

# ReGenESyS - REborned GENeric and Expansible SYstem Simulator

v22.01 (jude)

Generated by Doxygen 1.9.1

<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>9</b>
2.1 Class List	9
<b>3 File Index</b>	<b>16</b>
3.1 File List	16
<b>4 Class Documentation</b>	<b>31</b>
4.1 Access Class Reference	31
4.1.1 Detailed Description	33
4.2 AnElectronicAssemblyAndTestSystem Class Reference	34
4.2.1 Detailed Description	35
4.3 Assign Class Reference	35
4.3.1 Detailed Description	37
4.4 Assign::Assignment Class Reference	37
4.4.1 Detailed Description	38
4.5 SequenceStep::Assignment Class Reference	38
4.5.1 Detailed Description	38
4.6 Attribute Class Reference	38
4.6.1 Detailed Description	40
4.7 BaseConsoleGenesysApplication Class Reference	40
4.7.1 Detailed Description	41
4.8 yy::genesyspp_parser::basic_symbol< Base > Struct Template Reference	41
4.8.1 Detailed Description	43
4.9 Batch Class Reference	43
4.9.1 Detailed Description	45
4.10 Book_Cap02_Example01 Class Reference	46
4.10.1 Detailed Description	46
4.11 SamplerBoostImpl::BoostImplRNG_Parameters Struct Reference	47
4.11.1 Detailed Description	47
4.12 yy::genesyspp_parser::by_kind Struct Reference	48
4.12.1 Detailed Description	49
4.12.2 Member Function Documentation	49
4.12.3 Member Data Documentation	49
4.13 CelularAutomata Class Reference	50
4.13.1 Detailed Description	51
4.14 Collector_if Class Reference	51
4.14.1 Detailed Description	53
4.15 CollectorDatafile_if Class Reference	53
4.15.1 Detailed Description	55
4.15.2 Member Function Documentation	55

4.16 CollectorDatafileDefaultImpl1 Class Reference	56
4.16.1 Detailed Description	57
4.16.2 Member Function Documentation	57
4.17 CollectorDefaultImpl1 Class Reference	58
4.17.1 Detailed Description	58
4.18 ComponentManager Class Reference	59
4.18.1 Detailed Description	59
4.18.2 Constructor & Destructor Documentation	59
4.19 HypothesisTester_if::ConfidenceInterval Class Reference	59
4.19.1 Detailed Description	60
4.20 ConnectionManager Class Reference	60
4.20.1 Detailed Description	60
4.20.2 Member Function Documentation	60
4.21 yy::genesyspp_parser::context Class Reference	61
4.21.1 Detailed Description	61
4.21.2 Member Function Documentation	61
4.22 Counter Class Reference	62
4.22.1 Detailed Description	63
4.23 Create Class Reference	63
4.23.1 Detailed Description	65
4.24 Decide Class Reference	66
4.24.1 Detailed Description	67
4.25 ModelComponent::DEFAULT_VALUES Struct Reference	68
4.25.1 Detailed Description	68
4.26 SourceModelComponent::DEFAULT_VALUES Struct Reference	68
4.26.1 Detailed Description	68
4.27 WriteExpression::DEFAULT_VALUES Struct Reference	68
4.27.1 Detailed Description	69
4.28 WriteInExpression::DEFAULT_VALUES Struct Reference	69
4.28.1 Detailed Description	69
4.29 WriteInText::DEFAULT_VALUES Struct Reference	69
4.29.1 Detailed Description	69
4.30 WriteText::DEFAULT_VALUES Struct Reference	70
4.30.1 Detailed Description	70
4.31 SamplerDefaultImpl1::DefaultImpl1RNG_Parameters Struct Reference	70
4.31.1 Detailed Description	71
4.32 Delay Class Reference	71
4.32.1 Detailed Description	73
4.33 DialogAbout Class Reference	74
4.33.1 Detailed Description	74
4.34 Ui::DialogAbout Class Reference	75
4.34.1 Detailed Description	75

4.35 DialogModelInformation Class Reference . . . . .	76
4.35.1 Detailed Description . . . . .	76
4.36 Ui::DialogModelInformation Class Reference . . . . .	77
4.36.1 Detailed Description . . . . .	77
4.37 Dispose Class Reference . . . . .	78
4.37.1 Detailed Description . . . . .	79
4.38 DropOff Class Reference . . . . .	79
4.38.1 Detailed Description . . . . .	81
4.39 Dummy Class Reference . . . . .	81
4.39.1 Detailed Description . . . . .	83
4.40 ElementManager Class Reference . . . . .	83
4.40.1 Detailed Description . . . . .	84
4.40.2 Constructor & Destructor Documentation . . . . .	84
4.41 ElementManager_if Class Reference . . . . .	84
4.41.1 Detailed Description . . . . .	84
4.42 Enter Class Reference . . . . .	85
4.42.1 Detailed Description . . . . .	86
4.43 Entity Class Reference . . . . .	87
4.43.1 Detailed Description . . . . .	89
4.44 EntityGroup Class Reference . . . . .	90
4.44.1 Detailed Description . . . . .	91
4.44.2 Member Function Documentation . . . . .	91
4.45 EntityType Class Reference . . . . .	92
4.45.1 Detailed Description . . . . .	93
4.45.2 Member Function Documentation . . . . .	93
4.46 Event Class Reference . . . . .	94
4.46.1 Detailed Description . . . . .	94
4.47 Exact Class Reference . . . . .	94
4.47.1 Detailed Description . . . . .	95
4.48 Exit Class Reference . . . . .	95
4.48.1 Detailed Description . . . . .	97
4.49 ExperimentManager Class Reference . . . . .	97
4.49.1 Detailed Description . . . . .	98
4.50 ExperimentManager_if Class Reference . . . . .	98
4.50.1 Detailed Description . . . . .	99
4.51 ExperimentManagerDefaultImpl1 Class Reference . . . . .	100
4.51.1 Detailed Description . . . . .	101
4.52 Failure Class Reference . . . . .	101
4.52.1 Detailed Description . . . . .	103
4.52.2 Member Function Documentation . . . . .	103
4.53 File Class Reference . . . . .	104
4.53.1 Detailed Description . . . . .	105

4.53.2 Member Function Documentation	105
4.54 formPlugins Class Reference	106
4.54.1 Detailed Description	106
4.55 Ui::formPlugins Class Reference	107
4.55.1 Detailed Description	107
4.56 Formula Class Reference	108
4.56.1 Detailed Description	109
4.57 FullSimulationOfComplexModel Class Reference	110
4.57.1 Detailed Description	110
4.57.2 Member Function Documentation	111
4.58 ParserManager::GenerateNewParserResult Struct Reference	111
4.58.1 Detailed Description	111
4.59 GenesysApplication_if Class Reference	111
4.59.1 Detailed Description	112
4.60 genesyspp_driver Class Reference	112
4.60.1 Detailed Description	112
4.61 yy::genesyspp_parser Class Reference	113
4.61.1 Detailed Description	115
4.61.2 Member Function Documentation	115
4.62 GenesysQtGUI Class Reference	117
4.62.1 Detailed Description	117
4.63 GenesysShell_if Class Reference	118
4.63.1 Detailed Description	119
4.64 GenesysTerminalApp Class Reference	120
4.64.1 Detailed Description	121
4.65 Hold Class Reference	121
4.65.1 Detailed Description	123
4.66 HypothesisTester_if Class Reference	124
4.66.1 Detailed Description	125
4.67 HypothesisTesterDefaultImpl1 Class Reference	126
4.67.1 Detailed Description	127
4.67.2 Member Function Documentation	127
4.68 Leave Class Reference	130
4.68.1 Detailed Description	132
4.69 LicenceManager Class Reference	133
4.69.1 Detailed Description	133
4.70 List< T > Class Template Reference	134
4.70.1 Detailed Description	134
4.71 yy::location Class Reference	135
4.71.1 Detailed Description	135
4.72 LODE Class Reference	136
4.72.1 Detailed Description	137

4.73 MainWindow Class Reference . . . . .	138
4.73.1 Detailed Description . . . . .	138
4.74 Ui::MainWindow Class Reference . . . . .	139
4.74.1 Detailed Description . . . . .	139
4.75 MapKernelAndUI Class Reference . . . . .	139
4.75.1 Detailed Description . . . . .	140
4.76 MarkovChain Class Reference . . . . .	141
4.76.1 Detailed Description . . . . .	142
4.77 Match Class Reference . . . . .	142
4.77.1 Detailed Description . . . . .	144
4.78 Model Class Reference . . . . .	144
4.78.1 Detailed Description . . . . .	146
4.78.2 Constructor & Destructor Documentation . . . . .	146
4.79 ModelChecker_if Class Reference . . . . .	146
4.79.1 Detailed Description . . . . .	147
4.79.2 Member Function Documentation . . . . .	147
4.80 ModelCheckerDefaultImpl1 Class Reference . . . . .	149
4.80.1 Detailed Description . . . . .	149
4.80.2 Member Function Documentation . . . . .	149
4.81 ModelComponent Class Reference . . . . .	151
4.81.1 Detailed Description . . . . .	152
4.82 ModelElement Class Reference . . . . .	152
4.82.1 Detailed Description . . . . .	153
4.82.2 Member Function Documentation . . . . .	153
4.83 ModelInfo Class Reference . . . . .	155
4.83.1 Detailed Description . . . . .	156
4.84 ModelManager Class Reference . . . . .	156
4.84.1 Detailed Description . . . . .	156
4.85 ModelPersistence_if Class Reference . . . . .	157
4.85.1 Detailed Description . . . . .	157
4.86 ModelPersistenceDefaultImpl1 Class Reference . . . . .	158
4.86.1 Detailed Description . . . . .	158
4.87 ModelSimulation Class Reference . . . . .	158
4.87.1 Detailed Description . . . . .	160
4.87.2 Member Function Documentation . . . . .	160
4.88 ParserManager::NewParser Struct Reference . . . . .	160
4.88.1 Detailed Description . . . . .	161
4.89 obj_t Class Reference . . . . .	161
4.89.1 Detailed Description . . . . .	161
4.90 ODEfunction Class Reference . . . . .	161
4.90.1 Detailed Description . . . . .	162
4.91 OLD_ODEelement Class Reference . . . . .	163

4.91.1 Detailed Description . . . . .	164
4.92 OnEventManager Class Reference . . . . .	164
4.92.1 Detailed Description . . . . .	165
4.93 OperatingSystem02 Class Reference . . . . .	166
4.93.1 Detailed Description . . . . .	166
4.94 OperatingSystem03 Class Reference . . . . .	167
4.94.1 Detailed Description . . . . .	168
4.95 Parser_if Class Reference . . . . .	168
4.95.1 Detailed Description . . . . .	169
4.96 ParserChangesInformation Class Reference . . . . .	169
4.96.1 Detailed Description . . . . .	169
4.97 ParserDefaultImpl1 Class Reference . . . . .	170
4.97.1 Detailed Description . . . . .	170
4.98 ParserDefaultImpl2 Class Reference . . . . .	171
4.98.1 Detailed Description . . . . .	171
4.99 ParserManager Class Reference . . . . .	172
4.99.1 Detailed Description . . . . .	172
4.100 PersistentObject_base Class Reference . . . . .	172
4.100.1 Detailed Description . . . . .	172
4.101 PickStation Class Reference . . . . .	173
4.101.1 Detailed Description . . . . .	174
4.102 PickUp Class Reference . . . . .	175
4.102.1 Detailed Description . . . . .	176
4.103 Plugin Class Reference . . . . .	176
4.103.1 Detailed Description . . . . .	177
4.104 PluginConnector_if Class Reference . . . . .	177
4.104.1 Detailed Description . . . . .	178
4.105 PluginConnectorDummyImpl1 Class Reference . . . . .	178
4.105.1 Detailed Description . . . . .	178
4.105.2 Member Function Documentation . . . . .	179
4.106 PluginInformation Class Reference . . . . .	179
4.106.1 Detailed Description . . . . .	180
4.106.2 Constructor & Destructor Documentation . . . . .	180
4.107 PluginManager Class Reference . . . . .	180
4.107.1 Detailed Description . . . . .	181
4.108 yy::position Class Reference . . . . .	181
4.108.1 Detailed Description . . . . .	182
4.109 ProbabilityDistribution Class Reference . . . . .	182
4.109.1 Detailed Description . . . . .	182
4.109.2 Member Function Documentation . . . . .	182
4.110 Process Class Reference . . . . .	183
4.110.1 Detailed Description . . . . .	185

4.111 Queue Class Reference . . . . .	186
4.111.1 Detailed Description . . . . .	187
4.111.2 Member Function Documentation . . . . .	188
4.112 QueueableItem Class Reference . . . . .	188
4.112.1 Detailed Description . . . . .	189
4.113 Record Class Reference . . . . .	189
4.113.1 Detailed Description . . . . .	191
4.114 Release Class Reference . . . . .	192
4.114.1 Detailed Description . . . . .	193
4.115 Remove Class Reference . . . . .	194
4.115.1 Detailed Description . . . . .	195
4.116 Resource Class Reference . . . . .	195
4.116.1 Detailed Description . . . . .	197
4.116.2 Constructor & Destructor Documentation . . . . .	198
4.116.3 Member Function Documentation . . . . .	198
4.117 Sampler_if::RNG_Parameters Struct Reference . . . . .	199
4.117.1 Detailed Description . . . . .	199
4.118 Route Class Reference . . . . .	199
4.118.1 Detailed Description . . . . .	201
4.119 Sampler_if Class Reference . . . . .	202
4.119.1 Detailed Description . . . . .	203
4.120 SamplerBoostImpl Class Reference . . . . .	204
4.120.1 Detailed Description . . . . .	205
4.121 SamplerDefaultImpl1 Class Reference . . . . .	206
4.121.1 Detailed Description . . . . .	207
4.122 ScenarioExperiment_if Class Reference . . . . .	207
4.122.1 Detailed Description . . . . .	207
4.123 Schedule Class Reference . . . . .	208
4.123.1 Detailed Description . . . . .	209
4.124 Search Class Reference . . . . .	210
4.124.1 Detailed Description . . . . .	211
4.125 SeizableItem Class Reference . . . . .	211
4.125.1 Detailed Description . . . . .	212
4.126 Seize Class Reference . . . . .	212
4.126.1 Detailed Description . . . . .	214
4.127 yy::genesyspp_parser::semantic_type Class Reference . . . . .	215
4.127.1 Detailed Description . . . . .	216
4.127.2 Member Function Documentation . . . . .	216
4.128 Separate Class Reference . . . . .	217
4.128.1 Detailed Description . . . . .	219
4.129 Sequence Class Reference . . . . .	220
4.129.1 Detailed Description . . . . .	221



4.130 SequenceStep Class Reference . . . . .	222
4.130.1 Detailed Description . . . . .	223
4.131 Set Class Reference . . . . .	223
4.131.1 Detailed Description . . . . .	225
4.131.2 Member Function Documentation . . . . .	225
4.132 Signal Class Reference . . . . .	226
4.132.1 Detailed Description . . . . .	227
4.133 SimulationControl Class Reference . . . . .	227
4.133.1 Detailed Description . . . . .	228
4.134 SimulationEvent Class Reference . . . . .	229
4.134.1 Detailed Description . . . . .	229
4.135 SimulationExperiment Class Reference . . . . .	230
4.135.1 Detailed Description . . . . .	230
4.136 SimulationReporter_if Class Reference . . . . .	230
4.136.1 Detailed Description . . . . .	231
4.137 SimulationReporterDefaultImpl1 Class Reference . . . . .	231
4.137.1 Detailed Description . . . . .	232
4.137.2 Member Function Documentation . . . . .	232
4.138 SimulationResponse Class Reference . . . . .	232
4.138.1 Detailed Description . . . . .	234
4.139 SimulationScenario Class Reference . . . . .	234
4.139.1 Detailed Description . . . . .	234
4.139.2 Member Function Documentation . . . . .	234
4.140 Simulator Class Reference . . . . .	235
4.140.1 Detailed Description . . . . .	235
4.141 SinkModelComponent Class Reference . . . . .	236
4.141.1 Detailed Description . . . . .	237
4.142 yy::genesyspp_parser::stack< T, S >::slice Class Reference . . . . .	237
4.142.1 Detailed Description . . . . .	237
4.143 Smart_AssignWriteSeizes Class Reference . . . . .	238
4.143.1 Detailed Description . . . . .	238
4.144 Smart_Delay Class Reference . . . . .	239
4.144.1 Detailed Description . . . . .	240
4.144.2 Member Function Documentation . . . . .	240
4.145 Smart_Dummy Class Reference . . . . .	241
4.145.1 Detailed Description . . . . .	241
4.145.2 Member Function Documentation . . . . .	242
4.146 Smart_HoldSignal Class Reference . . . . .	243
4.146.1 Detailed Description . . . . .	243
4.146.2 Member Function Documentation . . . . .	244
4.147 Smart_ModelInfoModelSimulation Class Reference . . . . .	245
4.147.1 Detailed Description . . . . .	245

