9.35.1 Methods

***Inherited from *nMOLDYN.Analysis.Structure.PairDistributionFunction* (Section 8.1)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

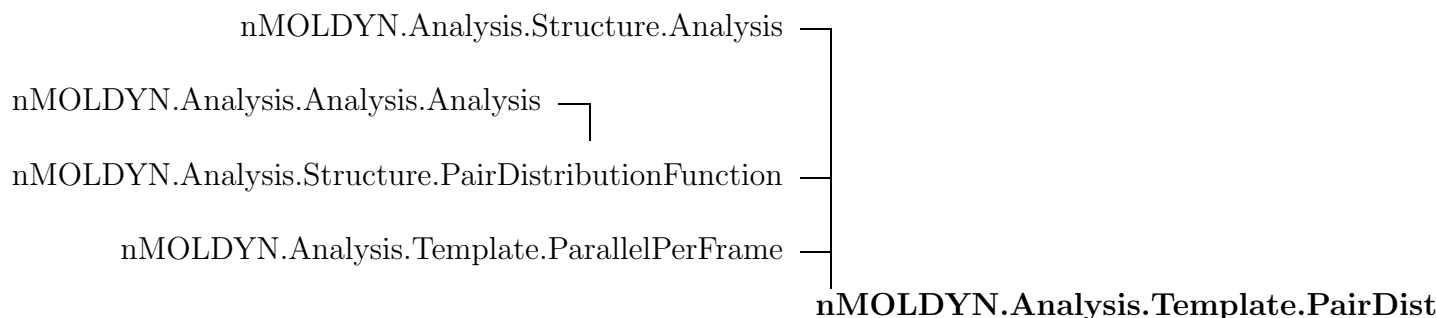
***Inherited from *nMOLDYN.Analysis.Template.SerialPerFrame* (Section 9.5)***

`internalRun()`

#### 9.35.2 Class Variables

| Name                                                                                           | Description |
|------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Structure.PairDistributionFunction</i> (Section 8.1)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>       |             |

### 9.36 Class *PairDistributionFunction\_parallel*



#### 9.36.1 Methods

***Inherited from nMOLDYN.Analysis.Structure.PairDistributionFunction (Section 8.1)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

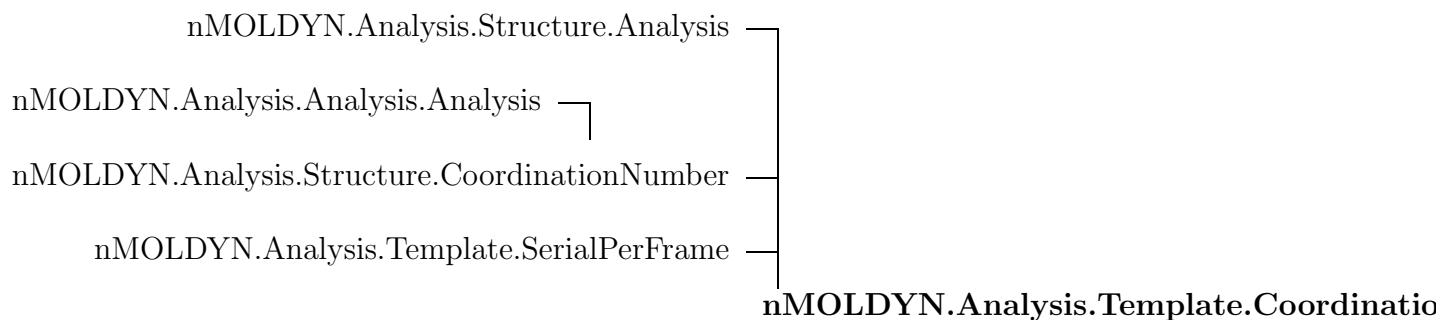
***Inherited from nMOLDYN.Analysis.Template.ParallelPerFrame (Section 9.6)***

`internalRun()`

#### 9.36.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Structure.PairDistributionFunction (Section 8.1)</i>  |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.37 Class CoordinationNumber\_serial



#### 9.37.1 Methods

*Inherited from nMOLDYN.Analysis.Structure.CoordinationNumber (Section 8.2)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

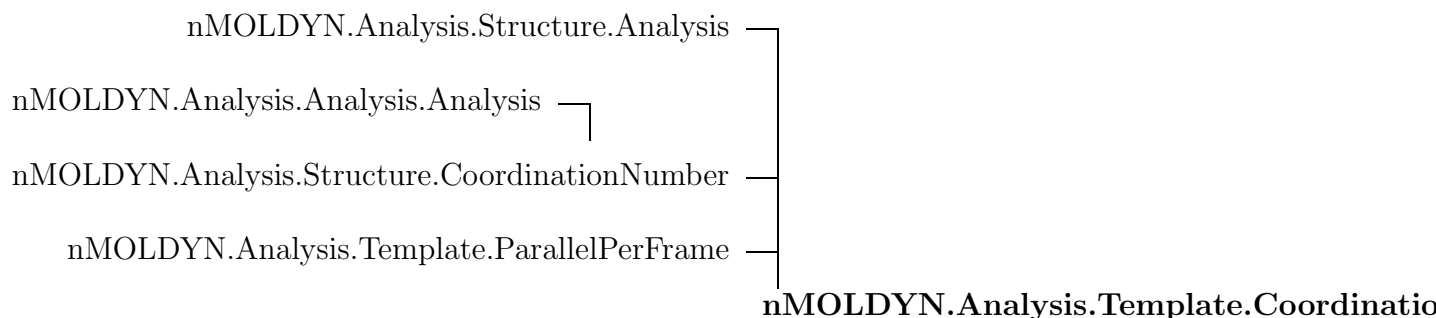
*Inherited from nMOLDYN.Analysis.Template.SerialPerFrame (Section 9.5)*

`internalRun()`

#### 9.37.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Structure.CoordinationNumber (Section 8.2)</i>        |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.38 Class CoordinationNumber\_parallel



#### 9.38.1 Methods

**Inherited from *nMOLDYN.Analysis.Structure.CoordinationNumber* (Section 8.2)**

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

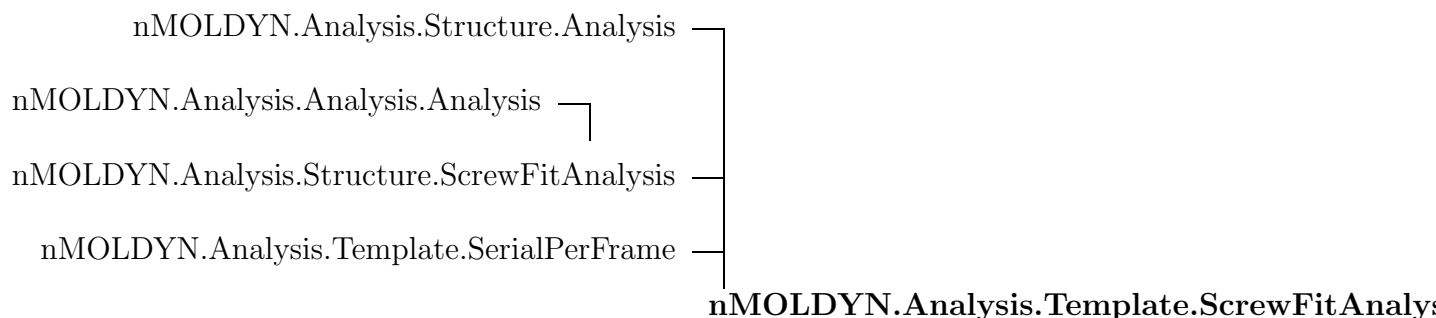
**Inherited from *nMOLDYN.Analysis.Template.ParallelPerFrame* (Section 9.6)**

`internalRun()`

#### 9.38.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Structure.CoordinationNumber (Section 8.2)</i>        |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.39 Class *ScrewFitAnalysis\_serial*



#### 9.39.1 Methods

***Inherited from nMOLDYN.Analysis.Structure.ScrewFitAnalysis (Section 8.3)***

`__init__()`, `angularDistance()`, `calc()`, `combine()`, `finalize()`, `findGenericTransformation()`, `findQuaternionMatrix()`, `initialize()`, `screwMotionAnalysis()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

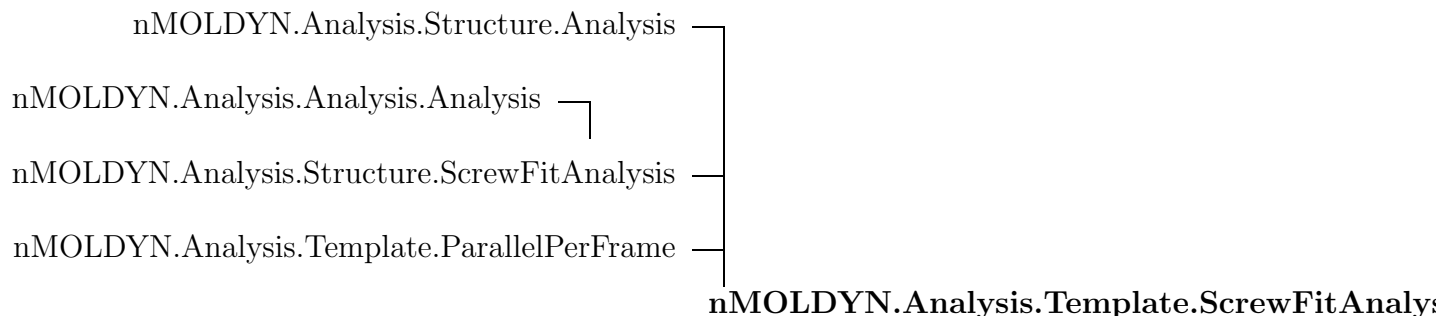
***Inherited from nMOLDYN.Analysis.Template.SerialPerFrame (Section 9.5)***

`internalRun()`

#### 9.39.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Structure.ScrewFitAnalysis (Section 8.3)</i>          |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.40 Class *ScrewFitAnalysis\_parallel*



### 9.40.1 Methods

***Inherited from *nMOLDYN.Analysis.Structure.ScrewFitAnalysis* (Section 8.3)***

`__init__()`, `angularDistance()`, `calc()`, `combine()`, `finalize()`, `findGenericTransformation()`, `findQuaternionMatrix()`, `initialize()`, `screwMotionAnalysis()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

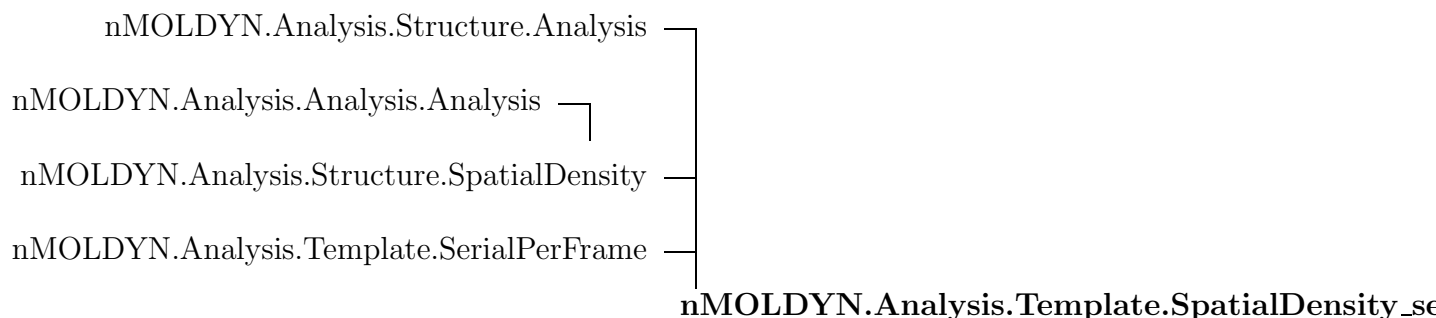
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerFrame* (Section 9.6)***

`internalRun()`

### 9.40.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Structure.ScrewFitAnalysis</i> (Section 8.3)</i>   |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.41 Class *SpatialDensity\_serial*



### 9.41.1 Methods

***Inherited from *nMOLDYN.Analysis.Structure.SpatialDensity* (Section 8.4)***

`__init__()`, `calc()`, `combine()`, `constructBasisFromAtoms()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

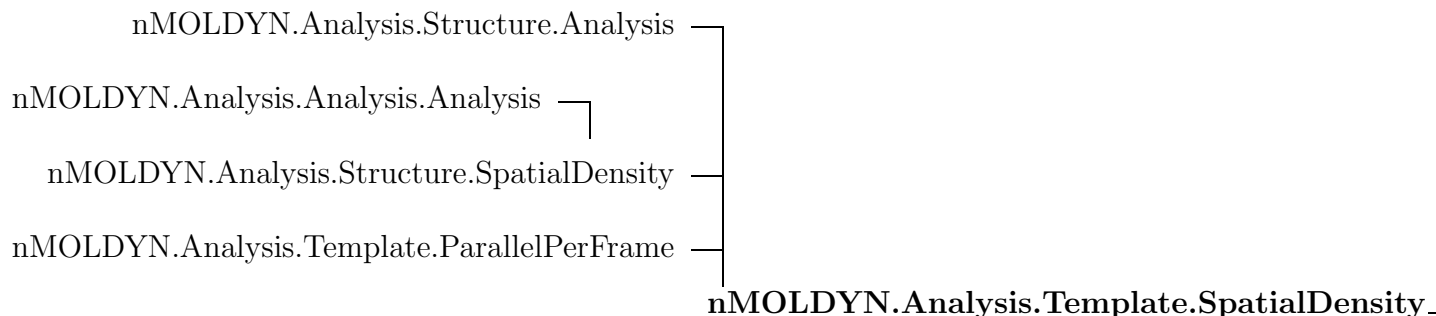
***Inherited from *nMOLDYN.Analysis.Template.SerialPerFrame* (Section 9.5)***

`internalRun()`

### 9.41.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Structure.SpatialDensity</i> (Section 8.4)</i>     |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.42 Class *SpatialDensity\_parallel*



### 9.42.1 Methods

***Inherited from *nMOLDYN.Analysis.Structure.SpatialDensity* (Section 8.4)***

`__init__()`, `calc()`, `combine()`, `constructBasisFromAtoms()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

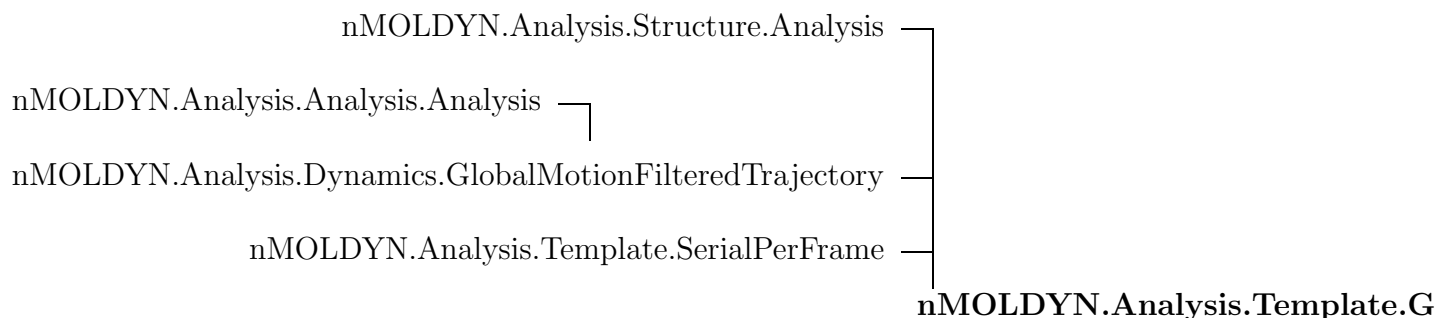
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerFrame* (Section 9.6)***

`internalRun()`

### 9.42.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Structure.SpatialDensity</i> (Section 8.4)</i>     |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.43 Class *GlobalMotionFilteredTrajectory\_serial*



#### 9.43.1 Methods

***Inherited from `nMOLDYN.Analysis.Dynamics.GlobalMotionFilteredTrajectory` (Section 4.8)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

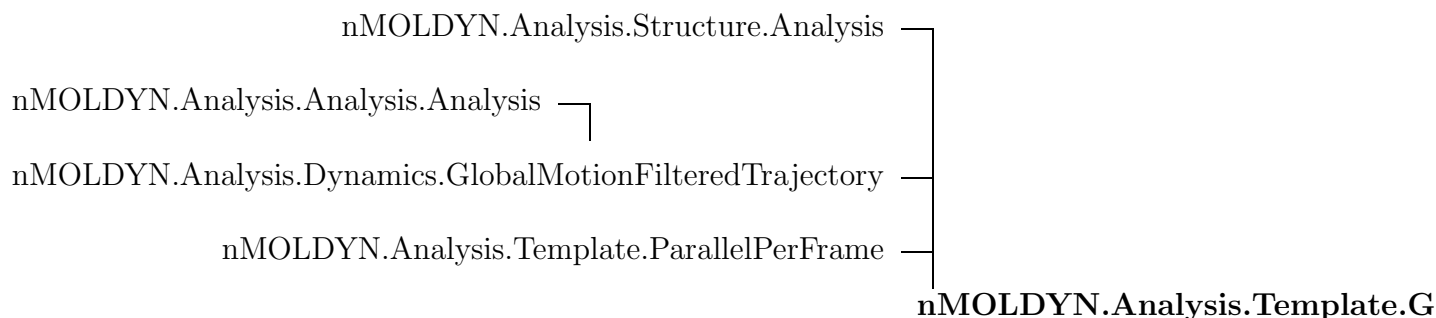
***Inherited from `nMOLDYN.Analysis.Template.SerialPerFrame` (Section 9.5)***

`internalRun()`

#### 9.43.2 Class Variables

| Name                                                                                                      | Description |
|-----------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.Dynamics.GlobalMotionFilteredTrajectory</code> (Section 4.8)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>                  |             |

## 9.44 Class *GlobalMotionFilteredTrajectory\_parallel*



### 9.44.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.GlobalMotionFilteredTrajectory* (Section 4.8)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

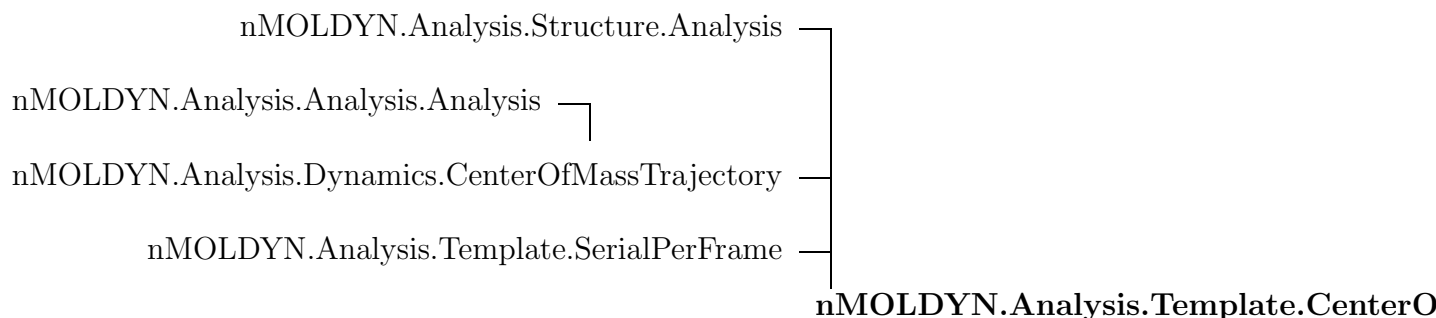
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerFrame* (Section 9.6)***

`internalRun()`

### 9.44.2 Class Variables

| Name                                                                                                | Description |
|-----------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.GlobalMotionFilteredTrajectory</i> (Section 4.8)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>            |             |

## 9.45 Class *CenterOfMassTrajectory\_serial*



### 9.45.1 Methods

***Inherited from `nMOLDYN.Analysis.Dynamics.CenterOfMassTrajectory` (Section 4.9)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)***

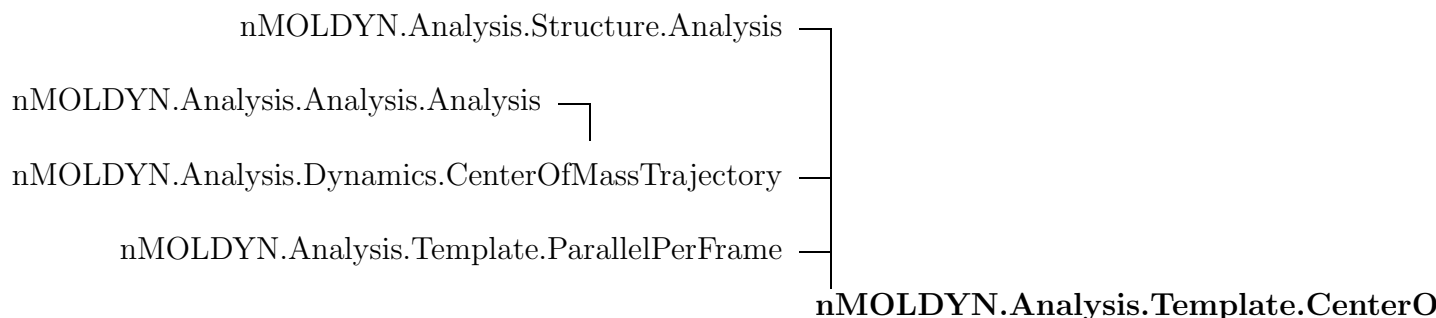
`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

***Inherited from `nMOLDYN.Analysis.Template.SerialPerFrame` (Section 9.5)***

`internalRun()`

### 9.45.2 Class Variables

| Name                                                                                              | Description |
|---------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.Dynamics.CenterOfMassTrajectory</code> (Section 4.9)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>          |             |

**9.46 Class *CenterOfMassTrajectory\_parallel*****9.46.1 Methods**

***Inherited from `nMOLDYN.Analysis.Dynamics.CenterOfMassTrajectory` (Section 4.9)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

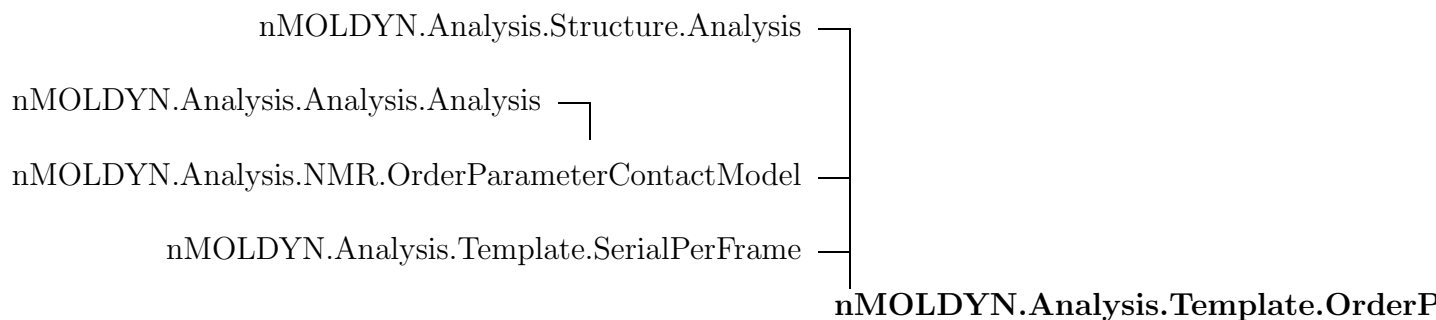
***Inherited from `nMOLDYN.Analysis.Template.ParallelPerFrame` (Section 9.6)***

`internalRun()`

**9.46.2 Class Variables**

| Name                                                                                              | Description                                                                              |
|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <i>Inherited from <code>nMOLDYN.Analysis.Dynamics.CenterOfMassTrajectory</code> (Section 4.9)</i> | <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |

## 9.47 Class `OrderParameterContactModel_serial`



### 9.47.1 Methods

*Inherited from `nMOLDYN.Analysis.NMR.OrderParameterContactModel` (Section 5.2)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)*

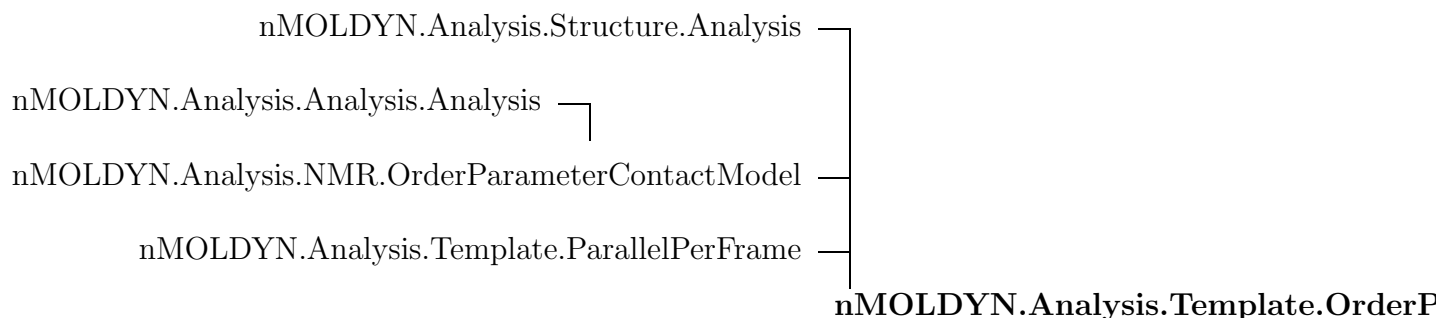
`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

*Inherited from `nMOLDYN.Analysis.Template.SerialPerFrame` (Section 9.5)*

`internalRun()`

### 9.47.2 Class Variables

| Name                                                                                             | Description |
|--------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.NMR.OrderParameterContactModel</code> (Section 5.2)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>         |             |

**9.48 Class *OrderParameterContactModel\_parallel*****9.48.1 Methods**

***Inherited from `nMOLDYN.Analysis.NMR.OrderParameterContactModel` (Section 5.2)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from `nMOLDYN.Analysis.Analysis.Analysis` (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

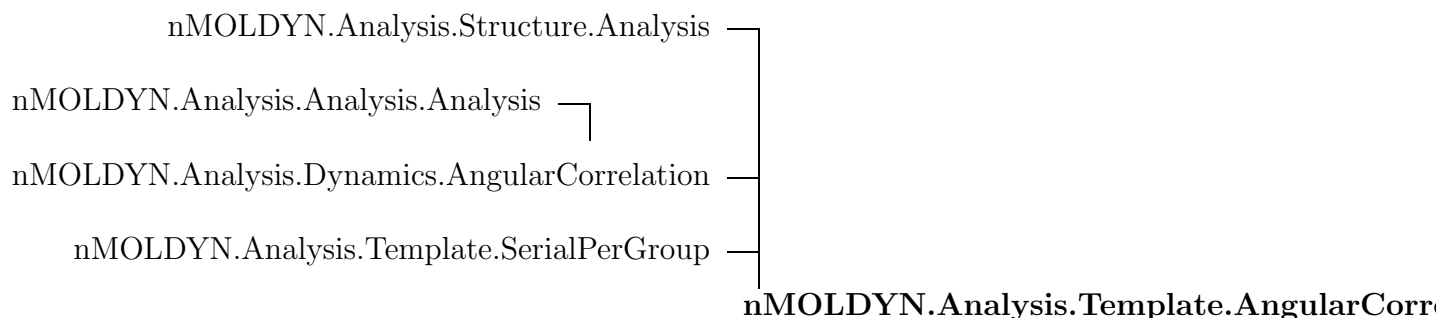
***Inherited from `nMOLDYN.Analysis.Template.ParallelPerFrame` (Section 9.6)***

`internalRun()`

**9.48.2 Class Variables**

| Name                                                                                             | Description |
|--------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <code>nMOLDYN.Analysis.NMR.OrderParameterContactModel</code> (Section 5.2)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>         |             |

## 9.49 Class *AngularCorrelation\_serial*



### 9.49.1 Methods

***Inherited from nMOLDYN.Analysis.Dynamics.AngularCorrelation (Section 4.11)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

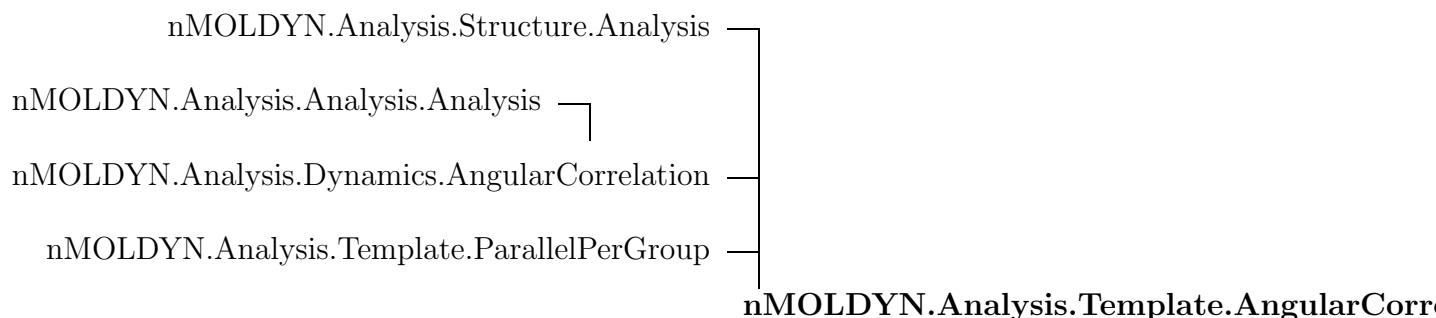
***Inherited from nMOLDYN.Analysis.Template.SerialPerGroup (Section 9.7)***

`internalRun()`

### 9.49.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.AngularCorrelation (Section 4.11)</i>        |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.50 Class *AngularCorrelation\_parallel*



### 9.50.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.AngularCorrelation* (Section 4.11)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

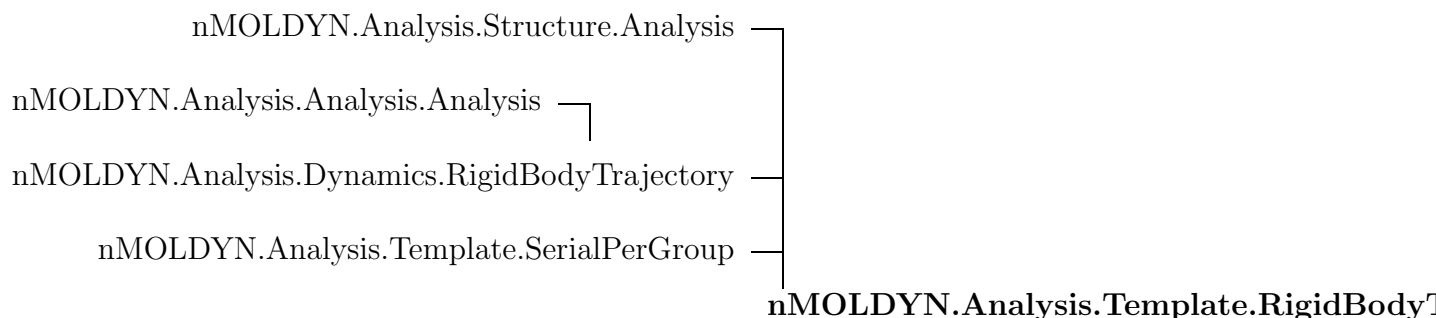
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerGroup* (Section 9.8)***

`internalRun()`

### 9.50.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.AngularCorrelation</i> (Section 4.11)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.51 Class *RigidBodyTrajectory\_serial*



### 9.51.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.RigidBodyTrajectory*(Section 4.12)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis*(Section 3.3)***

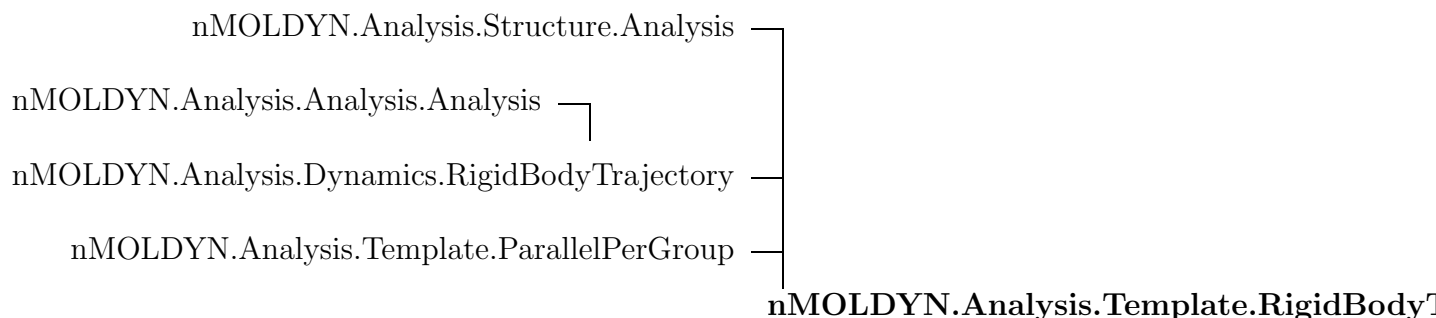
`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

***Inherited from *nMOLDYN.Analysis.Template.SerialPerGroup*(Section 9.7)***

`internalRun()`

### 9.51.2 Class Variables

| Name                                                                                      | Description |
|-------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.RigidBodyTrajectory</i> (Section 4.12)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>  |             |

**9.52 Class RigidBodyTrajectory\_parallel****9.52.1 Methods**

***Inherited from nMOLDYN.Analysis.Dynamics.RigidBodyTrajectory (Section 4.12)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

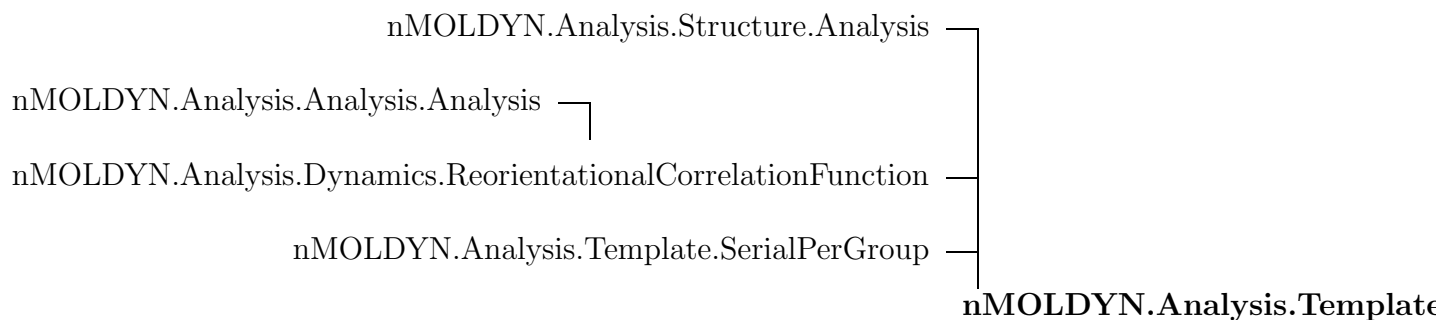
***Inherited from nMOLDYN.Analysis.Template.ParallelPerGroup (Section 9.8)***

`internalRun()`

**9.52.2 Class Variables**

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.RigidBodyTrajectory (Section 4.12)</i>       |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

### 9.53 Class *ReorientationalCorrelationFunction\_serial*



#### 9.53.1 Methods

*Inherited from nMOLDYN.Analysis.Dynamics.ReorientationalCorrelationFunction (Section 4.13)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)*

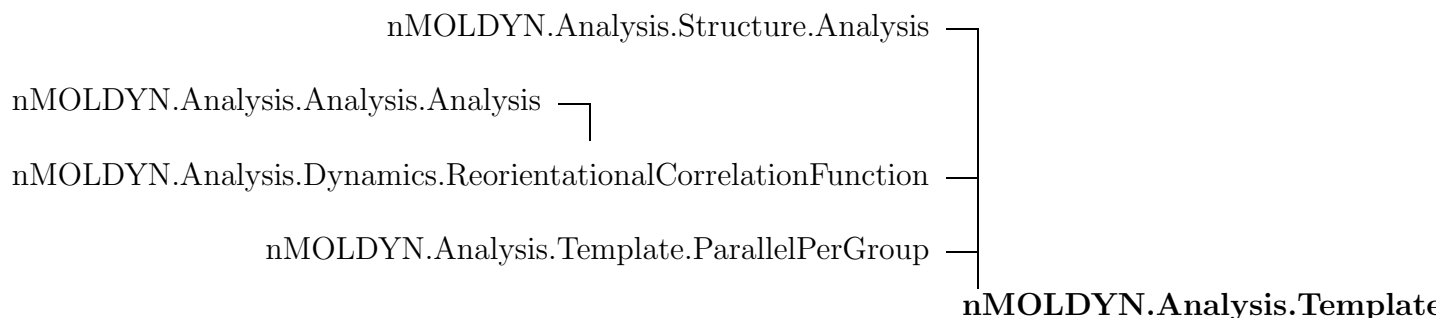
`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

*Inherited from nMOLDYN.Analysis.Template.SerialPerGroup (Section 9.7)*

`internalRun()`

#### 9.53.2 Class Variables

| Name                                                                                              | Description |
|---------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.ReorientationalCorrelationFunction (Section 4.13)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>          |             |

**9.54 Class *ReorientationalCorrelationFunction\_parallel*****9.54.1 Methods**

***Inherited from *nMOLDYN.Analysis.Dynamics.ReorientationalCorrelationFunction* (Section 4.13)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

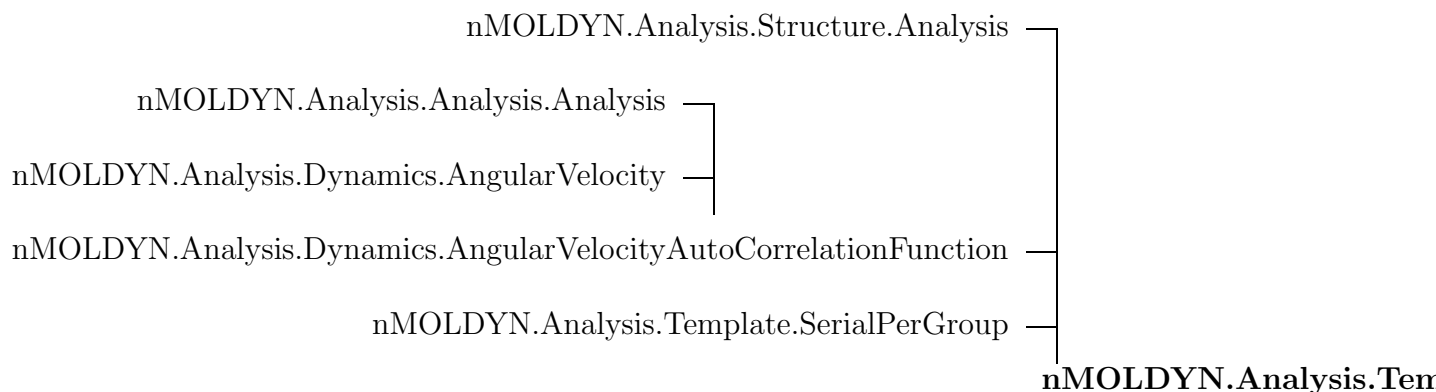
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerGroup* (Section 9.8)***