4.147.2 Member Function Documentation . . . . .	246
4.148 Smart_OnEvent Class Reference . . . . .	247
4.148.1 Detailed Description . . . . .	248
4.148.2 Member Function Documentation . . . . .	248
4.149 Smart_Parser Class Reference . . . . .	249
4.149.1 Detailed Description . . . . .	249
4.149.2 Member Function Documentation . . . . .	250
4.150 Smart_ParserModelFunctions Class Reference . . . . .	251
4.150.1 Detailed Description . . . . .	251
4.150.2 Member Function Documentation . . . . .	252
4.151 Smart_Process Class Reference . . . . .	253
4.151.1 Detailed Description . . . . .	253
4.151.2 Member Function Documentation . . . . .	254
4.152 Smart_ProcessSet Class Reference . . . . .	255
4.152.1 Detailed Description . . . . .	255
4.152.2 Member Function Documentation . . . . .	256
4.153 Smart_RouteStation Class Reference . . . . .	257
4.153.1 Detailed Description . . . . .	257
4.153.2 Member Function Documentation . . . . .	258
4.154 Smart_SeizeDelayRelease Class Reference . . . . .	259
4.154.1 Detailed Description . . . . .	259
4.154.2 Member Function Documentation . . . . .	260
4.155 Smart_SeizeDelayReleaseMany Class Reference . . . . .	261
4.155.1 Detailed Description . . . . .	261
4.155.2 Member Function Documentation . . . . .	262
4.156 Smart_Sequence Class Reference . . . . .	263
4.156.1 Detailed Description . . . . .	263
4.157 Solver_if Class Reference . . . . .	264
4.157.1 Detailed Description . . . . .	265
4.158 SolverDefaultImpl1 Class Reference . . . . .	266
4.158.1 Detailed Description . . . . .	267
4.159 SortFile Class Reference . . . . .	267
4.159.1 Detailed Description . . . . .	267
4.160 SourceModelComponent Class Reference . . . . .	268
4.160.1 Detailed Description . . . . .	269
4.161 Start Class Reference . . . . .	270
4.161.1 Detailed Description . . . . .	271
4.162 Station Class Reference . . . . .	272
4.162.1 Detailed Description . . . . .	273
4.162.2 Member Function Documentation . . . . .	273
4.163 Statistics_if Class Reference . . . . .	274
4.163.1 Detailed Description . . . . .	275

4.164 StatisticsCollector Class Reference . . . . .	275
4.164.1 Detailed Description . . . . .	277
4.165 StatisticsDatafile_if Class Reference . . . . .	278
4.165.1 Detailed Description . . . . .	279
4.166 StatisticsDatafileDefaultImpl1 Class Reference . . . . .	280
4.166.1 Detailed Description . . . . .	281
4.167 StatisticsDefaultImpl1 Class Reference . . . . .	282
4.167.1 Detailed Description . . . . .	283
4.168 Stop Class Reference . . . . .	283
4.168.1 Detailed Description . . . . .	285
4.169 Storage Class Reference . . . . .	286
4.169.1 Detailed Description . . . . .	287
4.169.2 Member Function Documentation . . . . .	287
4.170 Store Class Reference . . . . .	288
4.170.1 Detailed Description . . . . .	289
4.171 Submodel Class Reference . . . . .	289
4.171.1 Detailed Description . . . . .	291
4.172 yy::genesyspp_parser::symbol_kind Struct Reference . . . . .	291
4.172.1 Detailed Description . . . . .	292
4.172.2 Member Enumeration Documentation . . . . .	292
4.173 yy::genesyspp_parser::symbol_type Struct Reference . . . . .	292
4.173.1 Detailed Description . . . . .	294
4.174 yy::genesyspp_parser::syntax_error Struct Reference . . . . .	294
4.174.1 Detailed Description . . . . .	295
4.175 HypothesisTester_if::TestResult Class Reference . . . . .	295
4.175.1 Detailed Description . . . . .	295
4.176 yy::genesyspp_parser::token Struct Reference . . . . .	295
4.176.1 Detailed Description . . . . .	296
4.177 TraceErrorEvent Class Reference . . . . .	297
4.177.1 Detailed Description . . . . .	297
4.178 TraceEvent Class Reference . . . . .	298
4.178.1 Detailed Description . . . . .	298
4.179 TraceManager Class Reference . . . . .	298
4.179.1 Detailed Description . . . . .	299
4.180 TraceSimulationEvent Class Reference . . . . .	300
4.180.1 Detailed Description . . . . .	300
4.181 TraceSimulationProcess Class Reference . . . . .	301
4.181.1 Detailed Description . . . . .	301
4.182 Traits< T > Struct Template Reference . . . . .	301
4.182.1 Detailed Description . . . . .	301
4.183 Traits< GenesysApplication_if > Struct Reference . . . . .	302
4.183.1 Detailed Description . . . . .	302

4.184 Traits< HypothesisTester_if > Struct Reference . . . . .	302
4.184.1 Detailed Description . . . . .	302
4.185 Traits< Solver_if > Struct Reference . . . . .	303
4.185.1 Detailed Description . . . . .	303
4.186 TraitsKernel< T > Struct Template Reference . . . . .	303
4.186.1 Detailed Description . . . . .	303
4.187 TraitsKernel< Collector_if > Struct Reference . . . . .	303
4.187.1 Detailed Description . . . . .	304
4.188 TraitsKernel< Model > Struct Reference . . . . .	304
4.188.1 Detailed Description . . . . .	304
4.189 TraitsKernel< ModelChecker_if > Struct Reference . . . . .	304
4.189.1 Detailed Description . . . . .	304
4.190 TraitsKernel< ModelComponent > Struct Reference . . . . .	305
4.190.1 Detailed Description . . . . .	305
4.191 TraitsKernel< ModelElement > Struct Reference . . . . .	305
4.191.1 Detailed Description . . . . .	305
4.192 TraitsKernel< ModelPersistence_if > Struct Reference . . . . .	305
4.192.1 Detailed Description . . . . .	306
4.193 TraitsKernel< Parser_if > Struct Reference . . . . .	306
4.193.1 Detailed Description . . . . .	306
4.194 TraitsKernel< PluginConnector_if > Struct Reference . . . . .	306
4.194.1 Detailed Description . . . . .	306
4.195 TraitsKernel< Sampler_if > Struct Reference . . . . .	307
4.195.1 Detailed Description . . . . .	307
4.196 TraitsKernel< SimulationReporter_if > Struct Reference . . . . .	307
4.196.1 Detailed Description . . . . .	307
4.197 TraitsKernel< Statistics_if > Struct Reference . . . . .	307
4.197.1 Detailed Description . . . . .	308
4.198 TraitsKernel< StatisticsDatafile_if > Struct Reference . . . . .	308
4.198.1 Detailed Description . . . . .	308
4.199 Ui_DialogAbout Class Reference . . . . .	309
4.199.1 Detailed Description . . . . .	310
4.200 Ui_DialogModelInformation Class Reference . . . . .	310
4.200.1 Detailed Description . . . . .	311
4.201 Ui_formPlugins Class Reference . . . . .	311
4.201.1 Detailed Description . . . . .	312
4.202 Ui_MainWindow Class Reference . . . . .	312
4.202.1 Detailed Description . . . . .	315
4.203 Unstore Class Reference . . . . .	315
4.203.1 Detailed Description . . . . .	317
4.204 Util Class Reference . . . . .	317
4.204.1 Detailed Description . . . . .	318

4.204.2 Member Function Documentation	318
4.205 Variable Class Reference	319
4.205.1 Detailed Description	320
4.206 Waiting Class Reference	321
4.206.1 Detailed Description	322
4.207 WaitingResource Class Reference	322
4.207.1 Detailed Description	323
4.208 Write Class Reference	323
4.208.1 Detailed Description	325
4.209 WriteExpression Class Reference	326
4.209.1 Detailed Description	326
4.210 WriteInExpression Class Reference	327
4.210.1 Detailed Description	327
4.211 WriteInText Class Reference	328
4.211.1 Detailed Description	328
4.212 WriteText Class Reference	329
4.212.1 Detailed Description	330
<b>5 File Documentation</b>	<b>330</b>
5.1 <a href="#">/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy← S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/GenesysParser.h</a> File Reference	330
5.1.1 Detailed Description	331
5.2 <a href="#">/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy← S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/location.hh</a> File Reference	331
5.2.1 Detailed Description	332
5.2.2 Function Documentation	332

# Hierarchical Index

## 1.1 Class Hierarchy

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

<b>Assign::Assignment</b>	<b>37</b>
<b>yy::genesyspp_parser::basic_symbol&lt; Base &gt;</b>	<b>41</b>
<b>yy::genesyspp_parser::basic_symbol&lt; by_state &gt;</b>	<b>41</b>
<b>yy::genesyspp_parser::by_kind</b>	<b>48</b>
<b>yy::genesyspp_parser::basic_symbol&lt; by_kind &gt;</b>	<b>41</b>
<b>yy::genesyspp_parser::symbol_type</b>	<b>292</b>
<b>Collector_if</b>	<b>51</b>
<b>CollectorDatafile_if</b>	<b>53</b>

<b>ComponentManager</b>	<b>59</b>
<b>ConnectionManager</b>	<b>60</b>
<b>ElementManager</b>	<b>83</b>
<b>Event</b>	<b>94</b>
<b>ExperimentManager_if</b>	<b>98</b>
<b>yy::genesyspp_parser</b>	<b>113</b>
<b>HypothesisTester_if</b>	<b>124</b>
<b>LicenceManager</b>	<b>133</b>
<b>List&lt; T &gt;</b>	<b>134</b>
<b>List&lt; Assign::Assignment * &gt;</b>	<b>134</b>
<b>List&lt; Connection * &gt;</b>	<b>134</b>
<b>List&lt; Counter * &gt;</b>	<b>134</b>
<b>List&lt; double &gt;</b>	<b>134</b>
<b>List&lt; Entity * &gt;</b>	<b>134</b>
<b>List&lt; Event * &gt;</b>	<b>134</b>
<b>List&lt; Model * &gt;</b>	<b>134</b>
<b>List&lt; ModelComponent * &gt;</b>	<b>134</b>
<b>List&lt; ModelElement * &gt;</b>	<b>134</b>
<b>List&lt; ODEfunction * &gt;</b>	<b>134</b>
<b>List&lt; Plugin * &gt;</b>	<b>134</b>
<b>List&lt; SeizableItem * &gt;</b>	<b>134</b>
<b>List&lt; SequenceStep * &gt;</b>	<b>134</b>
<b>List&lt; SimulationControl * &gt;</b>	<b>134</b>
<b>List&lt; simulationEventHandler &gt;</b>	<b>134</b>
<b>List&lt; simulationEventHandlerMethod &gt;</b>	<b>134</b>
<b>List&lt; SimulationExperiment * &gt;</b>	<b>134</b>
<b>List&lt; SimulationResponse * &gt;</b>	<b>134</b>
<b>List&lt; SortedResourceEventHandler * &gt;</b>	<b>134</b>
<b>List&lt; StatisticsCollector * &gt;</b>	<b>134</b>
<b>List&lt; std::map&lt; std::string, double &gt; * &gt;</b>	<b>134</b>
<b>List&lt; std::string &gt;</b>	<b>134</b>
<b>List&lt; traceErrorListener &gt;</b>	<b>134</b>

<b>List&lt; traceErrorListenerMethod &gt;</b>	<b>134</b>
<b>List&lt; traceListener &gt;</b>	<b>134</b>
<b>List&lt; traceListenerMethod &gt;</b>	<b>134</b>
<b>List&lt; traceSimulationListener &gt;</b>	<b>134</b>
<b>List&lt; traceSimulationListenerMethod &gt;</b>	<b>134</b>
<b>List&lt; unsigned int &gt;</b>	<b>134</b>
<b>List&lt; Waiting * &gt;</b>	<b>134</b>
<b>List&lt; WriteText * &gt;</b>	<b>134</b>
<b>yy::location</b>	<b>135</b>
<b>Model</b>	<b>144</b>
<b>ModelChecker_if</b>	<b>146</b>
<b>ModelElement</b>	<b>152</b>
<b>Attribute</b>	<b>38</b>
<b>Counter</b>	<b>62</b>
<b>Entity</b>	<b>87</b>
<b>Failure</b>	<b>101</b>
<b>File</b>	<b>104</b>
<b>ModelComponent</b>	<b>151</b>
<b>Access</b>	<b>31</b>
<b>Assign</b>	<b>35</b>
<b>Batch</b>	<b>43</b>
<b>Decide</b>	<b>66</b>
<b>Delay</b>	<b>71</b>
<b>DropOff</b>	<b>79</b>
<b>Dummy</b>	<b>81</b>
<b>Enter</b>	<b>85</b>
<b>Exit</b>	<b>95</b>
<b>Hold</b>	<b>121</b>
<b>LSODE</b>	<b>136</b>
<b>Leave</b>	<b>130</b>
<b>Match</b>	<b>142</b>
<b>PickStation</b>	<b>173</b>

PickUp	175
Process	183
Record	189
Release	192
Remove	194
Route	199
Search	210
Seize	212
Separate	217
Signal	226
SinkModelComponent	236
Dispose	78
SourceModelComponent	268
Create	63
Start	270
Stop	283
Store	288
Submodel	289
Unstore	315
Write	323
Queue	186
Resource	195
Schedule	208
Sequence	220
Set	223
StatisticsCollector	275
Variable	319
ModelInfo	155
ModelPersistence_if	157
ModelSimulation	158
OnEventManager	164
Plugin	176



<a href="#">yy::position</a>	<a href="#">181</a>
<a href="#">Sampler_if::RNG_Parameters</a>	<a href="#">199</a>
<a href="#">Sampler_if</a>	<a href="#">202</a>
<a href="#">yy::genesyspp_parser::semantic_type</a>	<a href="#">215</a>
<a href="#">SimulationEvent</a>	<a href="#">229</a>
<a href="#">SimulationReporterDefaultImpl1</a>	<a href="#">231</a>
<a href="#">SimulationResponse</a>	<a href="#">232</a>
<a href="#">SimulationControl</a>	<a href="#">227</a>
<a href="#">SimulationScenario</a>	<a href="#">234</a>
<a href="#">Simulator</a>	<a href="#">235</a>
<a href="#">yy::genesyspp_parser::stack&lt; T, S &gt;::slice</a>	<a href="#">237</a>
<a href="#">Solver_if</a>	<a href="#">264</a>
<a href="#">Statistics_if</a>	<a href="#">274</a>
<a href="#">yy::genesyspp_parser::symbol_kind</a>	<a href="#">291</a>
<a href="#">yy::genesyspp_parser::syntax_error</a>	<a href="#">294</a>
<a href="#">yy::genesyspp_parser::token</a>	<a href="#">295</a>
<a href="#">TraceManager</a>	<a href="#">298</a>
<a href="#">TraceSimulationProcess</a>	<a href="#">301</a>
<a href="#">Traits&lt; GenesysApplication_if &gt;</a>	<a href="#">302</a>
<a href="#">Traits&lt; HypothesisTester_if &gt;</a>	<a href="#">302</a>
<a href="#">Traits&lt; Solver_if &gt;</a>	<a href="#">303</a>

## 2 Class Index

### 2.1 Class List

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

<a href="#">Access</a>	<a href="#">31</a>
<a href="#">Assign</a>	<a href="#">35</a>
<a href="#">Assign::Assignment</a>	<a href="#">37</a>
<a href="#">Attribute</a>	<a href="#">38</a>
<a href="#">yy::genesyspp_parser::basic_symbol&lt; Base &gt;</a>	<a href="#">41</a>
<a href="#">Batch</a>	<a href="#">43</a>

<b>yy::genesyspp_parser::by_kind</b>	
Type access provider for token (enum) based symbols	48
Collector_if	51
CollectorDatafile_if	53
ComponentManager	59
ConnectionManager	60
Counter	62
Create	63
Decide	66
Delay	71
Dispose	78
DropOff	79
Dummy	81
ElementManager	83
Enter	85
Entity	87
Event	94
Exit	95
ExperimentManager_if	98
Failure	101
File	104
<b>yy::genesyspp_parser</b>	
A Bison parser	113
Hold	121
HypothesisTester_if	124
Leave	130
LicenceManager	133
List< T >	134
<b>yy::location</b>	
Two points in a source file	135
LSODE	136
Match	142
Model	144

<a href="#">ModelChecker_if</a>	146
<a href="#">ModelComponent</a>	151
<a href="#">ModelElement</a>	152
<a href="#">ModelInfo</a>	155
<a href="#">ModelPersistence_if</a>	157
<a href="#">ModelSimulation</a>	158
<a href="#">OnEventManager</a>	164
<a href="#">PickStation</a>	173
<a href="#">PickUp</a>	175
<a href="#">Plugin</a>	176
<a href="#">yy::position</a>	
A point in a source file	181
<a href="#">Process</a>	183
<a href="#">Queue</a>	186
<a href="#">Record</a>	189
<a href="#">Release</a>	192
<a href="#">Remove</a>	194
<a href="#">Resource</a>	195
<a href="#">Sampler_if::RNG_Parameters</a>	199
<a href="#">Route</a>	199
<a href="#">Sampler_if</a>	202
<a href="#">Schedule</a>	208
<a href="#">Search</a>	210
<a href="#">Seize</a>	212
<a href="#">yy::genesyspp_parser::semantic_type</a>	215
<a href="#">Separate</a>	217
<a href="#">Sequence</a>	220
<a href="#">Set</a>	223
<a href="#">Signal</a>	226
<a href="#">SimulationControl</a>	227
<a href="#">SimulationEvent</a>	229
<a href="#">SimulationReporterDefaultImpl1</a>	231

<a href="#">SimulationResponse</a>	232
<a href="#">SimulationScenario</a>	234
<a href="#">Simulator</a>	235
<a href="#">SinkModelComponent</a>	236
<a href="#">yy::genesyspp_parser::stack&lt; T, S &gt;::slice</a> Present a slice of the top of a stack	237
<a href="#">Solver_if</a>	264
<a href="#">SourceModelComponent</a>	268
<a href="#">Start</a>	270
<a href="#">Statistics_if</a>	274
<a href="#">StatisticsCollector</a>	275
<a href="#">Stop</a>	283
<a href="#">Store</a>	288
<a href="#">Submodel</a>	289
<a href="#">yy::genesyspp_parser::symbol_kind</a> Symbol kinds	291
<a href="#">yy::genesyspp_parser::symbol_type</a> "External" symbols: returned by the scanner	292
<a href="#">yy::genesyspp_parser::syntax_error</a> Syntax errors thrown from user actions	294
<a href="#">yy::genesyspp_parser::token</a> Token kinds	295
<a href="#">TraceManager</a>	298
<a href="#">TraceSimulationProcess</a>	301
<a href="#">Traits&lt; GenesysApplication_if &gt;</a>	302
<a href="#">Traits&lt; HypothesisTester_if &gt;</a>	302
<a href="#">Traits&lt; Solver_if &gt;</a>	303
<a href="#">Unstore</a>	315
<a href="#">Variable</a>	319
<a href="#">Write</a>	323

## 3 File Index

### 3.1 File List

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

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/.dep.inc	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/main.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/moc_dialogabout.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/moc_dialogmodelinformation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/moc_formPlugins.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/moc_mainwindow.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/moc_predefs.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/qrc_guiresources.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/Traits.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/ui_dialogabout.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/ui_dialogmodelinformation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/ui_formPlugins.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/ui_mainwindow.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/BaseConsoleGenesysApplication.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/BaseConsoleGenesysApplication.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/GenesysApplication_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/gui/qt/GenesysQtGUI.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/gui/qt/GenesysQtGUI.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/gui/qt/mapkernelandui.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/gui/qt/mapkernelandui.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/applications/gui/qt/dialogs/dialogabout.cpp	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/dialogs/dialogabout.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/dialogs/dialogmodelinformation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/dialogs/dialogmodelinformation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/dialogs/ui_dialogabout.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/dialogs/ui_dialogmodelinformation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/forms/formPlugins.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/forms/formPlugins.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/forms/ui_formPlugins.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/mainWindow/mainwindow.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/mainWindow/mainwindow.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/gui/qt/mainWindow/ui_mainwindow.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/terminal/GenesysShell_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/terminal/GenesysTerminalApp.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/applications/terminal/GenesysTerminalApp.h	??
html/dynsections.js	??
html/jquery.js	??
html/menu.js	??
html/menudata.js	??
html/search/search.js	??
html/search/searchdata.js	??
user/html/dynsections.js	??
user/html/jquery.js	??
user/html/menu.js	??
user/html/menudata.js	??

user/html/navtree.js	??
user/html/navtreedata.js	??
user/html/navtreeindex0.js	??
user/html/resize.js	??
user/html/search/search.js	??
user/html/search/searchdata.js	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/book/Book_Cap02_Example01.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/book/Book_Cap02_Example01.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_AssignWriteSeizes.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_AssignWriteSeizes.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Delay.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Delay.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Dummy.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Dummy.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_HoldSignal.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_HoldSignal.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_ModelInfoModelSimulation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_ModelInfoModelSimulation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_OnEvent.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_OnEvent.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Parser.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Parser.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_ParserModelFunctions.cpp	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_ParserModelFunctions.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Process.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Process.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_ProcessSet.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_ProcessSet.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_RouteStation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_RouteStation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_SeizeDelayRelease.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_SeizeDelayRelease.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_SeizeDelayReleaseMany.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_SeizeDelayReleaseMany.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Sequence.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/smarts/Smart_Sequence.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/AnElectronicAssemblyAndTestSystem.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/AnElectronicAssemblyAndTestSystem.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/FullSimulationOfComplexModel.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/FullSimulationOfComplexModel.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/OperatingSystem02.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/OperatingSystem02.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/OperatingSystem03.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/examples/teaching/OperatingSystem03.h	??



/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/TraitsKernel.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Attribute.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Attribute.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ComponentManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ComponentManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ConnectionManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ConnectionManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Counter.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Counter.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/DefineGetterSetter.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ElementManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ElementManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ElementManager_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Entity.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Entity.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/EntityType.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/EntityType.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Event.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/Event.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ExperimentManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/kernel/simulator/ExperimentManager.h	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ExperimentManagerDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ExperimentManagerDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ExperimetManager_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/LicenceManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/LicenceManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/Model.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/Model.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelChecker_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelCheckerDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelCheckerDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelComponent.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelComponent.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelElement.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelElement.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelInfo.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelInfo.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelPersistence_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelPersistenceDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelPersistenceDefaultImpl1.h	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelSimulation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ModelSimulation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/OnEventManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/OnEventManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/Parser_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserChangesInformation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserChangesInformation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserDefaultImpl2.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserDefaultImpl2.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ParserManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PersistentObject_base.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/Plugin.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/Plugin.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PluginConnector_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PluginConnectorDummyImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PluginConnectorDummyImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PluginInformation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PluginInformation.h	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PluginManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/PluginManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/ScenarioExperiment_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationControl.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationControl.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationExperiment.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationExperiment.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationReporter_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationReporterDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationReporterDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationResponse.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationResponse.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationScenario.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SimulationScenario.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/Simulator.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/Simulator.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SinkModelComponent.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SinkModelComponent.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SourceModelComponent.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/SourceModelComponent.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/StatisticsCollector.cpp	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/StatisticsCollector.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/TraceManager.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/TraceManager.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/Genesys++-driver.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/Genesys++-driver.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/Genesys++-scanner.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/GenesysParser.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/ <a href="#">GenesysParser.h</a>	330
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/ <a href="#">location.hh</a>	331
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/obj_t.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/obj_t.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/position.hh	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/simulator/parserBisonFlex/stack.hh	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/Collector_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/CollectorDatafile_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/CollectorDatafileDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/CollectorDatafileDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/CollectorDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/CollectorDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/Sampler_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/SamplerBoostImpl.cpp	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/SamplerBoostImpl.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/SamplerDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/SamplerDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/SorttFile.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/SorttFile.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/Statistics_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/StatisticsDataFile_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/StatisticsDataFileDefaultImpl.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/StatisticsDataFileDefaultImpl.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/StatisticsDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/statistics/StatisticsDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/util/Exact.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/util/List.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/util/Util.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/kernel/util/Util.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/nbproject/private/c_standard_headers_indexer.c	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/nbproject/private/cpp_standard_headers_indexer.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Access.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Access.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Assign.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Assign.h	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Batch.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Batch.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/CellularAutomata.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/CellularAutomata.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Create.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Create.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Decide.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Decide.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Delay.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Delay.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Dispose.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Dispose.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/DropOff.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/DropOff.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Dummy.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Dummy.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Enter.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Enter.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Exit.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Exit.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Hold.cpp	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Hold.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Leave.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Leave.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/LSODE.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/LSODE.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/MarkovChain.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/MarkovChain.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Match.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Match.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/OLD_ODEelement.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/OLD_ODEelement.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/PickStation.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/PickStation.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/PickUp.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/PickUp.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Process.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Process.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/QueueableItem.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/QueueableItem.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Record.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Record.h	??



/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Release.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Release.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Remove.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Remove.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Route.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Route.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Search.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Search.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/SeizableItem.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/SeizableItem.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Seize.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Seize.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Separate.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Separate.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Signal.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Signal.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Start.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Start.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Stop.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Stop.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/components/Store.cpp	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Store.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Submodel.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Submodel.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Unstore.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Unstore.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Write.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/components/Write.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/EntityGroup.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/EntityGroup.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Failure.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Failure.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/File.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/File.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Formula.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Formula.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Queue.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Queue.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Resource.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Resource.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Schedule.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↔ S/2019_2022_GenESyS/plugins/elements/Schedule.h	??

/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Sequence.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Sequence.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Set.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Set.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Station.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Station.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Storage.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Storage.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Variable.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/plugins/elements/Variable.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tests/testProbabilityDistribution.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/HypothesisTester_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/HypothesisTesterDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/HypothesisTesterDefaultImpl1.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/ProbabilityDistribution.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/ProbabilityDistribution.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/solver_if.h	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/SolverDefaultImpl1.cpp	??
/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESy↵ S/2019_2022_GenESyS/tools/SolverDefaultImpl1.h	??

## 4 Class Documentation

### 4.1 Access Class Reference

```
#include <Access.h>
```

#### Public Member Functions

- **Access** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

#### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

#### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

#### Additional Inherited Members

##### 4.1.1 Detailed Description

[Access](#) module DESCRIPTION The [Access](#) module allocates one or more cells of a conveyor to an entity for movement from one station to another. Once the entity has control of the cells on the conveyor, it may then be conveyed to the next station. When an entity arrives at an [Access](#) module, it will wait until the appropriate number of contiguous cells on the conveyor are empty and aligned with the entity's station location. TYPICAL USES Parts accessing a conveyor to be sent to a paint booth Glass accessing a conveyor to be transferred to a cutting station PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart. Conveyor Name Name of the conveyor that the entity desires.

#### of Cells Number of contiguous conveyor cells the entity requires for

movement on the conveyor. [Queue](#) Type Determines the type of queue used to hold the entities, either an individual [Queue](#), a queue [Set](#), and Internal queue or an [Attribute](#) or Expression that evaluate to the queue name. [Queue](#) Name Name of the queue that will hold the entity until it accesses the conveyor. [Set](#) Name Name of the set of queues. [Set](#) Index Defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3. [Attribute](#) Name Defines the name of the attribute that stores the queue name to which entities will reside. Expression Defines the name of the expression that stores the queue name to which entities will reside.

Definition at line 53 of file Access.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Access.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Access.cpp

## 4.2 Assign Class Reference

```
#include <Assign.h>
```

### Classes

- class [Assignment](#)

### Public Member Functions

- **Assign** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- [List](#)< [Assignment](#) \* > \* **getAssignments** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.2.1 Detailed Description

[Assign](#) module DESCRIPTION This module is used for assigning new values to variables, entity attributes, entity types, entity pictures, or other system variables. Multiple assignments can be made with a single [Assign](#) module. TYPICAL USES Accumulate the number of subassemblies added to a part Change an entity's type to represent the customer copy of a multi-page form Establish a customer's priority PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Assignments Specifies the one or more assignments that will be made when an entity executes the module. Type Type of assignment to be made. Other can include system variables, such as resource capacity or simulation end time. [Variable](#) Name Name of the variable that will be assigned a new value when an entity enters the module. Applies only when Type is [Variable](#), [Variable](#) Array (1D), or [Variable](#) Array (2D). Row Specifies the row index for a variable array. Column Specifies the column index for a variable array. [Attribute](#) Name Name of the entity attribute that will be assigned a new value when the entity enters the module. Applies only when Type is [Attribute](#). [Entity](#) Type New entity type that will be assigned to the entity when the entity enters the module. Applies only when Type is [Entity](#) Type. [Entity](#) Picture New entity picture that will be assigned to the entity when the entity enters the module. Applies only when Type is [Entity](#) Picture. Other Identifies the special system variable that will be assigned a new value when an entity enters the module. Applies only when Type is Other. New Value [Assignment](#) value of the attribute, variable, or other system variable. Does not apply when Type is [Entity](#) Type or [Entity](#) Picture.

Definition at line 58 of file Assign.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Assign.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Assign.cpp

## 4.3 Assign::Assignment Class Reference

```
#include <Assign.h>
```

### Public Member Functions

- **Assignment** (std::string destination, std::string expression)
- void **setDestination** (std::string \_destination)
- std::string **getDestination** () const
- void **setExpression** (std::string \_expression)
- std::string **getExpression** () const

#### 4.3.1 Detailed Description

While the assign class allows you to perform multiple assignments, the assignment class defines an assignment itself.

Definition at line 64 of file Assign.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Assign.h

## 4.4 Attribute Class Reference

```
#include <Attribute.h>
```

### Public Member Functions

- **Attribute** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.4.1 Detailed Description

**Attribute** module DESCRIPTION This data module is used to define an attribute's dimension, data type and initial value(s). An attribute is a characteristic of all entities created, but with a specific value that can differ from one entity to another. Attributes can be referenced in other modules (for example, the **Decide** module), can be reassigned a new value with the **Assign** module, and can be used in any expression. **Attribute** values are unique for each entity, as compared to Variables which are global to the simulation module. There are three methods for manually editing the Initial Values of an **Attribute** module: Using the standard spreadsheet interface. In the module spreadsheet, rightclick on the Initial Values cell and select the Edit via spreadsheet menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value. Using the module dialog box. In the module spreadsheet, right-click on any cell and select the Edit via dialog menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value. Using the two-dimensional (2-D) spreadsheet interface. In the module spreadsheet, click on the Initial Values cell. TYPICAL USES Due date of an order (entity) Priority of an order (entity) Color of a part (entity) PROMPTS Prompt Description Name The unique name of the attribute being defined. Rows Number of rows in a one- or two-dimensional attribute. Columns Number of columns in a two-dimensional attribute. Data Type The data type of the values stored in the attribute. Valid types are Real and String. The default type is Real. Initial Values Lists the initial value or values of the attribute. You can assign new values to the attribute by using the **Assign** module. Initial Value **Entity** attribute value when entity is created and enters the system.