`internalRun()`

**9.54.2 Class Variables**

| Name                                                                                                     | Description |
|----------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.ReorientationalCorrelationFunction</i> (Section 4.13)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>                 |             |

## 9.55 Class *AngularVelocityAutoCorrelationFunction\_serial*



### 9.55.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.AngularVelocityAutoCorrelationFunction* (Section 4.15)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

***Inherited from *nMOLDYN.Analysis.Dynamics.AngularVelocity* (Section 4.14)***

`getAngularVelocity()`, `qMatrix()`

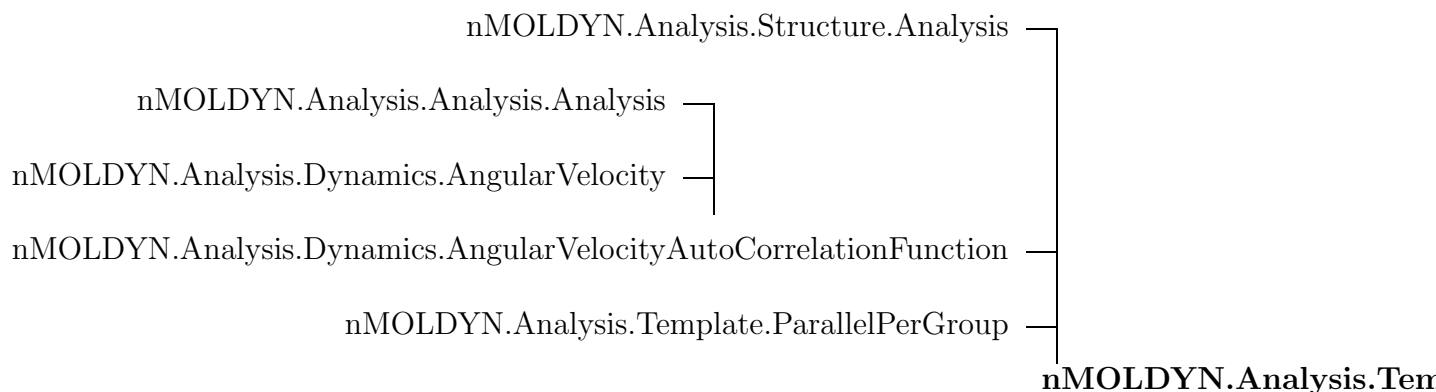
***Inherited from *nMOLDYN.Analysis.Template.SerialPerGroup* (Section 9.7)***

`internalRun()`

### 9.55.2 Class Variables

| Name                                                                                                         | Description |
|--------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.AngularVelocityAutoCorrelationFunction</i> (Section 4.15)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>                     |             |

## 9.56 Class *AngularVelocityAutoCorrelationFunction\_parallel*



### 9.56.1 Methods

***Inherited from *nMOLDYN.Analysis.Dynamics.AngularVelocityAutoCorrelationFunction* (Section 4.15)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

***Inherited from *nMOLDYN.Analysis.Dynamics.AngularVelocity* (Section 4.14)***

`getAngularVelocity()`, `qMatrix()`

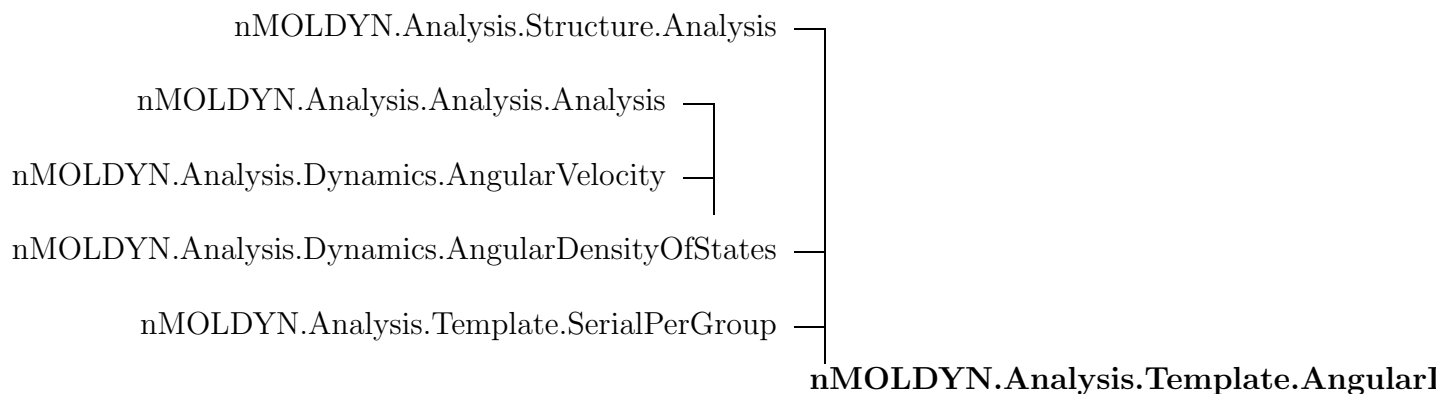
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerGroup* (Section 9.8)***

`internalRun()`

### 9.56.2 Class Variables

| Name                                                                                                         | Description |
|--------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Dynamics.AngularVelocityAutoCorrelationFunction</i> (Section 4.15)</i> |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code>                     |             |

## 9.57 Class AngularDensityOfStates\_serial



### 9.57.1 Methods

*Inherited from nMOLDYN.Analysis.Dynamics.AngularDensityOfStates (Section 4.16)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

*Inherited from nMOLDYN.Analysis.Dynamics.AngularVelocity (Section 4.14)*

`getAngularVelocity()`, `qMatrix()`

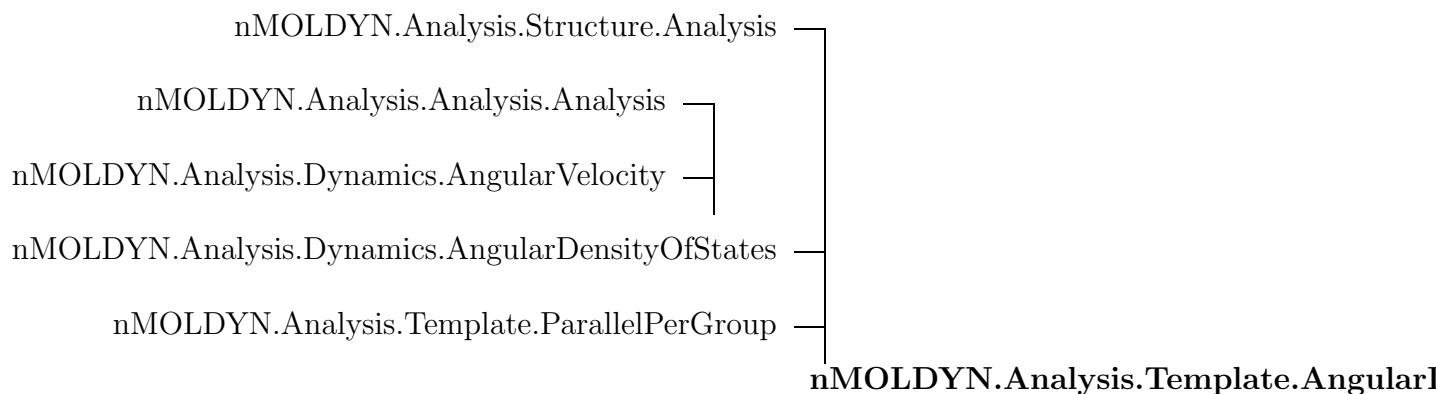
*Inherited from nMOLDYN.Analysis.Template.SerialPerGroup (Section 9.7)*

`internalRun()`

### 9.57.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.AngularDensityOfStates (Section 4.16)</i>    |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.58 Class AngularDensityOfStates\_parallel



### 9.58.1 Methods

***Inherited from nMOLDYN.Analysis.Dynamics.AngularDensityOfStates (Section 4.16)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

***Inherited from nMOLDYN.Analysis.Dynamics.AngularVelocity (Section 4.14)***

`getAngularVelocity()`, `qMatrix()`

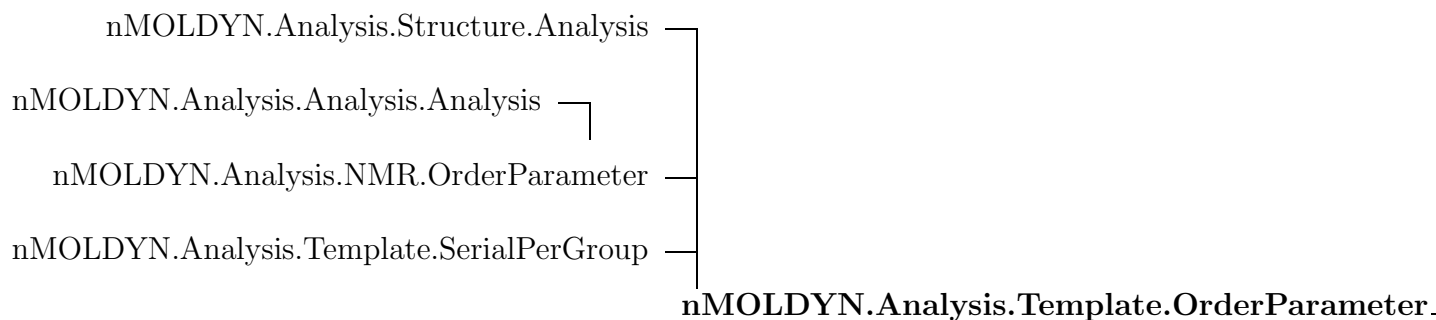
***Inherited from nMOLDYN.Analysis.Template.ParallelPerGroup (Section 9.8)***

`internalRun()`

### 9.58.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.AngularDensityOfStates (Section 4.16)</i>    |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.59 Class *OrderParameter\_serial*



### 9.59.1 Methods

*Inherited from nMOLDYN.Analysis.NMR.OrderParameter (Section 5.1)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

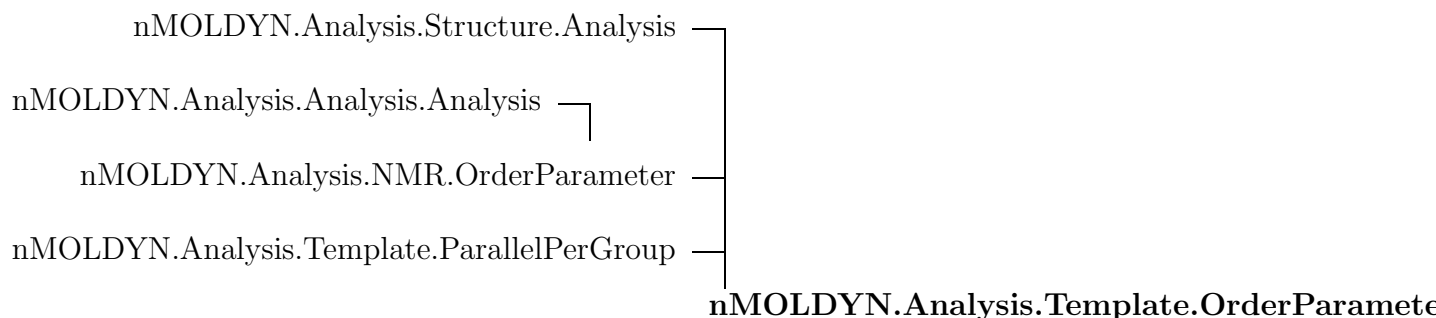
*Inherited from nMOLDYN.Analysis.Template.SerialPerGroup (Section 9.7)*

`internalRun()`

### 9.59.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.NMR.OrderParameter (Section 5.1)</i>                  |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.60 Class `OrderParameter_parallel`



### 9.60.1 Methods

**Inherited from *nMOLDYN.Analysis.NMR.OrderParameter* (Section 5.1)**

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)**

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

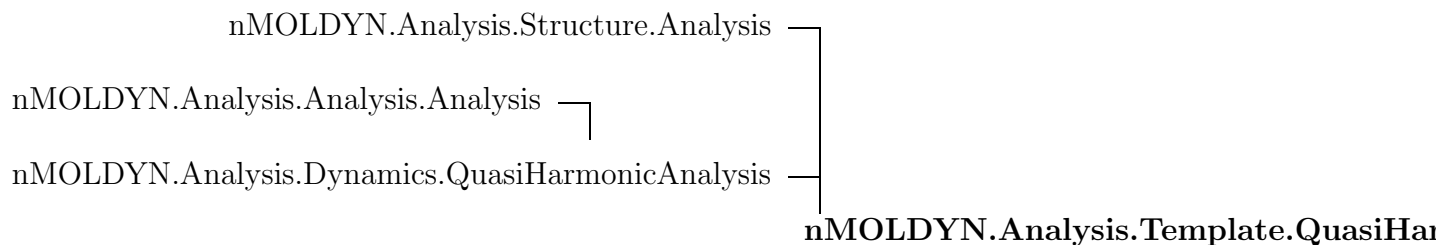
**Inherited from *nMOLDYN.Analysis.Template.ParallelPerGroup* (Section 9.8)**

`internalRun()`

### 9.60.2 Class Variables

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.NMR.OrderParameter (Section 5.1)</i>                  |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.61 Class `QuasiHarmonicAnalysis_serial`



**9.61.1 Methods**

*Inherited from nMOLDYN.Analysis.Dynamics.QuasiHarmonicAnalysis (Section 4.10)*

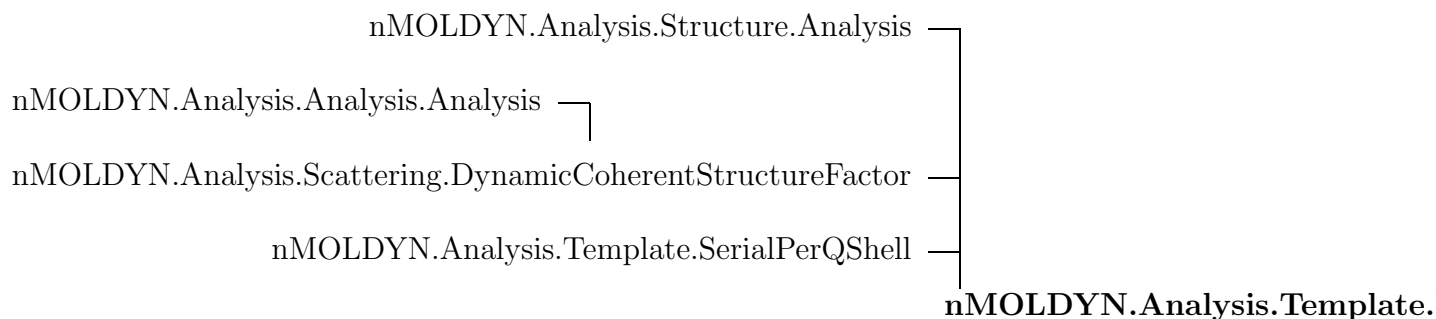
`__init__()`, `initialize()`, `internalRun()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

**9.61.2 Class Variables**

| Name                                                                                     | Description |
|------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Dynamics.QuasiHarmonicAnalysis (Section 4.10)</i>     |             |
| <code>canBeEstimated</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

**9.62 Class DynamicCoherentStructureFactor\_serial****9.62.1 Methods**

*Inherited from nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactor (Section 6.2)*

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

*Inherited from nMOLDYN.Analysis.Analysis.Analysis (Section 3.3)*

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

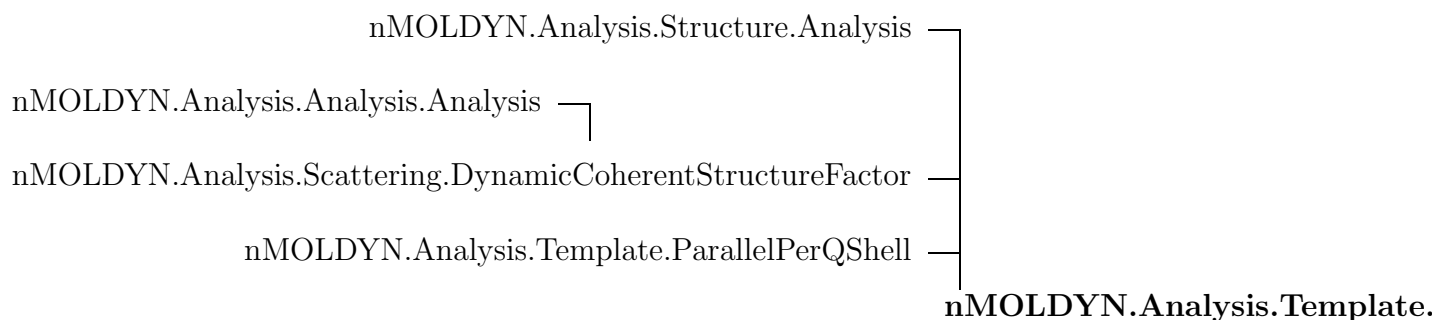
*Inherited from nMOLDYN.Analysis.Template.SerialPerQShell(Section 9.9)*

internalRun()

### 9.62.2 Class Variables

| Name                                                                                           | Description                                              |
|------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| <i>Inherited from nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactor (Section 6.2)</i> | canBeEstimated, default, inputParametersNames, shortName |

## 9.63 Class *DynamicCoherentStructureFactor\_parallel*



### 9.63.1 Methods

*Inherited from nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactor(Section 6.2)*

\_\_init\_\_(), calc(), combine(), finalize(), initialize()

*Inherited from nMOLDYN.Analysis.Analysis.Analysis(Section 3.3)*

analysisTime(), buildJobInfo(), buildTimeInfo(), deuterationSelection(), groupSelection(), parseInputParameters(), preLoadTrajectory(), runAnalysis(), saveAnalysis(), setInputParameters(), subsetSelection(), updateJobProgress(), weightingScheme()

*Inherited from nMOLDYN.Analysis.Template.ParallelPerQShell(Section 9.10)*

internalRun()

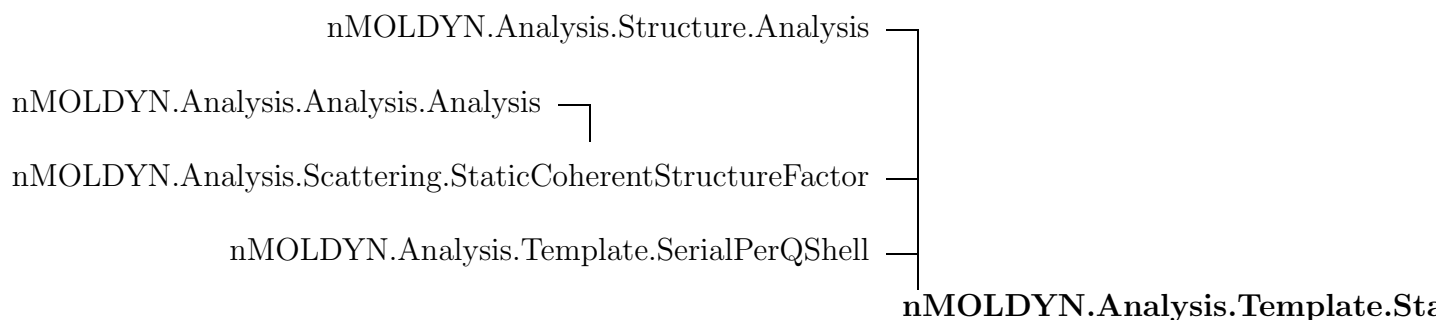
### 9.63.2 Class Variables

| Name                                                                                           | Description |
|------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactor (Section 6.2)</i> |             |

*continued on next page*

| Name                                                     | Description |
|----------------------------------------------------------|-------------|
| canBeEstimated, default, inputParametersNames, shortName |             |

## 9.64 Class *StaticCoherentStructureFactor\_serial*



### 9.64.1 Methods

**Inherited from *nMOLDYN.Analysis.Scattering.StaticCoherentStructureFactor* (Section 6.3)**

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

**Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)**

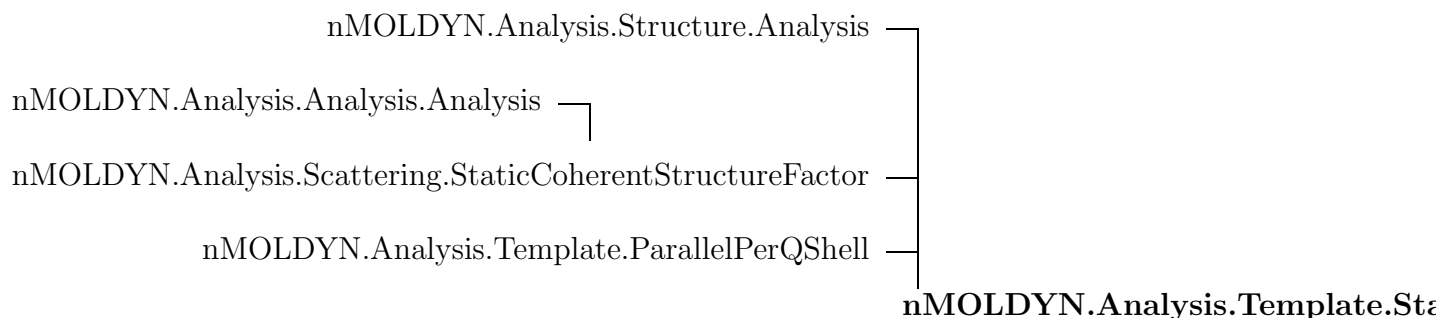
`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

**Inherited from *nMOLDYN.Analysis.Template.SerialPerQShell* (Section 9.9)**

`internalRun()`

### 9.64.2 Class Variables

| Name                                                                                          | Description |
|-----------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.Analysis.Scattering.StaticCoherentStructureFactor (Section 6.3)</i> |             |
| canBeEstimated, default, inputParametersNames, shortName                                      |             |

**9.65 Class *StaticCoherentStructureFactor\_parallel*****9.65.1 Methods**

***Inherited from *nMOLDYN.Analysis.Scattering.StaticCoherentStructureFactor* (Section 6.3)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

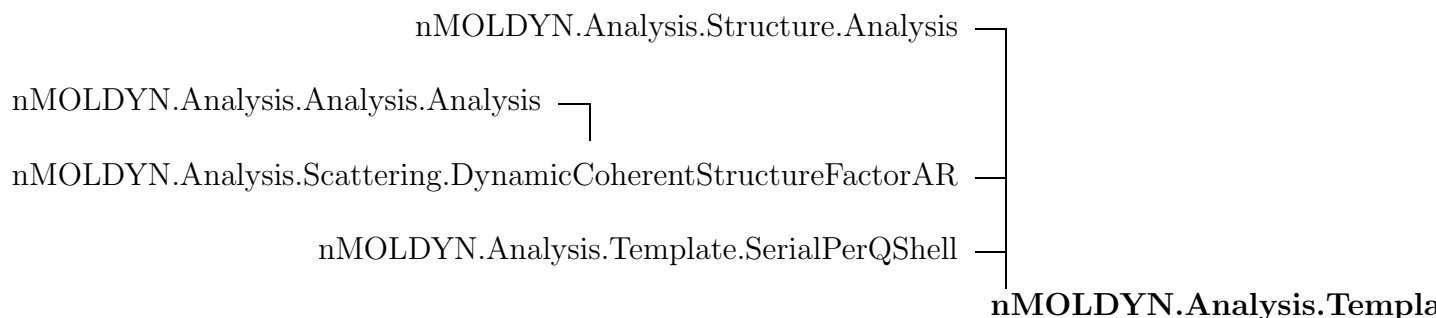
***Inherited from *nMOLDYN.Analysis.Template.ParallelPerQShell* (Section 9.10)***

`internalRun()`

**9.65.2 Class Variables**

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Scattering.StaticCoherentStructureFactor</i> (Section 6.3)</i>            |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 9.66 Class *DynamicCoherentStructureFactorAR\_serial*



### 9.66.1 Methods

***Inherited from *nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactorAR* (Section 6.4)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

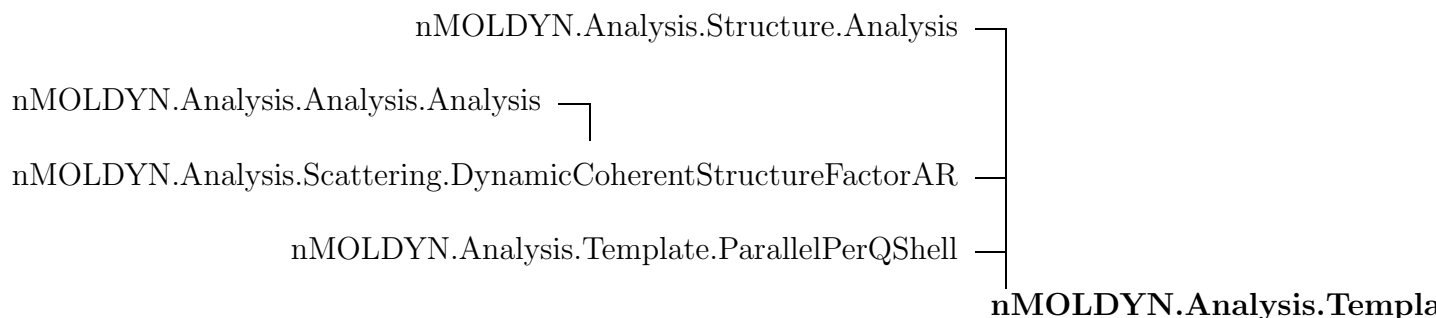
`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

***Inherited from *nMOLDYN.Analysis.Template.SerialPerQShell* (Section 9.9)***

`internalRun()`

### 9.66.2 Class Variables

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactorAR</i> (Section 6.4)</i>         |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

**9.67 Class *DynamicCoherentStructureFactorAR\_parallel*****9.67.1 Methods**

***Inherited from *nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactorAR* (Section 6.4)***

`__init__()`, `calc()`, `combine()`, `finalize()`, `initialize()`

***Inherited from *nMOLDYN.Analysis.Analysis.Analysis* (Section 3.3)***

`analysisTime()`, `buildJobInfo()`, `buildTimeInfo()`, `deuterationSelection()`, `groupSelection()`, `parseInputParameters()`, `preLoadTrajectory()`, `runAnalysis()`, `saveAnalysis()`, `setInputParameters()`, `subsetSelection()`, `updateJobProgress()`, `weightingScheme()`

***Inherited from *nMOLDYN.Analysis.Template.ParallelPerQShell* (Section 9.10)***

`internalRun()`

**9.67.2 Class Variables**

| Name                                                                                                            | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
| <i>Inherited from <i>nMOLDYN.Analysis.Scattering.DynamicCoherentStructureFactorAR</i> (Section 6.4)</i>         |             |
| <code>canBeEstimated</code> , <code>default</code> , <code>inputParametersNames</code> , <code>shortName</code> |             |

## 10 Package nMOLDYN.Core

### 10.1 Modules

- **Chemistry:** This module implements the functions and procedures that are related to chemistry.  
(Section 11, p. 145)
- **Config:** This module implements the procedures that handles nMOLDYN PREFERENCES.  
(Section 12, p. 146)
- **Error** (Section 13, p. 147)
- **IOFiles:** This module implements IO-related classes, functions and procedures.  
(Section 14, p. 148)
- **Logger:** This module implements the classes used to handle the nMOLDYN logger.  
(Section 15, p. 158)
- **Mathematics:** This module implements the mathematics-related classes, functions and procedures.  
(Section 16, p. 163)
- **Misc:** This module implements the functions and procedures that can not be classified anywhere else in the library.  
(Section 17, p. 170)
- **Preferences:** This module stores some of the nMOLDYN PREFERENCES variables that will be used throughout all nMOLDYN code.  
(Section 18, p. 172)

## 11 Module *nMOLDYN.Core.Chemistry*

This module implements the functions and procedures that are related to chemistry.

Functions:

- \* `belongToAnAmine` : this function determine whether or not `|atom|` is part of an amine group.
- \* `belongToAHydroxy` : this function determine whether or not `|atom|` is part of a hydroxy group.
- \* `belongToAMethyl` : this function determine whether or not `|atom|` is part of a methyl group.
- \* `belongToATHiol` : this function determine whether or not `|atom|` is part of a thiol group.

### 11.1 Functions

|                                                                                       |
|---------------------------------------------------------------------------------------|
| <b><code>belongToAMethyl(<i>atom</i>)</code></b>                                      |
| This function determine whether or not <code> atom </code> is part of a methyl group. |

|                                                                                       |
|---------------------------------------------------------------------------------------|
| <b><code>belongToAnAmine(<i>atom</i>)</code></b>                                      |
| This function determine whether or not <code> atom </code> is part of an amine group. |

|                                                                                      |
|--------------------------------------------------------------------------------------|
| <b><code>belongToATHiol(<i>atom</i>)</code></b>                                      |
| This function determine whether or not <code> atom </code> is part of a thiol group. |

|                                                                                        |
|----------------------------------------------------------------------------------------|
| <b><code>belongToAHydroxy(<i>atom</i>)</code></b>                                      |
| This function determine whether or not <code> atom </code> is part of a hydroxy group. |

## 12 Module nMOLDYN.Core.Config

This module implements the procedures that handles nMOLDYN PREFERENCES.

Procedures:

- \* saveConfigurationFile: saves a PREFERENCES file.
- \* loadConfigurationFile: loads a PREFERENCES file.

### 12.1 Functions

**saveConfigurationFile**(*cfgFilename*=None, *config*=PREFERENCES)

Saves |*config*| configuration to |*cfgFilename*| file name.

@param *cfgFilename*: if not None, the name of the PREFERENCES file to save else the configuration will be saved to a plat-form dependant location:

- \$USERPROFILE/Application Data/nMOLDYN/nMOLDYN.ini on Windows
- \$HOME/Library/Preferences/nMOLDYN/nMOLDYN.pref on MacOS
- \$HOME/.nMOLDYN on Linux

@type: *cfgFilename*: string.

@param *config*: the configuration to save. By default, the default PREFERENCES stored in nMOLDYN.Core.Preferences.nMOLDYNPreferences class.

@type *config*: instance of a dummy class whose attributes must be the ones defined in nMOLDYN.Core.Preferences.nMOLDYNPreferences class.

**loadConfigurationFile**(*cfgFilename*=None, *config*=PREFERENCES)

Loads a configuration from |*cfgFilename*| file name and updates |*config*| PREFERENCES.

@param *cfgFilename*: if not None, the name of the PREFERENCES file to load else the configuration will be loaded from a plat-form dependant location:

- \$USERPROFILE/Application Data/nMOLDYN/nMOLDYN.ini on Windows
- \$HOME/Library/Preferences/nMOLDYN/nMOLDYN.pref on MacOS
- \$HOME/.nMOLDYN on Linux

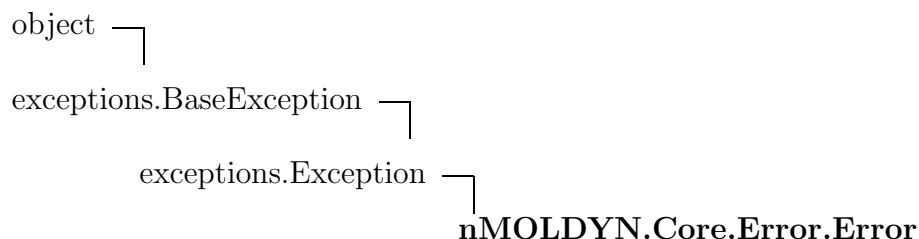
@type: *cfgFilename*: string.

@param *config*: the configuration to load. By default, the default PREFERENCES stored in nMOLDYN.Core.Preferences.nMOLDYNPreferences class.

@type *config*: instance of a dummy class whose attributes must be the ones defined in nMOLDYN.Core.Preferences.nMOLDYNPreferences class.

## 13 Module nMOLDYN.Core.Error

### 13.1 Class Error



#### 13.1.1 Methods

|                                             |
|---------------------------------------------|
| <b><code>__init__(self, message)</code></b> |
|---------------------------------------------|

|                  |
|------------------|
| The constructor. |
|------------------|

|                                         |
|-----------------------------------------|
| Overrides: <code>object.__init__</code> |
|-----------------------------------------|

*Inherited from `exceptions.Exception`*

`__new__()`

*Inherited from `exceptions.BaseException`*

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,  
`__setattr__()`, `__setstate__()`, `__str__()`

*Inherited from `object`*

`__hash__()`, `__reduce_ex__()`

#### 13.1.2 Properties

| Name                                                        | Description |
|-------------------------------------------------------------|-------------|
| <i>Inherited from <code>exceptions.BaseException</code></i> |             |
| <code>args</code> , <code>message</code>                    |             |
| <i>Inherited from <code>object</code></i>                   |             |
| <code>__class__</code>                                      |             |

## 14 Module nMOLDYN.Core.IOFiles

This module implements IO-related classes, functions and procedures.

Classes:

|                            |                                                           |
|----------------------------|-----------------------------------------------------------|
| * TemporaryFile            | : creates a temporary file storing the evolution of an a  |
| * EndOfFile                | : an empty dummy class used by  DCDReader .               |
| * FortranBinaryFile        | : sets up a binary file reader.                           |
| * DCDFile                  | : sets up a DCD file reader.                              |
| * AmberNetCDFConverter     | : converts a trajectory from Amber > 9 to a MMTK NetCDF t |
| * CHARMMConverter          | : converts a trajectory from CHARMM to a MMTK NetCDF traj |
| * DL_POLYConverter         | : converts a trajectory from DL_POLY > 9 to a MMTK NetCDF |
| * MaterialsStudioConverter | : converts a trajectory from MaterialsStudio > 9 to a MM  |
| * NAMDConverter            | : converts a trajectory from NAMD to a MMTK NetCDF trajec |
| * VASPConverter            | : converts a trajectory from VASP > 9 to a MMTK NetCDF tr |

Procedures:

|                         |                                            |
|-------------------------|--------------------------------------------|
| * convertNetCDFToASCII: | converts a NetCDF file into an ASCII file. |
| * convertASCIIToNetCDF: | converts an ASCII file into a NetCDF file. |

### 14.1 Functions

**convertNetCDFToASCII**(*inputFile*, *outputFile*, *variables*, *floatPrecision*=9, *doublePrecision*=17)

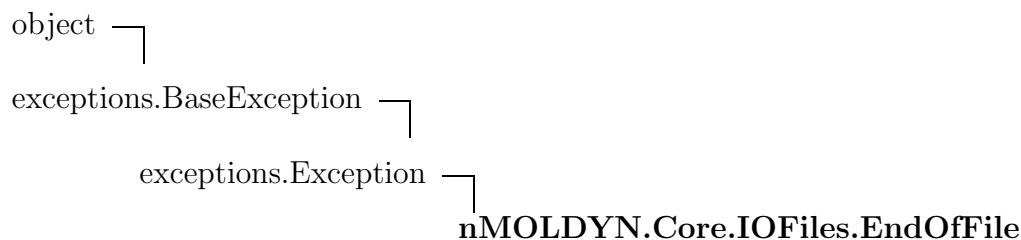
Converts a file in NetCDF format to a file in ASCII/CDL format using the `ncdump` program provided with the `netcdf` library.

**Parameters**

|                         |                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>inputFile:</b>       | the name of the NetCDF input file.<br>( <i>type=string</i> )                                                           |
| <b>outputFile:</b>      | the name of the CDL output file.<br>( <i>type=string</i> )                                                             |
| <b>variables:</b>       | list of the NetCDF variables names (string) to<br>extract from the NetCDF file for conversion.<br>( <i>type=list</i> ) |
| <b>floatPrecision:</b>  | the precision on the float numbers.<br>( <i>type=integer</i> )                                                         |
| <b>doublePrecision:</b> | the precision on the double numbers.<br>( <i>type=integer</i> )                                                        |

```
convertASCIIToNetCDF(inputFile, outputFile)
```

### 14.3 Class EndOfFile



#### 14.3.1 Methods

*Inherited from exceptions.Exception*

`__init__()`, `__new__()`

*Inherited from exceptions.BaseException*

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,  
`__setattr__()`, `__setstate__()`, `__str__()`

*Inherited from object*

`__hash__()`, `__reduce_ex__()`

#### 14.3.2 Properties

| Name                                                            | Description |
|-----------------------------------------------------------------|-------------|
| <i>Inherited from exceptions.BaseException</i><br>args, message |             |
| <i>Inherited from object</i><br>__class__                       |             |

### 14.4 Class FortranBinaryFile



Sets up a Fortran binary file reader.

Comments:

-written by Konrad Hinsen in the scope of a DCD file reader.

### 14.4.1 Methods

```
__init__(self, filename, byte_order='')
```

*x.\_\_init\_\_*(...) initializes *x*; see *x.\_\_class\_\_.\_\_doc\_\_* for signature

Overrides: object.\_\_init\_\_ extit(inherited documentation)

```
__iter__(self)
```

```
next(self)
```

```
skipRecord(self)
```

```
getRecord(self, format, repeat=False)
```

#### *Inherited from object*

```
__delattr__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __str__()
```

### 14.4.2 Properties

| Name                         | Description |
|------------------------------|-------------|
| <i>Inherited from object</i> |             |
| <code>__class__</code>       |             |

## 14.5 Class DCDFFile

```
object └─ nMOLDYN.Core.IOFiles.DCDFFile
```

Sets up a DCD file reader.

### 14.5.1 Methods

**`__init__(self, dcd_filename)`**

The constructor.

**Parameters**

**dcd\_filename:** the name of the DCD file to read.

(*type=string.*)

Overrides: `object.__init__`

**`readStep(self)`**

Reads a frame of the DCD file.

**`skipStep(self)`**

Skips a frame of the DCD file.

**`__iter__(self)`**

**`next(self)`**

***Inherited from object***

`__delattr__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,  
`__repr__()`, `__setattr__()`, `__str__()`

### 14.5.2 Properties

| Name                         | Description |
|------------------------------|-------------|
| <i>Inherited from object</i> |             |
| <code>__class__</code>       |             |

## 14.6 Class *AmberNetCDFConverter*

Converts an Amber NetCDF Trajectory into a MMTK NetCDFFile.

Comments:

- this code is an improved version of the original converter written by Paolo Calligari.

### 14.6.1 Methods

```
--init--(self, pdbFile, amberNetCDFFile, outputFile, timeStep=1.0)
```

The constructor. Will do the conversion.

**Parameters**

|                         |                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>pdbFile:</b>         | the Amber PDB file name of a frame of the trajectory to convert.<br>( <i>type=string</i> )                |
| <b>amberNetCDFFile:</b> | the Amber NetCDF file name of the trajectory to convert.<br>( <i>type=string</i> )                        |
| <b>outputFile:</b>      | the name of MMTK NetCDF trajectory output file.<br>( <i>type=string</i> )                                 |
| <b>timeStep:</b>        | the timestep that will used when building the MMTK trajectory. Default to 1 ps.<br>( <i>type=float.</i> ) |

## 14.7 Class CHARMMConverter

Converts a CHARMM Trajectory into a MMTK NetCDFFile.

Comments:

- this code is based on the original converter written by Konrad Hinsén.

### 14.7.1 Methods

**\_\_init\_\_**(*self*, *pdbFile*, *dcdFile*, *outputFile*)

The constructor. Will do the conversion.

**Parameters**

**pdbFile:** the CHARMM PDB file name of a frame of the trajectory to convert.  
(*type=string*)

**dcdFile:** the CHARMM DCD file name of the trajectory to convert.  
(*type=string*)

**outputFile:** the name of MMTK NetCDF trajectory output file.  
(*type=string*)

## 14.8 Class DL\_POLYConverter

Converts a DL\_POLY Trajectory into a MMTK NetCDFFile.

### 14.8.1 Methods

**\_\_init\_\_**(*self*, *fieldFile*, *historyFile*, *outputFile*, *specialAtoms*={})

The constructor. Will do the conversion.

**Parameters**

**fieldFile:** the DL\_POLY FIELD file name of the trajectory to convert.  
(*type=string*)

**historyFile:** the DL\_POLY HISTORY file name of the trajectory to convert.  
(*type=string*)

**outputFile:** the name of MMTK NetCDF trajectory output file.  
(*type=string*)

**specialAtoms:** dictionary of the form {s1 : e1, s2 : e2 ...} where 's1', 's2' ... and 'e1', 'e2' ... are respectively the DL\_POLY name and the symbol of atoms 1, 2 ...  
(*type=dict*)

## 14.9 Class *MaterialsStudioConverter*

Converts a MaterialsStudio Discover or Forcite Trajectory into a MMTK NetCDFFile.

### 14.9.1 Methods

**`__init__(self, module, xtdxsdFile, histrijFile, outputFile, subselection=None)`**

The constructor. Will do the conversion.

**Parameters**

**module:** a string being one of 'Discover' or 'Forcite' specifying which module of MaterialsStudio the trajectory is coming from.  
(*type=string*)

**xtdxsdFile:** the MaterialsStudio XTD or XSD file name of the trajectory to convert.  
(*type=string*)

**histrijFile:** the MaterialsStudio HIS (Discover) or TRJ (Forcite) file name of the trajectory to convert.  
(*type=string*)

**outputFile:** the name of MMTK NetCDF trajectory output file.  
(*type=string*)

**subselection:** if not None, list of the indexes (integer  $\geq 1$ ) of the atoms to select when writing out the MMTK trajectory. The order being the one defined in the XTD/XSD file.  
(*type=list*)

**`createCluster(self, at, clust)`**

**`readXTDFile(self)`**

Reads the Materials Studio XTD or XSD file and set up the universe from which the NetCDF MMTK trajectory will be written.

**Note:** the XTD and XSD file are xml file.

**`readHISFile(self)`**

Reads a Materials Studio HIS file and fills up the NetCDF trajectory file.

|                                 |
|---------------------------------|
| <b>readTRJFile(<i>self</i>)</b> |
|---------------------------------|

|                                                                            |
|----------------------------------------------------------------------------|
| Reads a Materials Studio HIS file and fills up the NetCDF trajectory file. |
|----------------------------------------------------------------------------|

#### 14.9.2 Class Variables

| Name           | Description                                                                  |
|----------------|------------------------------------------------------------------------------|
| atomLineFormat | <b>Value:</b><br>FortranFormat('A5,1X,F14.9,1X,F14.9,1X,F14.9,1X,A4,1X,A7,') |

### 14.10 Class NAMDCConverter

Converts a NAMD Trajectory into a MMTK NetCDFFile.

Comments:

- this code is based on the original converter written by Konrad Hinsen.

#### 14.10.1 Methods

|                                                                                                 |
|-------------------------------------------------------------------------------------------------|
| <b>__init__(<i>self</i>, <i>pdbFile</i>, <i>dcdFile</i>, <i>xstFile</i>, <i>outputFile</i>)</b> |
|-------------------------------------------------------------------------------------------------|

|                                          |
|------------------------------------------|
| The constructor. Will do the conversion. |
|------------------------------------------|

**Parameters**

|                    |                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------|
| <b>pdbFile:</b>    | the NAMD PDB file name of a frame of the trajectory to convert.<br>( <i>type=string</i> ) |
| <b>dcdFile:</b>    | the NAMD DCD file name of the trajectory to convert.<br>( <i>type=string</i> )            |
| <b>xstFile:</b>    | the NAMD XSTfile name of the trajectory to convert.<br>( <i>type=string</i> )             |
| <b>outputFile:</b> | the name of MMTK NetCDF trajectory output file.<br>( <i>type=string</i> )                 |

### 14.11 Class VASPConverter

Converts a VASP Trajectory into a MMTK NetCDFFile.

### 14.11.1 Methods

**`--init--(self, contcarFile, xdatcarFile, outputFile, atomContents)`**

The constructor. Will do the conversion.

**Parameters**

**contcarFile:** the VASP CONTCAR or POSCAR file name of the trajectory to convert.

*(type=string)*

**xdatcarFile:** the VASP XDATCAR file name of the trajectory to convert.

*(type=string)*

**outputFile:** the name of MMTK NetCDF trajectory output file.

*(type=string)*

**atomContents:** List of the element names (string) in the order they appear in the trajectory.

*(type=list)*

## 15 Module nMOLDYN.Core.Logger

This module implements the classes used to handle the nMOLDYN logger.

Classes:

- \* LogToGUI : sets up a GUI logger.
- \* LogToFile : sets up a file logger.
- \* LogToConsole : sets up a console logger.

Procedures:

- \* LogMessage : displays a logging message of a specified logging level to the spec

### 15.1 Functions

|                                                                                                           |                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LogMessage</b> ( <i>level</i> ='debug', <i>message</i> ='',<br><i>media</i> =['gui','file','console']) |                                                                                                                                                                                                                   |
| Displays the logging messahe  message  of logging level  level  to the logger(s)  media .                 |                                                                                                                                                                                                                   |
| <b>Parameters</b>                                                                                         |                                                                                                                                                                                                                   |
| <b>level:</b>                                                                                             | a string being one of 'debug', 'info', 'warning', 'error' or 'critical' specifying the logging level of the logging message. Will change the way the logging message will be displayed.<br>( <i>type=string</i> ) |
| <b>message:</b>                                                                                           | the logging message.<br>( <i>type=string</i> )                                                                                                                                                                    |
| <b>media:</b>                                                                                             | a list containing 'gui' and/or 'file' and/or 'console' specifying on which logger(s) the logging message should emitted.<br>( <i>type=list</i> )                                                                  |

### 15.2 Variables

| Name        | Description                                                                 |
|-------------|-----------------------------------------------------------------------------|
| LEVELS      | <b>Value:</b> {'debug': logging.DEBUG, 'info': logging.INFO, 'warning': ... |
| FILE_LOGGER | <b>Value:</b> logging.getLogger('NMOLDYN LOGFILE')                          |

*continued on next page*

| Name          | Description                                        |
|---------------|----------------------------------------------------|
| CONSOLELOGGER | <b>Value:</b> logging.getLogger('NMOLDYN CONSOLE') |
| GUILLOGGER    | <b>Value:</b> logging.getLogger('NMOLDYN GUI')     |

### 15.3 Class LogToGUI



Sets up a GUI handler for the nMOLDYN logger.

Emits the logging messages to a Tk dialog.

#### 15.3.1 Methods

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>__init__</b> ( <i>self</i> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| The constructor. Sets the logger.<br>Overrides: logging.Filterer.__init__                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>emit</b> ( <i>self</i> , <i>record</i> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Emits the logging message in a tkMessageBox.<br><b>Parameters</b><br><i>record</i> : the logging message.<br>( <i>type=instance of LogRecord class.</i> )<br>Overrides: logging.Handler.emit<br><b>Note:</b> the tkMessageBox called will depend on the logging level.<br><ul style="list-style-type: none"> <li>• tkMessageBox.showerror for 'ERROR' and 'CRITICAL' logging levels.</li> <li>• tkMessageBox.showwarning for 'WARNING' logging level.</li> <li>• tkMessageBox.showinfo for other logging levels.</li> </ul> |

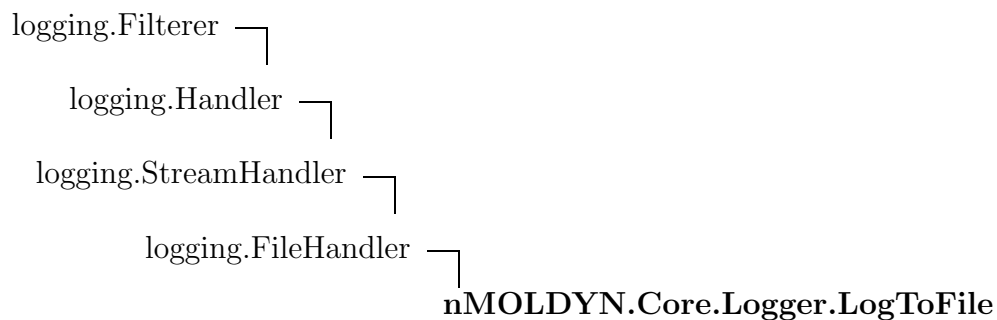
*Inherited from logging.Handler*

acquire(), close(), createLock(), flush(), format(), handle(), handleError(), release(), setFormatter(), setLevel()

*Inherited from logging.Filterer*

`addFilter()`, `filter()`, `removeFilter()`

## 15.4 Class **LogToFile**



Sets up a file logger.

Emits the logging messages to a file.

### 15.4.1 Methods

|                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------|
| <b><code>__init__(self, fileName)</code></b>                                                                    |
| The constructor. Sets the logger.                                                                               |
| <b>Parameters</b>                                                                                               |
| <b>fileName:</b> the name of the file where all the logging messages will be emitted.<br>( <i>type=string</i> ) |
| Overrides: <code>logging.Filterer.__init__</code>                                                               |

|                                                                                     |
|-------------------------------------------------------------------------------------|
| <b><code>emit(self, record)</code></b>                                              |
| Emits the logging message in a file.                                                |
| <b>Parameters</b>                                                                   |
| <b>record:</b> the logging message.<br>( <i>type=instance of LogRecord class.</i> ) |
| Overrides: <code>logging.Handler.emit</code>                                        |

|                                               |
|-----------------------------------------------|
| <b><code>close(self)</code></b>               |
| Closes the file logger.                       |
| Overrides: <code>logging.Handler.close</code> |

***Inherited from logging.StreamHandler***

flush()

***Inherited from logging.Handler***

acquire(), createLock(), format(), handle(), handleError(), release(), setFormatter(), setLevel()

***Inherited from logging.Filterer***

addFilter(), filter(), removeFilter()

**15.5 Class LogToConsole**

Sets up a console logger.

Emits the logging messages to the console.

**15.5.1 Methods****\_\_init\_\_(self)**

The constructor. Sets the logger.

Overrides: logging.Filterer.\_\_init\_\_

**emit(self, record)**

Emits the logging message to the console.

**Parameters****record:** the logging message.*(type=instance of LogRecord class.)*

Overrides: logging.Handler.emit

***Inherited from logging.StreamHandler***

flush()

***Inherited from logging.Handler***

acquire(), close(), createLock(), format(), handle(), handleError(), release(), set-  
Formatter(), setLevel()

***Inherited from logging.Filterer***

addFilter(), filter(), removeFilter()

## 16 Module nMOLDYN.Core.Mathematics

This module implements the mathematics-related classes, functions and procedures.

Classes:

- \* QVectors: the class that actually performs the q vectors generation.

Functions:

- \* differentiate : performs a numerical differentiation of 1D Numeric array
- \* correlation : performs the numerical correlation between two 1D Numeric arrays
- \* convolution : performs the numerical convolution between two 1D Numeric arrays
- \* FFT : performs the FFT of a 1D Numeric array
- \* invFFT : performs the inverse FFT of a 1D Numeric array
- \* gaussianWindow : performs a Gaussian smoothing of 1D Numeric array.
- \* factorial : computes factorial (n) where n is an integer.
- \* basisVectors : computes the basis vectors of the simulation cell from a set of vectors
- \* randomPointInCircle : returns a vector within a circle of radius |r| and orthogonal to the plane
- \* randomDirection2D : returns a normalized vector generated from a unit circle orthogonal to the plane
- \* randomPlane2D : generates a normalized random q-vector on a plane defined by two vectors
- \* qVectorGenerator : sets up and returns a set of q vectors generated from different directions
- \* sphericalHarmonics : calculates the spherical functions Y from a set of j, m, n with respect to a direction
- \* preparePP : sets up the calculation of spherical harmonics.

### 16.1 Functions

**differentiate**(*inputSeries*, *order*, *dx*)

Returns the numerical derivative of order |order| of the signal |inputSeries| using the differentiation step |dx|.

**Parameters**

- inputSeries:** the signal to differentiate.
- order:** an integer in [1,5] specifying the numerical differentiation order.
- dx:** a float specifying the differentiation step. Assumed to be constant over all the spectrum.

**Return Value**

the differentiated signal.

(*type=NumPy array*)

**See Also:** M. Abramowitz, I.A. Stegun; 'Handbook of mathematical functions', Dover, New-York, 1972 p.914.

**myautocorrelation**(*inputSeries*)

**correlation**(*inputSeries1*, *inputSeries2*=None)

Returns the numerical correlation between |inputSeries1| and |inputSeries2| multidimensional NumPy arrays.

**Parameters**

**inputSeries1**: the first signal

(*type*=NumPy array)

**inputSeries2**: if not None, the second signal to correlate with |inputSeries1| otherwise the correlation will be an autocorrelation.

(*type*=NumPy array)

**Return Value**

an array (length(|inputSeries1|)) storing the result of the correlation.

(*type*=NumPy array)

**Notes:**

- if |inputSeries1| is a multidimensional array the correlation calculation is performed on the first dimension.
- The correlation is computed using the FCA algorithm.

**convolution**(*inputSeries1*, *inputSeries2*)

Returns the numerical convolution between |inputSeries1| and |inputSeries2| one-dimensional NumPy arrays.

**Parameters**

**inputSeries1**: the first signal

(*type*=NumPy array)

**inputSeries2**: the second signal to convolve with |inputSeries1|.

(*type*=NumPy array)

**Return Value**

an array (length(|inputSeries1|)) storing the result of the convolution.

(*type*=NumPy array)

**Notes:**

- if |inputSeries1| is a multidimensional array the convolution calculation is performed on the first dimension.
- the convolution is computed using the convolve function of NumPy package.

**FFT**(*inputSeries*)

Returns the FFT of |inputSeries| multidimensional NumPy array.

**Parameters**

**inputSeries:** the array on which to computes the FFT.  
(*type=NumPy array*)

**Return Value**

the FFT transformed array.  
(*type=NumPy array*)

**Note:** the FFT is computed using the fft function of Scientific.FFT package.

**invFFT**(*inputSeries*)

Returns the inverse FFT of |inputSeries| multidimensional NumPy array.

**Parameters**

**inputSeries:** the array on which to computes the inverse FFT.  
(*type=NumPy array*)

**Return Value**

the inverse FFT transformed array.  
(*type=NumPy array*)

**Note:** the inverse FFT is computed using the inverse\_fft function of Scientific.FFT package.

**gaussianWindow**(*inputSeries, alpha*)

Returns a smoothed signal using |inputSeries| input signal and a gaussian kernel of width |alpha|.

**Parameters**

**inputSeries:** the signal to smooth.  
(*type=NumPy array*)

**alpha:** a float specifying the width of the Gaussian.  
(*type=float*)

**Return Value**

an array (length = 2\*len(|inputSeries|) - 1) containing the smoothed signal.  
(*type=NumPy array*)

**factorial**( $n$ )

Returns  $n!$

**Parameters**

**n**: the  $n$  of  $n!$ .

**Return Value**

$n!$ .

(*type=integer*)

**preparePP**( $j, m, n$ )

Intermediate function used to setup the calculation of spherical harmonics.

**sphericalCoordinates**( $x, y, z$ )

This function returns the  $r$ ,  $\theta$  and  $\phi$  spherical coordinates from the  $x, y, z$  cartesian coordinates.

**Parameters**

**x**: the cartesian  $x$ .

(*type=float*)

**y**: the cartesian  $y$ .

(*type=float*)

**z**: the cartesian  $z$ .

(*type=float*)

**Return Value**

the  $r$ ,  $\theta$  and  $\phi$  spherical coordinates..

(*type=a list of three floats*)

**changeBasis**(*pt, op, ip, jp, kp*)

This function return the coordinates

**Parameters**

**pt**: the coordinates of the point in the old basis.

(*type=Scientific Vector*)

**op**: the coordinates of the new origin in the old basis.

(*type=Scientific Vector*)

**ip**: the coordinates of the new x axis in the old basis.

(*type=Scientific Vector*)

**jp**: the coordinates of the new y axis in the old basis.

(*type=Scientific Vector*)

**kp**: the coordinates of the new z axis in the old basis.

(*type=Scientific Vector*)

**Return Value**

the coordinates of the point in the new basis.

(*type=Scientific Vector*)

**basisVectors**(*parameters*)

Returns the basis vectors for the simulation cell from the six crystallographic parameters.

**Parameters**

**parameters**: a list of six floats defining the simulation cell geometry.

**Return Value**

a list of three Scientific.Geometry.Vector objects representing respectively a, b and c basis vectors.

(*type=list*)

**randomPointInCircle**(*r*, *axis*)

Returns a vector drawn from an uniform distribution within a circle of radius  $|r|$  and orthogonal to vector  $|axis|$ .

**Parameters**

**r:** float specifying the radius of the circle.

(*type=float*)

**axis:** the axis orthogonal to the plane where the vectors have to be generated.

(*type=Scientific.Geometry.Vector object*)

**Return Value**

a vector pointing to a random point of the circle.

(*type=Scientific.Geometry.Vector object*)

**randomDirection2D**(*axis*)

Returns a normalized vector drawn from an uniform distribution on the surface of a unit circle on a plane orthogonal to  $|axis|$ .

**Parameters**

**axis:** the axis orthogonal to the plane where the vectors have to be generated.

(*type=Scientific.Geometry.Vector object*)

**Return Value**

A normalized vector defined in a unit disk orthogonal to  $|axis|$

(*type=Scientific.Geometry.Vector object*)

**randomVector**(*directions=*None)

Returns a normalized random vector on a plane or in space.

**Parameters**

**directions:** if not None, a list of 2 Scientific.Vector that will define the plane on which the vector should be generated.

(*type=list of 2 Scientific.Vector or None*)

**Return Value**

a normalized random vector on a plane defined by  $|directions|$  or in space ( $|directions| = \text{None}$ ).

(*type=Scientific.Geometry.Vector object*)

## 16.2 Variables

| Name | Description                                                                                                                                                |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a2   | a3 = array used to perform order 3 numerical differentiation scheme.<br><b>Value:</b> N.array([[ -3., 4., -1.], [ -1., 0., 1.], [ 1., -4., 3.]])           |
| a3   | a4 = array used to perform order 4 numerical differentiation scheme.<br><b>Value:</b> N.array([[ -11., 18., -9., 2.], [ -2., -3., 6., -1.], [ 1., -6., ... |
| a4   | a5 = N.array used to perform order 5 numerical differentiation scheme.<br><b>Value:</b> N.array([[ -50., 96., -72., 32., -6.], [ -6., -20., 36., -12., ... |
| a5   | <b>Value:</b> N.array([[ -274., 600., -600., 400., -150., 24.], [ -24., -130...                                                                            |

## 17 Module nMOLDYN.Core.Misc

This modules implements the functions and procedures that can not be classified anywhere in the library.

There should not be too much stuff here in order to not mess up the code.

Functions:

- \* `findNestedDirectories` : parses recursively a directory tree appending each nested
- \* `findExecutable` : searches for an executable in OS dependant classical paths

### 17.1 Functions

#### **findNestedDirectories**(*root*, *dirList*)

Parses recursively the directory tree starting from directory |rootDir| appending all the subdirectories found in |dirList| list.

##### **Parameters**

**root:** a string specifying the directory from which the directory tree will be generated.

(*type=string*)

**dirList:** a complete list of all the subdirectories found starting from directory |rootDir|.

(*type=list*)

**Note:** take care this is a recursive function.

#### **findExecutable**(*name*)

Searches for an executable in OS dependant classical paths.

##### **Parameters**

**name:** a string specifying the name of the executable.

(*type=string*)

##### **Return Value**

a string specifying the absolute name of the executable |name| if it could be found an empty string otherwise.

(*type=string*)

**determineNumberOfCPUs()**

Number of virtual or physical CPUs on this system, i.e. user/real as output by time(1) when called with an optimally scaling userspace-only program

**cpuInfo()**

Retrieves the total numbers of processors, the number of loaded and free processors on the host machine.

**Return Value**

the total number of processors on the host machine.

(*type=integer*)

**17.2 Variables**

| Name               | Description                |
|--------------------|----------------------------|
| scannedDirectories | <b>Value:</b> [sys.prefix] |

## 18 Module nMOLDYN.Core.Preferences

This modules stores some the nMOLDYN PREFERENCES variables that will be used throughout all nMOLDYN code.

### 18.1 Variables

| Name        | Description                        |
|-------------|------------------------------------|
| PREFERENCES | <b>Value:</b> nMOLDYNPreferences() |

### 18.2 Class nMOLDYNPreferences

object └─  
           **nMOLDYN.Core.Preferences.nMOLDYNPreferences**

Class whose attributes defines the nMOLDYN PREFERENCES variables.

This class is built on the Singleton principle. That means that one and just one instances of that class will be created.

#### 18.2.1 Methods

**\_\_new\_\_**(*c*)

**Return Value**

a new object with type S, a subtype of T

Overrides: object.\_\_new\_\_ extit(inherited documentation)

**\_\_getattr\_\_**(*self, name*)

**\_\_setattr\_\_**(*self, name, val*)

x.\_\_setattr\_\_('name', value) <==> x.name = value

Overrides: object.\_\_setattr\_\_ extit(inherited documentation)

#### *Inherited from object*

`__delattr__()`, `__getattr__()`, `__hash__()`, `__init__()`, `__reduce__()`, `__reduce_ex__()`,  
`__repr__()`, `__str__()`

**18.2.2 Properties**

| Name                                      | Description |
|-------------------------------------------|-------------|
| <i>Inherited from object</i><br>__class__ |             |

**18.2.3 Class Variables**

| Name     | Description        |
|----------|--------------------|
| instance | <b>Value:</b> None |

## 19 Package nMOLDYN.GUI

### 19.1 Modules

- **ASCIIToNetCDFConversionDialog**: This module implements I{File->Convert ASCII to NetCDF} dialog.  
(Section 20, p. 176)
- **AnalysisBenchmarkDialog**: This module implements I{Help->nMOLDYN benchmark} dialog.  
(Section 21, p. 178)
- **AnalysisDialog**: This module implements I{Analysis->selected analysis} dialog.  
(Section 22, p. 180)
- **AnimationDialog**: This module implements I{View->Animation} dialog.  
(Section 23, p. 182)
- **CheckJobsStatusDialog**: This module implements I{Help->Check job status} dialog.  
(Section 24, p. 185)
- **GeneralInformationsDialog**: This module implements I{Help->About nMOLDYN} dialog.  
(Section 25, p. 188)
- **HTMLReader** (Section 26, p. 191)
  - **mfxtools** (Section 27, p. 192)
  - **mfxutil** (Section 28, p. 193)
  - **tkconst** (Section 29, p. 197)
  - **tkfont** (Section 30, p. 198)
  - **tkhtml** (Section 31, p. 199)
  - **tkinit** (Section 32, p. 211)
  - **tkutil** (Section 33, p. 213)
  - **tkwidget** (Section 34, p. 214)
  - **util** (Section 35, p. 219)
  - **version** (Section 36, p. 221)
- **MainDialog**: This is where the main window of nMOLDYN is defined.  
(Section 37, p. 222)
- **NetCDFToASCIIConversionDialog**: This module implements I{File->Convert NetCDF to ASCII} dialog.  
(Section 38, p. 226)
- **PDBSnapshotGeneratorDialog**: This module implements I{File->Frame snapshot} dialog.  
(Section 39, p. 228)
- **PlotNetCDFVariableDialog**: This module implements I{View->Plot} dialog.  
(Section 40, p. 230)
- **PreferencesDialog**: This module implements I{File->Preferences} dialog.  
(Section 41, p. 234)
- **PyroServerDialog**: This module implements I{View->Animation} dialog.

(Section 42, p. 236)

- **SelectionDialog:** This module implements the atom selection dialog used in almost all nMOLDYN analysis.

(Section 43, p. 239)

- **Tags** (Section 44, p. 244)

- **TrajectoryConversionDialog:** This module implements I{File → Trajectory conversion → converter} dialog.

(Section 45, p. 245)

- **ViewEffectiveModeDialog:** This module implements I{View → Effective Mode} dialog.

(Section 46, p. 251)

- **Widgets:** This module implements all classes used for the generation of combo widgets.

(Section 47, p. 254)

## 20 Module `nMOLDYN.GUI.ASCIIToNetCDFConversionDialog`

This module implements `I{File-->Convert ASCII to NetCDF}` dialog.

Classes:

- \* `ASCIIToNetCDFConversionDialog`: creates `I{File-->Convert ASCII to NetCDF}` dialog used to convert a file in ASCII format to a file in NetCDF format.

### 20.1 Class `ASCIIToNetCDFConversionDialog`

`nMOLDYN.GUI.Widgets.Toplevel` └─ `nMOLDYN.GUI.ASCIIToNetCDFConversionDialog.ASCIIToNetCDFConversionDialog`

Sets up a dialog from where the user can convert a file with numeric data in ASCII or CDL format to NetCDF format.

The ASCII file may contain some comments introduced with the `#` character. These comments will also be written in the NetCDF output file (`|comment|` attribute). The numeric data has to be organized by column. The only restriction is that all the columns should have the same length.

#### 20.1.1 Methods

|                                                                          |
|--------------------------------------------------------------------------|
| <b><code>__init__(self, parent, title=None)</code></b>                   |
| The constructor.                                                         |
| <b>Parameters</b>                                                        |
| <b><code>parent</code></b> : the parent widget.                          |
| <b><code>title</code></b> : a string specifying the title of the dialog. |
| ( <i>type=string</i> )                                                   |

|                                                                   |
|-------------------------------------------------------------------|
| <b><code>body(self, master)</code></b>                            |
| Create dialog body. Return widget that should have initial focus. |

|                                     |
|-------------------------------------|
| <b><code>buttonbox(self)</code></b> |
| Add standard button box.            |

|                                          |
|------------------------------------------|
| <b><code>ok(self, event=None)</code></b> |
|------------------------------------------|

**cancel**(*self*, *event=None*)

**validate**(*self*)

**apply**(*self*)

This method is called when the user clicks on the OK button of the conversion dialog. It performs the conversion from the loaded NetCDF file to the selected ASCII file.

**openASCIIFile**(*self*, *event=None*)

This method/callback is called when the user press Return on the entry of the input file browser or browse directly from the file browser. It will set the filebrowser entry to the name of the browsed file and propose and set a name for the output file based on the basename of the browsed file.

## 21 Module `nMOLDYN.GUI.AnalysisBenchmarkDialog`

This module implements `I{Help-->nMOLDYN benchmark}` dialog.

Classes:

- \* `AnalysisBenchmarkDialog`: creates `I{Help-->nMOLDYN benchmark}` dialog used to run the benchmarks between `nMOLDYN v2.2.5` and the current version of `nMOLDYN`.

### 21.1 Class `AnalysisBenchmarkDialog`

`nMOLDYN.GUI.Widgets.Toplevel` └─ `nMOLDYN.GUI.AnalysisBenchmarkDialog.AnalysisBenchmarkDialog`

Builds the dialog for `nMOLDYN` analysis benchmarks.

These benchmarks will guarantee the stability of the current version with a reference one that is the version 2.2.5 the last official release of `nMOLDYN`.

**Note:** these benchmarks should be performed every time the code source of `nMOLDYN` has been touched for any reason.

#### 21.1.1 Methods

**`__init__(self, parent, title=None)`**

The constructor.

**Parameters**

**`parent`:** the parent widget.

**`title`:** a string specifying the title of the dialog.  
(*type=string*)

**`body(self, master)`**

Creates dialog body. Returns widget that should have initial focus.

**`buttonbox(self)`**

Adds standard button box.

**`ok(self, event=None)`**

**`cancel(self, event=None)`**

---

**validate**(*self*)

---

**apply**(*self*)

---

**infoAboutTest**(*self, event*)

This callback displays basic information about a selected test before running it by clicking on the right mouse button.

---

**displayTest**(*self, event, parameters*)

Displays in a text widget the nMOLDYN v2.2.5 input file of a selected failing test.

Arguments:

- event: the mouse left button click Tkinter event that will trigger the opening of the window.
- testSet: the test set to which belong the failing test.

---

**cancel1**(*self, event=None*)

---

**clearResults**(*self*)

This methods clears up all the results about the evaluated test sets.

## 22 Module *nMOLDYN.GUI.AnalysisDialog*

This module implements `I{Analysis-->selected analysis}` dialog.

Classes:

- \* `AnalysisDialog`: creates `I{Analysis-->selected analysis}` dialog used to setup and/or run an analysis.

### 22.1 Class `AnalysisDialog`



Builds the dialog for *nMOLDYN* analysis.

#### 22.1.1 Methods

**`__init__(self, parent, analysis, trajectory)`**

The constructor.

**Parameters**

- parent**: the parent widget.
- analysis**: the analysis to setup.  
(*type=a class object of one of the *nMOLDYN.Analysis* subclasses.*)
- trajectory**: the loaded trajectory.  
(*type=instance of *MMTK.Trajectory.Trajectory* class*)

**`body(self, master)`**

Create dialog body. Return widget that should have initial focus.

**`buttonbox(self)`**

Add standard button box.

**`ok(self, event=None, runMode='run')`**

**`cancel(self, event=None)`**

**validate**(*self*, *runMode*)

**apply**(*self*, *runMode*)

**estimateAnalysis**(*self*)

Estimates the time taken by the analysis directly from the GUI.

**saveAnalysis**(*self*)

Saves a python script of the analysis that can be run independantly of the GUI.

**runAnalysis**(*self*)

Runs the analysis directly from the GUI.

**saveAndRunAnalysis**(*self*)

This method is called when the user presses the 'Save and Run' button of an analysis dialog. It saves a python script of the analysis and run the analysis directly from the GUI.

**widgetsState**(*self*, *widget*, *state*=DISABLED)

This method is launched when an analysis is run from the GUI. Il will disable all the widgets of the analysis dialog.

## 23 Module *nMOLDYN.GUI.AnimationDialog*

This module implements `I{View-->Animation}` dialog.

Classes:

\* `AnimationDialog`: creates `I{View-->Animation}` dialog used to view an animation of a

### 23.1 Class `AnimationDialog`



Sets up a dialog used to visualize a loaded trajectory.

**Note:** if a trajectory has been previously loaded in *nMOLDYN* for simulation purposes this will be the one proposed for visualization by default. Otherwise, the user can still choose a trajectory to visualize from the dialog.

#### 23.1.1 Methods

**`__init__(self, parent, title=None, trajectory=None)`**

The constructor.

##### Parameters

**parent:** the parent widget.  
**title:** a string specifying the title of the dialog.  
*(type=string)*  
**trajectory:** an opened MMTK trajectory.  
*(type=an instance of MMTK.Trajectory.Trajectory class.)*

Overrides: `Tkinter.BaseWidget.__init__`

|                                             |
|---------------------------------------------|
| <b>body</b> ( <i>self</i> , <i>master</i> ) |
|---------------------------------------------|

|                                                                   |
|-------------------------------------------------------------------|
| Create dialog body. Return widget that should have initial focus. |
|-------------------------------------------------------------------|

|                                  |
|----------------------------------|
| <b>buttonbox</b> ( <i>self</i> ) |
|----------------------------------|

|                          |
|--------------------------|
| Add standard button box. |
|--------------------------|

|                                               |
|-----------------------------------------------|
| <b>ok</b> ( <i>self</i> , <i>event</i> =None) |
|-----------------------------------------------|

|                                                   |
|---------------------------------------------------|
| <b>cancel</b> ( <i>self</i> , <i>event</i> =None) |
|---------------------------------------------------|

|                                 |
|---------------------------------|
| <b>validate</b> ( <i>self</i> ) |
|---------------------------------|

|                              |
|------------------------------|
| <b>apply</b> ( <i>self</i> ) |
|------------------------------|

|                                                           |
|-----------------------------------------------------------|
| <b>openTrajectory</b> ( <i>self</i> , <i>event</i> =None) |
|-----------------------------------------------------------|

|                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The method is called when the user clicks on the 'Browse' button of the trajectory visualization dialog. It opens a file browser. After the file selection some of the dialog widgets are updated with the informations coming from the loaded trajectory. |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|            |
|------------|
| Arguments: |
|------------|

- |                                                                                  |
|----------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• <i>event</i>: Tkinter event.</li> </ul> |
|----------------------------------------------------------------------------------|

*Inherited from Tkinter.BaseWidget*

`destroy()`

*Inherited from Tkinter.Misc*

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`, `option_readfile()`, `pack_propagate()`, `pack_slaves()`, `place_slaves()`, `propagate()`, `quit()`, `register()`, `rowconfigure()`, `selection_clear()`, `selection_get()`, `selection_handle()`, `selection_own()`, `selection_own_get()`, `send()`, `setvar()`, `size()`, `slaves()`, `tk_bisque()`, `tk_focusFollowsMouse()`, `tk_focusNext()`, `tk_focusPrev()`, `tk_menuBar()`, `tk_setPalette()`,

tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Wm***

aspect(), attributes(), client(), colormapwindows(), command(), deiconify(), focus\_model(), frame(), geometry(), grid(), group(), iconbitmap(), iconify(), iconmask(), iconname(), iconposition(), iconwindow(), maxsize(), minsize(), overriddenredirect(), positionfrom(), protocol(), resizable(), sizefrom(), state(), title(), transient(), withdraw(), wm\_aspect(), wm\_attributes(), wm\_client(), wm\_colormapwindows(), wm\_command(), wm\_deiconify(), wm\_focusmodel(), wm\_frame(), wm\_geometry(), wm\_grid(), wm\_group(), wm\_iconbitmap(), wm\_iconify(), wm\_iconmask(), wm\_iconname(), wm\_iconposition(), wm\_iconwindow(), wm\_maxsize(), wm\_minsize(), wm\_overridedirect(), wm\_positionfrom(), wm\_protocol(), wm\_resizable(), wm\_sizefrom(), wm\_state(), wm\_title(), wm\_transient(), wm\_withdraw()

#### **23.1.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg-                            |             |

## 24 Module *nMOLDYN.GUI.CheckJobsStatusDialog*

This module implements I{Help-->Check job status} dialog.

Classes:

\* *CheckJobsStatusDialog*: creates I{Help-->Check job status} dialog used to check the status of the *nMOLDYN* running jobs.

### 24.1 Variables

| Name                              | Description                                                               |
|-----------------------------------|---------------------------------------------------------------------------|
| <code>nmoldyn_package_path</code> | <b>Value:</b><br><code>os.path.dirname(os.path.split(__file__)[0])</code> |

### 24.2 Class *CheckJobsStatusDialog*



Sets up a dialog used to check the status of the *nMOLDYN* running jobs.

#### 24.2.1 Methods

|                                                                                       |
|---------------------------------------------------------------------------------------|
| <b><code>__init__(self, parent, title=None)</code></b>                                |
| The constructor.                                                                      |
| <b>Parameters</b>                                                                     |
| <b>parent</b> : the parent widget.                                                    |
| <b>title</b> : a string specifying the title of the dialog.<br>( <i>type=string</i> ) |
| Overrides: <i>Tkinter.BaseWidget.__init__</i>                                         |

**body**(*self*, *master*)

Create dialog body. Return widget that should have initial focus.

**buttonbox**(*self*)

Add standard button box.

**ok**(*self*, *event*=None)

**cancel**(*self*, *event*=None)

**validate**(*self*)

**apply**(*self*)

**refresh**(*self*)

Refreshes the nMOLDYN running jobs list and its associated frame.

**findJobs**(*self*)

This method find the nMOLDYN active and inactive jobs.

Output:

- a list of the temporary nMOLDYN running job logfiles.

**killJobs**(*self*, *pid*)

This method is called when the user press the button 'Kill' of the dialog. It loops over all the running jobs and for those which have been selected to be killed asks the user to confirm that (s)he really wants to kill them. If so, kills them and updates the dialog.

Arguments:

- *pid*: the pid of the job to kill.

**Inherited from Tkinter.BaseWidget**

`destroy()`

**Inherited from Tkinter.Misc**

`_getitem__()`, `_setitem__()`, `_str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clip-`

board\_clear(), clipboard\_get(), colormodel(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointery(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Wm***

aspect(), attributes(), client(), colormapwindows(), command(), deiconify(), focus-model(), frame(), geometry(), grid(), group(), iconbitmap(), iconify(), iconmask(), iconname(), iconposition(), iconwindow(), maxsize(), minsize(), overriddenredirect(), positionfrom(), protocol(), resizable(), sizefrom(), state(), title(), transient(), withdraw(), wm\_aspect(), wm\_attributes(), wm\_client(), wm\_colormapwindows(), wm\_command(), wm\_deiconify(), wm\_focusmodel(), wm\_frame(), wm\_geometry(), wm\_grid(), wm\_group(), wm\_iconbitmap(), wm\_iconify(), wm\_iconmask(), wm\_iconname(), wm\_iconposition(), wm\_iconwindow(), wm\_maxsize(), wm\_minsize(), wm\_overriddenredirect(), wm\_positionfrom(), wm\_protocol(), wm\_resizable(), wm\_sizefrom(), wm\_state(), wm\_title(), wm\_transient(), wm\_withdraw()

#### **24.2.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg-                            |             |

## 25 Module *nMOLDYN.GUI.GeneralInformationsDialog*

This modules implements I{Help-->About *nMOLDYN*} dialog.

Classes:

- \* *CheckJobsStatusDialog*: creates I{Help-->About *nMOLDYN*} dialog used to display some general informations about *nMOLDYN*.

### 25.1 Variables

| Name                              | Description                                                               |
|-----------------------------------|---------------------------------------------------------------------------|
| <code>nmoldyn_package_path</code> | <b>Value:</b><br><code>os.path.dirname(os.path.split(__file__)[0])</code> |
| <code>aboutAuthors</code>         | <b>Value:</b> ...                                                         |
| <code>aboutHistory</code>         | <b>Value:</b> ...                                                         |
| <code>logo</code>                 | <b>Value:</b> "..."                                                       |

### 25.2 Class *GeneralInformationsDialog*



Sets up a dialog used to visualize some general informations about *NMOLDYN*.

**25.2.1 Methods****\_\_init\_\_**(*self*, *parent*, *title*=None)

The constructor.