Definition at line 63 of file Attribute.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/Attribute.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/Attribute.cpp

## 4.5 yy::genesyspp\_parser::basic\_symbol< Base > Struct Template Reference

```
#include <GenesysParser.h>
```

### Public Types

- typedef Base **super\_type**  
*Alias to Base.*

### Public Member Functions

- **basic\_symbol** ()  
*Default constructor.*
- **basic\_symbol** (const **basic\_symbol** &that)  
*Copy constructor.*
- **basic\_symbol** (typename Base::kind\_type t, const **location\_type** &l)  
*Constructors for typed symbols.*
- **basic\_symbol** (typename Base::kind\_type t, const obj\_t &v, const **location\_type** &l)
- ~**basic\_symbol** ()

- *Destroy the symbol.*  
void `clear` () YY\_NOEXCEPT
- *Destroy contents, and record that is empty.*  
std::string `name` () const YY\_NOEXCEPT
- *The user-facing name of this symbol.*  
symbol\_kind\_type `type_get` () const YY\_NOEXCEPT
- *Backward compatibility (Bison 3.6).*  
bool `empty` () const YY\_NOEXCEPT
- *Whether empty.*  
void `move` (basic\_symbol &s)
- *Destructive move, s is emptied into this.*

## Public Attributes

- semantic\_type value  
*The semantic value.*
- location\_type location  
*The location.*

### 4.5.1 Detailed Description

```
template<typename Base>
struct yy::genesyspp_parser::basic_symbol< Base >
```

A complete symbol.

Expects its Base type to provide access to the symbol kind via kind ().

Provide access to semantic value and location.

Definition at line 805 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)

## 4.6 Batch Class Reference

```
#include <Batch.h>
```

### Public Member Functions

- **Batch** (Model \*model, std::string name="")
- virtual std::string **show** ()



### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static ModelComponent \* **LoadInstance** (Model \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** (Entity \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.6.1 Detailed Description

**Batch** module DESCRIPTION This module is intended as the grouping mechanism within the simulation model. Batches can be permanently or temporarily grouped. Temporary batches must later be split using the **Separate** module. Batches may be made with any specified number of entering entities or may be matched together based on an attribute. Entities arriving at the **Batch** module are placed in a queue until the required number of entities has accumulated. Once accumulated, a new representative entity is created. TYPICAL USES Collect a number of parts before starting processing Reassemble previously separated copies of a form Bring together a patient and his record before commencing an appointment PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Type Method of batching entities together. **Batch** Size Number of entities to be batched. Save Criterion Method for assigning representative entity's user-defined attribute values. Rule Determines how incoming entities will be batched. Any **Entity** will take the first "Batch Size" number of entities and put them together. By **Attribute** signifies that the values of the specified attribute must match for entities to be grouped. For example, if **Attribute** Name is Color, all entities must have the same Color value to be grouped; otherwise, they will wait at the module for additional incoming entities. **Attribute** Name Name of the attribute whose value must match the value of the other incoming entities in order for a group to be made. Applies only when Rule is By **Attribute**. Representative **Entity** The entity type for the representative entity.

Definition at line 53 of file Batch.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Batch.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Batch.cpp

## 4.7 yy::genesyspp\_parser::by\_kind Struct Reference

Type access provider for token (enum) based symbols.

```
#include <GenesysParser.h>
```

### Public Types

- typedef token\_kind\_type kind\_type  
*The symbol kind as needed by the constructor.*

## Public Member Functions

- [by\\_kind](#) ()  
*Default constructor.*
- [by\\_kind](#) (const [by\\_kind](#) &that)  
*Copy constructor.*
- [by\\_kind](#) ([kind\\_type](#) t)  
*Constructor from (external) token numbers.*
- void [clear](#) () YY\_NOEXCEPT  
*Record that this symbol is empty.*
- void [move](#) ([by\\_kind](#) &that)  
*Steal the symbol kind from that.*
- [symbol\\_kind\\_type](#) [kind](#) () const YY\_NOEXCEPT
- [symbol\\_kind\\_type](#) [type\\_get](#) () const YY\_NOEXCEPT  
*Backward compatibility (Bison 3.6).*

## Public Attributes

- [symbol\\_kind\\_type](#) [kind\\_](#)

### 4.7.1 Detailed Description

Definition at line 1110 of file GenesysParser.h.

### 4.7.2 Member Function Documentation

**4.7.2.1** [kind\(\)](#) [genesyspp\\_parser::symbol\\_kind\\_type](#) [yy::genesyspp\\_parser::by\\_kind::kind](#) ( ) const

The (internal) type number (corresponding to *type*). *empty* when empty.

Definition at line 3227 of file GenesysParser.h.

### 4.7.3 Member Data Documentation

**4.7.3.1** [kind\\_](#) [symbol\\_kind\\_type](#) [yy::genesyspp\\_parser::by\\_kind::kind\\_](#)

The symbol kind. *S\_YYEMPTY* when empty.

Definition at line 1144 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)

## 4.8 Collector\_if Class Reference

```
#include <Collector_if.h>
```

### Public Member Functions

- virtual void **clear** ()=0
- virtual void **addValue** (double value)=0
- virtual double **getLastValue** ()=0
- virtual unsigned long **numElements** ()=0
- virtual void **setAddValueHandler** (CollectorAddValueHandler addValueHandler)=0
- virtual void **setClearHandler** (CollectorClearHandler clearHandler)=0

### 4.8.1 Detailed Description

Interface for collecting values of a single stochastic variable. Values collected can be used as base for statistical analysis.

Definition at line 37 of file Collector\_if.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/statistics/Collector\_if.h

## 4.9 CollectorDatafile\_if Class Reference

```
#include <CollectorDatafile_if.h>
```

### Public Member Functions

- virtual double **getValue** (unsigned int rank)=0
- virtual void **seekFirstValue** ()=0
- virtual double **getNextValue** ()=0
- virtual std::string **getDataFilename** ()=0
- virtual void **setDataFilename** (std::string filename)=0

### 4.9.1 Detailed Description

Interface for collecting values of a stochastic variable that will be stores in a datafile.

Definition at line 22 of file CollectorDatafile\_if.h.

### 4.9.2 Member Function Documentation

**4.9.2.1 getNextValue()** `virtual double CollectorDatafile_if::getNextValue ( ) [pure virtual]`

Get the next value in the file and advances the pointer

**4.9.2.2 getValue()** `virtual double CollectorDatafile_if::getValue ( unsigned int rank ) [pure virtual]`

Get a value from a specific position

**4.9.2.3 seekFirstValue()** `virtual void CollectorDatafile_if::seekFirstValue ( ) [pure virtual]`

[Set](#) the pointer to the first value in the file

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵  
2022_GenESyS/kernel/statistics/CollectorDatafile_if.h`

## 4.10 ComponentManager Class Reference

```
#include <ComponentManager.h>
```

### Public Member Functions

- **ComponentManager** ([Model](#) \*model)
- `bool insert` ([ModelComponent](#) \*comp)
- `void remove` ([ModelComponent](#) \*comp)
- [ModelComponent](#) \* `find` (std::string name)
- `void clear` ()
- `unsigned int getNumberOfComponents` ()
- `std::list< ModelComponent * >::iterator begin` ()
- `std::list< ModelComponent * >::iterator end` ()
- [ModelComponent](#) \* `front` ()
- [ModelComponent](#) \* `next` ()
- `bool hasChanged` () const
- `void setHasChanged` (bool \_hasChanged)

### 4.10.1 Detailed Description

[ComponentManager](#) allows to insert, access, find, remove and over Models

Definition at line 25 of file `ComponentManager.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵  
2022_GenESyS/kernel/simulator/ComponentManager.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵  
2022_GenESyS/kernel/simulator/ComponentManager.cpp`

## 4.11 ConnectionManager Class Reference

```
#include <ConnectionManager.h>
```

### Public Member Functions

- unsigned int **size** ()
- [ModelComponent](#) \* **front** ()
- [ModelComponent](#) \* **atRank** (unsigned int rank)
- Connection \* **getFrontConnection** ()
- Connection \* **getConnectionAtRank** (unsigned int rank)
- void **insert** ([ModelComponent](#) \*component, unsigned int inputNumber=0)
- void **insert** (Connection \*connection)
- void **insertAtRank** (unsigned int rank, Connection \*connection)
- std::list< Connection \* > \* **list** () const
- unsigned int **getCurrentOutputConnections** () const
- void **setMaxOutputConnections** (unsigned int \_maxOutputConnections)
- unsigned int **getMaxOutputConnections** () const
- void **setMinOutputConnections** (unsigned int \_minOutputConnections)
- unsigned int **getMinOutputConnections** () const
- unsigned int **getCurrentInputConnections** () const
- void **setMaxInputConnections** (unsigned int \_maxInputConnections)
- unsigned int **getMaxInputConnections** () const
- void **setMinInputConnections** (unsigned int \_minInputConnections)
- unsigned int **getMinInputConnections** () const

### 4.11.1 Detailed Description

[ConnectionManager](#) defines how a [ModelComponent](#) is output connected to none, one or more following [ModelComponent](#)s. It has a list of nextConnections, where each Connection is a pair, defining the next [ModelComponent](#) and an input port on that component (usefull only if the next component has more than one input). The number of the output connection is its rank in the nextConnections list. Min and max number of input and output connectons can be defined.

Definition at line 29 of file ConnectionManager.h.

### 4.11.2 Member Function Documentation

**4.11.2.1 atRank()** [ModelComponent](#) \* [ConnectionManager::atRank](#) ( unsigned int rank )

DEPRECTED. Use getConnectionAtRank instead

Definition at line 29 of file ConnectionManager.cpp.

#### 4.11.2.2 front() `ModelComponent * ConnectionManager::front ( )`

DEPRECTED. Use frontConnection instead

Definition at line 25 of file ConnectionManager.cpp.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/ConnectionManager.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/ConnectionManager.cpp

## 4.12 Counter Class Reference

```
#include <Counter.h>
```

### Public Member Functions

- **Counter** (`Model *model`, `std::string name=""`, `ModelElement *parent=nullptr`)
- virtual `std::string show ( )`
- void **clear** ( )
- void **incCountValue** (`double value=1.0`)
- double **getCountValue** ( ) const
- `ModelElement * getParent` ( ) const

### Static Public Member Functions

- static `PluginInformation * GetPluginInformation` ( )
- static `ModelElement * LoadInstance` (`Model *model`, `std::map< std::string, std::string > *fields`)

### Protected Member Functions

- virtual bool **\_loadInstance** (`std::map< std::string, std::string > *fields`)
- virtual `std::map< std::string, std::string > * _saveInstance` ( )
- virtual bool **\_check** (`std::string *errorMessage`)
- virtual void **\_initBetweenReplications** ( )

### Additional Inherited Members

#### 4.12.1 Detailed Description

The `Counter` element is used to count events, and its internal count value is added by a configurable amount, usually incremented by one.

Definition at line 25 of file Counter.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/Counter.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/Counter.cpp

## 4.13 Create Class Reference

```
#include <Create.h>
```

### Public Member Functions

- **Create** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

### Additional Inherited Members

#### 4.13.1 Detailed Description

[Create](#) is the most basic component to include the first entities into the model, and therefore is a source component (derived from [SourceModelComponent](#)) [Create](#) module DESCRIPTION This module is intended as the starting point for entities in a simulation model. Entities are created using a schedule or based on a time between arrivals. Entities then leave the module to begin processing through the system. The entity type is specified in this module. TYPICAL USES The start of a part's production in a manufacturing line A document's arrival (for example, order, check, application) into a business process A customer's arrival at a service process (for example, retail store, restaurant, information desk) PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. [Entity](#) Type Name of the entity type to be generated. Type Type of arrival stream to be generated. Types include Random (uses an exponential distribution, user specifies mean), [Schedule](#) (uses an exponential distribution, mean determined from the specified [Schedule](#) module), Constant (user specifies constant value; for example, 100), or Expression (drop-down list of various distributions). Value Determines the mean of the exponential distribution (if Random is used) or the constant value (if Constant is used) for the time between arrivals. Applies only when Type is Random or Constant. [Schedule](#) Name Identifies the name of the schedule to be used. The schedule defines the arrival pattern for entities arriving to the system. Applies only when Type is [Schedule](#). Expression Any distribution or value specifying the time between arrivals. Applies only when Type is Expression. Units Time units used for interarrival and first creation times. Does not apply when Type is [Schedule](#). Entities per Arrival Number of entities that will enter the system at a given time with each arrival. Max Arrivals Maximum number of entities that this module will generate. When this value is reached, the creation of new entities by this module ceases. First Creation Starting time for the first entity to arrive into the system. Does not apply when Type is [Schedule](#).

Definition at line 67 of file Create.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Create.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Create.cpp

## 4.14 Decide Class Reference

```
#include <Decide.h>
```

### Public Member Functions

- **Decide** ([Model](#) \*model, std::string name="")
- [List](#)< std::string > \* **getConditions** () const
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

### Additional Inherited Members

#### 4.14.1 Detailed Description

[Decide](#) module DESCRIPTION This module allows for decision-making processes in the system. It includes options to make decisions based on one or more conditions (for example, if entity type is Gold Card) or based on one or more probabilities (for example, 75%, true; 25%, false). Conditions can be based on attribute values (for example, Priority), variable values (for example, Number Denied), the entity type, or an expression (for example, NQ(ProcessA.Queue)). There are two exit points out of the [Decide](#) module when its specified type is either 2-way by Chance or 2-way by Condition. There is one exit point for "true" entities and one for "false" entities. When the N-way by Chance or by Condition type is specified, multiple exit points are shown for each condition or probability and a single "else" exit. The number of entities that exit from each type (true/false) is displayed for 2-way by Chance or by Condition modules only. TYPICAL USES Dispatching a faulty part for rework Branching accepted vs. rejected checks Sending priority customers to a dedicated process Prompt Description Name Unique module identifier displayed on the module shape. Type Indicates whether the decision is based on a condition (if X>Y) or by chance/percentage (for example, 60%, yes; 40%, no). The type can be specified as either 2-way or N-way. 2-way allows for one condition or probability (plus the "false" exit). N-way allows for any number of conditions or probabilities to be specified as well as an "else" exit. Conditions Defines one or more conditions used to direct entities to different modules. Applies only when Type is N-way by Condition. Percentages Defines one or more percentages used to direct entities to different modules. Applies only when Type is N-way by Chance. Percent True Value that will be checked to determine the percentage of entities sent out a given True exit. If Types of conditions that are available for evaluation: [Variable](#), [Variable](#) Array (1D), [Variable](#) Array (2D), [Attribute](#), [Entity](#) Type, Expression. Named Specifies the name of the variable, attribute, or entity type that will be evaluated when an entity enters



the module. Does not apply when Type is Expression. Is Evaluator for the condition. Applies only to [Attribute](#) and [Variable](#) conditions. Row Specifies the row index for a variable array. Applies only when Type is N-way by Condition or 2-way by Condition and [Variable](#) is Array 1-D or Array 2-D. Column Specifies the column index for a variable array. Applies only when Type is N-way by Condition or 2-way by Condition and [Variable](#) is Array 1-D or Array 2-D. Value Expression that will be either compared to an attribute or variable or that will be evaluated as a single expression to determine if it is true or false. Does not apply to [Entity](#) Type condition. If Type is Expression, this value must also include the evaluator (for example, Color<>Red).

Definition at line 74 of file Decide.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Decide.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Decide.cpp

## 4.15 Delay Class Reference

```
#include <Delay.h>
```

### Public Member Functions

- **Delay** ([Model](#) \*model, std::string name="")
- void **setDelayExpression** (std::string \_delayExpression)
- std::string **delayExpression** () const
- void **setDelay** (double delay)
- double **delay** () const
- void **setDelayTimeUnit** (Util::TimeUnit \_delayTimeUnit)
- Util::TimeUnit **delayTimeUnit** () const
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

## Additional Inherited Members

### 4.15.1 Detailed Description

**Delay** module DESCRIPTION The **Delay** module delays an entity by a specified amount of time. When an entity arrives at a **Delay** module, the time delay expression is evaluated and the entity remains in the module for the resulting time period. The time is then allocated to the entity's value-added, non-value added, transfer, wait, or other time. Associated costs are calculated and allocated as well. TYPICAL USES Processing a check at a bank Performing a setup on a machine Transferring a document to another department PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Allocation Type of category to which the entity's incurred delay time and cost will be added. **Delay** Time Determines the value of the delay for the entity. Units Time units used for the delay time.

Definition at line 41 of file Delay.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Delay.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Delay.cpp

## 4.16 Dispose Class Reference

```
#include <Dispose.h>
```

### Public Member Functions

- **Dispose** (**Model** \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static **ModelComponent** \* **LoadInstance** (**Model** \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** (**Entity** \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

## Additional Inherited Members

### 4.16.1 Detailed Description

**Dispose** module DESCRIPTION This module is intended as the ending point for entities in a simulation model. **Entity** statistics may be recorded before the entity is disposed of. TYPICAL USES Parts leaving the modeled facility The termination of a business process Customers departing from the store Prompt Description Name Unique module identifier displayed on the module shape. **Record Entity** Statistics Determines whether or not the incoming entity's statistics will be recorded. Statistics include value-added time, non-value-added time, wait time, transfer time, other time, total time, value-added cost, non-value-added cost, wait cost, transfer cost, other cost, and total cost.