**Parameters****parent**: the parent widget.**title**: a string specifying the title of the dialog.*(type=string)*

Overrides: Tkinter.BaseWidget.\_\_init\_\_

**body**(*self*, *master*)

Create dialog body. Return widget that should have initial focus.

**buttonbox**(*self*)

Add standard button box.

**ok**(*self*, *event*=None)**cancel**(*self*, *event*=None)**validate**(*self*)**apply**(*self*)**homepage**(*self*, *website*)***Inherited from Tkinter.BaseWidget***

destroy()

***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get\_boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(),

option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Wm***

aspect(), attributes(), client(), colormapwindows(), command(), deiconify(), focus\_model(), frame(), geometry(), grid(), group(), iconbitmap(), iconify(), iconmask(), iconname(), iconposition(), iconwindow(), maxsize(), minsize(), overriddenredirect(), positionfrom(), protocol(), resizable(), sizefrom(), state(), title(), transient(), withdraw(), wm\_aspect(), wm\_attributes(), wm\_client(), wm\_colormapwindows(), wm\_command(), wm\_deiconify(), wm\_focusmodel(), wm\_frame(), wm\_geometry(), wm\_grid(), wm\_group(), wm\_iconbitmap(), wm\_iconify(), wm\_iconmask(), wm\_iconname(), wm\_iconposition(), wm\_iconwindow(), wm\_maxsize(), wm\_minsize(), wm\_overriddenredirect(), wm\_positionfrom(), wm\_protocol(), wm\_resizable(), wm\_sizefrom(), wm\_state(), wm\_title(), wm\_transient(), wm\_withdraw()

#### **25.2.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg_</code>               |             |

## 26 Package nMOLDYN.GUI.HTMLReader

### 26.1 Modules

- **mfxttools** (*Section 27, p. 192*)
- **mfxutil** (*Section 28, p. 193*)
- **tkconst** (*Section 29, p. 197*)
- **tkfont** (*Section 30, p. 198*)
- **tkhtml** (*Section 31, p. 199*)
- **tkinit** (*Section 32, p. 211*)
- **tkutil** (*Section 33, p. 213*)
- **tkwidget** (*Section 34, p. 214*)
- **util** (*Section 35, p. 219*)
- **version** (*Section 36, p. 221*)

## 27 Module `nMOLDYN.GUI.HTMLReader.mfxtools`

### 27.1 Functions

|                             |
|-----------------------------|
| <code>ustr(<i>s</i>)</code> |
|-----------------------------|

|                                     |
|-------------------------------------|
| <code>indices(<i>object</i>)</code> |
|-------------------------------------|

|                                                                       |
|-----------------------------------------------------------------------|
| <code>trange(<i>start</i>, <i>stop</i>=None, <i>step</i>=None)</code> |
|-----------------------------------------------------------------------|

|                                       |
|---------------------------------------|
| <code>range_len(<i>object</i>)</code> |
|---------------------------------------|

|                                       |
|---------------------------------------|
| <code>reverse(<i>sequence</i>)</code> |
|---------------------------------------|

|                                                         |
|---------------------------------------------------------|
| <code>irange(<i>object</i>, <i>indices</i>=None)</code> |
|---------------------------------------------------------|

|                                                       |
|-------------------------------------------------------|
| <code>count(<i>condition</i>, <i>sequence</i>)</code> |
|-------------------------------------------------------|

|                                                        |
|--------------------------------------------------------|
| <code>exists(<i>condition</i>, <i>sequence</i>)</code> |
|--------------------------------------------------------|

|                                                        |
|--------------------------------------------------------|
| <code>forall(<i>condition</i>, <i>sequence</i>)</code> |
|--------------------------------------------------------|

|                               |
|-------------------------------|
| <code>sgn(<i>expr</i>)</code> |
|-------------------------------|

|                                            |
|--------------------------------------------|
| <code>mfxtools_main(<i>args</i>=[])</code> |
|--------------------------------------------|

### 27.2 Variables

| Name              | Description                               |
|-------------------|-------------------------------------------|
| <code>bool</code> | <b>Value:</b> <code>operator.truth</code> |

## 28 Module `nMOLDYN.GUI.HTMLReader.mfxutil`

### 28.1 Functions

```
static(f, *args, **kw)
```

```
ifelse(expr, val1, val2)
```

```
merge_dict(dict1, dict2, merge_none=1)
```

```
latin1_to_ascii(n)
```

```
latin1_to_html(n)
```

```
hexify(s)
```

```
getusername()
```

```
gethomedir()
```

```
getprefdir(package, home=None)
```

```
destruct(obj)
```

```
kwdefault(kw, **defaults)
```

```
pickle(obj, filename, binmode=0)
```

```
unpickle(filename)
```

```
spawnv(file, args=())
```

```
spawnvp(file, args=())
```

```
spawnSystemSoundMixer(query=0)
```

```
spawnSystemDisplaySettings()
```

```
openURL(url)
```

dumpmem(*dump\_all\_objects*=1)

callername()

callerglobals()

uplevel(*name*)

## 28.2 Variables

| Name             | Description                               |
|------------------|-------------------------------------------|
| thread           | <b>Value:</b> None                        |
| win32api         | <b>Value:</b> None                        |
| EnvError         | <b>Value:</b> IOError, OSError, os.error, |
| htmlentitydefs.i | <b>Value:</b> {}                          |
| usleep           | <b>Value:</b> time.sleep                  |
| uclock           | <b>Value:</b> time.time                   |

## 28.3 Class SubclassResponsibility



### 28.3.1 Methods

#### *Inherited from exceptions.Exception*

`__init__()`, `__new__()`

#### *Inherited from exceptions.BaseException*

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,  
`__setattr__()`, `__setstate__()`, `__str__()`

#### *Inherited from object*

`__hash__()`, `__reduce_ex__()`

### 28.3.2 Properties

| Name                                           | Description |
|------------------------------------------------|-------------|
| <i>Inherited from exceptions.BaseException</i> |             |
| args, message                                  |             |
| <i>Inherited from object</i>                   |             |
| __class__                                      |             |

## 28.4 Class Struct

### 28.4.1 Methods

```
__init__(self, **kw)
```

```
__str__(self)
```

```
__setattr__(self, key, value)
```

```
addattr(self, **kw)
```

```
update(self, dict)
```

```
clear(self)
```

```
copy(self)
```

## 28.5 Class KwStruct

### 28.5.1 Methods

```
__init__(self, kw={}, **defaults)
```

```
__setattr__(self, key, value)
```

```
__getitem__(self, key)
```

```
get(self, key, default=None)
```

|                              |
|------------------------------|
| <b>getKw</b> ( <i>self</i> ) |
|------------------------------|

## 29 Module nMOLDYN.GUI.HTMLReader.tkconst

### 29.1 Variables

| Name            | Description           |
|-----------------|-----------------------|
| tkname          | Value: "tk"           |
| tkversion       | Value: tuple(m [: 4]) |
| TK_DASH_PATCH   | Value: 0              |
| EVENT_HANDLED   | Value: "break"        |
| EVENT_PROPAGATE | Value: None           |
| CURSOR_DRAG     | Value: "hand1"        |
| CURSOR_WATCH    | Value: "watch"        |
| ANCHOR_CENTER   | Value: Tkinter.CENTER |
| ANCHOR_N        | Value: Tkinter.N      |
| ANCHOR_NW       | Value: Tkinter.NW     |
| ANCHOR_NE       | Value: Tkinter.NE     |
| ANCHOR_S        | Value: Tkinter.S      |
| ANCHOR_SW       | Value: Tkinter.SW     |
| ANCHOR_SE       | Value: Tkinter.SE     |
| ANCHOR_W        | Value: Tkinter.W      |
| ANCHOR_E        | Value: Tkinter.E      |

## 30 Module *nMOLDYN.GUI.HTMLReader.tkfont*

### 30.1 Functions

|                                                   |
|---------------------------------------------------|
| <code>getFont(<i>name</i>, <i>cardw</i>=0)</code> |
|---------------------------------------------------|

### 30.2 Variables

| Name          | Description |
|---------------|-------------|
| getFont_cache | Value: {}   |

## 31 Module **nMOLDYN.GUI.HTMLReader.tkhtml**

### 31.1 Functions

`tkhtml_main(args)`

### 31.2 Class **MfxScrolledText**



### 31.2.1 Methods

**`__init__`**(*self*, *parent*=None, ***\*\*cnf***)

Construct a text widget with the parent MASTER.

STANDARD OPTIONS

background, borderwidth, cursor,  
exportselection, font, foreground,  
highlightbackground, highlightcolor,  
highlightthickness, insertbackground,  
insertborderwidth, insertofftime,  
insertontime, insertwidth, padx, pady,  
relief, selectbackground,  
selectborderwidth, selectforeground,  
setgrid, takefocus,  
xscrollcommand, yscrollcommand,

WIDGET-SPECIFIC OPTIONS

autoseparators, height, maxundo,  
spacing1, spacing2, spacing3,  
state, tabs, undo, width, wrap,

Overrides: Tkinter.BaseWidget.\_\_init\_\_ `exitit`(inherited documentation)

**`xview_moveto`**(*self*, *fraction*)

Adjusts the view in the window so that FRACTION of the total width of the canvas is off-screen to the left.

Overrides: Tkinter.Text.xview\_moveto `exitit`(inherited documentation)

**`xview_scroll`**(*self*, *number*, *what*)

Shift the x-view according to NUMBER which is measured in "units" or "pages" (WHAT).

Overrides: Tkinter.Text.xview\_scroll `exitit`(inherited documentation)

**yview\_moveto**(*self*, *fraction*)

Adjusts the view in the window so that FRACTION of the total height of the canvas is off-screen to the top.

Overrides: Tkinter.Text.yview\_moveto exitit(inherited documentation)

**yview\_scroll**(*self*, *number*, *what*)

Shift the y-view according to NUMBER which is measured in "units" or "pages" (WHAT).

Overrides: Tkinter.Text.yview\_scroll exitit(inherited documentation)

### ***Inherited from Tkinter.Text***

bbox(), compare(), debug(), delete(), dlineinfo(), dump(), edit(), edit\_modified(), edit\_redo(), edit\_reset(), edit\_separator(), edit\_undo(), get(), image\_cget(), image\_configure(), image\_create(), image\_names(), index(), insert(), mark\_gravity(), mark\_names(), mark\_next(), mark\_previous(), mark\_set(), mark\_unset(), scan\_dragto(), scan\_mark(), search(), see(), tag\_add(), tag\_bind(), tag\_cget(), tag\_config(), tag\_configure(), tag\_delete(), tag\_lower(), tag\_names(), tag\_nextrange(), tag\_prevrange(), tag\_raise(), tag\_ranges(), tag\_remove(), tag\_unbind(), tk\_textBackspace(), tk\_textIndexCloser(), tk\_textResetAnchor(), tk\_textSelectTo(), window\_cget(), window\_config(), window\_configure(), window\_create(), window\_names(), xview(), yview(), yview\_pickplace()

### ***Inherited from Tkinter.BaseWidget***

destroy()

### ***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), getboolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_types(), keys(), lift(), lower(), mainloop(), name\_towidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), wininfo\_atom(), wininfo\_atomname(), wininfo\_cells(), wininfo\_children(), wininfo\_class(), wininfo\_colormapfull(), wininfo\_containing(), wininfo\_depth(), wininfo\_exists(),

winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(),  
 winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(),  
 winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(),  
 winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(),  
 winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(),  
 winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(),  
 winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(),  
 winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **31.2.2 Class Variables**

| Name                                          | Description |
|-----------------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i><br>_noarg- |             |

### 31.3 Class **MfxReadonlyScrolledText**



**31.3.1 Methods**

```

__init__(self, parent=None, **cnf)

Construct a text widget with the parent MASTER.

STANDARD OPTIONS

 background, borderwidth, cursor,
 exportselection, font, foreground,
 highlightbackground, highlightcolor,
 highlightthickness, insertbackground,
 insertborderwidth, insertofftime,
 insertontime, insertwidth, padx, pady,
 relief, selectbackground,
 selectborderwidth, selectforeground,
 setgrid, takefocus,
 xscrollcommand, yscrollcommand,

WIDGET-SPECIFIC OPTIONS

 autoseparators, height, maxundo,
 spacing1, spacing2, spacing3,
 state, tabs, undo, width, wrap,

Overrides: Tkinter.BaseWidget.__init__ extit(inherited documentation)

```

***Inherited from nMOLDYN.GUI.HTMLReader.tkhtml.MfxScrolledText (Section 31.2)***

xview\_moveto(), xview\_scroll(), yview\_moveto(), yview\_scroll()

***Inherited from Tkinter.Text***

bbox(), compare(), debug(), delete(), dlineinfo(), dump(), edit(), edit\_modified(), edit\_redo(), edit\_reset(), edit\_separator(), edit\_undo(), get(), image\_cget(), image\_configure(), image\_create(), image\_names(), index(), insert(), mark\_gravity(), mark\_names(), mark\_next(), mark\_previous(), mark\_set(), mark\_unset(), scan\_dragto(), scan\_mark(), search(), see(), tag\_add(), tag\_bind(), tag\_cget(), tag\_config(), tag\_configure(), tag\_delete(), tag\_lower(), tag\_names(), tag\_nextrange(), tag\_prevrange(), tag\_raise(), tag\_ranges(), tag\_remove(), tag\_unbind(), tk\_textBackspace(), tk\_textIndexCloser(), tk\_textResetAnchor(), tk\_textSelectTo(), window\_cget(), window\_config(), window\_configure(), window\_create(), window\_names(), xview(), yview(), yview\_pickplace()

***Inherited from Tkinter.BaseWidget***

destroy()

***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), getboolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), wininfo\_atom(), wininfo\_atomname(), wininfo\_cells(), wininfo\_children(), wininfo\_class(), wininfo\_colormapfull(), wininfo\_containing(), wininfo\_depth(), wininfo\_exists(), wininfo\_fpixels(), wininfo\_geometry(), wininfo\_height(), wininfo\_id(), wininfo\_interps(), wininfo\_ismapped(), wininfo\_manager(), wininfo\_name(), wininfo\_parent(), wininfo\_pathname(), wininfo\_pixels(), wininfo\_pointerx(), wininfo\_pointerxy(), wininfo\_pointery(), wininfo\_reqheight(), wininfo\_reqwidth(), wininfo\_rgb(), wininfo\_rootx(), wininfo\_rooty(), wininfo\_screen(), wininfo\_screencells(), wininfo\_screendepth(), wininfo\_screenheight(), wininfo\_screenmmheight(), wininfo\_screenmmwidth(), wininfo\_screenvisual(), wininfo\_screenwidth(), wininfo\_server(), wininfo\_toplevel(), wininfo\_viewable(), wininfo\_visual(), wininfo\_visualid(), wininfo\_visualsavailable(), wininfo\_vrootheight(), wininfo\_vrootwidth(), wininfo\_vrootx(), wininfo\_vrooty(), wininfo\_width(), wininfo\_x(), wininfo\_y()

***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

**31.3.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

### 31.4 Class **tkHTMLWriter**



#### 31.4.1 Methods

**\_\_init\_\_**(*self*, *text*, *viewer*)

Overrides: *formatter.NullWriter.\_\_init\_\_*

**createCallback**(*self*, *href*)

**write**(*self*, *data*)

**anchor\_bgn**(*self*, *href*, *name*, *type*)

**anchor\_end**(*self*)

**anchor\_enter**(*self*, \**args*)

**anchor\_leave**(*self*, \**args*)

**new\_font**(*self*, *font*)

Overrides: *formatter.NullWriter.new\_font*

**new\_margin**(*self*, *margin*, *level*)

Overrides: *formatter.NullWriter.new\_margin*

**send\_label\_data**(*self*, *data*)

Overrides: *formatter.NullWriter.send\_label\_data*

**send\_paragraph**(*self*, *blankline*)

Overrides: *formatter.NullWriter.send\_paragraph*

**send\_hor\_rule**(*self*, \**args*)

Overrides: *formatter.NullWriter.send\_hor\_rule*

***Inherited from formatter.DumbWriter***

reset(), send\_flowin\_data(), send\_line\_break(), send\_literal\_data()

***Inherited from formatter.NullWriter***

flush(), new\_alignment(), new\_spacing(), new\_styles()

**31.5 Class tkHTMLParser****31.5.1 Methods**

**anchor\_bgn**(*self*, *href*, *name*, *type*)

This method is called at the start of an anchor region.

The arguments correspond to the attributes of the <A> tag with the same names. The default implementation maintains a list of hyperlinks (defined by the HREF attribute for <A> tags) within the document. The list of hyperlinks is available as the data attribute anchorlist.

Overrides: htmlib.HTMLParser.anchor\_bgn extit(inherited documentation)

**anchor\_end**(*self*)

This method is called at the end of an anchor region.

The default implementation adds a textual footnote marker using an index into the list of hyperlinks created by the anchor\_bgn() method.

Overrides: htmlib.HTMLParser.anchor\_end extit(inherited documentation)

**do\_dt**(*self*, *attrs*)

Overrides: htmlib.HTMLParser.do\_dt

**handle\_image**(*self, src, alt, ismap, align, width, height*)

This method is called to handle images.

The default implementation simply passes the alt value to the `handle_data()` method.

Overrides: `htmllib.HTMLParser.handle_image` extit(inherited documentation)

### ***Inherited from `htmllib.HTMLParser`***

`__init__()`, `ddpop()`, `do_base()`, `do_br()`, `do_dd()`, `do_hr()`, `do_img()`, `do_isindex()`, `do_li()`, `do_link()`, `do_meta()`, `do_nextid()`, `do_p()`, `do_plaintext()`, `end_a()`, `end_address()`, `end_b()`, `end_blockquote()`, `end_body()`, `end_cite()`, `end_code()`, `end_dir()`, `end_dl()`, `end_em()`, `end_h1()`, `end_h2()`, `end_h3()`, `end_h4()`, `end_h5()`, `end_h6()`, `end_head()`, `end_html()`, `end_i()`, `end_kbd()`, `end_listing()`, `end_menu()`, `end_ol()`, `end_pre()`, `end_samp()`, `end_strong()`, `end_title()`, `end_tt()`, `end_ul()`, `end_var()`, `end_xmp()`, `error()`, `handle_data()`, `reset()`, `save_bgn()`, `save_end()`, `start_a()`, `start_address()`, `start_b()`, `start_blockquote()`, `start_body()`, `start_cite()`, `start_code()`, `start_dir()`, `start_dl()`, `start_em()`, `start_h1()`, `start_h2()`, `start_h3()`, `start_h4()`, `start_h5()`, `start_h6()`, `start_head()`, `start_html()`, `start_i()`, `start_kbd()`, `start_listing()`, `start_menu()`, `start_ol()`, `start_pre()`, `start_samp()`, `start_strong()`, `start_title()`, `start_tt()`, `start_ul()`, `start_var()`, `start_xmp()`, `unknown_endtag()`, `unknown_starttag()`

### ***Inherited from `sgmlib.SGMLParser`***

`close()`, `convert_charref()`, `convert_codepoint()`, `convert_entityref()`, `feed()`, `finish_endtag()`, `finish_shorttag()`, `finish_starttag()`, `get_starttag_text()`, `goahead()`, `handle_charref()`, `handle_comment()`, `handle_decl()`, `handle_endtag()`, `handle_entityref()`, `handle_pi()`, `handle_starttag()`, `parse_endtag()`, `parse_pi()`, `parse_starttag()`, `report_unbalanced()`, `setliteral()`, `setnomoretags()`, `unknown_charref()`, `unknown_entityref()`

### ***Inherited from `markupbase.ParserBase`***

`getpos()`, `parse_comment()`, `parse_declaration()`, `parse_marked_section()`, `unknown_decl()`, `updatepos()`

#### **31.5.2 Class Variables**

| Name                                                 | Description |
|------------------------------------------------------|-------------|
| <i>Inherited from <code>sgmlib.SGMLParser</code></i> |             |
| <code>entity_or_charref</code>                       |             |

## 31.6 Class *tkHTMLViewer*

### 31.6.1 Methods

**`__init__(self, parent)`**

**`initBindings(self)`**

**`destroy(self, *event)`**

**`page_up(self, *event)`**

**`page_down(self, *event)`**

**`unit_up(self, *event)`**

**`unit_down(self, *event)`**

**`scroll_top(self, *event)`**

**`scroll_bottom(self, *event)`**

**`basejoin(self, url, baseurl=None, relpath=1)`**

**`openfile(self, url)`**

**`display(self, url, add=1, relpath=1, xview=0, yview=0)`**

**`addHistory(self, url, xview=0, yview=0)`**

**`updateHistoryXYView(self)`**

**`goBack(self, *event)`**

**`goForward(self, *event)`**

**`goHome(self, *event)`**

**`errorDialog(self, msg)`**

|                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------|
| <b>showImage</b> ( <i>self</i> , <i>src</i> , <i>alt</i> , <i>ismap</i> , <i>align</i> , <i>width</i> , <i>height</i> ) |
|-------------------------------------------------------------------------------------------------------------------------|

## 32 Module nMOLDYN.GUI.HTMLReader.tkinit

### 32.1 Functions

`Misc__destroy(self)`

`Canvas__tag_bind(self, tagOrId, sequence=None, func=None, add=None)`

`Canvas__xview(self, *args)`

`Canvas__xview_moveto(self, fraction)`

`Canvas__xview_scroll(self, number, what)`

`Canvas__yview(self, *args)`

`Canvas__yview_moveto(self, fraction)`

`Canvas__yview_scroll(self, number, what)`

`Wm__wm_state(self, newstate=None)`

`Text__xview_moveto(self, fraction)`

`Text__xview_scroll(self, number, what)`

`Text__yview_moveto(self, fraction)`

`Text__yview_scroll(self, number, what)`

`CanvasItem__bbox(self)`

`Group__bbox(self)`

`CanvasItem__bind(self, sequence=None, command=None, add=None)`

`Group__bind(self, sequence=None, command=None, add=None)`

`CanvasItem__unbind(self, sequence, funcid=None)`

**Group\_\_unbind**(*self*, *sequence*, *funcid*=None)

**CanvasItem\_\_tkraise**(*self*, *abovethis*=None)

**CanvasItem\_\_lower**(*self*, *belowthis*=None)

**Group\_\_tkraise**(*self*, *abovethis*=None)

**Group\_\_lower**(*self*, *belowthis*=None)

**CallWrapper\_\_\_\_call\_\_**(*self*, \**args*)

## 33 Module `nMOLDYN.GUI.HTMLReader.tkutil`

### 33.1 Functions

`wm_withdraw(window)`

`wm_deiconify(window)`

`wm_map(window, maximized=0)`

`wm_set_icon(window, filename)`

`wm_get_geometry(window)`

`setTransient(window, parent, relx=None, rely=None, expose=1)`

`makeToplevel(parent, title=None, class_=None)`

`makeHelpToplevel(parent, title=None, class_=None)`

`bind(widget, sequence, func, add=None)`

`unbind_destroy(widget)`

`after(widget, ms, func, *args)`

`after_idle(widget, func, *args)`

`after_cancel(t)`

`makeImage(file=None, data=None, dither=None, alpha=None)`

`loadImage(file=None, data=None, dither=None, alpha=None)`

`copyImage(image, x, y, width, height)`

`fillImage(image, fill, outline=None)`

`createImage(width, height, fill, outline=None)`

## 34 Module nMOLDYN.GUI.HTMLReader.tkwidget

### 34.1 Functions

|                                         |
|-----------------------------------------|
| <code>tkwidget_main(<i>args</i>)</code> |
|-----------------------------------------|

### 34.2 Class MfxDialog

nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog

```

graph LR
 A[nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog] --| B[nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog]

```

#### 34.2.1 Methods

|                                                                              |
|------------------------------------------------------------------------------|
| <code>__init__(<i>self</i>, <i>parent</i>, <i>title</i>, **<i>kw</i>)</code> |
|------------------------------------------------------------------------------|

Overrides: nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog.\_\_init\_\_

|                                             |
|---------------------------------------------|
| <code>initKw(<i>self</i>, <i>kw</i>)</code> |
|---------------------------------------------|

|                                                   |
|---------------------------------------------------|
| <code>createFrames(<i>self</i>, <i>kw</i>)</code> |
|---------------------------------------------------|

|                                                                  |
|------------------------------------------------------------------|
| <code>createBitmaps(<i>self</i>, <i>frame</i>, <i>kw</i>)</code> |
|------------------------------------------------------------------|

|                                                                  |
|------------------------------------------------------------------|
| <code>createButtons(<i>self</i>, <i>frame</i>, <i>kw</i>)</code> |
|------------------------------------------------------------------|

|                                                |
|------------------------------------------------|
| <code>mDone(<i>self</i>, <i>button</i>)</code> |
|------------------------------------------------|

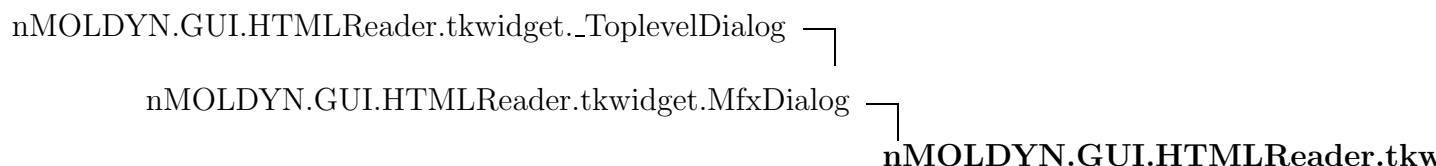
*Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog*

`destroy()`, `getDefaultFont()`, `mCancel()`, `mTimeout()`, `mainloop()`, `wmDeleteWindow()`

#### 34.2.2 Class Variables

| Name                                                                  | Description |
|-----------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog</i> |             |
| <code>img</code>                                                      |             |

### 34.3 Class MfxExceptionDialog



#### 34.3.1 Methods

|                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><code>__init__</code></b> ( <i>self</i> , <i>parent</i> , <i>ex</i> , <i>title</i> ="Error", <i>**kw</i> )<br>Overrides: nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog.__init__ |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

*Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.MfxDialog(Section 34.2)*

createBitmaps(), createButtons(), createFrames(), initKw(), mDone()

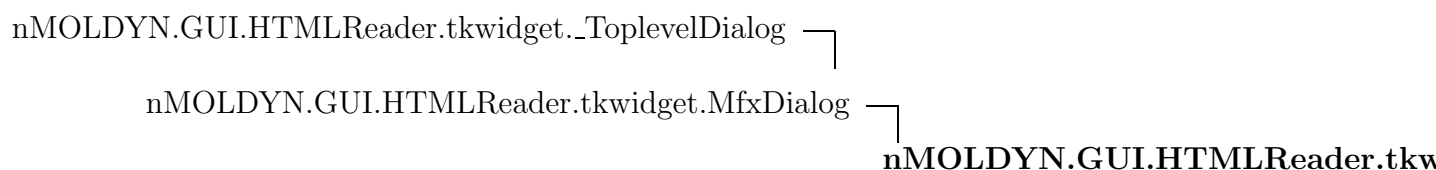
*Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog*

destroy(), getDefaultFont(), mCancel(), mTimeout(), mainloop(), wmDeleteWindow()

#### 34.3.2 Class Variables

| Name                                                                  | Description |
|-----------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog</i> |             |
| img                                                                   |             |

### 34.4 Class MfxSimpleSlider



#### 34.4.1 Methods

|                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b><code>__init__</code></b> ( <i>self</i> , <i>parent</i> , <i>title</i> , <i>label</i> , <i>value</i> , <i>from_</i> , <i>to</i> , <i>resolution</i> , <i>**kw</i> )<br>Overrides: nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog.__init__ |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```
initKw(self, kw)
```

Overrides: nMOLDYN.GUI.HTMLReader.tkwidget.MfxDialog.initKw

```
mDone(self, button)
```

Overrides: nMOLDYN.GUI.HTMLReader.tkwidget.MfxDialog.mDone

*Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.MfxDialog(Section 34.2)*

createBitmaps(), createButtons(), createFrames()

*Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog*

destroy(), getDefaultFont(), mCancel(), mTimeout(), mainloop(), wmDeleteWindow()

### 34.4.2 Class Variables

| Name                                                                  | Description |
|-----------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog</i> |             |
| img                                                                   |             |

## 34.5 Class MfxSimpleEntry

nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog

nMOLDYN.GUI.HTMLReader.tkwidget.MfxDialog

nMOLDYN.GUI.HTMLReader.tkwidget.MfxSimpleSlider

nMOLDYN.GUI.HTMLReader

### 34.5.1 Methods

```
__init__(self, parent, title, label, value, **kw)
```

Overrides: nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog.\_\_init\_\_

*Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.MfxSimpleSlider(Section 34.4)*

initKw(), mDone()

*Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.MfxDialog(Section 34.2)*

createBitmaps(), createButtons(), createFrames()

**Inherited from nMOLDYN.GUI.HTMLReader.tkwidget.\_ToplevelDialog**

destroy(), getDefaultFont(), mCancel(), mTimeout(), mainloop(), wmDeleteWindow()

### 34.5.2 Class Variables

| Name                                                                  | Description |
|-----------------------------------------------------------------------|-------------|
| <i>Inherited from nMOLDYN.GUI.HTMLReader.tkwidget._ToplevelDialog</i> | img         |

## 34.6 Class MfxTooltip

### 34.6.1 Methods

|                                     |
|-------------------------------------|
| <code>__init__(self, widget)</code> |
| <code>setText(self, text)</code>    |
| <code>destroy(self)</code>          |

## 34.7 Class MfxScrolledCanvas

### 34.7.1 Methods

|                                                           |
|-----------------------------------------------------------|
| <code>__init__(self, parent, hbar=2, vbar=2, **kw)</code> |
| <code>destroy(self)</code>                                |
| <code>pack(self, **kw)</code>                             |
| <code>unbind_all(self)</code>                             |
| <code>createFrame(self, kw)</code>                        |
| <code>createCanvas(self, kw)</code>                       |

```
createHbar(self, bg)
```

```
createVbar(self, bg)
```

```
bindHbar(self, w=None)
```

```
bindVbar(self, w=None)
```

```
showHbar(self, show=-1)
```

```
showVbar(self, show=-1)
```

```
page_up(self, *event)
```

```
page_down(self, *event)
```

```
unit_up(self, *event)
```

```
unit_down(self, *event)
```

```
page_left(self, *event)
```

```
page_right(self, *event)
```

```
unit_left(self, *event)
```

```
unit_right(self, *event)
```

```
scroll_top(self, *event)
```

```
scroll_bottom(self, *event)
```

## 35 Module nMOLDYN.GUI.HTMLReader.util

### 35.1 Functions

```
get_version_tuple(version_string)
```

### 35.2 Variables

| Name              | Description                                                          |
|-------------------|----------------------------------------------------------------------|
| PACKAGE           | Value: "PySol"                                                       |
| PACKAGE_URL       | Value: "http://www.oberhumer.com/pysol"                              |
| SUITS             | Value: "Club", "Spade", "Heart", "Diamond"                           |
| COLORS            | Value: "black", "red"                                                |
| RANKS             | Value: "Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jac..." |
| ACE               | Value: 0                                                             |
| JACK              | Value: 10                                                            |
| QUEEN             | Value: 11                                                            |
| KING              | Value: 12                                                            |
| ANY_SUIT          | Value: -1                                                            |
| ANY_COLOR         | Value: -1                                                            |
| ANY_RANK          | Value: -1                                                            |
| NO_SUIT           | Value: 999999                                                        |
| NO_COLOR          | Value: 999999                                                        |
| NO_RANK           | Value: 999999                                                        |
| NO_REDEAL         | Value: 0                                                             |
| UNLIMITED_REDEALS | Value: -1                                                            |
| VARIABLE_REDEALS  | Value: -2                                                            |
| CARDSET           | Value: "cardset"                                                     |
| IMAGE_EXTENSIONS  | Value: ".png", ".gif", ".ppm", ".jpg",                               |
| bundle            | Value: 0                                                             |
| cyclops           | Value: None                                                          |

### 35.3 Class Timer

#### 35.3.1 Methods

```
__init__(self, msg="")
```

```
reset(self)
```

```
get(self)
```

```
__repr__(self)
```

## 35.4 Class *DataLoader*

### 35.4.1 Methods

```
__init__(self, argv0, filenames, path=[])
```

```
findFile(self, filename, subdirs=None)
```

```
findImage(self, filename, subdirs=None)
```

```
findIcon(self, filename=None, subdirs=None)
```

```
findDir(self, filename, subdirs=None)
```

## 36 Module nMOLDYN.GUI.HTMLReader.version

### 36.1 Variables

| Name          | Description          |
|---------------|----------------------|
| VERSION       | Value: "4.82"        |
| VERSION_DATE  | Value: "20 Aug 2003" |
| VERSION_MAJOR | Value: 4             |
| VERSION_MINOR | Value: 82            |
| VERSION_TUPLE | Value: 4, 82         |

## 37 Module nMOLDYN.GUI.MainDialog

This is where the main window of nMOLDYN is defined.

Classes:

- \* MainDialog: The class that defines the nMOLDYN GUI main window and its associated

### 37.1 Variables

| Name                 | Description                                                  |
|----------------------|--------------------------------------------------------------|
| nmoldyn_package_path | <b>Value:</b><br>os.path.dirname(os.path.split(__file__)[0]) |

### 37.2 Class MainDialog



This is the base class for the nMoldyn GUI. It launches the main window of nMoldyn from which different menus can be accessed.

#### 37.2.1 Methods

|                                                                                            |
|--------------------------------------------------------------------------------------------|
| <b><code>__init__(self, netcdf_filename=None)</code></b>                                   |
| The constructor. Displays the main window.                                                 |
| <b>Parameters</b>                                                                          |
| <b>netcdf_filename:</b> a string specifying a NetCDF file, nMOLDYN should be started with. |
| <b>Return Value</b>                                                                        |
| string                                                                                     |
| Overrides: Tkinter.Tk.__init__                                                             |

|                                        |
|----------------------------------------|
| <b><code>body(self, master)</code></b> |
|----------------------------------------|

---

**cancel**(*self*, *event*=None)

---

**loadNetCDF**(*self*, *event*=None, *filename*=None)

This method is launched when the user clicks on the |Load NetCDF| of the |File| menu. It loads the NetCDF file and displays its main informations in the information window.

---

**extractTrajectoryFrame**(*self*, *event*=None)

This method pops up a dialog from where the user can extract a PDB file from a NetCDF trajectory frame.

---

**convertNetCDFToASCII**(*self*, *event*=None)

This method pops up a dialog where the user can proceed to a conversion from a file in NetCDF format to ASCII format.

---

**convertASCIIToNetCDF**(*self*, *event*=None)

This method pops up a dialog where the user can proceed to a conversion from a file in ASCII format to NetCDF format.

---

**checkConfiguration**(*self*)

This method checks for missing external programs and display some warning if it found some.

---

**setPreferences**(*self*, *event*=None)

This method pops up a dialog from where the user can edit the nMOLDYN configuration file.

---

**analysisDialog**(*self*, *analysis*)

---

**plotNetCDF**(*self*, *event*=None)

This method pops up a dialog from where the user can display any numeric 2D or 3D NetCDF variables.

---

**animateTrajectory**(*self*, *event*=None)

This method pops up a dialog from where the user can animate a trajectory. If a trajectory has been loaded for analysis this will be the default one. Otherwise the user can still browse one from the dialog. The animation requires VMD.

**viewEffectiveMode**(*self*, *event*=None)

This method pops up a dialog from where the user can animate an effective mode coming from a QHA analysis. The animation require VMD.

**traceAnalysis**(*self*, *event*=None)

This method pops up a dialog from where the user can check the march of the running jobs. The dialog can be updated dynamically by pressing its button 'refresh'.

**analysisBenchmark**(*self*)

This method pops up a dialog from where the user can perform some analysis benchmark. The benchmark is done between the current version and a reference version that is the version 2.2.5 the last official release of nMOLDYN.

**displayDocumentation**(*self*, *event*=None)

This methode opens the nMOLDYN pdf users guide. The users guide was written by E. Pellegrini, V. Calandrini, P. Calligari, K. Hinsén and G.R. Kneller.

**displayMailingList**(*self*)

This methode opens the nMOLDYN mailing list.

**displayAPI**(*self*, *event*=None)

**aboutNMOLDYN**(*self*, *event*=None)

This method displays general informations about the program such as the developer, the main versions ...

### ***Inherited from Tkinter.Tk***

`__getattr__()`, `destroy()`, `loadtk()`, `readprofile()`, `report_callback_exception()`

### ***Inherited from Tkinter.Misc***

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`,

lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Wm***

aspect(), attributes(), client(), colormapwindows(), command(), deiconify(), focus\_model(), frame(), geometry(), grid(), group(), iconbitmap(), iconify(), iconmask(), iconname(), iconposition(), iconwindow(), maxsize(), minsize(), overriddenredirect(), positionfrom(), protocol(), resizable(), sizefrom(), state(), title(), transient(), withdraw(), wm\_aspect(), wm\_attributes(), wm\_client(), wm\_colormapwindows(), wm\_command(), wm\_deiconify(), wm\_focusmodel(), wm\_frame(), wm\_geometry(), wm\_grid(), wm\_group(), wm\_iconbitmap(), wm\_iconify(), wm\_iconmask(), wm\_iconname(), wm\_iconposition(), wm\_iconwindow(), wm\_maxsize(), wm\_minsize(), wm\_overriddenredirect(), wm\_positionfrom(), wm\_protocol(), wm\_resizable(), wm\_sizefrom(), wm\_state(), wm\_title(), wm\_transient(), wm\_withdraw()

### **37.2.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg-                            |             |

## 38 Module `nMOLDYN.GUI.NetCDFToASCIIConversionDialog`

This module implements I{File-->Convert NetCDF to ASCII} dialog.

Classes:

- \* `NetCDFToASCIIConversionDialog`: creates I{File-->Convert NetCDF to ASCII} dialog used to convert a file in NetCDF format to a file in ASCII format.

### 38.1 Class `NetCDFToASCIIConversionDialog`

`nMOLDYN.GUI.Widgets.Toplevel` └─ `nMOLDYN.GUI.NetCDFToASCIIConversionDialog.NetCDFToASCIIConversionDialog`

Sets up a dialog that allows the conversion of any numeric variables present in a NetCDF file into an ASCII file.

#### 38.1.1 Methods

|                                                                     |
|---------------------------------------------------------------------|
| <b><code>__init__(self, parent, title=None, netcdf=None)</code></b> |
|---------------------------------------------------------------------|

The constructor.

**Parameters**

**parent**: the parent widget.

**title**: a string specifying the title of the dialog.

(*type=string*)

|                                        |
|----------------------------------------|
| <b><code>body(self, master)</code></b> |
|----------------------------------------|

Create dialog body. Return widget that should have initial focus.

|                                     |
|-------------------------------------|
| <b><code>buttonbox(self)</code></b> |
|-------------------------------------|

Add standard button box.

|                                          |
|------------------------------------------|
| <b><code>ok(self, event=None)</code></b> |
|------------------------------------------|

|                                              |
|----------------------------------------------|
| <b><code>cancel(self, event=None)</code></b> |
|----------------------------------------------|

|                                    |
|------------------------------------|
| <b><code>validate(self)</code></b> |
|------------------------------------|

**apply**(*self*)

This method is called when the user clicks on the OK button of the conversion dialog. It performs the conversion from the loaded NetCDF file to the selected ASCII/CDL file.

**openNetCDFFile**(*self*, *event*=None)

This method opens a NetCDF file and updates the dialog with the data read from that

Arguments:

-event: Tkinter event.

**selectVariable**(*self*)**displayNetCDFContents**(*self*)

This method display the variables found in the NetCDF file.

## 39 Module *nMOLDYN.GUI.PDBSnapshotGeneratorDialog*

This module implements I{File-->Frame snapshot} dialog.

Classes:

- \* *PDBSnapshotGeneratorDialog*: creates I{File-->Frame snapshot} dialog used to extract a PDB file from one or several trajectory frame(s).

### 39.1 Class *PDBSnapshotGeneratorDialog*

*nMOLDYN.GUI.Widgets.Toplevel* └─ *nMOLDYN.GUI.PDBSnapshotGeneratorDialog.PDBSnapshotGeneratorDialog*

Sets up a dialog used to export one or several trajectory frames into a PDB file.

**Note:** if a trajectory has been previously loaded in *nMOLDYN* this will be the one proposed for extraction by default. Otherwise, the user can still choose a trajectory to visualize from the dialog.

#### 39.1.1 Methods

|                                                                         |
|-------------------------------------------------------------------------|
| <b><code>__init__(self, parent, title=None, trajectory=None)</code></b> |
|-------------------------------------------------------------------------|

The constructor.

**Parameters**

**parent:** the parent widget.

**title:** a string specifying the title of the dialog.

(*type=string*)

|                                        |
|----------------------------------------|
| <b><code>body(self, master)</code></b> |
|----------------------------------------|

Create dialog body. Return widget that should have initial focus.

|                                     |
|-------------------------------------|
| <b><code>buttonbox(self)</code></b> |
|-------------------------------------|

Add standard button box.

|                                          |
|------------------------------------------|
| <b><code>ok(self, event=None)</code></b> |
|------------------------------------------|

|                                              |
|----------------------------------------------|
| <b><code>cancel(self, event=None)</code></b> |
|----------------------------------------------|

**validate**(*self*)**apply**(*self*)**openTrajectory**(*self*, *event=None*)

This method is called when the user clicks on the 'Browse' button of the trajectory visualization dialog. It opens a file browser. After the file selection some of the dialog widgets are updated with the informations coming from the loaded trajectory.

## 40 Module nMOLDYN.GUI.PlotNetCDFVariableDialog

This modules implements I{View-->Plot} dialog.

Classes:

- \* SettingsDialog: sets up the settings dialog.
- \* ASCIIToNetCDFConversionDialog: creates I{View-->Plot} dialog used to plot NetCDF v

### 40.1 Variables

| Name           | Description                                                                  |
|----------------|------------------------------------------------------------------------------|
| interpolations | <b>Value:</b> ['bessel', 'bilinear', 'bicubic', 'blackman', 'catrom', '...'] |
| colorMaps      | <b>Value:</b> ['autumn', 'bone', 'cool', 'copper', 'flag', 'gray', 'hot...'] |
| lineStyles     | <b>Value:</b> ['- ', '--', '-.', ':', 'None']                                |
| markerStyles   | <b>Value:</b> ['+', '.', '<', '>', 'o', 'p', 's', 'v', 'x', ' ', 'None']     |
| axisScales     | <b>Value:</b> ['linear', 'log']                                              |

### 40.2 Class SettingsDialog

nMOLDYN.GUI.Widgets.Toplevel — nMOLDYN.GUI.PlotNetCDFVariableDialog.SettingsDialog

Sets up a dialog tp perform some settings on the plots.

#### 40.2.1 Methods

|                                                                   |
|-------------------------------------------------------------------|
| <b>__init__</b> ( <i>self</i> , <i>parent</i> )                   |
| The constructor.                                                  |
| <b>Parameters</b>                                                 |
| <i>parent</i> : the parent widget.                                |
| <b>body</b> ( <i>self</i> , <i>master</i> )                       |
| Create dialog body. Return widget that should have initial focus. |

**buttonbox**(*self*)

Add standard button box.

**ok**(*self*, *event*=None)**cancel**(*self*, *event*=None)**validate**(*self*)**apply**(*self*)**initGlobalSettings**(*self*)**initPlotSettings**(*self*)**storeSettings**(*self*)**selectColor**(*self*, *widget*)**changeSettings**(*self*, *event*, *widgetName*)

Argument:

- *event*: either a Tkinter event, either a Tkinter control variable value that has been traced for changes.

**removePreviousPlotSettings**(*self*)

This method removes the previous plot settings widgets.

**addPlotSettingsWidgets**(*self*, *event*=None)

### 40.3 Class *PlotNetCDFVariableDialog*

nMOLDYN.GUI.Widgets.Toplevel

nMOLDYN.GUI.PlotNetCDFVariableDialog.PlotNetCDFV

Sets up a dialog used to plot variables present in a NetCDF file.

**40.3.1 Methods**

**\_\_init\_\_**(*self*, *parent*, *title*=None, *netcdf*=None, *xVar*=None, *yVar*=None, *zVar*=None)

The constructor.

**Parameters**

**parent**: the parent widget.

**title**: a string specifying the title of the dialog.

(*type*=string)

**netcdf**: the name of a NetCDF file to plot (string) or an opened NetCDF trajectory file.

(*type*=a string or a *Scientific.IO.NetCDF.\_NetCDFFile* object)

**xVar**: the NetCDF variable name of the X variable to plot.

(*type*=string)

**yVar**: the NetCDF variable name of the Y variable to plot.

(*type*=)

**zVar**: the NetCDF variable name of the Z variable to plot.

(*type*=)

**body**(*self*, *master*)

Create dialog body. Return widget that should have initial focus.

**buttonbox**(*self*)

Add standard button box.

**ok**(*self*, *event*=None)

**cancel**(*self*, *dialog*, *event*=None)

**validate**(*self*)

**apply**(*self*)

**openSettingsDialog**(*self*)

This method will open the dialog to set up the global settings.

**resetPlots**(*self*)

This method will clear up all the displayed plots.

**exportPlotDialog**(*self*)

This method pops up a dialog from which the plotted datas can be exported to an ASCII file.

**exportPlot**(*self*, *event*=None)

This method exports plotted datas to an ASCII file.

**selectXVariable**(*self*, *event*)**displayVariables**(*self*)

This method display the numeric variables found in the NetCDF file into their appropriate listbox.

**openNetCDF**(*self*, *event*=None)

This method opens a NetCDF file and updates the dialog with the data read from that

Arguments:

    -event: Tkinter event.

**plotXY**(*self*)

This method display a 2D plot.

**displayPlotSlices**(*self*, *event*)

This call back plot the orthogonal slices defined by the moving cursor of a 3D plot.

**plotXYZ**(*self*)

This method display a 2D plot.

## 41 Module `nMOLDYN.GUI.PreferencesDialog`

This module implements `I{File-->Preferences}` dialog.

Classes:

\* `PreferencesDialog`: creates `I{File-->Preferences}` dialog used to to set up `nMOLDYN`

### 41.1 Class `dummy`

#### 41.1.1 Methods

|                             |
|-----------------------------|
| <code>__init__(self)</code> |
|-----------------------------|

### 41.2 Class `PreferencesDialog`

`nMOLDYN.GUI.Widgets.Toplevel` └─ `nMOLDYN.GUI.PreferencesDialog.PreferencesDialog`

Sets up a dialog used to change some PREFERENCES.

#### 41.2.1 Methods

|                                                 |
|-------------------------------------------------|
| <code>__init__(self, parent, title=None)</code> |
|-------------------------------------------------|

The constructor.

**Parameters**

`parent`: the parent widget.

`title`: a string specifying the title of the dialog.

*(type=string)*

|                                 |
|---------------------------------|
| <code>body(self, master)</code> |
|---------------------------------|

Create dialog body. Return widget that should have initial focus.

|                              |
|------------------------------|
| <code>buttonbox(self)</code> |
|------------------------------|

Add standard button box.

|                                   |
|-----------------------------------|
| <code>ok(self, event=None)</code> |
|-----------------------------------|

|                                                   |
|---------------------------------------------------|
| <b>cancel</b> ( <i>self</i> , <i>event</i> =None) |
|---------------------------------------------------|

|                                 |
|---------------------------------|
| <b>validate</b> ( <i>self</i> ) |
|---------------------------------|

|                              |
|------------------------------|
| <b>apply</b> ( <i>self</i> ) |
|------------------------------|

|                                                 |
|-------------------------------------------------|
| This method will set the nMOLDYN configuration. |
|-------------------------------------------------|

|                                   |
|-----------------------------------|
| <b>changePage</b> ( <i>self</i> ) |
|-----------------------------------|

|                                                       |
|-------------------------------------------------------|
| <b>openDirectory</b> ( <i>self</i> , <i>prefKey</i> ) |
|-------------------------------------------------------|

|                                                      |
|------------------------------------------------------|
| <b>openFilename</b> ( <i>self</i> , <i>prefKey</i> ) |
|------------------------------------------------------|

|                                        |
|----------------------------------------|
| <b>savePreferences</b> ( <i>self</i> ) |
|----------------------------------------|

|                                                                                        |
|----------------------------------------------------------------------------------------|
| This method will save the preferences defined in the configuration dialog into a file. |
|----------------------------------------------------------------------------------------|

|                                        |
|----------------------------------------|
| <b>loadPreferences</b> ( <i>self</i> ) |
|----------------------------------------|

|                                                                                                   |
|---------------------------------------------------------------------------------------------------|
| This method will load a preferences file and display its contents into the configuration dialog.. |
|---------------------------------------------------------------------------------------------------|

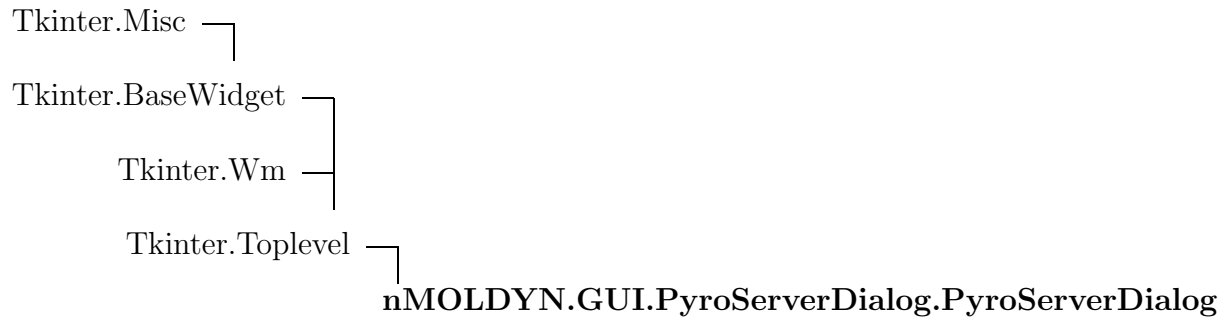
## 42 Module *nMOLDYN.GUI.PyroServerDialog*

This module implements I{View-->Animation} dialog.

Classes:

- \* *PyroServerDialog*: creates the dialog used to set up the Pyro server.

### 42.1 Class *PyroServerDialog*



Sets up a dialog used to set up the Pyro server.

#### 42.1.1 Methods

***\_\_init\_\_***(*self*, *parent*)

The constructor.

**Parameters**

**parent**: the parent widget.

Overrides: *Tkinter.BaseWidget.\_\_init\_\_*

***body***(*self*, *master*)

Create dialog body. Return widget that should have initial focus.

***buttonbox***(*self*)

Add standard button box.

***ok***(*self*, *event=None*)

***cancel***(*self*, *event=None*)

---

**validate**(*self*)

---



---

**apply**(*self*)

---

Builds the pyro server configuration string. This can be: -'monoprocessor' for monoprocessor running mode -'multiprocessor::hostname:number of allocated cpus' -'cluster::node name 1:# allocated cpus for node 1,node name 2:# allocated cpus for node 2 ...'

---

**getValue**(*self*)

---

Returns the pyro server specification string.

---

**readPyroServerConfigFile**(*self, filename*)

---

Reads a pyro server configuration file that contains information about the nodes of be checked for availability.

A pyro server configuration file is an ASCII file that contains either:

- a single line 'host node-name-1 node-name-2 node-name-3...' for an explicit declaration of the nodes that will be checked,

either

- two lines:

- \*'basename basename-for-the-node-list'

- \*'number number1-number2 number3-number4 ...' for a contracted declaration of the nodes to be checked. E.g. convenient if the node names are node1 node2 node3 ...

---

**getCPUInfo**(*self*)

---

Sets the total numbers of processors, the number of loaded and free processors on the host machine or on the different nodes of a cluster.

---

**changePyroServerConfigFile**(*self, event=None*)

---

Loads another pyro server configuration file.

---

**refreshServerInfo**(*self*)

---

Updates the text widget that contains the informations about the pyro server.

**Inherited from *Tkinter.BaseWidget***

destroy()

**Inherited from *Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clip-

board\_clear(), clipboard\_get(), colormodel(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointery(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Wm***

aspect(), attributes(), client(), colormapwindows(), command(), deiconify(), focus-model(), frame(), geometry(), grid(), group(), iconbitmap(), iconify(), iconmask(), iconname(), iconposition(), iconwindow(), maxsize(), minsize(), overriddenredirect(), positionfrom(), protocol(), resizable(), sizefrom(), state(), title(), transient(), withdraw(), wm\_aspect(), wm\_attributes(), wm\_client(), wm\_colormapwindows(), wm\_command(), wm\_deiconify(), wm\_focusmodel(), wm\_frame(), wm\_geometry(), wm\_grid(), wm\_group(), wm\_iconbitmap(), wm\_iconify(), wm\_iconmask(), wm\_iconname(), wm\_iconposition(), wm\_iconwindow(), wm\_maxsize(), wm\_minsize(), wm\_overriddenredirect(), wm\_positionfrom(), wm\_protocol(), wm\_resizable(), wm\_sizefrom(), wm\_state(), wm\_title(), wm\_transient(), wm\_withdraw()

#### **42.1.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

## 43 Module *nMOLDYN.GUI.SelectionDialog*

This module implements the atom selection dialog used in almost all *nMOLDYN* analysis.

The atom selection can be performed for various purposes such as selection of atoms for of hydrogen atoms to deuterate or selection of several group of atoms on which an analysis collectively.

Classes:

- \* *SelectionDialog*: sets up a selection dialog in the scope of an analysis dialog.

### 43.1 Class *SelectionDialog*



Sets up a dialog from which the user can perform an atom selection.

#### 43.1.1 Methods

***\_\_init\_\_(self, parent, selectionType, univContents)***

The constructor.

##### Parameters

- parent:** the parent widget.
- selectionType:** a string being one of 'subset', 'deuteration' or 'group' specifying the atom selection type that will be performed.  
(*type=string*)
- univContents:** a dictionary that contains the universe contents.  
(*type=dict*)

Overrides: *Tkinter.BaseWidget.\_\_init\_\_*

**body**(*self*, *master*)

Create dialog body. Return widget that should have initial focus.

**buttonbox**(*self*)

Add standard button box.

**ok**(*self*, *event*=None)

This method is called when the user clicks on the 'OK' button of the selection editor dialog. It closes the selection editor dialog and build the selection string.

**cancel**(*self*, *event*=None)

Cancel the selection setting up the selection string to a selection type-dependant value.

**validate**(*self*)**apply**(*self*)

Performs a last checking of the selection string before closing the selection dialog.

**getValue**(*self*)

This method returns the self.selectionString class attributes.

Thanks to this method, the selection dialog can be used like any other ComboWidget for which the getValue allows to fetch their contents.

**setDefaultSelectionString**(*self*)

Sets the selection string to its default value. This value depends on the the selection type.

**changeSelectionMedia**(*self*)

Changes the 'media' from which the selection will be performed.

It can be either from a selection file, either from the loaded trajectory or from an expression string. When changing selection media, the previous selection is cleared.

**clear(*self*)**

This methods clears all the listboxes of the 'Selection from the loaded trajectory' browser. It resets the selection listboxes and their associated variables, it resets the selection string, and updates the 'Selection preview' text widget.

**buildSelectionString(*self*)**

This method actually build the selection string out of the `|self.selection|` dictionary.

**displaySelectionString(*self*)**

Displays in the 'Selection preview' textwidget the selection string under process.

**selectFromExpression(*self*, *event*)**

This callback performs a selection from a expression string by writing the expression in its corresponding text widget.

The expression must be a set of valid ;-separated python instructions the last one being 'selection = ...' as the selection string parser will search for the selection variables when executing the expression string.

To refer to the universe just use the variable 'self.universe'.

**selectFromFile(*self*, *event*=None)**

This method/callback performs a selection from a file by selection the file from a browser.

**selectPrefixName(*self*, *event*)**

This callback is called when the user clicks on one item of the 'Group number' listbox of the selection editor.

**deletePrefixName(*self*, *event*)**

This callback will remove the selection string associated to the selected prefix name.

**createNewGroup(*self*)**

This callback will create a new group entry in the 'Group number' listbox.

**selectObjectName(*self*, *event*)**

This callback is called when the user clicks on one item of the 'Object name' listbox of the selection editor. It will display into the 'Selection keyword' listbox all the selection keywords corresponding to the selected object type.

**selectGroupingLevel(*self*, *event*)****selectKeyword(*self*, *event*)**

This callback is called whenever the user clicks on one item of the 'Selection keyword' listbox. It will display in the 'Selection value' listbox the selection values available for the selected keyword.

**deleteObjectName(*self*, *event*)**

This method will delete the object name from the selection string if it was previously selected.

**selectValue(*self*, *event*)**

This callback is called whenever the user clicks on one entry of the 'Selection value' listbox. It will update the current selection.

**appendLinker(*self*, *linker*)**

This method is called when the user press the '(', ')', 'AND' or 'OR' buttons of the atom selection dialog. It:

- checks whether the selected linker can be actually be appended
- appends the linker if so.

**Inherited from *Tkinter.BaseWidget***

`destroy()`

**Inherited from *Tkinter.Misc***

`_getitem__()`, `_setitem__()`, `_str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`,

option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Wm***

aspect(), attributes(), client(), colormapwindows(), command(), deiconify(), focus\_model(), frame(), geometry(), grid(), group(), iconbitmap(), iconify(), iconmask(), iconname(), iconposition(), iconwindow(), maxsize(), minsize(), overriddenredirect(), positionfrom(), protocol(), resizable(), sizefrom(), state(), title(), transient(), withdraw(), wm\_aspect(), wm\_attributes(), wm\_client(), wm\_colormapwindows(), wm\_command(), wm\_deiconify(), wm\_focusmodel(), wm\_frame(), wm\_geometry(), wm\_grid(), wm\_group(), wm\_iconbitmap(), wm\_iconify(), wm\_iconmask(), wm\_iconname(), wm\_iconposition(), wm\_iconwindow(), wm\_maxsize(), wm\_minsize(), wm\_overriddenredirect(), wm\_positionfrom(), wm\_protocol(), wm\_resizable(), wm\_sizefrom(), wm\_state(), wm\_title(), wm\_transient(), wm\_withdraw()

#### **43.1.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg_</code>               |             |

## 44 Module nMOLDYN.GUI.Tags

### 44.1 Variables

| Name | Description |
|------|-------------|
| tags | Value: {}   |

## 45 Module `nMOLDYN.GUI.TrajectoryConversionDialog`

This module implements `I{File --> Trajectory conversion --> converter}` dialog.

Classes:

- \* `AmberNetCDFConverterDialog`: sets up `I{File-->Trajectory conversion --> Amber NetCDF}` dialog to convert an AMBER trajectory to a MMTK NetCDF trajectory.
- \* `CHARMMConverterDialog`: sets up `I{File-->Trajectory conversion --> CHARMM/X-PLOR t}` dialog to convert a CHARMM or X-PLOR trajectory to a MMTK NetCDF trajectory.
- \* `DL_POLYConverterDialog`: sets up `I{File-->Trajectory conversion --> DL_POLY to MMTK}` dialog to convert a DL\_POLY trajectory to a MMTK NetCDF trajectory.
- \* `MaterialsStudioConverterDialog`: sets up `I{File-->Trajectory conversion --> MaterialsStudio}` dialog to convert a MaterialsStudio trajectory to a MMTK NetCDF trajectory.
- \* `ForciteConverterDialog`: sets up `I{File-->Trajectory conversion --> MaterialsStudio --> Forcite to MMTK}` dialog to convert a Forcite trajectory to a MMTK NetCDF trajectory.
- \* `NAMDConverterDialog`: sets up `I{File-->Trajectory conversion --> NAMD to MMTK}` dialog to convert a NAMD trajectory to a MMTK NetCDF trajectory.
- \* `VASPConverterDialog`: sets up `I{File-->Trajectory conversion --> VASP to MMTK}` dialog to convert a VASP trajectory to a MMTK NetCDF trajectory.

### 45.1 Class `AmberNetCDFConverterDialog`

`nMOLDYN.GUI.Widgets.Toplevel`

`nMOLDYN.GUI.TrajectoryConversionDialog.AmberNetCDFConverterDialog`

Sets up a dialog for the conversion from an Amber NetCDF trajectory to a MMTK NetCDF trajectory.

**Note:** the conversion requires a AMber NetCDF and PDB file and the time step in ps between two frames.

#### 45.1.1 Methods

**`__init__(self, parent)`**

The constructor.

**Parameters**

`parent`: the parent widget.

**`body(self, master)`**

Create dialog body. Return widget that should have initial focus.

|                                  |
|----------------------------------|
| <b>buttonbox</b> ( <i>self</i> ) |
|----------------------------------|

|                          |
|--------------------------|
| Add standard button box. |
|--------------------------|

|                                               |
|-----------------------------------------------|
| <b>ok</b> ( <i>self</i> , <i>event</i> =None) |
|-----------------------------------------------|

|                                                   |
|---------------------------------------------------|
| <b>cancel</b> ( <i>self</i> , <i>event</i> =None) |
|---------------------------------------------------|

|                                 |
|---------------------------------|
| <b>validate</b> ( <i>self</i> ) |
|---------------------------------|

|                              |
|------------------------------|
| <b>apply</b> ( <i>self</i> ) |
|------------------------------|

|                                    |
|------------------------------------|
| This method starts the conversion. |
|------------------------------------|

|                                                               |
|---------------------------------------------------------------|
| <b>suggestOutputFilename</b> ( <i>self</i> , * <i>dummy</i> ) |
|---------------------------------------------------------------|

|                                                                                           |
|-------------------------------------------------------------------------------------------|
| This method will propose a name for the output file based on the name of the loaded file. |
|-------------------------------------------------------------------------------------------|

## 45.2 Class CHARMMConverterDialog

nMOLDYN.GUI.Widgets.Toplevel

nMOLDYN.GUI.TrajectoryConversionDialog.CHARMMC

Sets up a dialog used for the conversion from a CHARMM/X-PLOR/NAMD trajectory to a MMTK NetCDF trajectory.

**Note:** the conversion requires CHARMM/X-PLOR DCD and PDB files.

### 45.2.1 Methods

|                                                 |
|-------------------------------------------------|
| <b>__init__</b> ( <i>self</i> , <i>parent</i> ) |
|-------------------------------------------------|

|                  |
|------------------|
| The constructor. |
|------------------|

**Parameters**

**parent:** the parent widget.

|                                             |
|---------------------------------------------|
| <b>body</b> ( <i>self</i> , <i>master</i> ) |
|---------------------------------------------|

|                                                                   |
|-------------------------------------------------------------------|
| Create dialog body. Return widget that should have initial focus. |
|-------------------------------------------------------------------|

|                                  |
|----------------------------------|
| <b>buttonbox</b> ( <i>self</i> ) |
|----------------------------------|

|                          |
|--------------------------|
| Add standard button box. |
|--------------------------|

|                                   |
|-----------------------------------|
| <code>ok(self, event=None)</code> |
|-----------------------------------|

|                                       |
|---------------------------------------|
| <code>cancel(self, event=None)</code> |
|---------------------------------------|

|                             |
|-----------------------------|
| <code>validate(self)</code> |
|-----------------------------|

|                          |
|--------------------------|
| <code>apply(self)</code> |
|--------------------------|

|                                    |
|------------------------------------|
| This method starts the conversion. |
|------------------------------------|

|                                                  |
|--------------------------------------------------|
| <code>suggestOutputFilename(self, *dummy)</code> |
|--------------------------------------------------|

|                                                                                           |
|-------------------------------------------------------------------------------------------|
| This method will propose a name for the output file based on the name of the loaded file. |
|-------------------------------------------------------------------------------------------|

### 45.3 Class `DL_POLYConverterDialog`

`nMOLDYN.GUI.Widgets.Toplevel`

`nMOLDYN.GUI.TrajectoryConversionDialog.DL_POLYCo`

Sets up a dialog used for the conversion from a DL\_POLY trajectory to a MMTK NetCDF trajectory.

**Note:** the conversion requires the DL\_POLY FIELD and HISTORY files and additional information to specify the atoms whose names is not sufficient to determine which element they are.

#### 45.3.1 Methods

|                                     |
|-------------------------------------|
| <code>__init__(self, parent)</code> |
|-------------------------------------|

|                  |
|------------------|
| The constructor. |
|------------------|

**Parameters**

**parent:** the parent widget.

|                                 |
|---------------------------------|
| <code>body(self, master)</code> |
|---------------------------------|

|                                                                   |
|-------------------------------------------------------------------|
| Create dialog body. Return widget that should have initial focus. |
|-------------------------------------------------------------------|

|                              |
|------------------------------|
| <code>buttonbox(self)</code> |
|------------------------------|

|                          |
|--------------------------|
| Add standard button box. |
|--------------------------|

|                                               |
|-----------------------------------------------|
| <b>ok</b> ( <i>self</i> , <i>event</i> =None) |
|-----------------------------------------------|

|                                                   |
|---------------------------------------------------|
| <b>cancel</b> ( <i>self</i> , <i>event</i> =None) |
|---------------------------------------------------|

|                                 |
|---------------------------------|
| <b>validate</b> ( <i>self</i> ) |
|---------------------------------|

|                              |
|------------------------------|
| <b>apply</b> ( <i>self</i> ) |
|------------------------------|

## 45.4 Class *MaterialsStudioConverterDialog*

nMOLDYN.GUI.Widgets.Toplevel

nMOLDYN.GUI.TrajectoryConversionDialog.MaterialsStudioConverterDialog

Sets up a dialog used for the conversion from a MaterialsStudio trajectory to a MMTK NetCDF trajectory.

**Note:** the conversion requires MaterialsStudio XTD or XSD file and HIS or TRJ files according to the MaterialsStudio modules used to produce the trajectory (HIS for Discover and TRJ for Forcite).

### 45.4.1 Methods

|                                                                 |
|-----------------------------------------------------------------|
| <b>__init__</b> ( <i>self</i> , <i>parent</i> , <i>module</i> ) |
|-----------------------------------------------------------------|

The constructor.

**Parameters**

**parent:** the parent widget.

|                                             |
|---------------------------------------------|
| <b>body</b> ( <i>self</i> , <i>master</i> ) |
|---------------------------------------------|

Create dialog body. Return widget that should have initial focus.

|                                  |
|----------------------------------|
| <b>buttonbox</b> ( <i>self</i> ) |
|----------------------------------|

Add standard button box.

|                                               |
|-----------------------------------------------|
| <b>ok</b> ( <i>self</i> , <i>event</i> =None) |
|-----------------------------------------------|

|                                                   |
|---------------------------------------------------|
| <b>cancel</b> ( <i>self</i> , <i>event</i> =None) |
|---------------------------------------------------|

|                                 |
|---------------------------------|
| <b>validate</b> ( <i>self</i> ) |
|---------------------------------|

|                              |
|------------------------------|
| <b>apply</b> ( <i>self</i> ) |
|------------------------------|

|                                                               |
|---------------------------------------------------------------|
| <b>suggestOutputFilename</b> ( <i>self</i> , * <i>dummy</i> ) |
|---------------------------------------------------------------|

|                                                                                           |
|-------------------------------------------------------------------------------------------|
| This method will propose a name for the output file based on the name of the loaded file. |
|-------------------------------------------------------------------------------------------|

## 45.5 Class NAMDCConverterDialog

nMOLDYN.GUI.Widgets.Toplevel

nMOLDYN.GUI.TrajectoryConversionDialog.NAMDCConv

Sets up a dialog used for the conversion from a CHARMM/X-PLOR/NAMD trajectory to a MMTK NetCDF trajectory.

**Note:** the conversion requires CHARMM/X-PLOR/NAMD DCD and PDB files and optionally the simulation box dimensions.

### 45.5.1 Methods

|                                                 |
|-------------------------------------------------|
| <b>__init__</b> ( <i>self</i> , <i>parent</i> ) |
|-------------------------------------------------|

|                  |
|------------------|
| The constructor. |
|------------------|

|                                             |
|---------------------------------------------|
| <b>body</b> ( <i>self</i> , <i>master</i> ) |
|---------------------------------------------|

|                                                                   |
|-------------------------------------------------------------------|
| Create dialog body. Return widget that should have initial focus. |
|-------------------------------------------------------------------|

|                                  |
|----------------------------------|
| <b>buttonbox</b> ( <i>self</i> ) |
|----------------------------------|

|                          |
|--------------------------|
| Add standard button box. |
|--------------------------|

|                                               |
|-----------------------------------------------|
| <b>ok</b> ( <i>self</i> , <i>event</i> =None) |
|-----------------------------------------------|

|                                                   |
|---------------------------------------------------|
| <b>cancel</b> ( <i>self</i> , <i>event</i> =None) |
|---------------------------------------------------|

|                                 |
|---------------------------------|
| <b>validate</b> ( <i>self</i> ) |
|---------------------------------|

|                              |
|------------------------------|
| <b>apply</b> ( <i>self</i> ) |
|------------------------------|

|                                    |
|------------------------------------|
| This method starts the conversion. |
|------------------------------------|

|                                                               |
|---------------------------------------------------------------|
| <b>suggestOutputFilename</b> ( <i>self</i> , * <i>dummy</i> ) |
|---------------------------------------------------------------|

|                                                                                           |
|-------------------------------------------------------------------------------------------|
| This method will propose a name for the output file based on the name of the loaded file. |
|-------------------------------------------------------------------------------------------|

## 45.6 Class *VASPCConverterDialog*

nMOLDYN.GUI.Widgets.Toplevel

nMOLDYN.GUI.TrajectoryConversionDialog.VASPCConver

Sets up a dialog used for the conversion from a VASP trajectory to a MMTK NetCDF trajectory.

**Note:** the conversion requires the VASP CONTCAR or POSCAR and XDATCAR files.

### 45.6.1 Methods

|                                                 |
|-------------------------------------------------|
| <b>__init__</b> ( <i>self</i> , <i>parent</i> ) |
|-------------------------------------------------|

|                  |
|------------------|
| The constructor. |
|------------------|

|                                             |
|---------------------------------------------|
| <b>body</b> ( <i>self</i> , <i>master</i> ) |
|---------------------------------------------|

|                                                                   |
|-------------------------------------------------------------------|
| Create dialog body. Return widget that should have initial focus. |
|-------------------------------------------------------------------|

|                                  |
|----------------------------------|
| <b>buttonbox</b> ( <i>self</i> ) |
|----------------------------------|

|                          |
|--------------------------|
| Add standard button box. |
|--------------------------|

|                                               |
|-----------------------------------------------|
| <b>ok</b> ( <i>self</i> , <i>event</i> =None) |
|-----------------------------------------------|

|                                                   |
|---------------------------------------------------|
| <b>cancel</b> ( <i>self</i> , <i>event</i> =None) |
|---------------------------------------------------|

|                                 |
|---------------------------------|
| <b>validate</b> ( <i>self</i> ) |
|---------------------------------|

|                              |
|------------------------------|
| <b>apply</b> ( <i>self</i> ) |
|------------------------------|

## 46 Module `nMOLDYN.GUI.ViewEffectiveModeDialog`

This module implements `I{View --> Effective Mode}` dialog.

Classes:

- \* `ViewEffectiveModeDialog`: creates `I{View --> Effective Mode}` dialog used to view an animation of the effective modes resulting from a Quasi Harmonic Analysis.

### 46.1 Class `ViewEffectiveModeDialog`



Sets up a dialog used to visualize the effective modes resulting from a QHA analysis.

#### 46.1.1 Methods

**`__init__(self, parent, title=None)`**

The constructor.

**Parameters**

**parent**: the parent widget.

**title**: a string specifying the title of the dialog.

(*type=string*)

Overrides: `Tkinter.BaseWidget.__init__`

**`body(self, master)`**

Create dialog body. Return widget that should have initial focus.

**`buttonbox(self)`**

Add standard button box.

**`ok(self, event=None)`**

**cancel**(*self*, *event*=None)

**validate**(*self*)

**apply**(*self*)

**openNetCDFFile**(*self*, *event*=None)

This method open the NetCDF that contains the effective modes. Arguments:

- *event*: Tkinter event.

**Inherited from Tkinter.BaseWidget**

`destroy()`

**Inherited from Tkinter.Misc**

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletcommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`, `option_readfile()`, `pack_propagate()`, `pack_slaves()`, `place_slaves()`, `propagate()`, `quit()`, `register()`, `rowconfigure()`, `selection_clear()`, `selection_get()`, `selection_handle()`, `selection_own()`, `selection_own_get()`, `send()`, `setvar()`, `size()`, `slaves()`, `tk_bisque()`, `tk_focusFollowsMouse()`, `tk_focusNext()`, `tk_focusPrev()`, `tk_menuBar()`, `tk_setPalette()`, `tk_strictMotif()`, `tkraise()`, `unbind()`, `unbind_all()`, `unbind_class()`, `update()`, `update_idletasks()`, `wait_variable()`, `wait_visibility()`, `wait_window()`, `waitvar()`, `winfo_atom()`, `winfo_atomname()`, `winfo_cells()`, `winfo_children()`, `winfo_class()`, `winfo_colormapfull()`, `winfo_containing()`, `winfo_depth()`, `winfo_exists()`, `winfo_fpixels()`, `winfo_geometry()`, `winfo_height()`, `winfo_id()`, `winfo_interps()`, `winfo_ismapped()`, `winfo_manager()`, `winfo_name()`, `winfo_parent()`, `winfo_pathname()`, `winfo_pixels()`, `winfo_pointerx()`, `winfo_pointerxy()`, `winfo_pointery()`, `winfo_reqheight()`, `winfo_reqwidth()`, `winfo_rgb()`, `winfo_rootx()`, `winfo_rooty()`, `winfo_screen()`, `winfo_screencells()`, `winfo_screendepth()`, `winfo_screenheight()`, `winfo_screenmmheight()`, `winfo_screenmmwidth()`, `winfo_screenvisual()`, `winfo_screenwidth()`, `winfo_server()`, `winfo_toplevel()`, `winfo_viewable()`, `winfo_visual()`, `winfo_visualid()`, `winfo_visualsavailable()`, `winfo_vrootheight()`, `winfo_vrootwidth()`, `winfo_vrootx()`, `winfo_vrooty()`, `winfo_width()`, `winfo_x()`, `winfo_y()`

**Inherited from Tkinter.Wm**

`aspect()`, `attributes()`, `client()`, `colormapwindows()`, `command()`, `deiconify()`, `focus-model()`, `frame()`, `geometry()`, `grid()`, `group()`, `iconbitmap()`, `iconify()`, `iconmask()`, `iconname()`, `iconposition()`, `iconwindow()`, `maxsize()`, `minsize()`, `overrideredirect()`, `positionfrom()`, `protocol()`, `resizable()`, `sizefrom()`, `state()`, `title()`, `transient()`, `withdraw()`, `wm_aspect()`, `wm_attributes()`, `wm_client()`, `wm_colormapwindows()`, `wm_command()`, `wm_deiconify()`, `wm_focusmodel()`, `wm_frame()`, `wm_geometry()`, `wm_grid()`, `wm_group()`, `wm_iconbitmap()`, `wm_iconify()`, `wm_iconmask()`, `wm_iconname()`, `wm_iconposition()`, `wm_iconwindow()`, `wm_maxsize()`, `wm_minsize()`, `wm_overrideredirect()`, `wm_positionfrom()`, `wm_protocol()`, `wm_resizable()`, `wm_sizefrom()`, `wm_state()`, `wm_title()`, `wm_transient()`, `wm_withdraw()`

#### 46.1.2 Class Variables

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

## 47 Module *nMOLDYN.GUI.Widgets*

This module implements all classes used for the generation of combo widgets.

A combo widget can be defined as a label frame encapsulating several other Tkinter widgets.

Classes:

- \* `ValidatingEntry` : sets up a Tkinter Entry widget that checks its contents whenever the content changes.
- \* `IntegerEntry` : sets up a Tkinter Entry widget that checks that the input is an integer.
- \* `FloatEntry` : sets up a Tkinter Entry widget that checks that the input is a float.
- \* `StringEntry` : sets up a Tkinter Entry widget that checks that the input is a string.
- \* `ComboLabelEntry` : sets up a combo widget made of a Tkinter Label widget embedded in a Tkinter LabelFrame widget.
- \* `ComboIntegerEntry` : sets up a combo widget made of a Tkinter Label widget and an IntegerEntry widget embedded in a Tkinter LabelFrame widget.
- \* `ComboFloatEntry` : sets up a combo widget made of a Tkinter Label widget and a FloatEntry widget embedded in a Tkinter LabelFrame widget.
- \* `ComboStringEntry` : sets up a combo widget made of a Tkinter Label widget and a StringEntry widget embedded in a Tkinter LabelFrame widget.
- \* `ComboRadiobutton` : sets up a combo widget made of a Tkinter Label widget and a Tkinter Radiobutton widget embedded in a Tkinter LabelFrame widget.
- \* `ComboCheckbutton` : sets up a combo widget made of a Tkinter Label widget and a Tkinter Checkbutton widget embedded in a Tkinter LabelFrame widget.
- \* `ComboOptionMenu` : sets up a combo widget made of a Tkinter Label widget and a Tkinter OptionMenu widget embedded in a Tkinter LabelFrame widget.
- \* `ComboSpinbox` : sets up a combo widget made of a Tkinter Label widget and a Tkinter Spinbox widget embedded in a Tkinter LabelFrame widget.
- \* `ComboScale` : sets up a combo widget made of a Tkinter Label widget and a Tkinter Scale widget embedded in a Tkinter LabelFrame widget.
- \* `ComboButton` : sets up a combo widget made of a Tkinter Button widget and optional Tkinter widgets embedded in a Tkinter LabelFrame widget.
- \* `ComboFileBrowser` : sets up a combo widget made of a Tkinter Label widget, a StringEntry widget for file browsing embedded in a Tkinter LabelFrame widget.
- \* `ComboText` : sets up a combo widget made of a Tkinter Text widget and a vertical scrollbar widget embedded in a Tkinter LabelFrame widget.
- \* `ComboListbox` : sets up a combo widget made of a Tkinter Listbox widget and a vertical scrollbar widget embedded in a Tkinter LabelFrame widget.
- \* `StatusBar` : sets up a combo widget made of a Tkinter Scale widget embedded in a Tkinter LabelFrame widget.

The following classes were adapted from the implementation of Frederic Lundh:

- `ValidatingEntry`
- `IntegerEntry`
- `FloatEntry`

## 47.1 Functions

### **changeMouseCursor**(*event*, *cursor*)

Changes the mouse cursor aspect.

#### **Parameters**

**event**: the event triggering the change of mouse cursor.

(*type=a Tkinter.Event object*)

**cursor**: the Tkinter name of the mouse cursor.

(*type=string*)

### **loadHelpTree**()

This method will load the XML file that stores the pdf and html links for the help file and store its contents into the `_helptree` nMOLDYN global variables..

### **informationAboutWidget**(*event*)

Checks the label of the Tkinter LabelFrame of a combo widget and displays the corresponding informations documented in nMOLDYN users guide.

#### **Parameters**

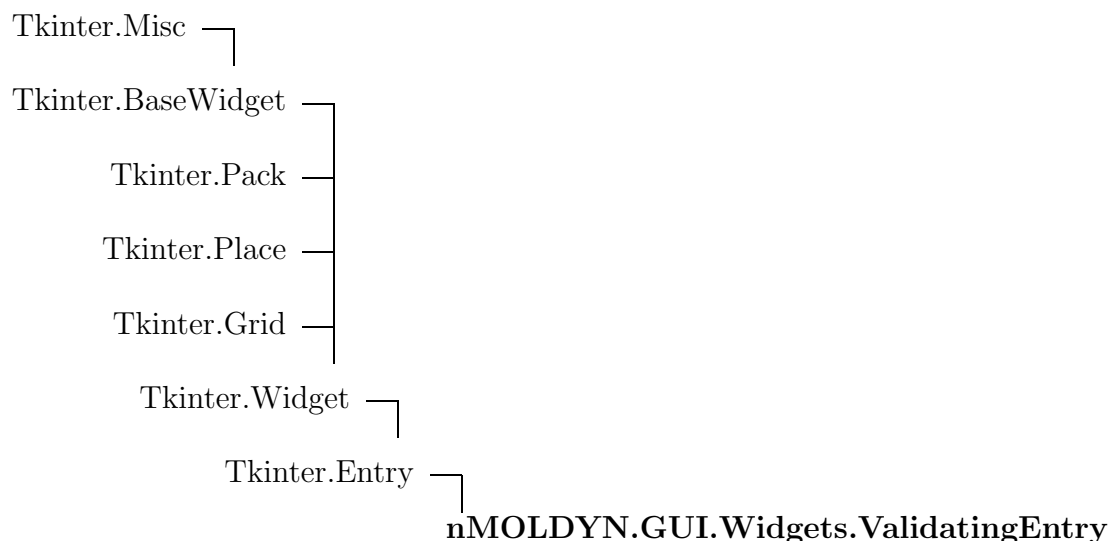
**event**: the event triggering the request for information about the combo widget.

(*type=a Tkinter.Event object*)

## 47.2 Variables

| Name                              | Description                                                               |
|-----------------------------------|---------------------------------------------------------------------------|
| <code>nmoldyn_package_path</code> | <b>Value:</b><br><code>os.path.dirname(os.path.split(__file__)[0])</code> |
| <code>nmoldyn_xml_tree</code>     | <b>Value:</b> <code>{}</code>                                             |

### 47.3 Class ValidatingEntry



Base class for `|IntegerEntry|`, `|FloatEntry|` and `|StringEntry|` widgets

#### 47.3.1 Methods

```
__init__(self, master, contents='', **kwargs)
```

The constructor.

**Parameters**

**master:** the parent widget.

**contents:** the contents of the Tkinter Entry widget.  
(*type=string*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

```
checkValue(self, event=None, contents=None)
```

```
setValue(self, value)
```

Sets the value of the control variable linked to the Tkinter Entry widget.