Definition at line 38 of file Dispose.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Dispose.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Dispose.cpp

## 4.17 DropOff Class Reference

```
#include <DropOff.h>
```

### Public Member Functions

- **DropOff** (**Model** \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static **ModelComponent** \* **LoadInstance** (**Model** \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** (**Entity** \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.17.1 Detailed Description

**Dropoff module DESCRIPTION** The Dropoff module removes a specified number of entities from the entity's group and sends them to another module, as specified by a graphical connection. Group user-defined attribute value and internal attributes may be given to the dropped-off entities based on a specified rule. **TYPICAL USES** Loading shelves with product Separating a form for use in various departments **PROMPTS** Prompt Description Name Unique module identifier displayed on the module shape. Quantity Number of entities that will be dropped off from an incoming representative grouped entity. Starting Rank Starting rank of the entities to be dropped off, based on the entities in the group. Member Attributes Method of determining how to assign the representative entity attribute values (other than costs/times) to the dropped-off original entities. **Attribute** Name Name of representative entity attribute(s) assigned to droppedoff original entities of the group

Definition at line 41 of file DropOff.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/DropOff.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/DropOff.cpp

## 4.18 Dummy Class Reference

```
#include <Dummy.h>
```

### Public Member Functions

- **Dummy** (**Model** \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static **ModelComponent** \* **LoadInstance** (**Model** \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** (**Entity** \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.18.1 Detailed Description

This component ...

Definition at line 22 of file Dummy.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Dummy.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Dummy.cpp

## 4.19 ElementManager Class Reference

```
#include <ElementManager.h>
```

### Public Member Functions

- **ElementManager** ([Model](#) \*model)
- bool **insert** ([ModelElement](#) \*anElement)
- void **remove** ([ModelElement](#) \*anElement)  
*Deprecated.*
- bool **insert** (std::string elementTypename, [ModelElement](#) \*anElement)  
*Deprecated.*
- void **remove** (std::string elementTypename, [ModelElement](#) \*anElement)
- bool **check** (std::string elementTypename, [ModelElement](#) \*anElement, std::string expressionName, std::string \*errorMessage)
- bool **check** (std::string elementTypename, std::string elementName, std::string expressionName, bool mandatory, std::string \*errorMessage)
- void **clear** ()
- [ModelElement](#) \* **getElement** (std::string elementTypename, Util::identification id)
- [ModelElement](#) \* **getElement** (std::string elementTypename, std::string name)
- unsigned int **getNumberOfElements** (std::string elementTypename)
- unsigned int **getNumberOfElements** ()
- int **getRankOf** (std::string elementTypename, std::string name)  
*returns the position (1st position=0) of the element if found, or negative value if not found*
- std::list< std::string > \* **getElementClassnames** () const
- [List](#)< [ModelElement](#) \* > \* **getElementList** (std::string elementTypename) const
- void **show** ()
- [Model](#) \* **getParentModel** () const
- bool **hasChanged** () const
- void **setHasChanged** (bool \_hasChanged)

### 4.19.1 Detailed Description

The [ElementManager](#) is responsible for inserting and removing elements ([ModelElement](#)) used by components, in a consistent way. TO FIX: No direct access for insertion or deletion should be allow

Definition at line 30 of file ElementManager.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/ElementManager.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/ElementManager.cpp

## 4.20 Enter Class Reference

```
#include <Enter.h>
```

### Public Member Functions

- **Enter** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- void **setStation** (Station \*\_station)
- Station \* **getStation** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

## Additional Inherited Members

### 4.20.1 Detailed Description

**Enter** module DESCRIPTION The **Enter** module defines a station (or a set of stations) corresponding to a physical or logical location where processing occurs. When an entity arrives at an **Enter** module, an unloading delay may occur and any transfer device used to transfer the entity to the **Enter** module's station may be released. The station (or each station within the defined set) has a matching Activity Area that is used to report all times and costs accrued by the entities in this station. This Activity Area's name is the same as the station. If a parent Activity Area is defined, then it also accrues any times and costs by the entities in this station. TYPICAL USES The start of a part's production in a series of parallel processes where the part's forklift needs to be released The start of a document's processing after the document has been created where the mail clerk resource needs to be released PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart Station Type Type of station, either a single Station or station **Set**. Station Name Name of the individual station. A given station can only exist once within a model. Parent Activity Area Name of the Activity Area's parent. Associated Intersection Name of the intersection associated with this station in a guided transporter network. Report Statistics Specifies whether or not statistics will automatically be collected and stored in the report database for this station and its corresponding activity area. **Set** Name Name of the station set. A given station set can only exist once within a model. Save **Attribute** Specifies the attribute to be used to store the index into the station set for an entity entering this module. **Set** Members This repeat group permits you to define the individual stations that are to be members of the specified station set. A station set must have at least one member station. Active when Station Type is **Set**. Station Name This field indicates the name of a station that is to be a member of this station set. A given station can only exist within a model once. Therefore, an individual station can only be the member of one station set, and that individual station may not be the name of a station in another module. Parent Activity Area Name of the Activity Area's parent for the station set member. Associated Intersection Name of the intersection associated with this station set in a guided transporter network. Report Statistics Specifies whether or not statistics will automatically be collected and stored in the report database for this station set member and its corresponding activity area. Allocation Type of category to which the entity's incurred delay time and cost will be added. **Delay** This field defines the delay that will be experienced by entities immediately upon arrival at the station. Units Time units used for the delay time. Transfer In If a resource, transporter, or conveyor was used to transfer the entity to this station, this can be used to release, free, or exit the device. If **Release Resource** is selected, the specified resource is released. If Free Transporter is selected, the specified transporter is freed. If **Exit Conveyor** is selected, the specified conveyor is exited. Transporter Name Name of the transporter to be freed upon arrival to the station. Active when Transfer Name is Free Transporter. Unit Number Unit number of the transporter if the transporter is multicapacity. Conveyor Name Name of the conveyor to exit upon arrival to the station. **Resource** Type Type of allocation, either single **Resource** or resource **Set**. **Resource** Name Name of the resource to release. Active when Transfer Name is **Release Resource**. **Set** Name Name of the resource set from which the resource is to be released. **Release** Rule Determines which member of the set is to be released, either the Last Member Seized, First Member Seized, or Specific Member. **Set** Index Index into the set that determines which member of the set is to be released. **Attribute** Name Name of the attribute that determines the instance number of the resource to release. Expression Expression value that determines the instance number of the resource to release.

Definition at line 99 of file Enter.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Enter.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Enter.cpp

## 4.21 Entity Class Reference

```
#include <Entity.h>
```

## Public Member Functions

- virtual std::string **show** ()
- void **setEntityTypeName** (std::string entityTypeName) throw ()
- std::string **getEntityTypeName** () const
- void **setEntityType** (EntityType \*entityType)
- EntityType \* **getEntityType** () const
- double **getAttributeValue** (std::string attributeName)
- double **getAttributeValue** (std::string index, std::string attributeName)
- double **getAttributeValue** (Util::identification attributeID)
- double **getAttributeValue** (std::string index, Util::identification attributeID)
- void **setAttributeValue** (std::string attributeName, double value)
- void **setAttributeValue** (std::string index, std::string attributeName, double value)
- void **setAttributeValue** (Util::identification attributeID, double value)
- void **setAttributeValue** (std::string index, Util::identification attributeID, double value)
- Util::identification **entityNumber** () const

## Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Friends

- class **Model**

## Additional Inherited Members

### 4.21.1 Detailed Description

**Entity** module DESCRIPTION This data module defines the various entity types and their initial picture values in a simulation. Initial costing information and holding costs are also defined for the entity. TYPICAL USES Items being produced or assembled (parts, pallets) Documents (forms, e-mails, faxes, reports) People moving through a process (customers, callers) PROMPTS Prompt Description Name The unique name of the attribute being defined. Rows Number of rows in a one- or two-dimensional attribute. Columns Number of columns in a two-dimensional attribute. Data Type The data type of the values stored in the attribute. Valid types are Real and String. The default type is Real. Initial Values Lists the initial value or values of the attribute. You can assign new values to the attribute by using the [Assign](#) module. Initial Value **Entity** attribute value when entity is created and enters the system. Prompt Description **Entity** Type The name of the entity type being defined. This name must be unique. Initial Picture Graphical representation of the entity at the start of the simulation. This value can be changed during the simulation using the [Assign](#) module. Holding Cost/Hour Hourly cost of processing the entity through the system. This cost is incurred when the entity is anywhere in the system. Initial VA Cost Initial cost value that will be assigned to the value-added cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a value-added activity. Initial NVA Cost Initial cost value that will be assigned to the non-value-added cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a non-value-added activity. Initial Waiting Cost Initial cost value that will be assigned to the waiting-cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a wait activity; for example, waiting to be batched or waiting for resource(s) at a [Process](#) module. Initial Transfer Cost Initial cost value that will be assigned to the transfer cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a transfer activity. Initial Other Cost Initial cost value that will be assigned to the other cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in another activity. Report Statistics Specifies whether or not statistics will be collected automatically and stored in the report database for this entity type.

Definition at line 76 of file Entity.h.

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



- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/simulator/Entity.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/simulator/Entity.cpp`

## 4.22 Event Class Reference

```
#include <Event.h>
```

### Public Member Functions

- **Event** (double time, [Entity](#) \*entity, [ModelComponent](#) \*component, unsigned int componentInputNumber=0)
- **Event** (double time, [Entity](#) \*entity, Connection \*connection)
- double **getTime** () const
- [ModelComponent](#) \* **getComponent** () const
- [Entity](#) \* **getEntity** () const
- unsigned int **getComponentInputNumber** () const
- std::string **show** ()

### 4.22.1 Detailed Description

An instantaneous event, triggered at a certain moment by an entity upon reaching a component. The simulated time advances in discrete points in time and that are the instants that an event is triggered.

Definition at line 28 of file Event.h.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/simulator/Event.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/simulator/Event.cpp`

## 4.23 Exit Class Reference

```
#include <Exit.h>
```

### Public Member Functions

- **Exit** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

## Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.23.1 Detailed Description

[Exit](#) module DESCRIPTION The [Exit](#) module releases the entity's cells on the specified conveyor. If another entity is waiting in queue for the conveyor at the same station when the cells are released, it will then access the conveyor. TYPICAL USES Cases exit a conveyor for packing Bad parts are removed from the conveyor and disposed Passengers remove luggage from the baggage claim conveyor PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart. Conveyor Name Name of the conveyor on which the entity will exit. If left blank, the previously accessed conveyor is assumed.

**of Cells Number of contiguous conveyor cells the entity will relinquish.**

Definition at line 37 of file Exit.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Exit.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Exit.cpp

## 4.24 ExperimentManager\_if Class Reference

```
#include <ExperimetManager_if.h>
```

### Public Member Functions

- virtual [List](#)< [SimulationScenario](#) \* > \* **getScenarios** () const =0
- virtual [List](#)< [SimulationControl](#) \* > \* **getControls** () const =0
- virtual [List](#)< [SimulationResponse](#) \* > \* **getResponses** () const =0
- virtual [List](#)< [SimulationControl](#) \* > \* **extractControlsFromModel** (std::string modelFilename) const =0
- virtual [List](#)< [SimulationResponse](#) \* > \* **extractResponsesFromModel** (std::string modelFilename) const =0
- virtual void **startSimulationOfScenario** ([SimulationScenario](#) \*scenario)=0
- virtual void **startExperiment** ()=0
- virtual void **stopExperiment** ()=0
- virtual void **addTraceSimulationHandler** (traceSimulationProcessListener traceSimulationProcess↔Listener)=0

### 4.24.1 Detailed Description

The experiment manager allows to extract controls and responses from a model, include some of them as controls and responses for a set of scenarios to be simulated

Definition at line 26 of file `ExperimetManager_if.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/simulator/ExperimetManager_if.h`

## 4.25 Failure Class Reference

```
#include <Failure.h>
```

### Public Member Functions

- **Failure** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)
- static PluginInformation \* **GetPluginInformation** ()

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual ParserChangesInformation \* **\_getParserChangesInformation** ()

### Additional Inherited Members

### 4.25.1 Detailed Description

**Failure** module DESCRIPTION The **Failure** module is designed for use with resources. When a failure occurs, the entire resource (regardless of its capacity) is failed. Failures are designed to be used with single-capacity resources or with multiple-capacity resources whose individual resource units all fail at the same time. TYPICAL USES Breakdown information for a machine Cash register tape refill every “x” customers Random computer shutdowns or restarts PROMPTS Recordset Name of the recordset in the specified file from which to read values. This field is available only if you specify a [File](#) Name with a file access type, path, and recordset. Arena uses the Rows and Columns properties to determine the amount of data to read from the recordset. A recordset is required for all file types except .xml. The recordset size must be equal to or greater than the number of rows and columns specified for the expression. Expression Values Lists the value or values of the expression. This property is not available if you specify a [File](#) Name from which to read expression values. Expression Value Expression value associated with the expression name. Prompt Description Name The name of the failure associated with one or more resources. Type Determines if the failure is time-based or count-based. Count Defines the number of resource releases for count-based failures. Valid when the Type is Count. Up Time Defines the time between failures for time-based failures. Valid when the Type is Time. Up Time Units Time units for the time between failures (Up Time) for timebased failures. Down Time Defines the duration of the failure. Down Time Units Time units for the duration of the failure (Down Time). Uptime in this State only Defines the state that should be considered for the time between failures (only for time-based failures). If state is not specified, then all states are considered (that is, the time between failures does not depend on the time spent in a specific state, but rather on the total simulation time). For example, you might want to define a failure to be based only on the state Busy, and therefore, the time between downtimes would be based on the amount of time that a resource is busy, not simulated clock time.

Definition at line 67 of file `Failure.h`.

## 4.25.2 Member Function Documentation

**4.25.2.1 `_getParserChangesInformation()`** `ParserChangesInformation * Failure::_getParserChangesInformation ( ) [protected], [virtual]`

This method returns all changes in the parser that are needed by plugins of this ModelElements. When connecting a new plugin, ParserChangesInformation are used to change parser source code, which is after compiled and dinamically linked to to simulator kernel to reflect the changes

Reimplemented from [ModelElement](#).

Definition at line 65 of file Failure.cpp.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_2022_GenESyS/plugins/elements/Failure.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_2022_GenESyS/plugins/elements/Failure.cpp`

## 4.26 File Class Reference

```
#include <File.h>
```

### Public Member Functions

- **File** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)
- static PluginInformation \* **GetPluginInformation** ()

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual ParserChangesInformation \* **\_getParserChangesInformation** ()

## Additional Inherited Members

### 4.26.1 Detailed Description

**File** module DESCRIPTION Use the **File** module to access external files for the ReadWrite module, **Variable** module, and Expression module. The **File** module identifies the system file name and defines the access type and operational characteristics of the file. TYPICAL USES **File** containing predefined airline flight data **File** specifying customer order times and relevant information **File** to write user model configuration data from menu input PROMPTS Prompt Description Name The name of the file whose characteristics are being defined. This name must be unique. **Access** Type The file type. Operating System **File** Name Name of the actual file that is being read from or to which it is being written. Connecting String Connection string used to open ADO connection to the data source. Structure **File** structure, which can be unformatted, free format, or a specific C or FORTRAN format. End of **File** Action Type of action to occur if an end of file condition is reached. Initialize Option Action to be taken on file at beginning of each simulation replication. Comment Character Character indicating comment record. Recordset Name Name used to identify the recordset in the Expression, ReadWrite, and **Variable** modules. This name must be unique within the file. This field is available for Microsoft Excel, Microsoft Excel 2007, Microsoft **Access**, Microsoft **Access** 2007, and ActiveX Data Objects files. CommandText Text of the command that will be used to open the recordset (for example, SQL statement, procedure name, table name.) This field is available for ActiveX Data Object files only. CommandType Type of command entered in the CommandText. Named Range The named range in the Excel workbook to which the recordset refers. Table Name The name of the table in the **Access** database to which the recordset refers.