```
getValue(self)
```

Returns the value of the control variable linked to the Tkinter Entry widget.

**Inherited from Tkinter.Entry**

`delete()`, `get()`, `icursor()`, `index()`, `insert()`, `scan_dragto()`, `scan_mark()`, `select_adjust()`,

select\_clear(), select\_from(), select\_present(), select\_range(), select\_to(), selection\_adjust(),  
 selection\_clear(), selection\_from(), selection\_present(), selection\_range(), selection\_to(),  
 xview(), xview\_moveto(), xview\_scroll()

### ***Inherited from Tkinter.BaseWidget***

destroy()

### ***Inherited from Tkinter.Misc***

\_getitem\_(), \_setitem\_(), \_str\_(), after(), after\_cancel(), after\_idle(), bbox(),  
 bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clip-  
 board\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), config-  
 ure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(),  
 focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-  
 boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(),  
 grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(),  
 grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(),  
 lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(),  
 option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(),  
 register(), rowconfigure(), selection\_get(), selection\_handle(), selection\_own(), se-  
 lection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(),  
 tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(),  
 unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(),  
 wait\_visibility(), wait\_window(), waitvar(), wininfo\_atom(), wininfo\_atomname(), wininfo\_cells(),  
 wininfo\_children(), wininfo\_class(), wininfo\_colormapfull(), wininfo\_containing(), wininfo\_depth(),  
 wininfo\_exists(), wininfo\_fpixels(), wininfo\_geometry(), wininfo\_height(), wininfo\_id(), wininfo\_interps(),  
 wininfo\_ismapped(), wininfo\_manager(), wininfo\_name(), wininfo\_parent(), wininfo\_pathname(),  
 wininfo\_pixels(), wininfo\_pointerx(), wininfo\_pointerxy(), wininfo\_pointery(), wininfo\_reqheight(),  
 wininfo\_reqwidth(), wininfo\_rgb(), wininfo\_rootx(), wininfo\_rooty(), wininfo\_screen(), wininfo\_screencells(),  
 wininfo\_screendepth(), wininfo\_screenheight(), wininfo\_screenmmheight(), wininfo\_screenmmwidth(),  
 wininfo\_screenvisual(), wininfo\_screenwidth(), wininfo\_server(), wininfo\_toplevel(), wininfo\_viewable(),  
 wininfo\_visual(), wininfo\_visualid(), wininfo\_visualsavailable(), wininfo\_vrootheight(), wininfo\_vrootwidth(),  
 wininfo\_vrootx(), wininfo\_vrooty(), wininfo\_width(), wininfo\_x(), wininfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

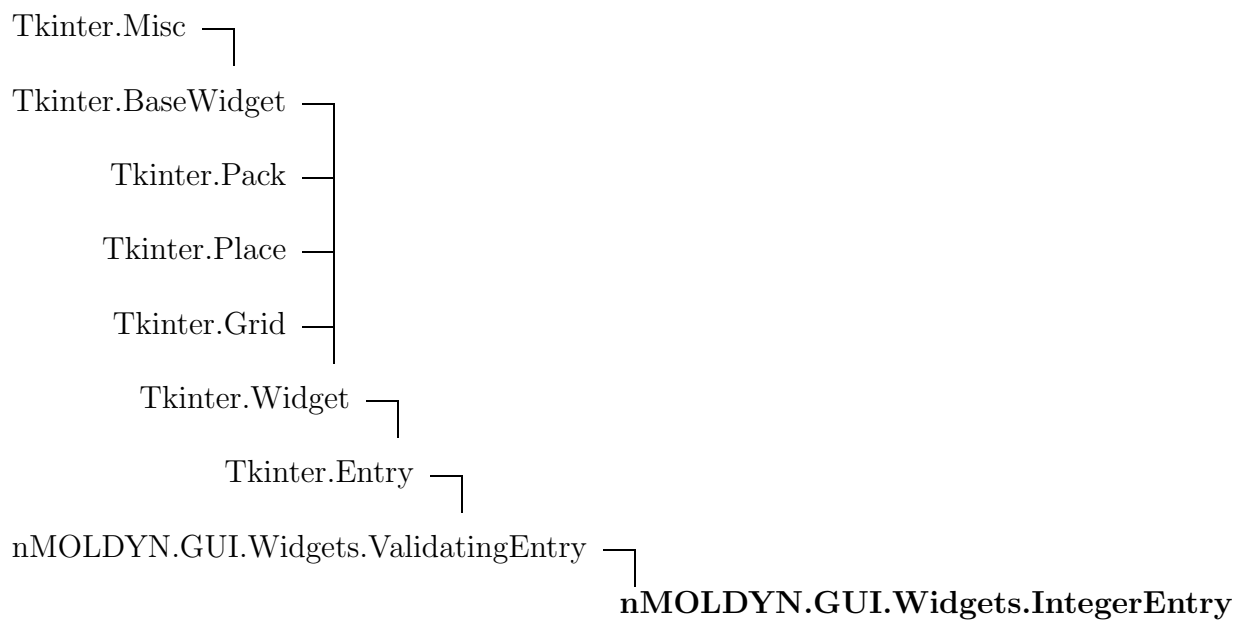
### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

## 47.3.2 Class Variables

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

## 47.4 Class IntegerEntry



Sets up a Tkinter Entry widget that accepts only integer values.

**Note:** subclass of `nMOLDYN.GUI.Widgets.ValidatingEntry` class.

## 47.4.1 Methods

**checkValue**(*self*, *event*=None, *contents*=None)

Check that the contents of the Tkinter Entry widget is actually an integer.

**Parameters**

**event:** the keyboard event that triggers the checking of the contents of the Tkinter Entry widget.

(*type*=a Tkinter.Event object)

**contents:** the contents of the Tkinter Entry widget to check.

(*type*=integer)

Overrides: nMOLDYN.GUI.Widgets.ValidatingEntry.checkValue

**setValue**(*self*, *value*=None)

Sets the value of the control variable linked to the Tkinter Entry widget.

Overrides: nMOLDYN.GUI.Widgets.ValidatingEntry.setValue

**getValue**(*self*)

Returns the value of the control variable linked to the Tkinter Entry widget if it is an integer otherwise throws an error.

Overrides: nMOLDYN.GUI.Widgets.ValidatingEntry.getValue

**Inherited from nMOLDYN.GUI.Widgets.ValidatingEntry (Section 47.3)**

`__init__()`

**Inherited from Tkinter.Entry**

`delete()`, `get()`, `icursor()`, `index()`, `insert()`, `scan_dragto()`, `scan_mark()`, `select_adjust()`, `select_clear()`, `select_from()`, `select_present()`, `select_range()`, `select_to()`, `selection_adjust()`, `selection_clear()`, `selection_from()`, `selection_present()`, `selection_range()`, `selection_to()`, `xview()`, `xview_moveto()`, `xview_scroll()`

**Inherited from Tkinter.BaseWidget**

`destroy()`

**Inherited from Tkinter.Misc**

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`,

focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

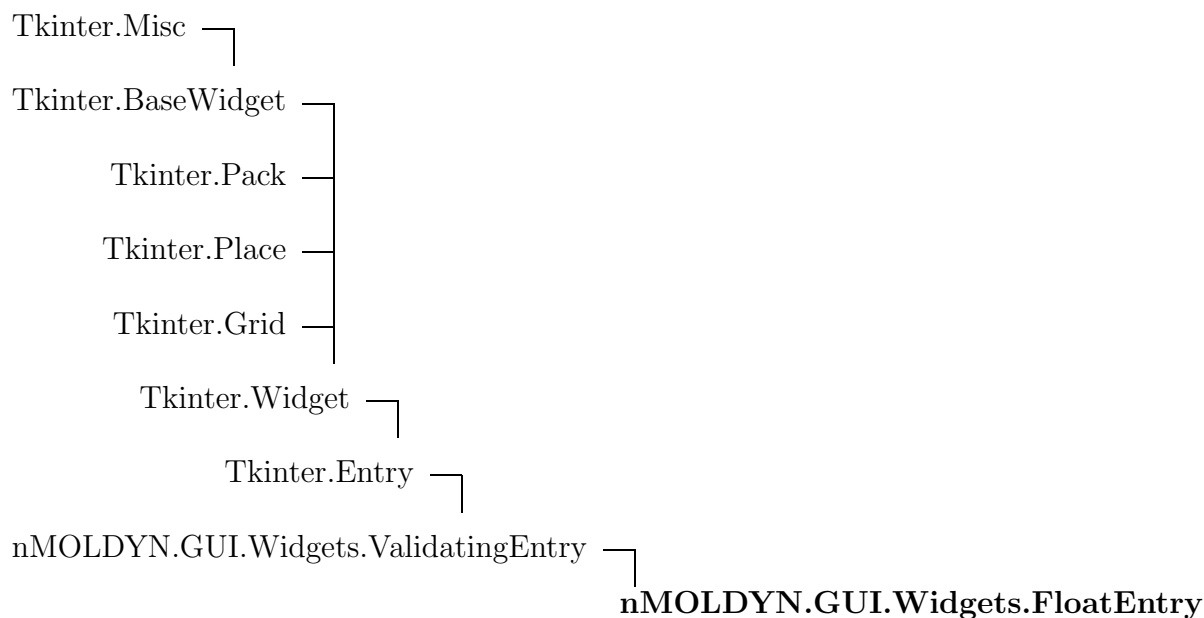
### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.4.2 Class Variables**

| Name                                          | Description |
|-----------------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i><br>_noarg- |             |

## 47.5 Class FloatEntry



Sets up a Tkinter Entry widget that accepts only float values.

**Note:** subclass of `nMOLDYN.GUI.Widgets.ValidatingEntry` class.

### 47.5.1 Methods

**checkValue**(*self*, *event=None*, *contents=None*)

Check that the contents of the Tkinter Entry widget is actually a float.

**Parameters**

**event:** the keyboard event that triggers the checking of the contents of the Tkinter Entry widget.

(*type=a Tkinter.Event object*)

**contents:** the contents of the Tkinter Entry widget to check.

(*type=float*)

Overrides: `nMOLDYN.GUI.Widgets.ValidatingEntry.checkValue`

**setValue**(*self*, *value*)

Sets the value of the control variable linked to the Tkinter Entry widget.

Overrides: `nMOLDYN.GUI.Widgets.ValidatingEntry.setValue`

**getValue(*self*)**

Returns the value of the control variable linked to the Tkinter Entry widget if it is a float otherwise throws an error.

Overrides: nMOLDYN.GUI.Widgets.ValidatingEntry.getValue

**Inherited from nMOLDYN.GUI.Widgets.ValidatingEntry(Section 47.3)**

`__init__()`

**Inherited from Tkinter.Entry**

`delete()`, `get()`, `icursor()`, `index()`, `insert()`, `scan_dragto()`, `scan_mark()`, `select_adjust()`, `select_clear()`, `select_from()`, `select_present()`, `select_range()`, `select_to()`, `selection_adjust()`, `selection_clear()`, `selection_from()`, `selection_present()`, `selection_range()`, `selection_to()`, `xview()`, `xview_moveto()`, `xview_scroll()`

**Inherited from Tkinter.BaseWidget**

`destroy()`

**Inherited from Tkinter.Misc**

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`, `option_readfile()`, `pack_propagate()`, `pack_slaves()`, `place_slaves()`, `propagate()`, `quit()`, `register()`, `rowconfigure()`, `selection_get()`, `selection_handle()`, `selection_own()`, `selection_own_get()`, `send()`, `setvar()`, `size()`, `slaves()`, `tk_bisque()`, `tk_focusFollowsMouse()`, `tk_focusNext()`, `tk_focusPrev()`, `tk_menuBar()`, `tk_setPalette()`, `tk_strictMotif()`, `tkraise()`, `unbind()`, `unbind_all()`, `unbind_class()`, `update()`, `update_idletasks()`, `wait_variable()`, `wait_visibility()`, `wait_window()`, `waitvar()`, `wininfo_atom()`, `wininfo_atomname()`, `wininfo_cells()`, `wininfo_children()`, `wininfo_class()`, `wininfo_colormapfull()`, `wininfo_containing()`, `wininfo_depth()`, `wininfo_exists()`, `wininfo_fpixels()`, `wininfo_geometry()`, `wininfo_height()`, `wininfo_id()`, `wininfo_interps()`, `wininfo_ismapped()`, `wininfo_manager()`, `wininfo_name()`, `wininfo_parent()`, `wininfo_pathname()`, `wininfo_pixels()`, `wininfo_pointerx()`, `wininfo_pointery()`, `wininfo_reqheight()`, `wininfo_reqwidth()`, `wininfo_rgb()`, `wininfo_rootx()`, `wininfo_rooty()`, `wininfo_screen()`, `wininfo_screencells()`, `wininfo_screendepth()`, `wininfo_screenheight()`, `wininfo_screenmmheight()`, `wininfo_screenmmwidth()`, `wininfo_screenvisual()`, `wininfo_screenwidth()`, `wininfo_server()`, `wininfo_toplevel()`, `wininfo_viewable()`, `wininfo_visual()`, `wininfo_visualid()`, `wininfo_visualsavailable()`, `wininfo_vrootheight()`, `wininfo_vrootwidth()`, `wininfo_vrootx()`, `wininfo_vrooty()`, `wininfo_width()`, `wininfo_x()`, `wininfo_y()`

***Inherited from Tkinter.Pack***

`forget()`, `info()`, `pack()`, `pack_configure()`, `pack_forget()`, `pack_info()`

***Inherited from Tkinter.Place***

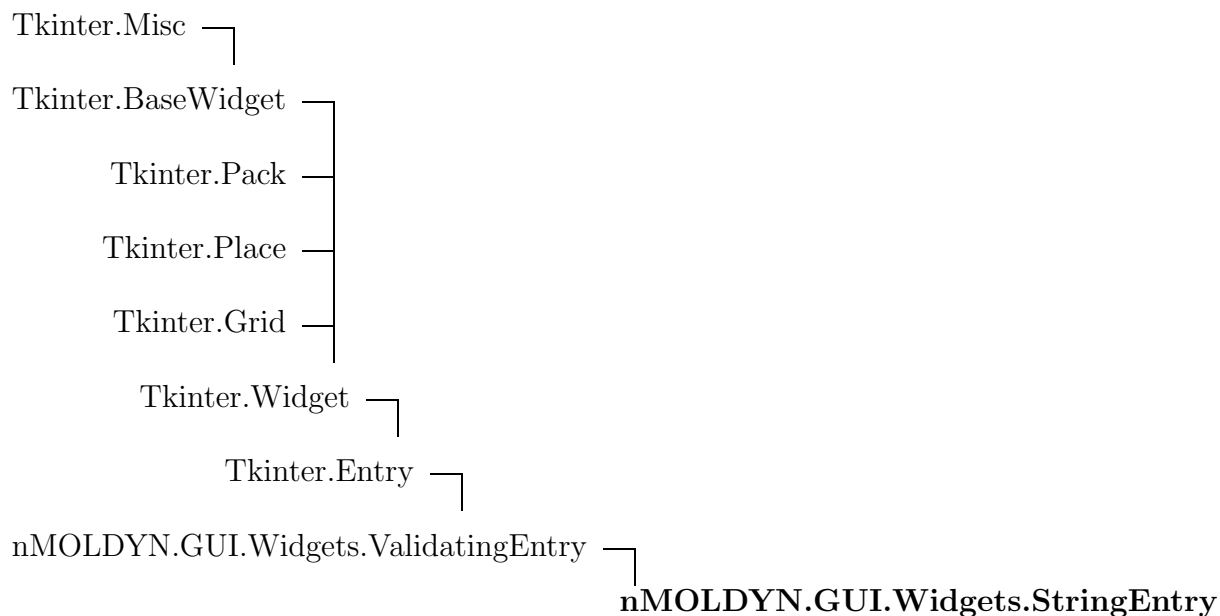
`place()`, `place_configure()`, `place_forget()`, `place_info()`

***Inherited from Tkinter.Grid***

`grid()`, `grid_configure()`, `grid_forget()`, `grid_info()`, `grid_remove()`, `location()`

**47.5.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

**47.6 Class `StringEntry`**

Sets up a Tkinter Entry widget that accepts only string values.

**Note:** subclass of `nMOLDYN.GUI.Widgets.ValidatingEntry` class.

**47.6.1 Methods**


---

**`__init__(self, master, contents='', pattern=None, **kwargs)`**


---

The constructor.

**Parameters**

**master:** the parent widget.

**contents:** the contents of the Tkinter Entry widget.  
(*type=string*)

**pattern:** if not None, a string specifying a pattern the contents of the Tkinter Entry widget has to fulfill.  
(*type=string*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

---

**`checkValue(self, event=None, contents=None)`**


---

Check that the contents of the Tkinter Entry widget is actually a string that possibly match the |self.pattern| pattern it is not None..

**Parameters**

**event:** the keyboard event that triggers the checking of the contents of the Tkinter Entry widget.  
(*type=a Tkinter.Event object*)

**contents:** the contents of the Tkinter Entry widget to check.  
(*type=string*)

Overrides: nMOLDYN.GUI.Widgets.ValidatingEntry.checkValue

---

**`getValue(self)`**


---

Returns the value of the control variable linked to the Tkinter Entry widget if it is a string otherwise throws an error.

Overrides: nMOLDYN.GUI.Widgets.ValidatingEntry.getValue

***Inherited from nMOLDYN.GUI.Widgets.ValidatingEntry(Section 47.3)***

setValue()

***Inherited from Tkinter.Entry***

delete(), get(), icursor(), index(), insert(), scan\_dragto(), scan\_mark(), select\_adjust(), select\_clear(), select\_from(), select\_present(), select\_range(), select\_to(), selection\_adjust(), selection\_clear(), selection\_from(), selection\_present(), selection\_range(), selection\_to(), xview(), xview\_moveto(), xview\_scroll()

***Inherited from Tkinter.BaseWidget***

`destroy()`

***Inherited from Tkinter.Misc***

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`, `option_readfile()`, `pack_propagate()`, `pack_slaves()`, `place_slaves()`, `propagate()`, `quit()`, `register()`, `rowconfigure()`, `selection_get()`, `selection_handle()`, `selection_own()`, `selection_own_get()`, `send()`, `setvar()`, `size()`, `slaves()`, `tk_bisque()`, `tk_focusFollowsMouse()`, `tk_focusNext()`, `tk_focusPrev()`, `tk_menuBar()`, `tk_setPalette()`, `tk_strictMotif()`, `tkraise()`, `unbind()`, `unbind_all()`, `unbind_class()`, `update()`, `update_idletasks()`, `wait_variable()`, `wait_visibility()`, `wait_window()`, `waitvar()`, `wininfo_atom()`, `wininfo_atomname()`, `wininfo_cells()`, `wininfo_children()`, `wininfo_class()`, `wininfo_colormapfull()`, `wininfo_containing()`, `wininfo_depth()`, `wininfo_exists()`, `wininfo_fpixels()`, `wininfo_geometry()`, `wininfo_height()`, `wininfo_id()`, `wininfo_interps()`, `wininfo_ismapped()`, `wininfo_manager()`, `wininfo_name()`, `wininfo_parent()`, `wininfo_pathname()`, `wininfo_pixels()`, `wininfo_pointerx()`, `wininfo_pointerxy()`, `wininfo_pointery()`, `wininfo_reqheight()`, `wininfo_reqwidth()`, `wininfo_rgb()`, `wininfo_rootx()`, `wininfo_rooty()`, `wininfo_screen()`, `wininfo_screencells()`, `wininfo_screendepth()`, `wininfo_screenheight()`, `wininfo_screenmmheight()`, `wininfo_screenmmwidth()`, `wininfo_screenvisual()`, `wininfo_screenwidth()`, `wininfo_server()`, `wininfo_toplevel()`, `wininfo_viewable()`, `wininfo_visual()`, `wininfo_visualid()`, `wininfo_visualsavailable()`, `wininfo_vrootheight()`, `wininfo_vrootwidth()`, `wininfo_vrootx()`, `wininfo_vrooty()`, `wininfo_width()`, `wininfo_x()`, `wininfo_y()`

***Inherited from Tkinter.Pack***

`forget()`, `info()`, `pack()`, `pack_configure()`, `pack_forget()`, `pack_info()`

***Inherited from Tkinter.Place***

`place()`, `place_configure()`, `place_forget()`, `place_info()`

***Inherited from Tkinter.Grid***

`grid()`, `grid_configure()`, `grid_forget()`, `grid_info()`, `grid_remove()`, `location()`

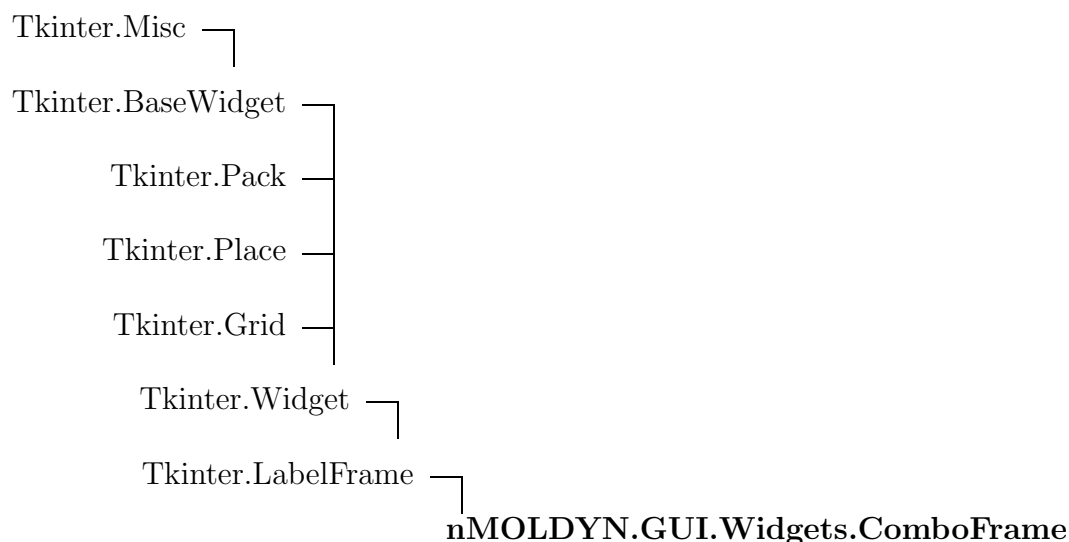
**47.6.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |

*continued on next page*

| Name                 | Description |
|----------------------|-------------|
| <code>_noarg-</code> |             |

## 47.7 Class ComboFrame



Sets up a Tkinter LabelFrame widget.

Some bindings are set by default:

- Tkinter '<Motion>' Event will change the mouse cursor to 'question\_arrow' symbol.
- Tkinter '<Leave>' Event will change the mouse cursor to 'arrow' symbol when leaving the widget.
- Tkinter '<Button-1>' Event will give some information about the widget when clicking on it.

### 47.7.1 Methods

|                                                                |
|----------------------------------------------------------------|
| <code>__init__(self, master, frameLabel='', tagName='')</code> |
|----------------------------------------------------------------|

The constructor.

Overrides: Tkinter.BaseWidget.\_\_init\_\_

*Inherited from Tkinter.BaseWidget*

`destroy()`

*Inherited from Tkinter.Misc*

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get\_boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

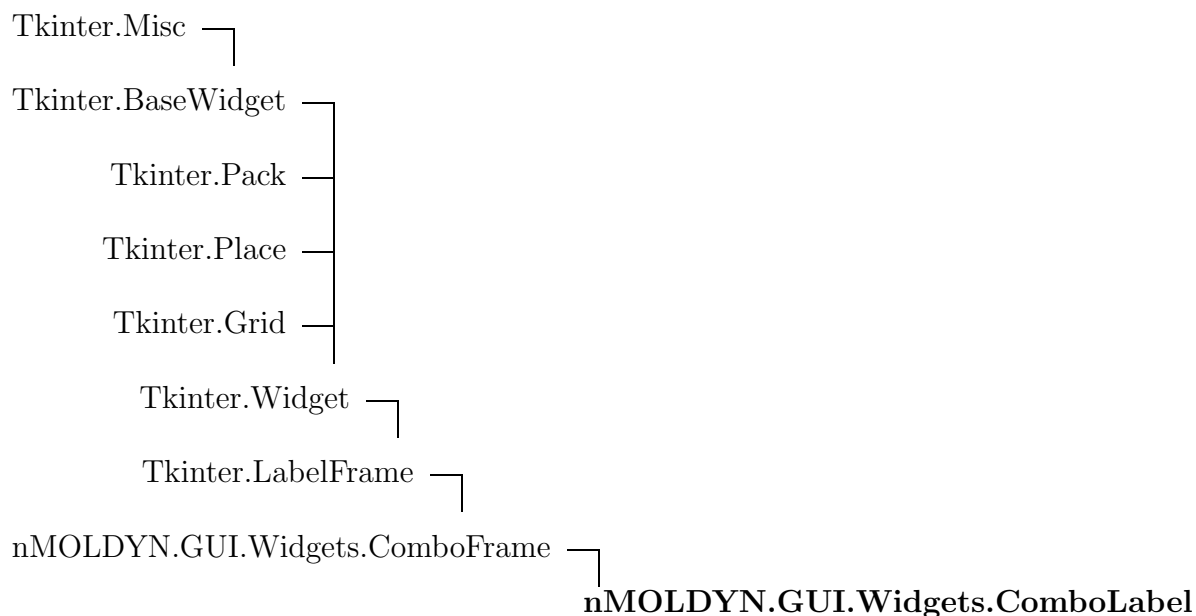
### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.7.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

## 47.8 Class ComboLabel



Sets up a combo widget made of a Tkinter Label widget embedded in a Tkinter LabelFrame widget.

### 47.8.1 Methods

```
__init__(self, master, frameLabel='', tagName='', contents='')
```

The constructor.

#### Parameters

**master:** the parent widget of the combo widget.

**frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)

**tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)

**contents:** the contents of the Tkinter Label widget.  
(*type=string*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

|                              |
|------------------------------|
| <b>getValue(<i>self</i>)</b> |
|------------------------------|

|                                                |
|------------------------------------------------|
| Returns the value of the Tkinter Label widget. |
|------------------------------------------------|

|                                            |
|--------------------------------------------|
| <b>setValue(<i>self</i>, <i>value</i>)</b> |
|--------------------------------------------|

|                                                |
|------------------------------------------------|
| Returns the value of the Tkinter Label widget. |
|------------------------------------------------|

### ***Inherited from Tkinter.BaseWidget***

destroy()

### ***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get\_boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), wininfo\_atom(), wininfo\_atomname(), wininfo\_cells(), wininfo\_children(), wininfo\_class(), wininfo\_colormapfull(), wininfo\_containing(), wininfo\_depth(), wininfo\_exists(), wininfo\_fpixels(), wininfo\_geometry(), wininfo\_height(), wininfo\_id(), wininfo\_interps(), wininfo\_ismapped(), wininfo\_manager(), wininfo\_name(), wininfo\_parent(), wininfo\_pathname(), wininfo\_pixels(), wininfo\_pointerx(), wininfo\_pointerxy(), wininfo\_pointery(), wininfo\_reqheight(), wininfo\_reqwidth(), wininfo\_rgb(), wininfo\_rootx(), wininfo\_rooty(), wininfo\_screen(), wininfo\_screencells(), wininfo\_screendepth(), wininfo\_screenheight(), wininfo\_screenmmheight(), wininfo\_screenmmwidth(), wininfo\_screenvisual(), wininfo\_screenwidth(), wininfo\_server(), wininfo\_toplevel(), wininfo\_viewable(), wininfo\_visual(), wininfo\_visualid(), wininfo\_visualsavailable(), wininfo\_vrootheight(), wininfo\_vrootwidth(), wininfo\_vrootx(), wininfo\_vrooty(), wininfo\_width(), wininfo\_x(), wininfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

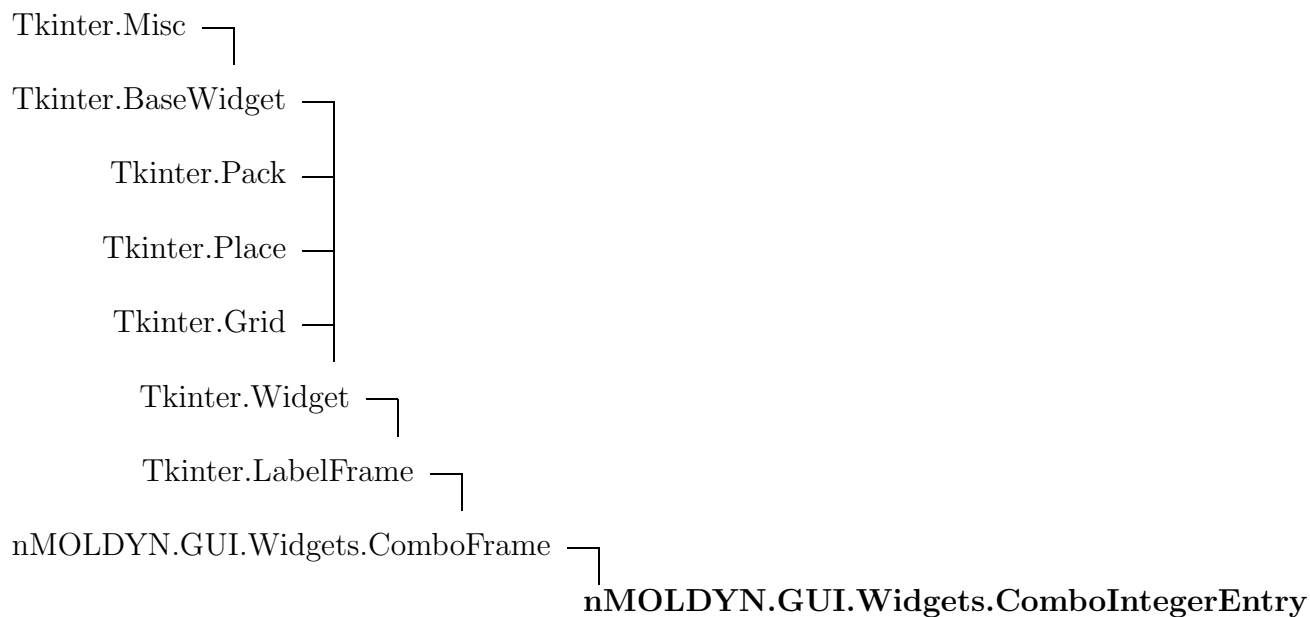
place(), place\_configure(), place\_forget(), place\_info()

***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

**47.8.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg-                            |             |

**47.9 Class ComboIntegerEntry**

Sets up a combo widget made of a Tkinter Label widget and a IntegerEntry widget embedded in a Tkinter LabelFrame widget.

**47.9.1 Methods**


---

**\_\_init\_\_**(*self*, *master*, *frameLabel*='', *tagName*='', *contents*='')

---

The constructor.

**Parameters**

**master:** the parent widget of the combo widget.

**frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)

**tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)

**contents:** the contents of the IntegerEntry widget.  
(*type=integer*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

---

**getValue**(*self*)

---

Returns the value of the IntegerEntry widget.

---

**setValue**(*self*, *value*)

---

Sets the value of the IntegerEntry widget to |value| (integer).

**Parameters**

**value:** the contents to insert in the IntegerEntry widget.  
(*type=integer*)

***Inherited from Tkinter.BaseWidget***

destroy()

***Inherited from Tkinter.Misc***

`_getitem_()`, `_setitem_()`, `_str_()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`, `option_readfile()`, `pack_propagate()`, `pack_slaves()`, `place_slaves()`, `propagate()`, `quit()`,

register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

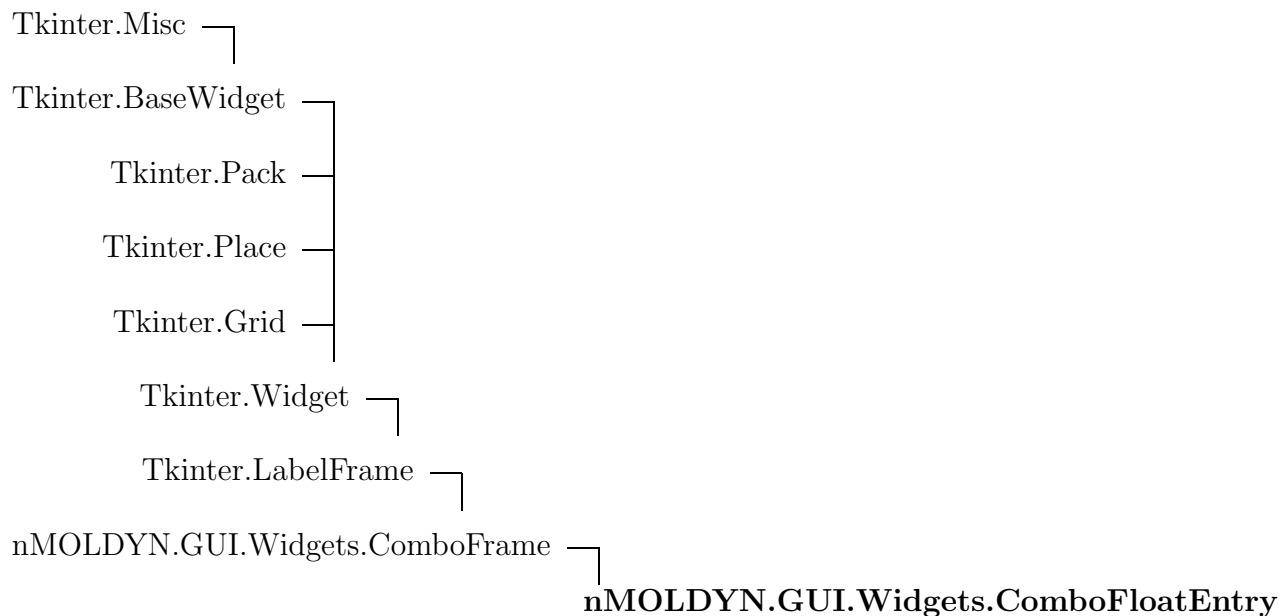
### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.9.2 Class Variables**

| Name                                          | Description |
|-----------------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i><br>_noarg- |             |

## 47.10 Class ComboFloatEntry



Sets up a combo widget made of a Tkinter Label widget and a FloatEntry widget embedded in a Tkinter LabelFrame widget.

### 47.10.1 Methods

**`__init__(self, master, frameLabel='', tagName='', contents='')`**

The constructor.

#### Parameters

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** the contents of the FloatEntry widget.  
(*type=float*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

|                                             |
|---------------------------------------------|
| <b>getValue(<i>self</i>)</b>                |
| Returns the value of the FloatEntry widget. |

|                                                                 |
|-----------------------------------------------------------------|
| <b>setValue(<i>self</i>, <i>value</i>)</b>                      |
| Sets the value of the FloatEntry widget to  value  (float).     |
| <b>Parameters</b>                                               |
| <i>value</i> : the contents to insert in the FloatEntry widget. |
| ( <i>type=float</i> )                                           |

***Inherited from Tkinter.BaseWidget***

destroy()

***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get\_boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

***Inherited from Tkinter.Place***

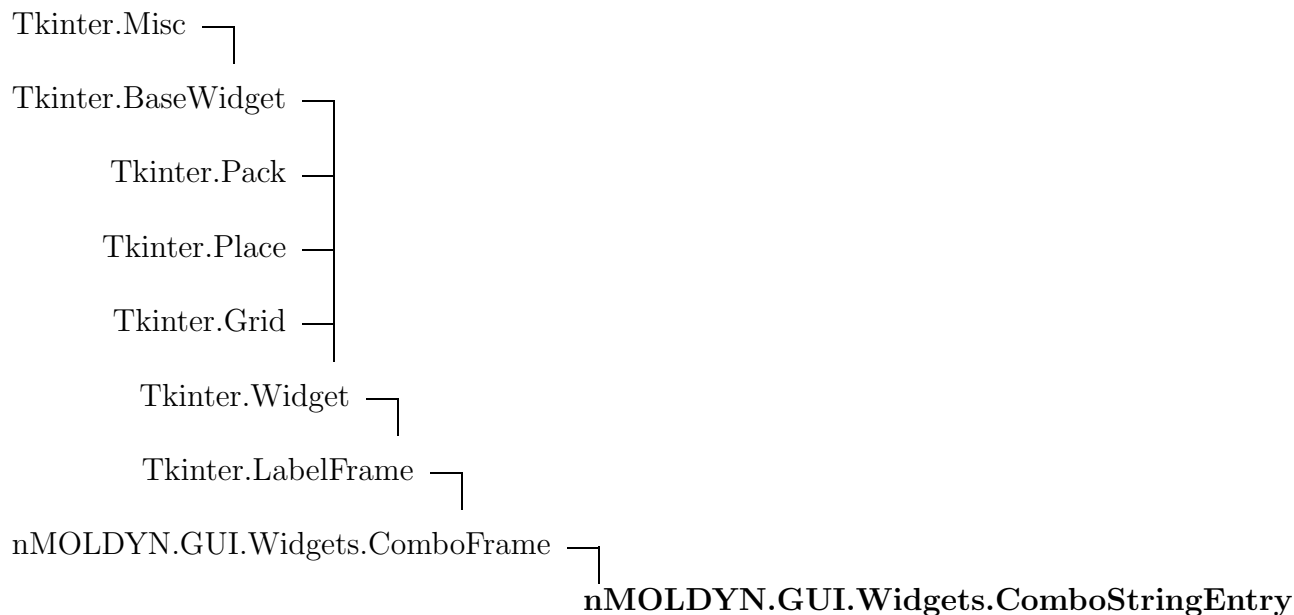
place(), place\_configure(), place\_forget(), place\_info()

***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

**47.10.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg_                            |             |

**47.11 Class ComboStringEntry**

Sets up a combo widget made of a Tkinter Label widget and a StringEntry widget embedded in a Tkinter LabelFrame widget.

## 47.11.1.1 Methods

---

**\_\_init\_\_**(*self*, *master*, *frameLabel*='', *tagName*='', *contents*='', *pattern*=None)

---

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** the contents of the StringEntry widget.  
(*type=string*)
- pattern:** if not None, a string specifying a pattern the contents of the Tkinter Entry widget has to fulfill.  
(*type=string*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

---



---

**getValue**(*self*)

---

Returns the value of the StringEntry widget.

---



---

**setValue**(*self*, *value*)

---

Sets the value of the StringEntry widget to |value| (string).

**Parameters**

- value:** the contents to insert in the StringEntry widget.  
(*type=string*)
- 

**Inherited from Tkinter.BaseWidget**

destroy()

**Inherited from Tkinter.Misc**

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get-`

boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

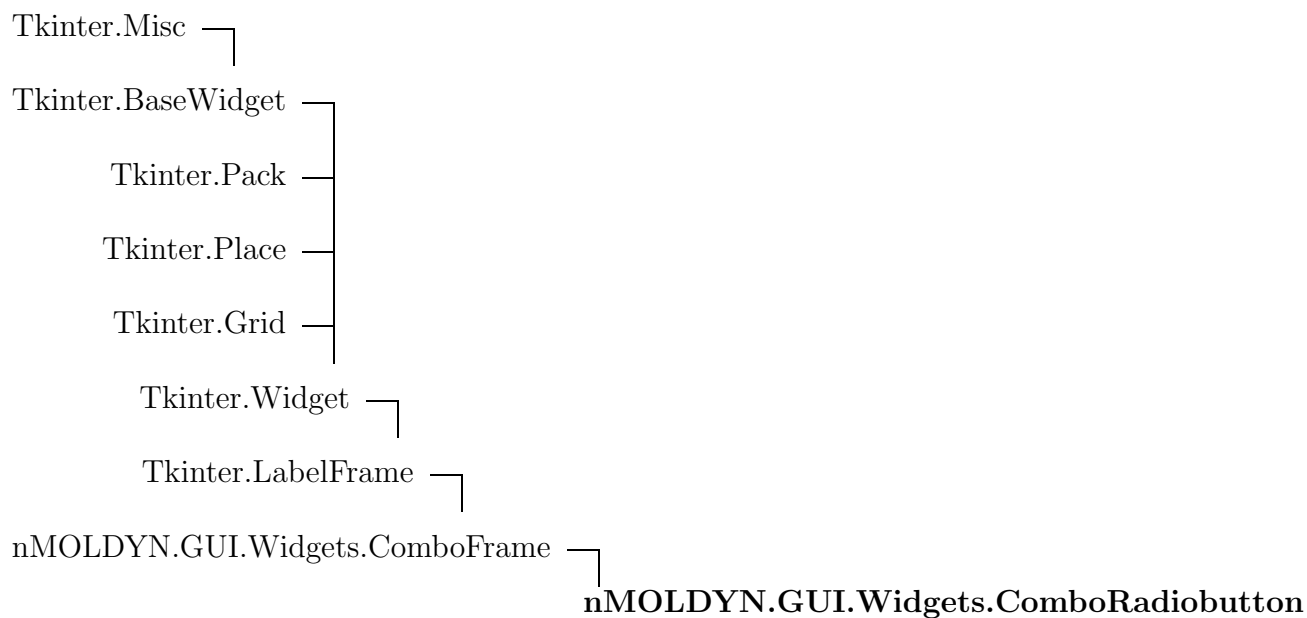
### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.11.2 Class Variables**

| Name                                          | Description |
|-----------------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i><br>_noarg- |             |

## 47.12 Class **ComboRadiobutton**



Sets up a combo widget made of a Tkinter Label widget and a set of Tkinter RadioButton widgets embedded in a Tkinter LabelFrame widget.

## 47.12.1 Methods

---

**\_\_init\_\_**(*self*, *master*, *frameLabel*='', *tagName*='', *contents*=[], *default*=0, *layout*=None)

---

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** a list (string) specifying the names of each Tkinter Radiobutton widget.  
(*type=list*)
- default:** an integer specifying which Tkinter Radiobutton widget will be switched on by default.  
(*type=integer*)
- layout:** a tuple (integer) of the form (nrows,ncolumns) specifying the way the set of Tkinter Radiobutton widget will be displayed.  
(*type=tuple*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

---



---

**getValue**(*self*)

---

Returns the value of the control variable linked to the set of Tkinter Radiobutton widgets.

---



---

**setValue**(*self*, *value*)

---

Sets the value of the control variable linked to the set of Tkinter Radiobutton widgets.

**Parameters**

- value:** the value the control variable linked to the set of Tkinter Radiobutton widgets will take. Must be an element of  
|self.contents|  
(*type=string*)
-

***Inherited from Tkinter.BaseWidget***

`destroy()`

***Inherited from Tkinter.Misc***

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`, `option_readfile()`, `pack_propagate()`, `pack_slaves()`, `place_slaves()`, `propagate()`, `quit()`, `register()`, `rowconfigure()`, `selection_clear()`, `selection_get()`, `selection_handle()`, `selection_own()`, `selection_own_get()`, `send()`, `setvar()`, `size()`, `slaves()`, `tk_bisque()`, `tk_focusFollowsMouse()`, `tk_focusNext()`, `tk_focusPrev()`, `tk_menuBar()`, `tk_setPalette()`, `tk_strictMotif()`, `tkraise()`, `unbind()`, `unbind_all()`, `unbind_class()`, `update()`, `update_idletasks()`, `wait_variable()`, `wait_visibility()`, `wait_window()`, `waitvar()`, `winfo_atom()`, `winfo_atomname()`, `winfo_cells()`, `winfo_children()`, `winfo_class()`, `winfo_colormapfull()`, `winfo_containing()`, `winfo_depth()`, `winfo_exists()`, `winfo_fpixels()`, `winfo_geometry()`, `winfo_height()`, `winfo_id()`, `winfo_interps()`, `winfo_ismapped()`, `winfo_manager()`, `winfo_name()`, `winfo_parent()`, `winfo_pathname()`, `winfo_pixels()`, `winfo_pointerx()`, `winfo_pointerxy()`, `winfo_pointery()`, `winfo_reqheight()`, `winfo_reqwidth()`, `winfo_rgb()`, `winfo_rootx()`, `winfo_rooty()`, `winfo_screen()`, `winfo_screencells()`, `winfo_screendepth()`, `winfo_screenheight()`, `winfo_screenmmheight()`, `winfo_screenmmwidth()`, `winfo_screenvisual()`, `winfo_screenwidth()`, `winfo_server()`, `winfo_toplevel()`, `winfo_viewable()`, `winfo_visual()`, `winfo_visualid()`, `winfo_visualsavailable()`, `winfo_vrootheight()`, `winfo_vrootwidth()`, `winfo_vrootx()`, `winfo_vrooty()`, `winfo_width()`, `winfo_x()`, `winfo_y()`

***Inherited from Tkinter.Pack***

`forget()`, `info()`, `pack()`, `pack_configure()`, `pack_forget()`, `pack_info()`

***Inherited from Tkinter.Place***

`place()`, `place_configure()`, `place_forget()`, `place_info()`

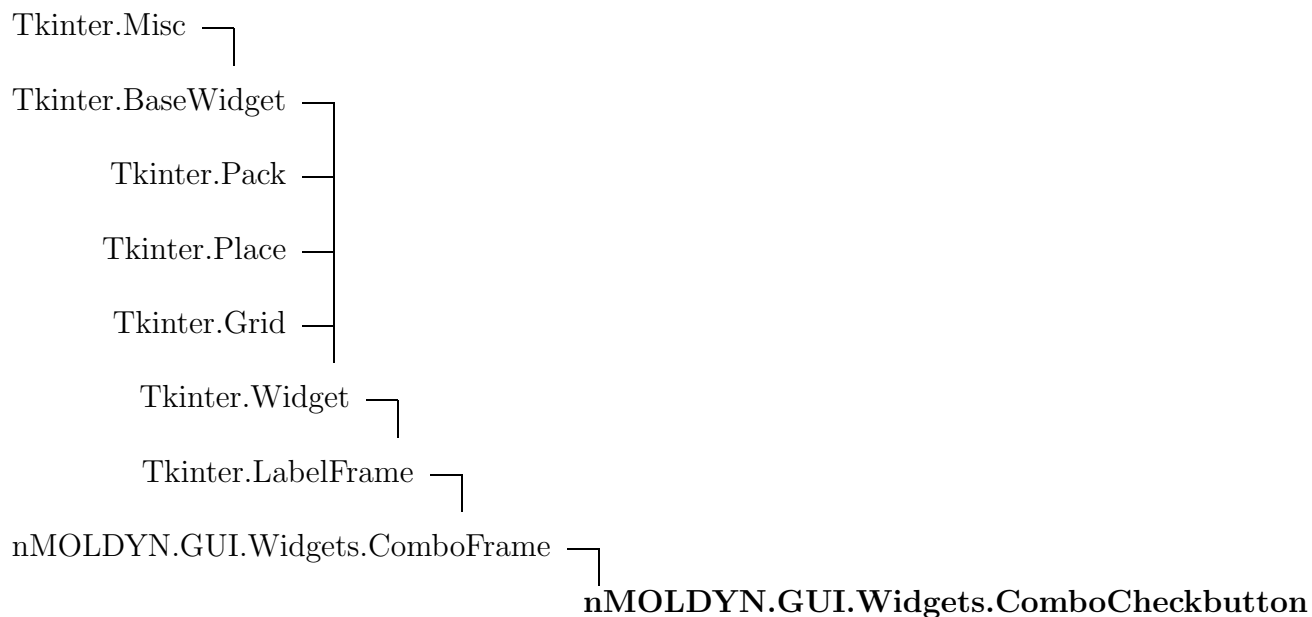
***Inherited from Tkinter.Grid***

`grid()`, `grid_configure()`, `grid_forget()`, `grid_info()`, `grid_remove()`, `location()`

**47.12.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

### 47.13 Class ComboCheckbutton



Sets up a combo widget made of a Tkinter Label widget and a set of Tkinter Checkbutton widget embedded in a Tkinter LabelFrame widget.

## 47.13.1 Methods

```
__init__(self, master, frameLabel='', tagName='', contents='',
onvalue=True, offvalue=False, default=False)
```

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** the name of the Tkinter Checkbutton widget.  
(*type=string*)
- onvalue:** the value of the Tkinter Checkbutton widget when switched on.  
(*type=boolean, integer or string*)
- offvalue:** the value of the Tkinter Checkbutton widget when switched off.  
(*type=boolean, integer or string*)
- default:** a boolean specifying whether the Tkinter Checkbutton widget is switched on (True) or switched off (False) by default.  
(*type=boolean*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

```
getValue(self)
```

Returns the value of the control variable linked to the Tkinter Checkbutton widget.

**setValue**(*self*, *value*)

Sets the value of the control variable linked to the Tkinter Checkbutton widget.

**Parameters**

**value:** the value the control variable linked to the Tkinter Checkbutton widget will take. Must be an element of `|self.contents|`  
(*type=string*)

**Inherited from Tkinter.BaseWidget**

`destroy()`

**Inherited from Tkinter.Misc**

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`, `focus()`, `focus_displayof()`, `focus_force()`, `focus_get()`, `focus_lastfor()`, `focus_set()`, `get_boolean()`, `getvar()`, `grab_current()`, `grab_release()`, `grab_set()`, `grab_set_global()`, `grab_status()`, `grid_bbox()`, `grid_columnconfigure()`, `grid_location()`, `grid_propagate()`, `grid_rowconfigure()`, `grid_size()`, `grid_slaves()`, `image_names()`, `image_types()`, `keys()`, `lift()`, `lower()`, `mainloop()`, `nametowidget()`, `option_add()`, `option_clear()`, `option_get()`, `option_readfile()`, `pack_propagate()`, `pack_slaves()`, `place_slaves()`, `propagate()`, `quit()`, `register()`, `rowconfigure()`, `selection_clear()`, `selection_get()`, `selection_handle()`, `selection_own()`, `selection_own_get()`, `send()`, `setvar()`, `size()`, `slaves()`, `tk_bisque()`, `tk_focusFollowsMouse()`, `tk_focusNext()`, `tk_focusPrev()`, `tk_menuBar()`, `tk_setPalette()`, `tk_strictMotif()`, `tkraise()`, `unbind()`, `unbind_all()`, `unbind_class()`, `update()`, `update_idletasks()`, `wait_variable()`, `wait_visibility()`, `wait_window()`, `waitvar()`, `winfo_atom()`, `winfo_atomname()`, `winfo_cells()`, `winfo_children()`, `winfo_class()`, `winfo_colormapfull()`, `winfo_containing()`, `winfo_depth()`, `winfo_exists()`, `winfo_fpixels()`, `winfo_geometry()`, `winfo_height()`, `winfo_id()`, `winfo_interps()`, `winfo_ismapped()`, `winfo_manager()`, `winfo_name()`, `winfo_parent()`, `winfo_pathname()`, `winfo_pixels()`, `winfo_pointerx()`, `winfo_pointerxy()`, `winfo_pointery()`, `winfo_reqheight()`, `winfo_reqwidth()`, `winfo_rgb()`, `winfo_rootx()`, `winfo_rooty()`, `winfo_screen()`, `winfo_screencells()`, `winfo_screendepth()`, `winfo_screenheight()`, `winfo_screenmmheight()`, `winfo_screenmmwidth()`, `winfo_screenvisual()`, `winfo_screenwidth()`, `winfo_server()`, `winfo_toplevel()`, `winfo_viewable()`, `winfo_visual()`, `winfo_visualid()`, `winfo_visualsavailable()`, `winfo_vrootheight()`, `winfo_vrootwidth()`, `winfo_vrootx()`, `winfo_vrooty()`, `winfo_width()`, `winfo_x()`, `winfo_y()`

**Inherited from Tkinter.Pack**

`forget()`, `info()`, `pack()`, `pack_configure()`, `pack_forget()`, `pack_info()`

***Inherited from Tkinter.Place***

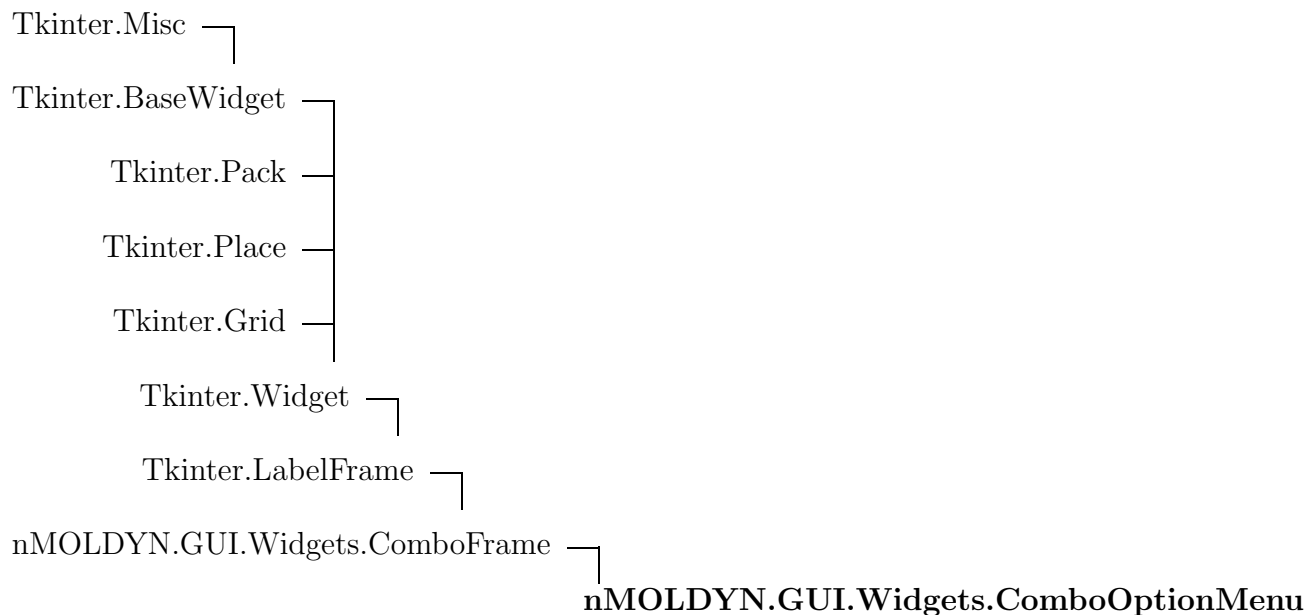
place(), place\_configure(), place\_forget(), place\_info()

***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

**47.13.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg_                            |             |

**47.14 Class ComboOptionsMenu**

Sets up a combo widget made of a Tkinter Label widget and a Tkinter OptionMenu widget embedded in a Tkinter LabelFrame widget.

## 47.14.1.1 Methods

---

**\_\_init\_\_**(*self*, *master*, *frameLabel*='', *tagName*='', *contents*=[], *default*=0)

---

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** a list (string) specifying the names of for the entries each Tkinter OptionMenu widget.  
(*type=list*)
- default:** an integer specifying which entry of the Tkinter optionMenu widget will be displayed by default.  
(*type=integer*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

---



---

**getValue**(*self*)

---

Returns the value of the control variable linked to the Tkinter OptionMenu widget.

---



---

**setValue**(*self*, *value*)

---

Sets the value of the control variable linked to the Tkinter OptionMenu widget.

**Parameters**

- value:** the value the control variable linked to the Tkinter OptionMenu widget will take. Must be an element of  
|self.contents|  
(*type=string*)
- 

**Inherited from Tkinter.BaseWidget**

destroy()

**Inherited from Tkinter.Misc**

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard.append(), clip-

board\_clear(), clipboard\_get(), colormodel(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.14.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg-                            |             |

### 47.15 Class **ComboSpinbox**



Sets up a combo widget made of a Tkinter Label widget and a Tkinter Spinbox widget embedded in a Tkinter LabelFrame widget.

## 47.15.1 Methods

---

**`__init__(self, master, frameLabel='', tagName='', contents=[], default=0)`**


---

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** a tuple (string) specifying the names of for the entries to insert in the Tkinter Spinbox widget.  
(*type=tuple*)
- default:** an integer specifying which entry of the Tkinter Spinbox widget will be displayed by default.  
(*type=integer*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

---



---

**`getValue(self)`**


---

Returns the value of the control variable linked to the Tkinter Spinbox widget.

---



---

**`setValue(self, value)`**


---

Sets the value of the control variable linked to the Tkinter Spinbox widget.

**Parameters**

- value:** the value the control variable linked to the Tkinter Spinbox widget will take. Must be an element of `|self.contents|`  
(*type=string*)
- 

***Inherited from Tkinter.BaseWidget***

`destroy()`

***Inherited from Tkinter.Misc***

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard_append()`, `clipboard_clear()`, `clipboard_get()`, `colormodel()`, `columnconfigure()`, `config()`, `configure()`, `deletecommand()`, `event_add()`, `event_delete()`, `event_generate()`, `event_info()`,

focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

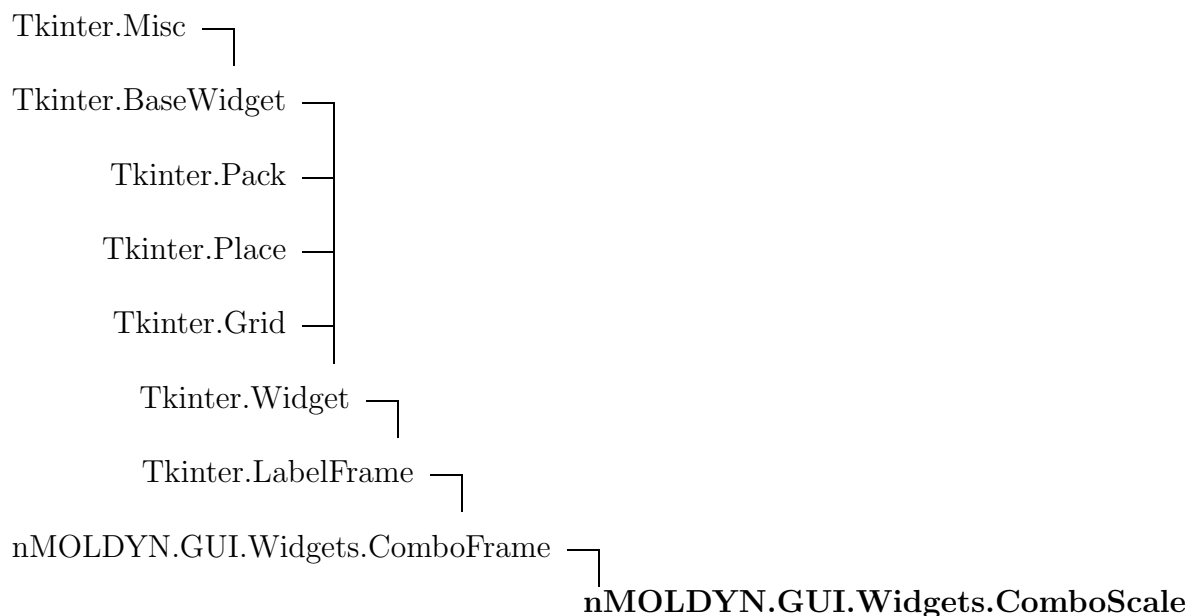
place(), place\_configure(), place\_forget(), place\_info()

### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.15.2 Class Variables**

| Name                                          | Description |
|-----------------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i><br>_noarg_ |             |

**47.16 Class ComboScale**

Sets up a combo widget made of a Tkinter Label widget and a Tkinter Scale widget embedded in a Tkinter LabelFrame widget.

**47.16.1 Methods**

**`__init__(self, master, frameLabel='', tagName='', contents=[])`**

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** a 3 elements list (integer) specifying respectively the first, the last and the resolution values for the Tkinter Scale widget.  
(*type=list*)

Overrides: `Tkinter.BaseWidget.__init__`

|                              |
|------------------------------|
| <b>getValue(<i>self</i>)</b> |
|------------------------------|

|                                                                               |
|-------------------------------------------------------------------------------|
| Returns the value of the control variable linked to the Tkinter Scale widget. |
|-------------------------------------------------------------------------------|

|                                            |
|--------------------------------------------|
| <b>setValue(<i>self</i>, <i>value</i>)</b> |
|--------------------------------------------|

|                                                                            |
|----------------------------------------------------------------------------|
| Sets the value of the control variable linked to the Tkinter Scale widget. |
|----------------------------------------------------------------------------|

|                   |
|-------------------|
| <b>Parameters</b> |
|-------------------|

|                                                             |
|-------------------------------------------------------------|
| <b>value:</b> the value the Tkinter Scale widget will take. |
|-------------------------------------------------------------|

|                         |
|-------------------------|
| ( <i>type=integer</i> ) |
|-------------------------|

***Inherited from Tkinter.BaseWidget***

destroy()

***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get\_boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

***Inherited from Tkinter.Place***

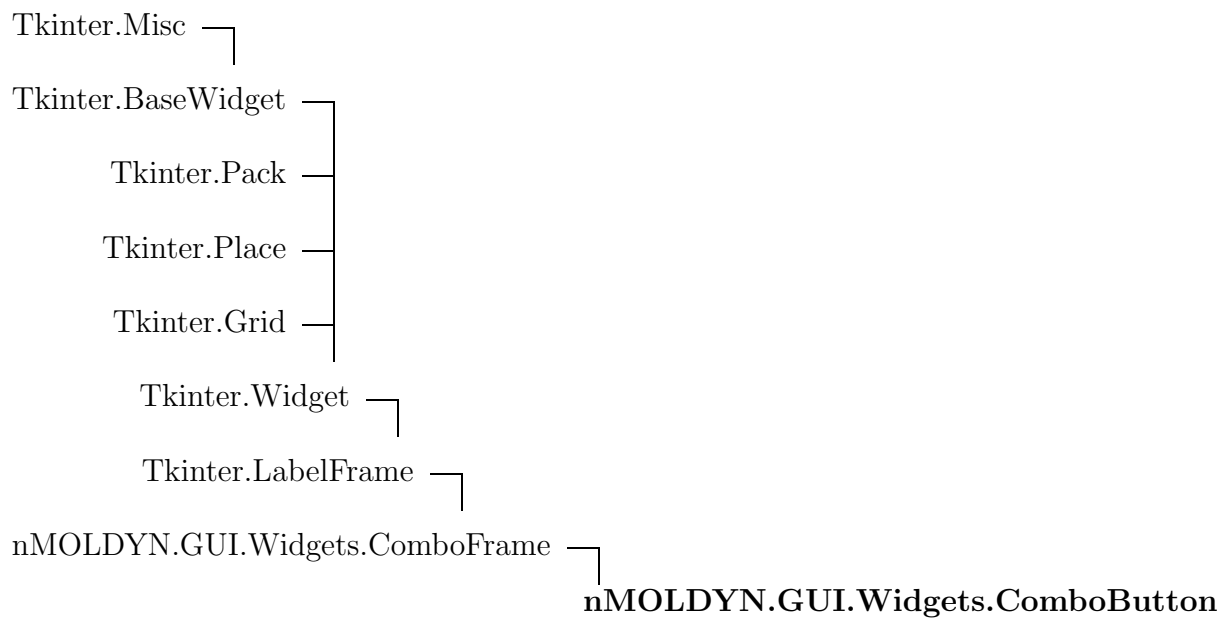
`place()`, `place_configure()`, `place_forget()`, `place_info()`

***Inherited from Tkinter.Grid***

`grid()`, `grid_configure()`, `grid_forget()`, `grid_info()`, `grid_remove()`, `location()`

**47.16.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg_</code>               |             |

**47.17 Class *ComboButton***

Sets up a combo widget made of a Tkinter Button widget and optionnally a Tkinter Entry widget embedded in a Tkinter LabelFrame widget.

**47.17.1 Methods**

```
__init__(self, master, frameLabel='', tagName='', contents='',
withEntry=None, command=None)
```

The constructor.

**Parameters**

**master:** the parent widget of the combo widget.

**frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)

**tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)

**contents:** string specifying the label of the Tkinter Button widget.  
(*type=string*)

**withEntry:** if not None, a Tkinter Entry widget will be placed beside the Tkinter Button with |withEntry| as contents.  
(*type=boolean*)

**command:** the function to call when clicking on the Tkinter Button widget.  
(*type=function*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

```
pushButton(self)
```

Executes the command linked to the Tkinter Button widget.

```
getValue(self)
```

Returns the output value of the |self.command| function.

```
setValue(self)
```

***Inherited from Tkinter.BaseWidget***

destroy()

***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(),

bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormodel(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.17.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

#### 47.18 Class **ComboFileBrowser**



Sets up a combo widget made of a Tkinter Label widget, a StringEntry widget and a Tkinter Button widget for file browsing embedded in a Tkinter LabelFrame widget.

## 47.18.1 Methods

```
__init__(self, master, frameLabel='', tagName='', contents='',
save=False, command=None, filetypes=[])
```

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** string specifying the contents of the Tkinter Entry widget.  
(*type=string*)
- save:** boolean specifying whether the file browser is for saving (True) or for loading (False).  
(*type=boolean*)
- command:** the function to call when browsing the file.  
(*type=function*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

```
browse(self)
```

Executes the command linked to the Tkinter Button widget.

```
getValue(self)
```

Returns the value of the StringEntry widget.

```
setValue(self, value)
```

Sets the value of the StringEntry widget to |value| (string).

**Parameters**

- value:** the contents to insert in the StringEntry widget.  
(*type=string*)

*Inherited from Tkinter.BaseWidget*

destroy()

***Inherited from Tkinter.Misc***

\_\_getitem\_\_(), \_\_setitem\_\_(), \_\_str\_\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get\_boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), wininfo\_atom(), wininfo\_atomname(), wininfo\_cells(), wininfo\_children(), wininfo\_class(), wininfo\_colormapfull(), wininfo\_containing(), wininfo\_depth(), wininfo\_exists(), wininfo\_fpixels(), wininfo\_geometry(), wininfo\_height(), wininfo\_id(), wininfo\_interps(), wininfo\_ismapped(), wininfo\_manager(), wininfo\_name(), wininfo\_parent(), wininfo\_pathname(), wininfo\_pixels(), wininfo\_pointerx(), wininfo\_pointerxy(), wininfo\_pointery(), wininfo\_reqheight(), wininfo\_reqwidth(), wininfo\_rgb(), wininfo\_rootx(), wininfo\_rooty(), wininfo\_screen(), wininfo\_screencells(), wininfo\_screendepth(), wininfo\_screenheight(), wininfo\_screenmmheight(), wininfo\_screenmmwidth(), wininfo\_screenvisual(), wininfo\_screenwidth(), wininfo\_server(), wininfo\_toplevel(), wininfo\_viewable(), wininfo\_visual(), wininfo\_visualid(), wininfo\_visualsavailable(), wininfo\_vrootheight(), wininfo\_vrootwidth(), wininfo\_vrootx(), wininfo\_vrooty(), wininfo\_width(), wininfo\_x(), wininfo\_y()

***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

***Inherited from Tkinter.Place***

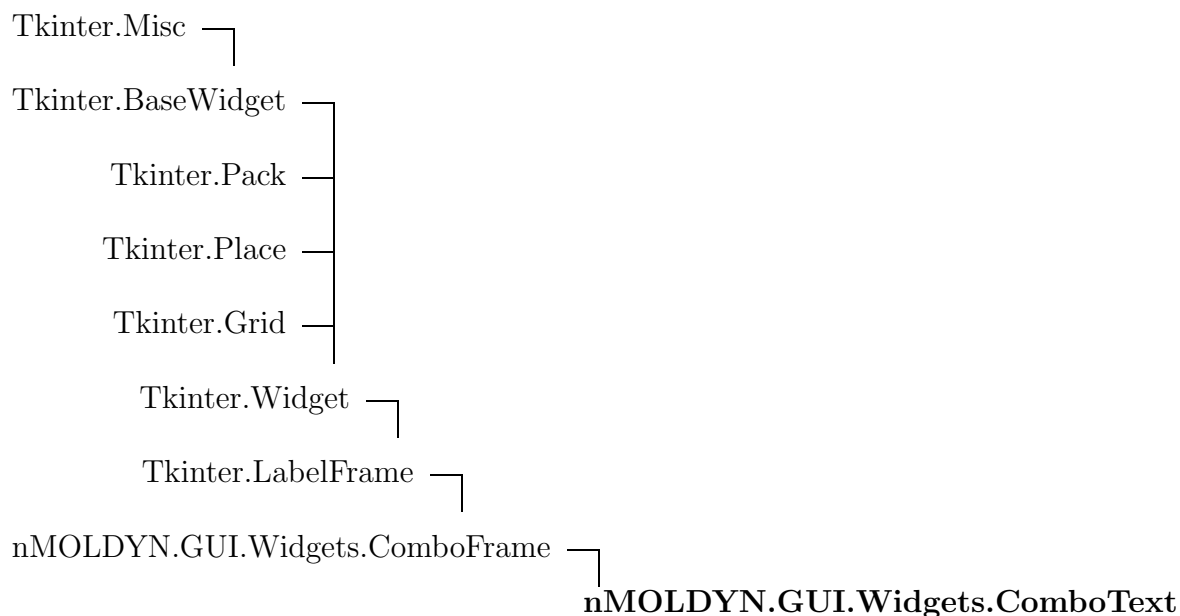
place(), place\_configure(), place\_forget(), place\_info()

***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

**47.18.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg-</code>               |             |

**47.19 Class ComboText**

Sets up a combo widget made of a Tkinter Text widget and a vertical Tkinter Scrollbar widget embedded in a Tkinter LabelFrame widget.

**47.19.1 Methods**

**`__init__(self, master=None, frameLabel='', tagName='', contents=None)`**

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** string specifying the contents to insert in the Tkinter Text widget.  
(*type=string*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

**cleanup**(*self*)

Deletes the contents of the Tkinter Text widget.

**insert**(*self*, *cursor*=END, *contents*=None, *tag*=None)

Inserts |contents| text in the Tkinter Text widget.

**Parameters**

**contents:** a string specifying the text to insert in the Tkinter Text widget.

(*type*=string)

**tag:** if not None, a string specifying the tag name to associate with the inserted text.

(*type*=string)

**tag**(*self*, *pattern*, *tag*)

Searches and tags all the occurrences of a given pattern.

**Parameters**

**pattern:** the pattern to search and tag.

(*type*=string)

**tag:** the tag to associate with |pattern|.

(*type*=string)

**Inherited from Tkinter.BaseWidget**

destroy()

**Inherited from Tkinter.Misc**

\_getitem\_(), \_setitem\_(), \_str\_(), after(), after\_cancel(), after\_idle(), bbox(), bell(), bind(), bind\_all(), bind\_class(), bindtags(), cget(), clipboard\_append(), clipboard\_clear(), clipboard\_get(), colormap(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get\_boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), up-

`date_idletasks()`, `wait_variable()`, `wait_visibility()`, `wait_window()`, `waitvar()`, `winfo_atom()`,  
`winfo_atomname()`, `winfo_cells()`, `winfo_children()`, `winfo_class()`, `winfo_colormapfull()`,  
`winfo_containing()`, `winfo_depth()`, `winfo_exists()`, `winfo_fpixels()`, `winfo_geometry()`,  
`winfo_height()`, `winfo_id()`, `winfo_interps()`, `winfo_ismapped()`, `winfo_manager()`, `winfo_name()`,  
`winfo_parent()`, `winfo_pathname()`, `winfo_pixels()`, `winfo_pointerx()`, `winfo_pointerxy()`,  
`winfo_pointery()`, `winfo_reqheight()`, `winfo_reqwidth()`, `winfo_rgb()`, `winfo_rootx()`,  
`winfo_rooty()`, `winfo_screen()`, `winfo_screencells()`, `winfo_screendepth()`, `winfo_screenheight()`,  
`winfo_screenmmheight()`, `winfo_screenmmwidth()`, `winfo_screenvisual()`, `winfo_screenwidth()`,  
`winfo_server()`, `winfo_toplevel()`, `winfo_viewable()`, `winfo_visual()`, `winfo_visualid()`,  
`winfo_visualsavailable()`, `winfo_vrootheight()`, `winfo_vrootwidth()`, `winfo_vrootx()`,  
`winfo_vrooty()`, `winfo_width()`, `winfo_x()`, `winfo_y()`

***Inherited from Tkinter.Pack***

`forget()`, `info()`, `pack()`, `pack_configure()`, `pack_forget()`, `pack_info()`

***Inherited from Tkinter.Place***

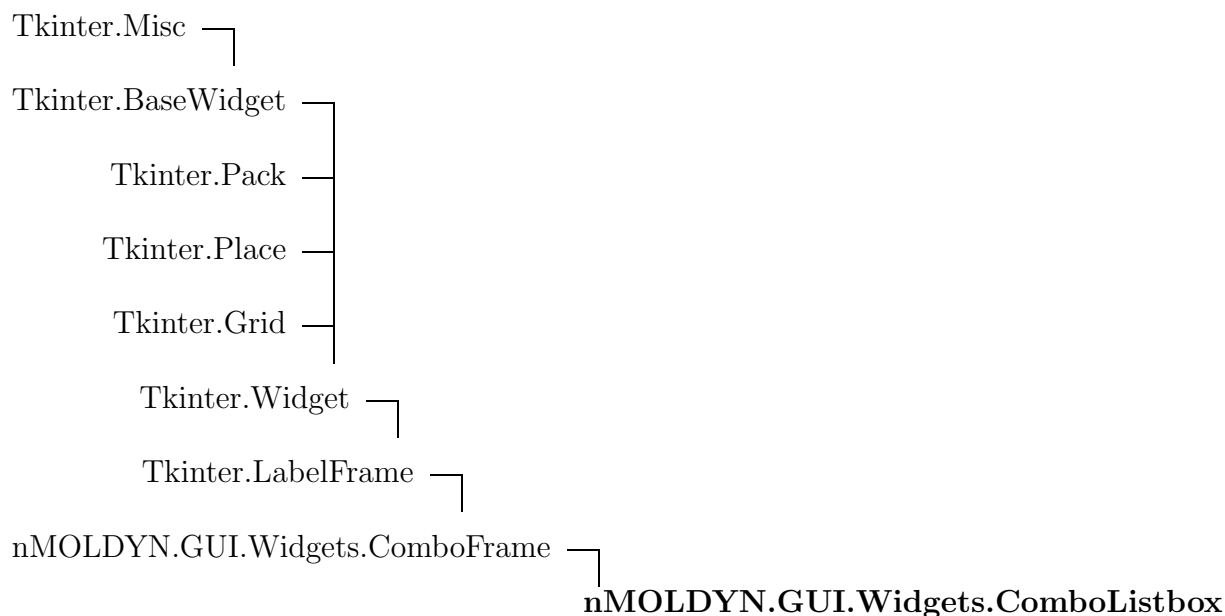
`place()`, `place_configure()`, `place_forget()`, `place_info()`

***Inherited from Tkinter.Grid***

`grid()`, `grid_configure()`, `grid_forget()`, `grid_info()`, `grid_remove()`, `location()`

#### 47.19.2 Class Variables

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| <code>_noarg_</code>               |             |

**47.20 Class ComboListbox**

Sets up a combo widget made of a Tkinter Listbox widget and a vertical Tkinter Scrollbar widget embedded in a Tkinter LabelFrame widget.

**47.20.1 Methods**