Definition at line 64 of file File.h.

### 4.26.2 Member Function Documentation

**4.26.2.1** **\_getParserChangesInformation()** ParserChangesInformation \* File::\_getParserChangesInformation ( ) [protected], [virtual]

This method returns all changes in the parser that are needed by plugins of this ModelElements. When connecting a new plugin, ParserChangesInformation are used to change parser source code, which is after compiled and dynamically linked to simulator kernel to reflect the changes

Reimplemented from **ModelElement**.

Definition at line 66 of file File.cpp.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/elements/File.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/elements/File.cpp

## 4.27 yy::genesyspp\_parser Class Reference

A Bison parser.

```
#include <GenesysParser.h>
```

## Classes

- struct [basic\\_symbol](#)
- struct [by\\_kind](#)  
*Type access provider for token (enum) based symbols.*
- class [semantic\\_type](#)
- struct [symbol\\_kind](#)  
*Symbol kinds.*
- struct [symbol\\_type](#)  
*"External" symbols: returned by the scanner.*
- struct [syntax\\_error](#)  
*Syntax errors thrown from user actions.*
- struct [token](#)  
*Token kinds.*

## Public Types

- typedef [location](#) [location\\_type](#)  
*Symbol locations.*
- typedef [token::yytokentype](#) [token\\_kind\\_type](#)  
*Token kind, as returned by yylex.*
- typedef [token\\_kind\\_type](#) [token\\_type](#)  
*Backward compatibility alias (Bison 3.6).*
- typedef [symbol\\_kind::symbol\\_kind\\_type](#) [symbol\\_kind\\_type](#)  
*(Internal) symbol kind.*
- typedef [by\\_kind](#) [by\\_type](#)  
*Backward compatibility for a private implementation detail (Bison 3.6).*
- typedef int [debug\\_level\\_type](#)  
*Type for debugging levels.*

## Public Member Functions

- [genesyspp\\_parser](#) ([genesyspp\\_driver](#) &[driver\\_yyarg](#))  
*Build a parser object.*
- int [operator\(\)](#) ()
- virtual int [parse](#) ()
- std::ostream & [debug\\_stream](#) () const YY\_ATTRIBUTE\_PURE  
*The current debugging stream.*
- void [set\\_debug\\_stream](#) (std::ostream &)  
*Set the current debugging stream.*
- [debug\\_level\\_type](#) [debug\\_level](#) () const YY\_ATTRIBUTE\_PURE  
*The current debugging level.*
- void [set\\_debug\\_level](#) ([debug\\_level\\_type](#) l)  
*Set the current debugging level.*
- virtual void [error](#) (const [location\\_type](#) &loc, const std::string &msg)
- void [error](#) (const [syntax\\_error](#) &err)  
*Report a syntax error.*

## Static Public Member Functions

- static std::string [symbol\\_name](#) ([symbol\\_kind\\_type](#) yysymbol)
- static [symbol\\_type](#) [make\\_END](#) (const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_YYerror](#) (const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_YYUNDEF](#) (const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_NUMD](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_NUMH](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_CTEZERO](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oLE](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oGE](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oEQ](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oNE](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oAND](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oOR](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oNAND](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oXOR](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_oNOT](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fSIN](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fCOS](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fROUND](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fMOD](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fTRUNC](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fFRAC](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fEXP](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fSQRT](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fLOG](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fLN](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fVAL](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fEVAL](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fLENG](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fRND1](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fEXPO](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fNORM](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fUNIF](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fWEIB](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fLOGN](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fGAMM](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fERLA](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fTRIA](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fBETA](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fDISC](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fTNOW](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fTFIN](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fMAXREP](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fNUMREP](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fIDENT](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_cIF](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_cELSE](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_cFOR](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_cTO](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_cDO](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_ATTRIB](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_CSTAT](#) (const obj\_t &v, const [location\\_type](#) &l)
- static [symbol\\_type](#) [make\\_fTAVG](#) (const obj\_t &v, const [location\\_type](#) &l)

- static `symbol_type` `make_ILLEGAL` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_RESOURCE` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fNR` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fMR` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fIRF` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fRESSEIZES` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fSTATE` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fSETSUM` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fRESUTIL` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_QUEUE` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fNQ` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fFIRSTINQ` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fLASTINQ` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fSAQUE` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fAQUE` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fENTATRANK` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_SET` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fNUMSET` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_VARI` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_FORM` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fNUMGR` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_fATRGR` (const `obj_t` &`v`, const `location_type` &`l`)
- static `symbol_type` `make_LPAREN` (const `location_type` &`l`)
- static `symbol_type` `make_RPAREN` (const `location_type` &`l`)
- static `symbol_type` `make_LBRACKET` (const `location_type` &`l`)
- static `symbol_type` `make_RBRACKET` (const `location_type` &`l`)
- static `symbol_type` `make_PLUS` (const `location_type` &`l`)
- static `symbol_type` `make_MINUS` (const `location_type` &`l`)
- static `symbol_type` `make_STAR` (const `location_type` &`l`)
- static `symbol_type` `make_POWER` (const `location_type` &`l`)
- static `symbol_type` `make_SLASH` (const `location_type` &`l`)
- static `symbol_type` `make_LESS` (const `location_type` &`l`)
- static `symbol_type` `make_GREATER` (const `location_type` &`l`)
- static `symbol_type` `make_ASSIGN` (const `location_type` &`l`)
- static `symbol_type` `make_COMMA` (const `location_type` &`l`)
- static `symbol_type` `make_NEG` (const `location_type` &`l`)

### Static Public Attributes

- static const `symbol_kind_type` `YYNTOKENS` = `symbol_kind::YYNTOKENS`

*The number of tokens.*

### 4.27.1 Detailed Description

Definition at line 222 of file GenesysParser.h.

### 4.27.2 Member Function Documentation

**4.27.2.1 error()** `void yy::genesyspp_parser::error (`  
`const location_type & loc,`  
`const std::string & msg ) [virtual]`

Report a syntax error.



## Parameters

<i>loc</i>	where the syntax error is found.
<i>msg</i>	a description of the syntax error.

Definition at line 2712 of file GenesysParser.cpp.

#### 4.27.2.2 operator()() `int yy::genesyspp_parser::operator() ( )`

Parse. An alias for parse ().

## Returns

0 iff parsing succeeded.

Definition at line 749 of file GenesysParser.cpp.

#### 4.27.2.3 parse() `int yy::genesyspp_parser::parse ( ) [virtual]`

Parse.

## Returns

0 iff parsing succeeded.

Definition at line 755 of file GenesysParser.cpp.

#### 4.27.2.4 symbol\_name() `std::string yy::genesyspp_parser::symbol_name ( symbol_kind_type yysymbol ) [static]`

The user-facing name of the symbol whose (internal) number is YYSYMBOL. No bounds checking.

Definition at line 1965 of file GenesysParser.cpp.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/GenesysParser.cpp

## 4.28 Hold Class Reference

```
#include <Hold.h>
```

## Public Member Functions

- **Hold** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

## Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

## Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.28.1 Detailed Description

[Hold](#) module DESCRIPTION This module will hold an entity in a queue to wait for a signal, wait for a specified condition to become true (scan), or be held infinitely (to be removed later with the [Remove](#) module). If the entity is holding for a signal, the [Signal](#) module is used elsewhere in the model to allow the entity to move on to the next module. If the entity is holding for a given condition to be true, the entity will remain at the module (either in a defined or internal queue) until the condition(s) becomes true. When the entity is in an infinite hold, the [Remove](#) module is used elsewhere in the model to allow the entity to continue processing. TYPICAL USES Waiting for a traffic light to turn green Holding a part for authorization Checking the status of a machine or operator to continue a process PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Type Indicates the reasoning for holding the entity within a specified or internal queue. Wait for [Signal](#) will hold the entity until a signal of the same value is received. Scan for Condition will hold the entity until the specified condition becomes true. Infinite [Hold](#) will hold the entity until it is removed from the queue by a [Remove](#) module. Wait for Value [Signal](#) code for the waiting entity. Applies only when Type is Wait for [Signal](#). Limit Maximum number of waiting entities that will be released upon receipt of a signal. Applies only when Type is Wait for [Signal](#). Condition Specifies the condition that will be evaluated to hold the entity at the module. If the condition is evaluated to true, the entity leaves the module immediately. If the condition is false, the entity will wait in the associated queue until the condition becomes true. Applies only when Type is Scan for Condition. [Queue](#) Type Determines the type of queue used to hold the entities. If [Queue](#) is selected, the queue name is specified. If [Set](#) is selected, the queue set and member in the set are specified. If Internal is selected, an internal queue is used to hold all waiting entities. [Attribute](#) and Expression are additional methods for defining the queue to be used. [Queue](#) Name This field is visible only if [Queue](#) Type is [Queue](#), and it defines the symbol name of the queue. [Set](#) Name This field is visible only if [Queue](#) Type is [Set](#), and it defines the queue set that contains the queue being referenced. [Set](#) Index This field is visible only if [Queue](#) Type is [Set](#), and it defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entry for a queue set containing three members is an expression that evaluates to 1, 2, or 3. [Attribute](#) This field is visible only if [Queue](#) Type is [Attribute](#). The attribute entered in this field will be evaluated to indicate which queue is to be used. Expression This field is visible only if [Queue](#) Type is Expression. The expression entered in this field will be evaluated to indicate which queue is to be used.

Definition at line 75 of file Hold.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Hold.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Hold.cpp

## 4.29 HypothesisTester\_if Class Reference

```
#include <HypothesisTester_if.h>
```

### Public Types

- enum **H1Comparition** { **DIFFERENT** = 1 , **LESS\_THAN** = 2 , **GREATER\_THAN** = 3 }

### Public Member Functions

- virtual HypothesisTester\_if::ConfidenceInterval **averageConfidenceInterval** (double avg, double stddev, unsigned int n, double confidenceLevel)=0
- virtual HypothesisTester\_if::ConfidenceInterval **proportionConfidenceInterval** (double prop, unsigned int n, double confidenceLevel)=0
- virtual HypothesisTester\_if::ConfidenceInterval **varianceConfidenceInterval** (double var, unsigned int n, double confidenceLevel)=0
- virtual HypothesisTester\_if::ConfidenceInterval **averageDifferenceConfidenceInterval** (double avg1, double stddev1, unsigned int n1, double avg2, double stddev2, unsigned int n2, double confidenceLevel)=0
- virtual HypothesisTester\_if::ConfidenceInterval **varianceRatioConfidenceInterval** (double var1, unsigned int n1, double var2, unsigned int n2, double confidenceLevel)=0
- virtual HypothesisTester\_if::ConfidenceInterval **averageConfidenceInterval** (std::string sampleDataFilename, double confidenceLevel)=0
- virtual HypothesisTester\_if::ConfidenceInterval **proportionConfidenceInterval** (std::string sampleDataFilename, checkProportionFunction function, double confidenceLevel)=0
- virtual HypothesisTester\_if::ConfidenceInterval **varianceConfidenceInterval** (std::string sampleDataFilename, double confidenceLevel)=0
- virtual HypothesisTester\_if::TestResult **testAverage** (double avg, double stddev, unsigned int n, double avgSample, double confidenceLevel, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testProportion** (double prop, unsigned int n, double proptest, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testVariance** (double var, unsigned int n, double vartest, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testAverage** (double avg1, unsigned int n1, double avg2, unsigned int n2, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testProportion** (double prop1, unsigned int n1, double prop2, unsigned int n2, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testVariance** (double var1, unsigned int n1, double var2, unsigned int n2, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testAverage** (std::string sampleDataFilename, double avgSample, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testProportion** (std::string sampleDataFilename, checkProportionFunction function, double proptest, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testVariance** (std::string sampleDataFilename, double vartest, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testAverage** (std::string firstSampleDataFilename, std::string secondSampleDataFilename, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testProportion** (std::string firstSampleDataFilename, std::string secondSampleDataFilename, checkProportionFunction function, HypothesisTester\_if::H1Comparition comp)=0
- virtual HypothesisTester\_if::TestResult **testVariance** (std::string firstSampleDataFilename, std::string secondSampleDataFilename, HypothesisTester\_if::H1Comparition comp)=0

### 4.29.1 Detailed Description

Interface for parametric hypothesis tests based on a datafile or parameters. All tests are supposed to be based on samples with unknown population parameters

Definition at line 25 of file HypothesisTester\_if.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/tools/HypothesisTester\_if.h

## 4.30 Leave Class Reference

```
#include <Leave.h>
```

### Public Member Functions

- **Leave** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- void **setStation** (Station \*\_station)
- Station \* **getStation** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

## Additional Inherited Members

### 4.30.1 Detailed Description

**Leave** module DESCRIPTION The **Leave** module is used to transfer an entity to a station or module. An entity may be transferred in two ways. It can be transferred to a module that defines a station by referencing the station and routing, conveying, or transporting to that station, or a graphical connection can be used to transfer an entity to another module. When an entity arrives at a **Leave** module, it may wait to obtain a transfer device (resource, transporter, or conveyor). When the transfer device has been obtained, the entity may experience a loading delay. Finally, the entity is transferred from this module to a destination module or station. TYPICAL USES The end of a part's production in a series of parallel processes where the part needs a forklift to be transferred to shipping PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart. Allocation Type of category to which the entity's incurred delay time and cost will be added. **Delay** Specifies a load time incurred after getting a transfer device. Units Time units used for the delay time. Transfer Out Determines whether a resource (**Seize Resource**), transporter (Request Transporter), or conveyor (**Access Conveyor**) is required prior to transferring the entity out of this module. Priority Indicates the priority of the module when either seizing a resource or requesting a transporter when there are entities waiting for that resource/transporter from other modules. This field is not visible when the Transfer Type is None or **Access Conveyor**. Transporter Name Name of the transporter to request. **Queue** Type Type of queue, either a single **Queue**, queue **Set**, Internal queue, **Attribute**, or Expression. **Queue** Name Name of the individual queue. **Queue Set** Name Name of the queue set. **Set** Index Defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. **Queue Attribute** Name The attribute name that will be evaluated to indicate which queue is to be used. **Queue** Expression The expression that will be evaluated to indicate which queue is to be used. Selection Rule Method of selecting among available transporters in a set. Cyclical will cycle through available members. Random will randomly select a member. Preferred Order will always select the first available member. Specific Member requires an input attribute value to specify which member of the set (previously saved in the Save **Attribute** field). Largest Distance selects the transporter farthest away, and Smallest Distance selects the closest transporter. Save **Attribute Attribute** name used to store the index number into the set of the member that is chosen. This attribute can later be referenced with the Specific Member selection rule. Active when Transfer Out is Request Transporter. Index **Set Attribute** name whose value identifies the index number into the set of the member requested. The entity must have a value for the attribute before utilizing this option. **Resource** Type Type of resource for seizing, either specifying a particular **Resource**, selecting from a pool of resources (that is, a resource **Set**), **Attribute**, or Expression. **Resource** Name Name of the resource to seize. Conveyor Name Name of the conveyor to access.

of Cells Number of contiguous cells the entity requires.

Connect Type Determines if the entity is to **Route**, Convey, or Transport to another station or Connect to another module. Move Time Time to route from this module to the destination station. Units Time units used for the move time. Station Type The entity's destination station type either an individual Station, a station based on an **Attribute** or Expression value, or By **Sequence**. Station Name Name of the individual destination station. **Attribute** Name The attribute name that will be evaluated to indicate the station. Expression The expression that will be evaluated to indicate the station.

Definition at line 94 of file Leave.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Leave.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Leave.cpp

## 4.31 LicenceManager Class Reference

```
#include <LicenceManager.h>
```

## Public Member Functions

- **LicenceManager** ([Simulator](#) \*simulator)
- const std::string **showLicence** () const
- const std::string **showLimits** () const
- const std::string **showActivationCode** () const
- bool **lookforActivationCode** ()
- bool **insertActivationCode** ()
- void **removeActivationCode** ()
- unsigned int **getModelComponentsLimit** ()
- unsigned int **getModelElementsLimit** ()
- unsigned int **getEntityLimit** ()
- unsigned int **getHostsLimit** ()
- unsigned int **getThreadsLimit** ()

### 4.31.1 Detailed Description

[LicenceManager](#) just prints a licence agreement and checks for some resource limits.