```
__init__(self, master=None, frameLabel='', tagName='', contents=[])
```

The constructor.

**Parameters**

- master:** the parent widget of the combo widget.
- frameLabel:** the label for the Tkinter LabelFrame widget.  
(*type=string or Tkinter.StringVar object.*)
- tagLabel:** the tag used for to document the widget. If set to "", the widget will not be documented.  
(*type=string.*)
- contents:** a list (string) specifying the items to insert in the Tkinter Listbox widget.  
(*type=list*)

Overrides: Tkinter.BaseWidget.\_\_init\_\_

**focus**(*self*, *event*)

Sets the focus to the Tkinter Listbox widget.

**Parameters**

**event**: the event triggering the focus.

(*type*=a *Tkinter.Event* object)

Overrides: *Tkinter.Misc.focus*

**insert**(*self*, *contents*=[])

Inserts a list of items (string) in the Tkinter Listbox widget.

**Parameters**

**contents**: a list (string) specifying the items to insert in the Tkinter Listbox widget.

(*type*=*list*)

**cleanup**(*self*)

Deletes all the items of the Tkinter Listbox widget.

**onListboxEntrySelection**(*self*, *event*)

Updates the selection of the Tkinter Listbox widget when |event| Tkinter Event occurs.

**Parameters**

**event**: the event triggering the callback.

(*type*=a *Tkinter.Event* object)

**Inherited from *Tkinter.BaseWidget***

*destroy*()

**Inherited from *Tkinter.Misc***

*\_\_getitem\_\_*(), *\_\_setitem\_\_*(), *\_\_str\_\_*(), *after*(), *after\_cancel*(), *after\_idle*(), *bbox*(), *bell*(), *bind*(), *bind\_all*(), *bind\_class*(), *bindtags*(), *cget*(), *clipboard\_append*(), *clipboard\_clear*(), *clipboard\_get*(), *colormodel*(), *columnconfigure*(), *config*(), *configure*(), *deletecommand*(), *event\_add*(), *event\_delete*(), *event\_generate*(), *event\_info*(), *focus\_displayof*(), *focus\_force*(), *focus\_get*(), *focus\_lastfor*(), *focus\_set*(), *getboolean*(), *getvar*(), *grab\_current*(), *grab\_release*(), *grab\_set*(), *grab\_set\_global*(), *grab\_status*(), *grid\_bbox*(), *grid\_columnconfigure*(), *grid\_location*(), *grid\_propagate*(), *grid\_rowconfigure*(), *grid\_size*(), *grid\_slaves*(), *image\_names*(), *image\_types*(), *keys*(), *lift*(), *lower*(), *mainloop*(), *nametowidget*(), *option\_add*(), *option\_clear*(), *option\_get*(), *option\_readfile*(), *pack\_propagate*(), *pack\_slaves*(), *place\_slaves*(), *propagate*(), *quit*(), *register*(), *row-*

`configure()`, `selection_clear()`, `selection_get()`, `selection_handle()`, `selection_own()`,  
`selection_own_get()`, `send()`, `setvar()`, `size()`, `slaves()`, `tk_bisque()`, `tk_focusFollowsMouse()`,  
`tk_focusNext()`, `tk_focusPrev()`, `tk_menuBar()`, `tk_setPalette()`, `tk_strictMotif()`, `tkraise()`,  
`unbind()`, `unbind_all()`, `unbind_class()`, `update()`, `update_idletasks()`, `wait_variable()`,  
`wait_visibility()`, `wait_window()`, `waitvar()`, `winfo_atom()`, `winfo_atomname()`, `winfo_cells()`,  
`winfo_children()`, `winfo_class()`, `winfo_colormapfull()`, `winfo_containing()`, `winfo_depth()`,  
`winfo_exists()`, `winfo_fpixels()`, `winfo_geometry()`, `winfo_height()`, `winfo_id()`, `winfo_interps()`,  
`winfo_ismapped()`, `winfo_manager()`, `winfo_name()`, `winfo_parent()`, `winfo_pathname()`,  
`winfo_pixels()`, `winfo_pointerx()`, `winfo_pointery()`, `winfo_reqheight()`,  
`winfo_reqwidth()`, `winfo_rgb()`, `winfo_rootx()`, `winfo_rooty()`, `winfo_screen()`, `winfo_screencells()`,  
`winfo_screendepth()`, `winfo_screenheight()`, `winfo_screenmmheight()`, `winfo_screenmmwidth()`,  
`winfo_screenvisual()`, `winfo_screenwidth()`, `winfo_server()`, `winfo_toplevel()`, `winfo_viewable()`,  
`winfo_visual()`, `winfo_visualid()`, `winfo_visualsavailable()`, `winfo_vrootheight()`, `winfo_vrootwidth()`,  
`winfo_vrootx()`, `winfo_vrooty()`, `winfo_width()`, `winfo_x()`, `winfo_y()`

### ***Inherited from Tkinter.Pack***

`forget()`, `info()`, `pack()`, `pack_configure()`, `pack_forget()`, `pack_info()`

### ***Inherited from Tkinter.Place***

`place()`, `place_configure()`, `place_forget()`, `place_info()`

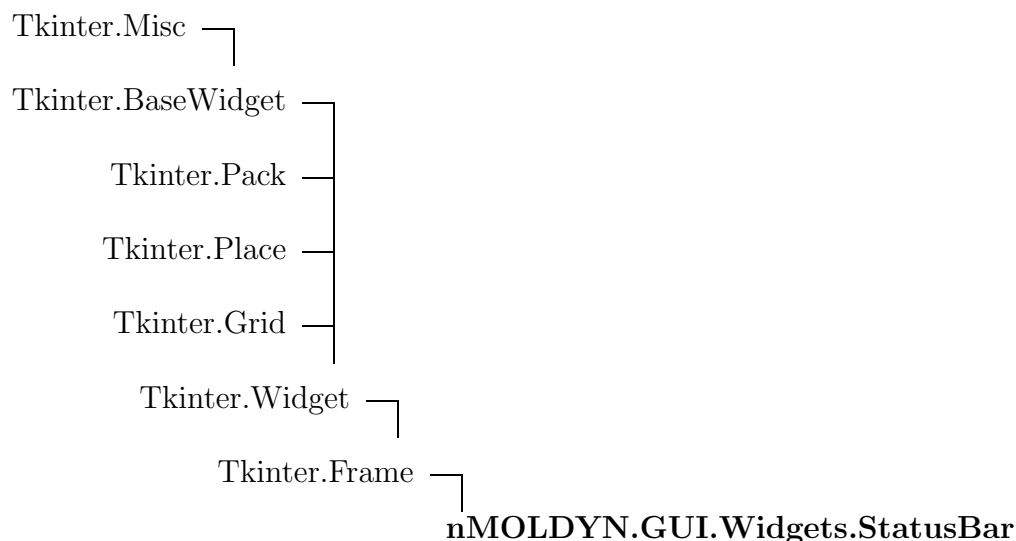
### ***Inherited from Tkinter.Grid***

`grid()`, `grid_configure()`, `grid_forget()`, `grid_info()`, `grid_remove()`, `location()`

#### **47.20.2 Class Variables**

| Name                                                       | Description |
|------------------------------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i><br><code>_noarg-</code> |             |

## 47.21 Class StatusBar



Sets up a combo widget made of a Tkinter Scale widget embedded in a Tkinter Frame widget.

### 47.21.1 Methods

**`__init__(self, master=None)`**

The constructor.

**Parameters**

**master**: the parent widget of the combo widget.

Overrides: Tkinter.BaseWidget.\_\_init\_\_

**`setValue(self, text)`**

Sets the text that will be displayed within the status bar.

**Parameters**

**text**: the value to display in the Status Bar.

(*type=string or float*)

**Inherited from Tkinter.BaseWidget**

`destroy()`

**Inherited from Tkinter.Misc**

`__getitem__()`, `__setitem__()`, `__str__()`, `after()`, `after_cancel()`, `after_idle()`, `bbox()`, `bell()`, `bind()`, `bind_all()`, `bind_class()`, `bindtags()`, `cget()`, `clipboard.append()`, `clip-`

board\_clear(), clipboard\_get(), colormodel(), columnconfigure(), config(), configure(), deletecommand(), event\_add(), event\_delete(), event\_generate(), event\_info(), focus(), focus\_displayof(), focus\_force(), focus\_get(), focus\_lastfor(), focus\_set(), get-boolean(), getvar(), grab\_current(), grab\_release(), grab\_set(), grab\_set\_global(), grab\_status(), grid\_bbox(), grid\_columnconfigure(), grid\_location(), grid\_propagate(), grid\_rowconfigure(), grid\_size(), grid\_slaves(), image\_names(), image\_types(), keys(), lift(), lower(), mainloop(), nametowidget(), option\_add(), option\_clear(), option\_get(), option\_readfile(), pack\_propagate(), pack\_slaves(), place\_slaves(), propagate(), quit(), register(), rowconfigure(), selection\_clear(), selection\_get(), selection\_handle(), selection\_own(), selection\_own\_get(), send(), setvar(), size(), slaves(), tk\_bisque(), tk\_focusFollowsMouse(), tk\_focusNext(), tk\_focusPrev(), tk\_menuBar(), tk\_setPalette(), tk\_strictMotif(), tkraise(), unbind(), unbind\_all(), unbind\_class(), update(), update\_idletasks(), wait\_variable(), wait\_visibility(), wait\_window(), waitvar(), winfo\_atom(), winfo\_atomname(), winfo\_cells(), winfo\_children(), winfo\_class(), winfo\_colormapfull(), winfo\_containing(), winfo\_depth(), winfo\_exists(), winfo\_fpixels(), winfo\_geometry(), winfo\_height(), winfo\_id(), winfo\_interps(), winfo\_ismapped(), winfo\_manager(), winfo\_name(), winfo\_parent(), winfo\_pathname(), winfo\_pixels(), winfo\_pointerx(), winfo\_pointerxy(), winfo\_pointery(), winfo\_reqheight(), winfo\_reqwidth(), winfo\_rgb(), winfo\_rootx(), winfo\_rooty(), winfo\_screen(), winfo\_screencells(), winfo\_screendepth(), winfo\_screenheight(), winfo\_screenmmheight(), winfo\_screenmmwidth(), winfo\_screenvisual(), winfo\_screenwidth(), winfo\_server(), winfo\_toplevel(), winfo\_viewable(), winfo\_visual(), winfo\_visualid(), winfo\_visualsavailable(), winfo\_vrootheight(), winfo\_vrootwidth(), winfo\_vrootx(), winfo\_vrooty(), winfo\_width(), winfo\_x(), winfo\_y()

### ***Inherited from Tkinter.Pack***

forget(), info(), pack(), pack\_configure(), pack\_forget(), pack\_info()

### ***Inherited from Tkinter.Place***

place(), place\_configure(), place\_forget(), place\_info()

### ***Inherited from Tkinter.Grid***

grid(), grid\_configure(), grid\_forget(), grid\_info(), grid\_remove(), location()

#### **47.21.2 Class Variables**

| Name                               | Description |
|------------------------------------|-------------|
| <i>Inherited from Tkinter.Misc</i> |             |
| _noarg-                            |             |

## 48 Package nMOLDYN.Tests

### 48.1 Modules

- **ARA** (*Section 49, p. 307*)
  - **TestsContents** (*Section 50, p. 308*)
  - **runTests** (*Section 51, p. 309*)
- **AVACF** (*Section 52, p. 310*)
  - **TestsContents** (*Section 53, p. 311*)
  - **runTests** (*Section 54, p. 312*)
- **AnalysisTests**: Test cases for analysis modules.  
(*Section 55, p. 313*)
- **BuildTestCases** (*Section 56, p. 314*)
- **DCSF** (*Section 57, p. 315*)
  - **TestsContents** (*Section 58, p. 316*)
  - **runTests** (*Section 59, p. 317*)
- **DISF** (*Section 60, p. 318*)
  - **TestsContents** (*Section 61, p. 319*)
  - **runTests** (*Section 62, p. 320*)
- **DISFG** (*Section 63, p. 321*)
  - **TestsContents** (*Section 64, p. 322*)
  - **runTests** (*Section 65, p. 323*)
- **DOS** (*Section 66, p. 324*)
  - **TestsContents** (*Section 67, p. 325*)
  - **runTests** (*Section 68, p. 326*)
- **EISF** (*Section 69, p. 327*)
  - **TestsContents** (*Section 70, p. 328*)
  - **runTests** (*Section 71, p. 329*)
- **MSD** (*Section 72, p. 330*)
  - **TestsContents** (*Section 73, p. 331*)
  - **runTests** (*Section 74, p. 332*)
- **StabilityTests**: Test cases for stability of the current version of nMOLDYN versus nMoldyn v2.1.0, the last stable release of nMoldyn.  
(*Section 75, p. 333*)
- **VACF** (*Section 76, p. 344*)
  - **TestsContents** (*Section 77, p. 345*)
  - **runTests** (*Section 78, p. 346*)

## 49 Package **nMOLDYN.Tests.ARA**

### 49.1 Modules

- **TestsContents** (*Section 50, p. 308*)
- **runTests** (*Section 51, p. 309*)

## 50 Module nMOLDYN.Tests.ARA.TestsContents

### 50.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 51 Module `nMOLDYN.Tests.ARA.runTests`

### 51.1 Variables

| Name                       | Description                                                                  |
|----------------------------|------------------------------------------------------------------------------|
| <code>pMoldyn</code>       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo.' |
| <code>selectedTests</code> | <b>Value:</b> <code>range(1, 15)</code>                                      |
| <code>a</code>             | <b>Value:</b> 'ARA'                                                          |

## 52 Package nMOLDYN.Tests.AVACF

### 52.1 Modules

- **TestsContents** (*Section 53, p. 311*)
- **runTests** (*Section 54, p. 312*)

## 53 Module *nMOLDYN.Tests.AVACF.TestsContents*

### 53.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 54 Module *nMOLDYN.Tests.AVACF.runTests*

### 54.1 Variables

| Name          | Description                                                                 |
|---------------|-----------------------------------------------------------------------------|
| pMoldyn       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo. |
| selectedTests | <b>Value:</b> range(1, 15)                                                  |
| a             | <b>Value:</b> 'AVACF'                                                       |

## 55 Module **nMOLDYN.Tests.AnalysisTests**

Test cases for analysis modules.

- `testCorrelation1`: test for the autocorrelation of a single time serie.
- `testCorrelation2`: test for the autocorrelation of a two dimensional time serie.
- `testCorrelation3`: another test for the autocorrelation of a two dimensional time serie.

### 55.1 Class **AnalysisTest**

```
unittest.TestCase └─ nMOLDYN.Tests.AnalysisTests.AnalysisTest
```

Tests Analysis module fonctionnalités.

#### 55.1.1 Methods

**setUp**(*self*)

Hook method for setting up the test fixture before exercising it.

Overrides: `unittest.TestCase.setUp` `exitit`(inherited documentation)

**testCorrelation1**(*self*)

**testCorrelation2**(*self*)

**testCorrelation3**(*self*)

*Inherited from **unittest.TestCase***

`__call__()`, `__init__()`, `__repr__()`, `__str__()`, `assertAlmostEqual()`, `assertAlmostEquals()`, `assertEqual()`, `assertEquals()`, `assertFalse()`, `assertNotAlmostEqual()`, `assertNotAlmostEquals()`, `assertNotEqual()`, `assertNotEquals()`, `assertRaises()`, `assertTrue()`, `assert_()`, `countTestCases()`, `debug()`, `defaultTestResult()`, `fail()`, `failIf()`, `failIfAlmostEqual()`, `failIfEqual()`, `failUnless()`, `failUnlessAlmostEqual()`, `failUnlessEqual()`, `failUnlessRaises()`, `id()`, `run()`, `shortDescription()`, `tearDown()`

## 56 Module `nMOLDYN.Tests.BuildTestCases`

### 56.1 Variables

| Name                              | Description                                                                          |
|-----------------------------------|--------------------------------------------------------------------------------------|
| <code>nmoldyn_package_path</code> | <b>Value:</b><br><code>'/home/cs/pellegrini/nMOLDYN/development'</code>              |
| <code>analysis</code>             | <b>Value:</b> <code>sys.argv [1]</code>                                              |
| <code>testContents</code>         | <b>Value:</b> <code>open(os.path.join(analysis,<br/>'TestsContents.py'), 'r')</code> |

## 57 Package nMOLDYN.Tests.DCSF

### 57.1 Modules

- **TestsContents** (*Section 58, p. 316*)
- **runTests** (*Section 59, p. 317*)

## 58 Module `nMOLDYN.Tests.DCSF.TestsContents`

### 58.1 Variables

| Name     | Description                                       |
|----------|---------------------------------------------------|
| template | <b>Value:</b> <code>{'REF': {}, 'NEW': {}}</code> |
| test     | <b>Value:</b> <code>[]</code>                     |

## 59 Module `nMOLDYN.Tests.DCSF.runTests`

### 59.1 Variables

| Name          | Description                                                                 |
|---------------|-----------------------------------------------------------------------------|
| pMoldyn       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo. |
| selectedTests | <b>Value:</b> range(1, 13)                                                  |

## 60 Package nMOLDYN.Tests.DISF

### 60.1 Modules

- **TestsContents** (*Section 61, p. 319*)
- **runTests** (*Section 62, p. 320*)

## 61 Module *nMOLDYN.Tests.DISF.TestsContents*

### 61.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 62 Module *nMOLDYN.Tests.DISF.runTests*

### 62.1 Variables

| Name          | Description                                                                 |
|---------------|-----------------------------------------------------------------------------|
| pMoldyn       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo. |
| selectedTests | <b>Value:</b> range(1, 13)                                                  |

## 63 Package `nMOLDYN.Tests.DISFG`

### 63.1 Modules

- `TestsContents` (*Section 64, p. 322*)
- `runTests` (*Section 65, p. 323*)

## 64 Module *nMOLDYN.Tests.DISFG.TestsContents*

### 64.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 65 Module *nMOLDYN.Tests.DISFG.runTests*

### 65.1 Variables

| Name          | Description                                                                  |
|---------------|------------------------------------------------------------------------------|
| a             | <b>Value:</b> 'DISFG'                                                        |
| pMoldyn       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo.' |
| selectedTests | <b>Value:</b> range(1, 13)                                                   |

## 66 Package nMOLDYN.Tests.DOS

### 66.1 Modules

- **TestsContents** (*Section 67, p. 325*)
- **runTests** (*Section 68, p. 326*)

## 67 Module *nMOLDYN.Tests.DOS.TestsContents*

### 67.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 68 Module `nMOLDYN.Tests.DOS.runTests`

### 68.1 Variables

| Name          | Description                                                                 |
|---------------|-----------------------------------------------------------------------------|
| pMoldyn       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo. |
| selectedTests | <b>Value:</b> range(1, 15)                                                  |

## 69 Package nMOLDYN.Tests.EISF

### 69.1 Modules

- **TestsContents** (*Section 70, p. 328*)
- **runTests** (*Section 71, p. 329*)

## 70 Module nMOLDYN.Tests.EISF.TestsContents

### 70.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 71 Module `nMOLDYN.Tests.EISF.runTests`

### 71.1 Variables

| Name                       | Description                                                                  |
|----------------------------|------------------------------------------------------------------------------|
| <code>pMoldyn</code>       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo.' |
| <code>selectedTests</code> | <b>Value:</b> <code>range(1, 13)</code>                                      |

## 72 Package `nMOLDYN.Tests.MSD`

### 72.1 Modules

- `TestsContents` (*Section 73, p. 331*)
- `runTests` (*Section 74, p. 332*)

## 73 Module nMOLDYN.Tests.MSD.TestsContents

### 73.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 74 Module `nMOLDYN.Tests.MSD.runTests`

### 74.1 Variables

| Name                       | Description                                                                  |
|----------------------------|------------------------------------------------------------------------------|
| <code>pMoldyn</code>       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo.' |
| <code>selectedTests</code> | <b>Value:</b> <code>range(1, 15)</code>                                      |

## 75 Module *nMOLDYN.Tests.StabilityTests*

Test cases for stability of the current version of *nMOLDYN* versus *nMoldyn* v2.1.0, the latest version of *nMoldyn*.

Each test case is made of a benchmark of  $N$  small tests corresponding to various input variables. For all of these small tests, the output of the tests with the reference version of *nMOLDYN* is stored into files with the name:

- \* `$PREFIX$TESTID_Reference.nc` where `$PREFIX` is the prefix for the analysis to test and `$TESTID` is the test ID. When running a stability test, the output of the current version are created on the same files.
- \* `$PREFIX$TESTID_Current.nc` where `$PREFIX` is the prefix for the analysis to test and `$TESTID` is the test ID. and each of the numeric variables of the reference and current NetCDF output file are compared with a tolerance set to  $1E-06$ .

### 75.1 Variables

| Name                            | Description                                                                              |
|---------------------------------|------------------------------------------------------------------------------------------|
| <code>nmoldyn_tests_path</code> | <b>Value:</b><br><code>os.path.abspath(os.path.split(__file__)[0])</code>                |
| <code>availableTests</code>     | <b>Value:</b> <code>['ARA', 'AVACF', 'DCSF', 'DISF', 'DISFG', 'EISF', 'MSD', ...]</code> |

### 75.2 Class *AnalysisTest*

`unittest.TestCase`  `nMOLDYN.Tests.StabilityTests.AnalysisTest`

#### 75.2.1 Methods

|                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b><code>setUp(self)</code></b>                                                                                                            |
| Overrides the <code>TestCase.setUp</code> method. Initialization of the test variables.<br>Overrides: <code>unittest.TestCase.setUp</code> |
| <b><code>runTest(self)</code></b>                                                                                                          |

|                                 |
|---------------------------------|
| <b>tearDown</b> ( <i>self</i> ) |
|---------------------------------|

|                                                                   |
|-------------------------------------------------------------------|
| Overrides the TestCase.tearDown method. Finalization of the test. |
|-------------------------------------------------------------------|

|                                       |
|---------------------------------------|
| Overrides: unittest.TestCase.tearDown |
|---------------------------------------|

**Inherited from *unittest.TestCase***

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

**75.3 Class *MSDTests***

```
unittest.TestCase └─
```

```
nMOLDYN.Tests.StabilityTests.AnalysisTest └─
 nMOLDYN.Tests.StabilityTests.MSDTests
```

**75.3.1 Methods**

|                                |
|--------------------------------|
| <b>runTest</b> ( <i>self</i> ) |
|--------------------------------|

|                                                              |
|--------------------------------------------------------------|
| Overrides: nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest |
|--------------------------------------------------------------|

**Inherited from *nMOLDYN.Tests.StabilityTests.AnalysisTest* (Section 75.2)**

setUp(), tearDown()

**Inherited from *unittest.TestCase***

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

**75.3.2 Class Variables**

| Name      | Description         |
|-----------|---------------------|
| shortName | <b>Value:</b> 'MSD' |

*continued on next page*

| Name          | Description                                                                   |
|---------------|-------------------------------------------------------------------------------|
| longName      | <b>Value:</b> 'Mean-Square Displacement'                                      |
| numberOfTests | <b>Value:</b> 14                                                              |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)     |
| variables     | <b>Value:</b> [['MSD', 'column1', 'MSD',<br>'time'], ['MSD', 'column2', 'M... |

## 75.4 Class *VACFTests*



### 75.4.1 Methods

**runTest**(*self*)

Overrides: *nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest*

*Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest(Section 75.2)*

setUp(), tearDown()

*Inherited from unittest.TestCase*

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

### 75.4.2 Class Variables

| Name          | Description                                          |
|---------------|------------------------------------------------------|
| shortName     | <b>Value:</b> 'VACF'                                 |
| longName      | <b>Value:</b> 'Velocity AutoCorrelation<br>Function' |
| numberOfTests | <b>Value:</b> 14                                     |

*continued on next page*

| Name      | Description                                                                    |
|-----------|--------------------------------------------------------------------------------|
| info      | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)      |
| variables | <b>Value:</b> [['VACF', 'column1', 'VACF',<br>'time'], ['VACF', 'column2', ... |

## 75.5 Class ARATests

unittest.TestCase

nMOLDYN.Tests.StabilityTests.AnalysisTest

nMOLDYN.Tests.StabilityTests.ARATests

### 75.5.1 Methods

**runTest(self)**

Overrides: nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest

*Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest(Section 75.2)*

setUp(), tearDown()

*Inherited from unittest.TestCase*

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

### 75.5.2 Class Variables

| Name          | Description                                                                   |
|---------------|-------------------------------------------------------------------------------|
| shortName     | <b>Value:</b> 'ARA'                                                           |
| longName      | <b>Value:</b> 'Auto-Regressive Analysis'                                      |
| numberOfTests | <b>Value:</b> 12                                                              |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)     |
| variables     | <b>Value:</b> [['MSD', 'column1', 'ARA',<br>'time_msd'], ['MSD', 'column2'... |

## 75.6 Class AVACFTests



### 75.6.1 Methods

**runTest(*self*)**

Overrides: nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest

*Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest (Section 75.2)*

setUp(), tearDown()

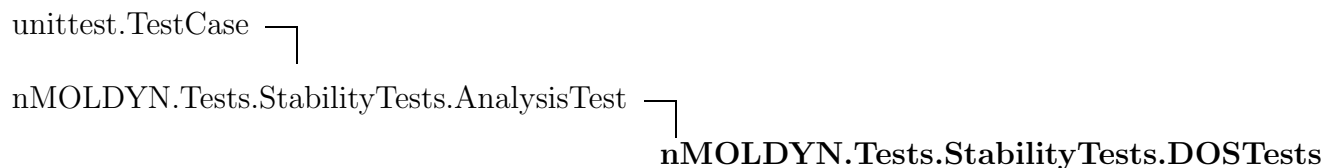
*Inherited from unittest.TestCase*

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

### 75.6.2 Class Variables

| Name          | Description                                                                |
|---------------|----------------------------------------------------------------------------|
| shortName     | <b>Value:</b> 'AVACF'                                                      |
| longName      | <b>Value:</b> 'Angular Velocity AutoCorrelation Function'                  |
| numberOfTests | <b>Value:</b> 14                                                           |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)  |
| variables     | <b>Value:</b> [['AVACF', 'column1', 'AVACF', 'time'], ['AVACF', 'column... |

## 75.7 Class *DOSTests*



### 75.7.1 Methods

**runTest(*self*)**

Overrides: *nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest*

*Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest(Section 75.2)*

setUp(), tearDown()

*Inherited from unittest.TestCase*

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

### 75.7.2 Class Variables

| Name          | Description                                                                   |
|---------------|-------------------------------------------------------------------------------|
| shortName     | <b>Value:</b> 'DOS'                                                           |
| longName      | <b>Value:</b> 'Density Of States'                                             |
| numberOfTests | <b>Value:</b> 14                                                              |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)     |
| variables     | <b>Value:</b> [['DOS', 'column1', 'DOS',<br>'frequency'], ['DOS', 'column2... |

## 75.8 Class DCSFTests



### 75.8.1 Methods

**runTest(*self*)**

Overrides: nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest

*Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest (Section 75.2)*

setUp(), tearDown()

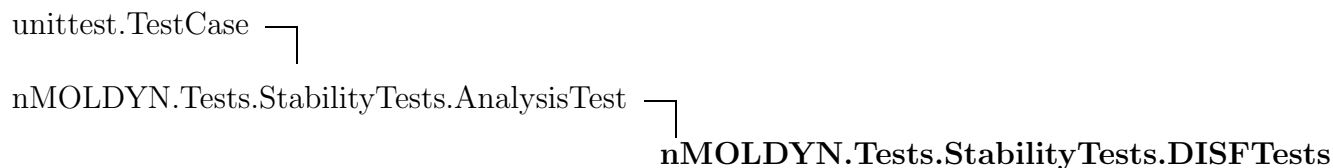
*Inherited from unittest.TestCase*

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

### 75.8.2 Class Variables

| Name          | Description                                                                     |
|---------------|---------------------------------------------------------------------------------|
| shortName     | <b>Value:</b> 'DCSF'                                                            |
| longName      | <b>Value:</b> 'Dynamic Coherent Structure Factor'                               |
| numberOfTests | <b>Value:</b> 12                                                                |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)       |
| variables     | <b>Value:</b> [['Fqt', 'q', 'DCSF', 'q'],<br>['Fqt', 'time', 'DCSF', 'time...]] |

## 75.9 Class *DISFTests*



### 75.9.1 Methods

**runTest(*self*)**

Overrides: *nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest*

*Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest (Section 75.2)*

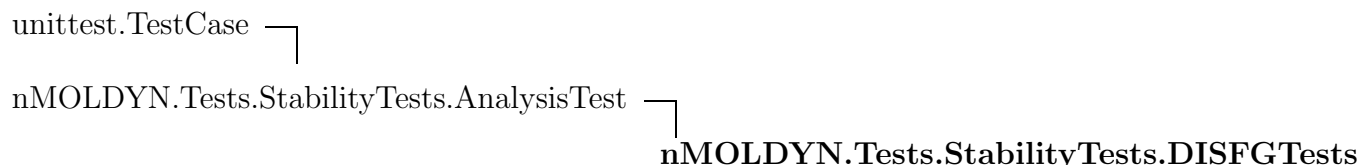
setUp(), tearDown()

*Inherited from unittest.TestCase*

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

### 75.9.2 Class Variables

| Name          | Description                                                                     |
|---------------|---------------------------------------------------------------------------------|
| shortName     | <b>Value:</b> 'DISF'                                                            |
| longName      | <b>Value:</b> 'Dynamic Incoherent Structure Factor'                             |
| numberOfTests | <b>Value:</b> 12                                                                |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)       |
| variables     | <b>Value:</b> [['Fqt', 'q', 'DISF', 'q'],<br>['Fqt', 'time', 'DISF', 'time...]] |

**75.10 Class DISFGTests****75.10.1 Methods****runTest(*self*)**Overrides: *nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest****Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest(Section 75.2)***

setUp(), tearDown()

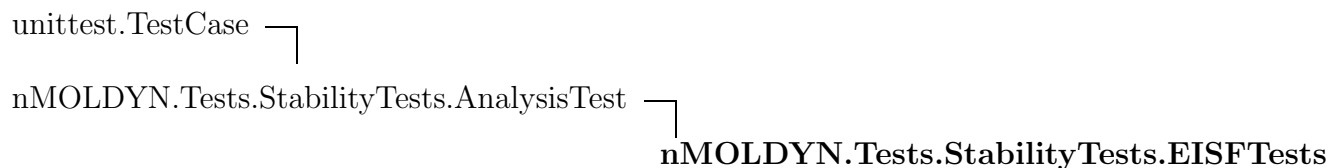
***Inherited from unittest.TestCase***

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

**75.10.2 Class Variables**

| Name          | Description                                                                     |
|---------------|---------------------------------------------------------------------------------|
| shortName     | <b>Value:</b> 'DISFG'                                                           |
| longName      | <b>Value:</b> 'Dynamic Incoherent Structure Factor'                             |
| numberOfTests | <b>Value:</b> 12                                                                |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)       |
| variables     | <b>Value:</b> [['Fqt', 'q', 'DISFG', 'q'],<br>['Fqt', 'time', 'DISFG', 'ti...]] |

## 75.11 Class *EISFTests*



### 75.11.1 Methods

**runTest(*self*)**

Overrides: *nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest*

*Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest(Section 75.2)*

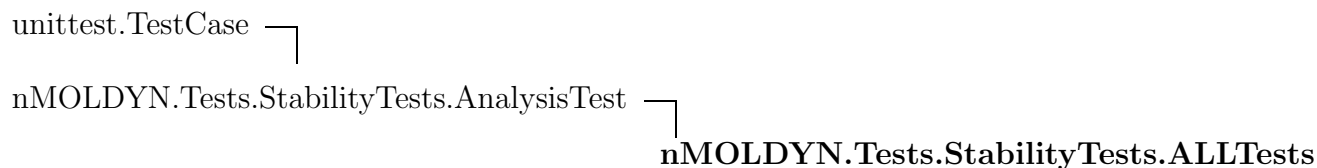
setUp(), tearDown()

*Inherited from unittest.TestCase*

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

### 75.11.2 Class Variables

| Name          | Description                                                                  |
|---------------|------------------------------------------------------------------------------|
| shortName     | <b>Value:</b> 'EISF'                                                         |
| longName      | <b>Value:</b> 'Elastic Incoherent Structure Factor'                          |
| numberOfTests | <b>Value:</b> 12                                                             |
| info          | <b>Value:</b> '%d tests about %s analysis.'<br>%(numberOfTests, longName)    |
| variables     | <b>Value:</b> [['EISF', 'column1', 'EISF', 'q'], ['EISF', 'column2', 'E...]] |

**75.12 Class *ALLTests*****75.12.1 Methods****runTest(*self*)**Overrides: *nMOLDYN.Tests.StabilityTests.AnalysisTest.runTest****Inherited from nMOLDYN.Tests.StabilityTests.AnalysisTest(Section 75.2)***

setUp(), tearDown()

***Inherited from unittest.TestCase***

\_\_call\_\_(), \_\_init\_\_(), \_\_repr\_\_(), \_\_str\_\_(), assertAlmostEqual(), assertAlmostEquals(), assertEquals(), assertEquals(), assertFalse(), assertNotAlmostEqual(), assertNotAlmostEquals(), assertNotEqual(), assertNotEquals(), assertRaises(), assertTrue(), assert\_(), countTestCases(), debug(), defaultTestResult(), fail(), failIf(), failIfAlmostEqual(), failIfEqual(), failUnless(), failUnlessAlmostEqual(), failUnlessEqual(), failUnlessRaises(), id(), run(), shortDescription()

**75.12.2 Class Variables**

| Name          | Description                         |
|---------------|-------------------------------------|
| shortName     | <b>Value:</b> 'ALL'                 |
| longName      | <b>Value:</b> 'All available tests' |
| numberOfTests | <b>Value:</b> 12                    |
| variables     | <b>Value:</b> None                  |
| info          | <b>Value:</b> ''                    |

## 76 Package nMOLDYN.Tests.VACF

### 76.1 Modules

- **TestsContents** (*Section 77, p. 345*)
- **runTests** (*Section 78, p. 346*)

## 77 Module *nMOLDYN.Tests.VACF.TestsContents*

### 77.1 Variables

| Name     | Description                            |
|----------|----------------------------------------|
| template | <b>Value:</b> { 'REF': {}, 'NEW': {} } |
| test     | <b>Value:</b> []                       |

## 78 Module `nMOLDYN.Tests.VACF.runTests`

### 78.1 Variables

| Name                       | Description                                                                  |
|----------------------------|------------------------------------------------------------------------------|
| <code>pMoldyn</code>       | <b>Value:</b><br>'/home/cs/pellegrini/nMOLDYN/nMOLDYN2.2.5/nMoldyn/bin/pMo.' |
| <code>selectedTests</code> | <b>Value:</b> range(1, 15)                                                   |

## **79    Module nMOLDYN.\_\_pkginfo\_\_**

**Version:** 3.0.5

## Index

- nMOLDYN (*package*), 9–11
  - nMOLDYN.\_\_pkginfo\_\_ (*module*), 347
  - nMOLDYN.Analysis (*package*), 12
    - nMOLDYN.Analysis.Analysis (*module*), 13–17
    - nMOLDYN.Analysis.Dynamics (*module*), 18–49
    - nMOLDYN.Analysis.NMR (*module*), 50–54
    - nMOLDYN.Analysis.Scattering (*module*), 55–74
    - nMOLDYN.Analysis.Slave (*module*), 75
    - nMOLDYN.Analysis.Structure (*module*), 76–85
    - nMOLDYN.Analysis.Template (*module*), 86–143
  - nMOLDYN.Core (*package*), 144
    - nMOLDYN.Core.Chemistry (*module*), 145
    - nMOLDYN.Core.Config (*module*), 146
    - nMOLDYN.Core.Error (*module*), 147
    - nMOLDYN.Core.IOFiles (*module*), 148–157
    - nMOLDYN.Core.Logger (*module*), 158–162
    - nMOLDYN.Core.Mathematics (*module*), 163–169
    - nMOLDYN.Core.Misc (*module*), 170–171
    - nMOLDYN.Core.Preferences (*module*), 172–173
  - nMOLDYN.GUI (*package*), 174–175
    - nMOLDYN.GUI.AnalysisBenchmarkDialog (*module*), 178–179
    - nMOLDYN.GUI.AnalysisDialog (*module*), 180–181
    - nMOLDYN.GUI.AnimationDialog (*module*), 182–184
    - nMOLDYN.GUI.ASCIIToNetCDFConversionDialog (*module*), 176–177
    - nMOLDYN.GUI.CheckJobsStatusDialog (*module*), 185–187
    - nMOLDYN.GUI.GeneralInformationsDialog (*module*), 188–190
    - nMOLDYN.GUI.HTMLReader (*package*), 191
    - nMOLDYN.GUI.MainDialog (*module*), 222–225
    - nMOLDYN.GUI.NetCDFToASCIIConversionDialog (*module*), 226–227
    - nMOLDYN.GUI.PDBSnapshotGeneratorDialog (*module*), 228–229
    - nMOLDYN.GUI.PlotNetCDFVariableDialog (*module*), 230–233
    - nMOLDYN.GUI.PreferencesDialog (*module*), 234–235
    - nMOLDYN.GUI.PyroServerDialog (*module*), 236–238
    - nMOLDYN.GUI.SelectionDialog (*module*), 239–243
    - nMOLDYN.GUI.Tags (*module*), 244
    - nMOLDYN.GUI.TrajectoryConversionDialog (*module*), 245–250
    - nMOLDYN.GUI.ViewEffectiveModeDialog (*module*), 251–253
    - nMOLDYN.GUI.Widgets (*module*), 254–305
  - nMOLDYN.Tests (*package*), 306
    - nMOLDYN.Tests.AnalysisTests (*module*), 313
    - nMOLDYN.Tests.ARA (*package*), 307
    - nMOLDYN.Tests.AVACF (*package*), 310
    - nMOLDYN.Tests.BuildTestCases (*module*), 314
    - nMOLDYN.Tests.DCSF (*package*), 315
    - nMOLDYN.Tests.DISF (*package*), 318
    - nMOLDYN.Tests.DISFG (*package*), 321
    - nMOLDYN.Tests.DOS (*package*), 324
    - nMOLDYN.Tests.EISF (*package*), 327
    - nMOLDYN.Tests.MSD (*package*), 330
    - nMOLDYN.Tests.StabilityTests (*module*), 333–343
    - nMOLDYN.Tests.VACF (*package*), 344