Definition at line 25 of file `LicenceManager.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔  
2022_GenESyS/kernel/simulator/LicenceManager.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔  
2022_GenESyS/kernel/simulator/LicenceManager.cpp`

## 4.32 List< T > Class Template Reference

```
#include <List.h>
```

### Public Types

- using **CompFunct** = std::function< bool(const T, const T) >

### Public Member Functions

- unsigned int **size** ()
- bool **empty** ()
- void **clear** ()
- void **pop\_front** ()
- template<class Compare >  
void **sort** (Compare comp)
- std::list< T > \* **list** () const
- T **create** ()
- template<typename U >  
T **create** (U arg)
- std::string **show** ()
- std::list< T >::iterator **find** (T element)

- void **insert** (T element)
- void **remove** (T element)
- void **setAtRank** (unsigned int rank, T element)
- T **getAtRank** (unsigned int rank)
- T **next** ()
- T **front** ()
- T **last** ()
- T **previous** ()
- T **current** ()
- void **setSortFunc** (CompFunct \_sortFunc)

#### 4.32.1 Detailed Description

```
template<typename T>
class List< T >
```

[List](#) corresponds to an extended version of the list that must guarantee the consistency of the elements that make up the simulation model.

Definition at line 32 of file List.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/util/List.h

## 4.33 yy::location Class Reference

Two points in a source file.

```
#include <location.hh>
```

### Public Types

- typedef [position::filename\\_type](#) filename\_type  
*Type for file name.*
- typedef [position::counter\\_type](#) counter\_type  
*Type for line and column numbers.*

### Public Member Functions

- [location](#) (const [position](#) &b, const [position](#) &e)  
*Construct a location from b to e.*
- [location](#) (const [position](#) &p=[position](#)())  
*Construct a 0-width location in p.*
- [location](#) (filename\_type \*f, counter\_type l=1, counter\_type c=1)  
*Construct a 0-width location in f, l, c.*
- void [initialize](#) (filename\_type \*f=YY\_NULLPTR, counter\_type l=1, counter\_type c=1)  
*Initialization.*

### Line and Column related manipulators

- void [step](#) ()  
*Reset initial location to final location.*
- void [columns](#) (counter\_type count=1)  
*Extend the current location to the COUNT next columns.*
- void [lines](#) (counter\_type count=1)  
*Extend the current location to the COUNT next lines.*

## Public Attributes

- [position begin](#)  
*Beginning of the located region.*
- [position end](#)  
*End of the located region.*

### 4.33.1 Detailed Description

Definition at line 165 of file `location.hh`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔  
2022_GenESyS/kernel/simulator/parserBisonFlex/location.hh`

## 4.34 LODE Class Reference

```
#include <LSODE.h>
```

### Public Member Functions

- **LSODE** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- void **setDiffEquations** (Formula \*formula)
- Formula \* **getDiffEquations** () const
- void **setTimeVariable** ([Variable](#) \*\_timeVariable)
- [Variable](#) \* **getTimeVariable** () const
- void **setStep** (double \_step)
- double **getStep** () const
- void **setVariables** ([Variable](#) \*\_variables)
- [Variable](#) \* **getVariables** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)



## Additional Inherited Members

### 4.34.1 Detailed Description

This component ...

Definition at line 24 of file LODE.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/LODE.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/LODE.cpp

## 4.35 Match Class Reference

```
#include <Match.h>
```

### Public Member Functions

- **Match** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.35.1 Detailed Description

**Match** module DESCRIPTION The **Match** module brings together a specified number of entities waiting in different queues. The match may be accomplished when there is at least one entity in each of the desired queues. Additionally, an attribute may be specified such that the entities waiting in the queues must have the same attribute values before the match is initiated. When an entity arrives at the **Match** module, it is placed in one of up to five associated queues, based on the entry point to which it is connected. Entities will remain in their respective queues until a match exists. Once a match exists, one entity from each queue is released to be matched. The matched entities are then synchronized to depart from the module. TYPICAL USES Assembling a part Gathering various products for a customer order Synchronizing a customer exit with a filled order Prompt Description Name Unique module identifier displayed on the module shape. Number to **Match** Number of matching entities that must reside in different queues before a match may be completed. Type Method for matching the incoming entities. If Type is Any Entities, one entity must reside in each queue for a match to be made. If Type is Based on **Attribute**, one entity must reside in each queue with the same attribute value. **Attribute** Name **Attribute** name that is used for identifying an arriving entity's match value. Applies only when Type is Based on **Attribute**.

Definition at line 47 of file Match.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Match.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Match.cpp

## 4.36 Model Class Reference

```
#include <Model.h>
```

### Public Member Functions

- **Model** (**Simulator** \*simulator)
- bool **save** (std::string filename)
- bool **load** (std::string filename)
- bool **check** ()  
*Checks the integrity and consistency of the model, possibly corrects some inconsistencies, and returns if the model is in position to the simulated.*
- void **clear** ()
- void **show** ()
- bool **insert** (**ModelElement** \*elemOrComp)  
*Insert a new **ModelElement** or **ModelComponent** into the model (since 20191015). It's a generic access to ComponentManager->insert() or ModelElement->insert()*
- void **remove** (**ModelElement** \*elemOrComp)  
*Remove a new **ModelElement** or **ModelComponent** into the model (since 20191015). It's a generic access to ComponentManager->remove() or ModelElement->remove()*
- **Entity** \* **createEntity** (std::string name, bool insertIntoModel)
- void **removeEntity** (**Entity** \*entity)
- void **sendEntityToComponent** (**Entity** \*entity, **Connection** \*connection, double timeDelay=0.0)  
*Used by components (**ModelComponent**) to send entities to another specific component, usually the next one connected to it, or used by the model itself, when processing an event (**Event**).*

- void **sendEntityToComponent** (**Entity** \*entity, **ModelComponent** \*component, double timeDelay=0.0, unsigned int componentInputNumber=0)
 

*Used by components (**ModelComponent**) to send entities to another specific component, usually the next one connected to it, or used by the model itself, when processing an event (**Event**).*
- double **parseExpression** (const std::string expression)
 

*Invokes the parser to evaluate tyhe expression. Result is always a double, even if expression has syntatic errors (returns 0)*
- double **parseExpression** (const std::string expression, bool \*success, std::string \*errorMessage)
 

*Invokes the parser to evaluate tyhe expression. Result is always a double, even if expression has syntatic errors (returns 0). Explicitly informs if there was an error.*
- bool **checkExpression** (const std::string expression, const std::string expressionName, std::string \*error←Message)
 

*This is invoked by ModelComponents and ModelElements in their private method \_check() to verify if an expression defined by user is valid or not.*
- Util::identification **getId** () const
- bool **hasChanged** () const
- **EventManager** \* **getOnEvents** () const
 

*Provides access to the class that manages events generated by the model, such as the beggining of a new simulation or replication, the processig of an event and much more.*
- **ElementManager** \* **getElements** () const
 

*Provides access to the class that manages the most basic elements of the simulation model (such as queues, re-sources, variables, etc.).*
- **ComponentManager** \* **getComponents** () const
- **ModelInfo** \* **getInfos** () const
- **Simulator** \* **getParentSimulator** () const
- **ModelSimulation** \* **getSimulation** () const
 

*Provides access to the class that manages the model simulation.*
- List< **Event** \* > \* **getFutureEvents** () const
 

*The future events list chronologically sorted; Events are scheduled by components when processing other events, and a replication evolves over time by sequentially processing the very first event in this list. It's initialized with events first described by source components (SourceComponentModel).*
- List< **SimulationControl** \* > \* **getControls** () const
 

*Returns a list of values that can be externally controlled (changed). They usually correspond to input parameters in the simulation model that must be changed for an experimental design.*
- List< **SimulationResponse** \* > \* **getResponses** () const
 

*Returns a list of exits or simulation results that can be read externally. They usually correspond to statistics resulting from the simulation that must be read for an experiment design.*
- void **setTracer** (**TraceManager** \*\_traceManager)
- **TraceManager** \* **getTracer** () const
 

*Provides access to the class that performs the trace of simulation and replications.*

#### 4.36.1 Detailed Description

**Model** is probably the most important class of Genesys kernel. It represents a discrete event-driven simulation model. Each model is responsible for controlling its own simulation, ie, for sequentially processing events and collecting statistical results. A model is mainly represented by a collection of components (**ModelComponent**), adequately configured and connected, and a collection of under layered element (**ModelElement**).

Definition at line 44 of file Model.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/Model.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/Model.cpp

## 4.37 ModelChecker\_if Class Reference

```
#include <ModelChecker_if.h>
```

### Public Member Functions

- virtual bool **checkAll** ()=0
- virtual bool **checkConnected** ()=0
- virtual bool **checkSymbols** ()=0
- virtual bool **checkActivationCode** ()=0
- virtual bool **checkLimits** ()=0

#### 4.37.1 Detailed Description

The ModelChecker is responsible for verifying the model consistency, fixing inconsistencies whenever possible

Definition at line 26 of file ModelChecker\_if.h.

#### 4.37.2 Member Function Documentation

**4.37.2.1 checkActivationCode()** `virtual bool ModelChecker_if::checkActivationCode ( ) [pure virtual]`

Checks if user-defined strings for symbols required by components, usually expressions or functions, are valid or references existing and valid elements.

**4.37.2.2 checkConnected()** `virtual bool ModelChecker_if::checkConnected ( ) [pure virtual]`

Invokes all other checks and returns true only if all of them returned true

**4.37.2.3 checkLimits()** `virtual bool ModelChecker_if::checkLimits ( ) [pure virtual]`

Checks if the installed version has acquired a valid activation code for commercial use

**4.37.2.4 checkSymbols()** `virtual bool ModelChecker_if::checkSymbols ( ) [pure virtual]`

Checks if components are consistently connected to other to form a valid process-oriented model, describing how entities proceed to the flow

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/ModelChecker\_if.h

## 4.38 ModelComponent Class Reference

```
#include <ModelComponent.h>
```

### Public Member Functions

- **ModelComponent** ([Model](#) \*model, std::string componentType, std::string name="")
- virtual std::string **show** ()
- [ConnectionManager](#) \* **getConnections** () const  
*Returns a list of components directly connected to the output. Usually the components have a single output, but they may have none (such as [Dispose](#)) or more than one (as [Decide](#)). In addition to the component, NextComponents specifies the inputNumber of the next component where the entity will be sent to. Usually the components have a single input, but they may have none (such as [Create](#)) or more than one (as [Match](#)).*
- void **setDescription** (std::string \_description)
- std::string **getDescription** () const

### Static Public Member Functions

- static void **Execute** ([Entity](#) \*entity, [ModelComponent](#) \*component, unsigned int inputNumber)  
*This method triggers the simulation of the behavior of the component. It is invoked when an event (corresponding to this component) is taken from the list of future events or when an entity arrives at this component by connection.*
- static void **CreateInternalElements** ([ModelComponent](#) \*component)
- static bool **Check** ([ModelComponent](#) \*component)
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)
- static std::map< std::string, std::string > \* **SaveInstance** ([ModelComponent](#) \*component)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)=0
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_createInternalElements** ()  
*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

### Protected Attributes

- const struct ModelComponent::DEFAULT\_VALUES **DEFAULT**
- std::string **\_description** = DEFAULT.description
- [ConnectionManager](#) \* **\_connections** = new [ConnectionManager](#)()

#### 4.38.1 Detailed Description

A component of the model is a block that represents a specific behavior to be simulated. The behavior is triggered when an entity arrives at the component, which corresponds to the occurrence of an event. A simulation model corresponds to a set of interconnected components to form the process by which the entity is submitted.

## Parameters

<i>model</i>	The model this component belongs to
--------------	-------------------------------------

Definition at line 33 of file ModelComponent.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/ModelComponent.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/ModelComponent.cpp

## 4.39 ModelElement Class Reference

```
#include <ModelElement.h>
```

### Public Member Functions

- **ModelElement** (**Model** \*model, std::string elementTypename, std::string name="", bool insertInto↔Model=true)
- Util::identification **getId** () const
- void **setName** (std::string name)
- std::string **getName** () const
- std::string **getClassname** () const
- bool **isReportStatistics** () const
- void **setReportStatistics** (bool reportStatistics)
- virtual std::string **show** ()
- std::list< std::string > \* **getChildrenElementKeys** () const
- **ModelElement** \* **getChildElement** (std::string key) const
- bool **hasChanged** () const

### Static Public Member Functions

- static **ModelElement** \* **LoadInstance** (**Model** \*model, std::map< std::string, std::string > \*fields, bool insert↔IntoModel)
- static std::map< std::string, std::string > \* **SaveInstance** (**ModelElement** \*element)
- static bool **Check** (**ModelElement** \*element, std::string \*errorMessage)
- static void **CreateInternalElements** (**ModelElement** \*element)
- static void **InitBetweenReplications** (**ModelElement** \*element)

### Protected Member Functions

- void **\_setChildElement** (std::string key, **ModelElement** \*child)
- void **\_removeChildrenElements** ()
- void **\_removeChildElement** (std::string key)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual ParserChangesInformation \* **\_getParserChangesInformation** ()
- virtual void **\_initBetweenReplications** ()
- virtual void **\_createInternalElements** ()

### Protected Attributes

- Util::identification **\_id**
- std::string **\_typename**
- bool **\_reportStatistics**
- bool **\_hasChanged**
- [Model](#) \* **\_parentModel**
- std::map< std::string, [ModelElement](#) \* > \* **\_childrenElements** = new std::map<std::string, [ModelElement](#)\*>()

#### 4.39.1 Detailed Description

This class is the basis for any element of the model (such as [Queue](#), [Resource](#), [Variable](#), etc.) and also for any component of the model. It has the infrastructure to read and write on file and to verify symbols.

Definition at line 33 of file ModelElement.h.

#### 4.39.2 Member Function Documentation

**4.39.2.1 `_createInternalElements()`** `void ModelElement::_createInternalElements ( ) [protected], [virtual]`

This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist before checking such expression

Reimplemented in [Resource](#), [Queue](#), [Route](#), [Record](#), [Process](#), [Leave](#), [Enter](#), [Dispose](#), [Delay](#), [Decide](#), [Create](#), and [ModelComponent](#).

Definition at line 257 of file ModelElement.cpp.

**4.39.2.2 `_getParserChangesInformation()`** `ParserChangesInformation * ModelElement::_getParserChangesInformation ( ) [protected], [virtual]`

This method returns all changes in the parser that are needed by plugins of this ModelElements. When connecting a new plugin, ParserChangesInformation are used to change parser source code, which is after compiled and dynamically linked to to simulator kernel to reflect the changes

Reimplemented in [Set](#), [Queue](#), [File](#), and [Failure](#).

Definition at line 112 of file ModelElement.cpp.

**4.39.2.3 Check()** `bool ModelElement::Check (`  
    `ModelElement * element,`  
    `std::string * errorMessage ) [static]`

This class method takes an instance of a [ModelElement](#) and invokes the private method `_check()` method of that instance, which checks itself

Definition at line 229 of file `ModelElement.cpp`.

**4.39.2.4 CreateInternalElements()** `void ModelElement::CreateInternalElements (`  
    `ModelElement * element ) [static]`

This class method is responsible for invoking the protected method `_check()` of the instance element, which creates any internal [ModelElement](#) (such as childrenElements) or even other external needed ModelElements, such as attributes or variables

Definition at line 247 of file `ModelElement.cpp`.

**4.39.2.5 getChildrenElementKeys()** `std::list< std::string > * ModelElement::getChildrenElement↵`  
`Keys ( ) const`

Returns a list of keys (names) of internal ModelElements, such as Counters, StatisticsCollectors and others. ChildrenElements are ModelElements used by this [ModelElement](#) that are needed before model checking

Definition at line 140 of file `ModelElement.cpp`.

**4.39.2.6 isReportStatistics()** `bool ModelElement::isReportStatistics ( ) const`

Return true if this [ModelElement](#) generates statistics for simulation reports

Definition at line 268 of file `ModelElement.cpp`.

**4.39.2.7 LoadInstance()** `ModelElement * ModelElement::LoadInstance (`  
    `Model * model,`  
    `std::map< std::string, std::string > * fields,`  
    `bool insertIntoModel ) [static]`

This class method receives a map of fields readed from a file (or somewhere else) creates an instance of the Model↵ Elements and invokes the protected method `_loadInstance()` of that instance, which fills the field values. The instance can be automatically inserted into the simulation model if required

Definition at line 202 of file `ModelElement.cpp`.



**4.39.2.8 SaveInstance()** `std::map< std::string, std::string > * ModelElement::SaveInstance ( ModelElement * element ) [static]`

This class method takes an instance of a [ModelElement](#), invokes the protected method `_saveInstance()` of that instance and returns a map of fields (name=value) that can be saved on a file (or somewhere else)

Definition at line 219 of file `ModelElement.cpp`.

**4.39.2.9 setReportStatistics()** `void ModelElement::setReportStatistics ( bool reportStatistics )`

Defines if this [ModelElement](#) generates statistics for simulation reports

Definition at line 261 of file `ModelElement.cpp`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔2022_GenESyS/kernel/simulator/ModelElement.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔2022_GenESyS/kernel/simulator/ModelElement.cpp`

## 4.40 ModelInfo Class Reference

```
#include <ModelInfo.h>
```

### Public Member Functions

- `std::string show ()`
- `void setName (std::string _name)`
- `std::string getName () const`
- `void setAnalystName (std::string _analystName)`
- `std::string getAnalystName () const`
- `void setDescription (std::string _description)`
- `std::string getDescription () const`
- `void setProjectTitle (std::string _projectTitle)`
- `std::string getProjectTitle () const`
- `void setVersion (std::string _version)`
- `std::string getVersion () const`
- `void loadInstance (std::map< std::string, std::string > *fields)`
- `std::map< std::string, std::string > * saveInstance ()`
- `bool hasChanged () const`

### 4.40.1 Detailed Description

[ModelInfo](#) stores basic model project information, such as the project and analyst names.

Definition at line 25 of file `ModelInfo.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔2022_GenESyS/kernel/simulator/ModelInfo.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔2022_GenESyS/kernel/simulator/ModelInfo.cpp`

## 4.41 ModelPersistence\_if Class Reference

```
#include <ModelPersistence_if.h>
```

### Public Member Functions

- virtual bool **save** (std::string filename)=0
- virtual bool **load** (std::string filename)=0
- virtual bool **hasChanged** ()=0

### 4.41.1 Detailed Description

First and inadequate interface for model persistence. It should use the best pattern for the DAO approach

Definition at line 22 of file ModelPersistence\_if.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/ModelPersistence\_if.h

## 4.42 ModelSimulation Class Reference

```
#include <ModelSimulation.h>
```

### Public Member Functions

- **ModelSimulation** ([Model](#) \*model)
- std::string **show** ()
- void **start** ()  
*Starts a sequential execution of a simulation, ie, a set of replications of this model.*
- void **pause** ()
- void **step** ()  
*Executes the processing of a single event, the next one in the future events list.*
- void **stop** ()
- void **setNumberOfReplications** (unsigned int \_numberOfReplications)
- unsigned int **getNumberOfReplications** () const
- void **setReplicationLength** (double \_replicationLength)
- double **getReplicationLength** () const
- void **setReplicationLengthTimeUnit** (Util::TimeUnit \_replicationLengthTimeUnit)
- Util::TimeUnit **getReplicationLengthTimeUnit** () const
- void **setReplicationReportBaseTimeUnit** (Util::TimeUnit \_replicationReportBaseTimeUnit)
- Util::TimeUnit **getReplicationBaseTimeUnit** () const
- void **setWarmUpPeriod** (double \_warmUpPeriod)
- double **getWarmUpPeriod** () const
- void **setWarmUpPeriodTimeUnit** (Util::TimeUnit \_warmUpPeriodTimeUnit)
- Util::TimeUnit **getWarmUpPeriodTimeUnit** () const
- void **setTerminatingCondition** (std::string \_terminatingCondition)
- std::string **getTerminatingCondition** () const

- void **setPauseOnEvent** (bool \_pauseOnEvent)
- bool **isPauseOnEvent** () const
- void **setStepByStep** (bool \_stepByStep)
- bool **isStepByStep** () const
- void **setInitializeStatistics** (bool \_initializeStatistics)
- bool **isInitializeStatistics** () const
- void **setInitializeSystem** (bool \_initializeSystem)
- bool **isInitializeSystem** () const
- void **setPauseOnReplication** (bool \_pauseBetweenReplications)
- bool **isPauseOnReplication** () const
- void **setReporter** (SimulationReporter\_if \* \_simulationReporter)
- SimulationReporter\_if \* **getReporter** () const
- double **getSimulatedTime** () const
- bool **isRunning** () const
- bool **isPaused** () const
- unsigned int **getCurrentReplicationNumber** () const
- [ModelComponent](#) \* **getCurrentComponent** () const
- [Entity](#) \* **getCurrentEntity** () const
- unsigned int **getCurrentInputNumber** () const
- void **setShowReportsAfterReplication** (bool showReportsAfterReplication)
- bool **isShowReportsAfterReplication** () const
- void **setShowReportsAfterSimulation** (bool showReportsAfterSimulation)
- bool **isShowReportsAfterSimulation** () const
- [List](#)< double > \* **getBreakpointsOnTime** () const
- [List](#)< [Entity](#) \* > \* **getBreakpointsOnEntity** () const
- [List](#)< [ModelComponent](#) \* > \* **getBreakpointsOnComponent** () const
- void **loadInstance** (std::map< std::string, std::string > \*fields)
- std::map< std::string, std::string > \* **saveInstance** ()
- [Event](#) \* **getCurrentEvent** () const
- void **setShowSimulationResponsesInReport** (bool \_showSimulationResponsesInReport)
- bool **isShowSimulationResponsesInReport** () const
- void **setShowSimulationControlsInReport** (bool \_showSimulationControlsInReport)
- bool **isShowSimulationControlsInReport** () const

## Friends

- class **Model**

### 4.42.1 Detailed Description

The [ModelSimulation](#) controls the simulation of a model, allowing to start, pause, resume e stop a simulation, composed by a set of replications.

Definition at line 33 of file ModelSimulation.h.

### 4.42.2 Member Function Documentation

#### 4.42.2.1 **isRunning()** `bool ModelSimulation::isRunning ( ) const`

The current time in the model being simulated, i.e., the instant when the current event was triggered

Definition at line 528 of file ModelSimulation.cpp.

#### 4.42.2.2 **start()** `void ModelSimulation::start ( )`

Checks the model and if ok then initialize the simulation, execute repeatedly each replication and then show simulation statistics

Definition at line 90 of file ModelSimulation.cpp.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/ModelSimulation.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/ModelSimulation.cpp

## 4.43 OnEventManager Class Reference

```
#include <OnEventManager.h>
```

### Public Member Functions

- void **addOnReplicationStartHandler** (simulationEventHandler EventHandler)
- void **addOnReplicationStepHandler** (simulationEventHandler EventHandler)
- void **addOnReplicationEndHandler** (simulationEventHandler EventHandler)
- void **addOnProcessEventHandler** (simulationEventHandler EventHandler)
- void **addOnEntityCreateHandler** (simulationEventHandler EventHandler)
- void **addOnEntityMoveHandler** (simulationEventHandler EventHandler)
- void **addOnEntityRemoveHandler** (simulationEventHandler EventHandler)
- void **addOnSimulationStartHandler** (simulationEventHandler EventHandler)
- void **addOnSimulationPausedHandler** (simulationEventHandler EventHandler)
- void **addOnSimulationResumeHandler** (simulationEventHandler EventHandler)
- void **addOnSimulationEndHandler** (simulationEventHandler EventHandler)
- void **addOnBreakpointHandler** (simulationEventHandler EventHandler)
- template<typename Class >  
void **addOnReplicationStartHandler** (Class \*object, void(Class::\*function)(SimulationEvent \*))
- template<typename Class >  
void **addOnReplicationStepHandler** (Class \*object, void(Class::\*function)(SimulationEvent \*))
- template<typename Class >  
void **addOnReplicationEndHandler** (Class \*object, void(Class::\*function)(SimulationEvent \*))
- template<typename Class >  
void **addOnProcessEventHandler** (Class \*object, void(Class::\*function)(SimulationEvent \*))
- template<typename Class >  
void **addOnEntityCreateHandler** (Class \*object, void(Class::\*function)(SimulationEvent \*))
- template<typename Class >  
void **addOnEntityMoveHandler** (Class \*object, void(Class::\*function)(SimulationEvent \*))

- `template<typename Class >`  
`void addOnEntityRemoveHandler (Class *object, void(Class::*function)(SimulationEvent *))`
- `template<typename Class >`  
`void addOnSimulationStartHandler (Class *object, void(Class::*function)(SimulationEvent *))`
- `template<typename Class >`  
`void addOnSimulationPausedHandler (Class *object, void(Class::*function)(SimulationEvent *))`
- `template<typename Class >`  
`void addOnSimulationResumeHandler (Class *object, void(Class::*function)(SimulationEvent *))`
- `template<typename Class >`  
`void addOnSimulationEndHandler (Class *object, void(Class::*function)(SimulationEvent *))`
- `template<typename Class >`  
`void addOnBreakpointHandler (Class *object, void(Class::*function)(SimulationEvent *))`
- `void NotifyReplicationStartHandlers (SimulationEvent *se)`
- `void NotifyReplicationStepHandlers (SimulationEvent *se)`
- `void NotifyReplicationEndHandlers (SimulationEvent *se)`
- `void NotifyProcessEventHandlers (SimulationEvent *se)`
- `void NotifyEntityCreateHandlers (SimulationEvent *se)`
- `void NotifyEntityMoveHandlers (SimulationEvent *se)`
- `void NotifyEntityRemoveHandlers (SimulationEvent *se)`
- `void NotifySimulationStartHandlers (SimulationEvent *se)`
- `void NotifySimulationPausedHandlers (SimulationEvent *se)`
- `void NotifySimulationResumeHandlers (SimulationEvent *se)`
- `void NotifySimulationEndHandlers (SimulationEvent *se)`
- `void NotifyBreakpointHandlers (SimulationEvent *se)`

#### 4.43.1 Detailed Description

[OnEventManager](#) allows external methods to hook interval simulation events as listeners (or observers) of specific events. All methods added as listeners of an event will be invoked when that event is triggered.

Definition at line 140 of file `OnEventManager.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/simulator/OnEventManager.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/simulator/OnEventManager.cpp`

## 4.44 PickStation Class Reference

```
#include <PickStation.h>
```

### Public Member Functions

- `PickStation (Model *model, std::string name="")`
- `virtual std::string show ()`

### Static Public Member Functions

- `static PluginInformation * GetPluginInformation ()`
- `static ModelComponent * LoadInstance (Model *model, std::map< std::string, std::string > *fields)`

## Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.44.1 Detailed Description

[PickStation](#) module DESCRIPTION The [PickStation](#) module allows an entity to select a particular station from the multiple stations specified. This module picks among the group of stations based on the selection logic defined with the module. The entity may then route, transport, convey, or connect to the station specified. If the method chosen is connect, the selected station is assigned to an entity attribute. The station selection process is based on the minimum or maximum value of a variety of system variables and expressions. TYPICAL USES A part sent to a processing station based on machine's availability at each station A loan application sent to a set of loan officers based on the number sent to each officer A customer selecting among cashier lines based on the least number waiting in each line PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart. Test Condition Test condition to use for the station selection process, either Minimum or Maximum. Number En [Route](#) to Station The number of entities transferring to the station is considered in the station selection process. Number in [Queue](#) The number of entities in the queue at the station is considered in the station selection process. Number of Resources Busy The number of busy resources at the station is considered in the station selection process. Expression Determines if an additional user-defined expression is considered in the station selection process. Transfer Type Determines how an entity will be transferred out of this module to its next destination station—either [Route](#), Convey, Transport, or Connect. Save [Attribute](#) Defines the name of the attribute that will store the station name that is selected, visible when the transfer method is Connect. [Route](#) Time Move time of the entity from its current station to the station determined through this module. Units Time units for route-time parameters.

Definition at line 62 of file [PickStation.h](#).

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

- [/home/rlcancian/Laboratory/Software\\_Lab/IA32\\_Architecture/GccProjects/RebornedGenESyS/2019\\_↔2022\\_GenESyS/plugins/components/PickStation.h](#)
- [/home/rlcancian/Laboratory/Software\\_Lab/IA32\\_Architecture/GccProjects/RebornedGenESyS/2019\\_↔2022\\_GenESyS/plugins/components/PickStation.cpp](#)

## 4.45 PickUp Class Reference

```
#include <PickUp.h>
```

### Public Member Functions

- **PickUp** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.45.1 Detailed Description

Pickup module DESCRIPTION The Pickup module removes a number of consecutive entities from a given queue starting at a specified rank in the queue. The entities that are picked up are added to the end of the incoming entity's group. TYPICAL USES Gathering an order from various queue locations Gathering completed forms for an office order Picking up students at a bus stop for school PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Quantity Number of entities to pick up. [Queue](#) Name Name of the queue from which the entities will be picked up, starting at the specified rank. Starting Rank Starting rank of the entities to pick up from the queue, [Queue](#) Name.

Definition at line 38 of file Pickup.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/PickUp.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/PickUp.cpp

## 4.46 Plugin Class Reference

```
#include <Plugin.h>
```

### Public Member Functions

- **Plugin** (std::string filename\_so\_dll)
- **Plugin** (StaticGetPluginInformation getInformation)
- bool **isIsValidPlugin** () const
- PluginInformation \* **getPluginInfo** () const
- [ModelElement](#) \* **loadNew** ([Model](#) \*model, std::map< std::string, std::string > \*fields)  
*creates a new [ModelElement](#) from fields loaded from a file*
- bool **loadAndInsertNew** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

#### 4.46.1 Detailed Description

A [Plugin](#) represents a dynamically linked component class ([ModelComponent](#)) or element class ([ModelElement](#)); It gives access to a [ModelComponent](#) so it can be used by the model. Classes like [Create](#), [Delay](#), and [Dispose](#) are examples of Plugins. It corresponds directly to the "Expansible" part (the capitalized 'E') of the GenESyS acronymous Plugins are NOT implemented yet

Definition at line 30 of file Plugin.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/Plugin.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/Plugin.cpp

#### 4.47 yy::position Class Reference

A point in a source file.

```
#include <location.hh>
```

##### Public Types

- typedef const std::string [filename\\_type](#)  
*Type for file name.*
- typedef int [counter\\_type](#)  
*Type for line and column numbers.*

##### Public Member Functions

- [position](#) ([filename\\_type](#) \*f=YY\_NULLPTR, [counter\\_type](#) l=1, [counter\\_type](#) c=1)  
*Construct a position.*
- void [initialize](#) ([filename\\_type](#) \*fn=YY\_NULLPTR, [counter\\_type](#) l=1, [counter\\_type](#) c=1)  
*Initialization.*

##### Line and Column related manipulators

- void [lines](#) ([counter\\_type](#) count=1)  
*(line related) Advance to the COUNT next lines.*
- void [columns](#) ([counter\\_type](#) count=1)  
*(column related) Advance to the COUNT next columns.*

##### Public Attributes

- [filename\\_type](#) \* [filename](#)  
*File name to which this position refers.*
- [counter\\_type](#) [line](#)  
*Current line number.*
- [counter\\_type](#) [column](#)  
*Current column number.*



### 4.47.1 Detailed Description

Definition at line 60 of file location.hh.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/location.hh

## 4.48 Process Class Reference

```
#include <Process.h>
```

### Public Member Functions

- **Process** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- void **setPriority** (unsigned short \_priority)
- unsigned short **getPriority** () const
- void **setAllocationType** (unsigned int \_allocationType)
- unsigned int **getAllocationType** () const
- [List](#)< SeizableItem \* > \* **getSeizeRequests** () const
- void **setQueueableItem** (QueueableItem \*\_queueableItem)
- QueueableItem \* **getQueueableItem** () const
- void **setSaveAttribute** (std::string \_saveAttribute)
- std::string **getSaveAttribute** () const
- void **setDelayExpression** (std::string \_delayExpression)
- std::string **delayExpression** () const
- void **setDelayTimeUnit** (Util::TimeUnit \_delayTimeUnit)
- Util::TimeUnit **delayTimeUnit** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

## Additional Inherited Members

### 4.48.1 Detailed Description

This component ...

Definition at line 25 of file Process.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Process.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Process.cpp

## 4.49 Queue Class Reference

```
#include <Queue.h>
```

### Public Types

- enum class **OrderRule** : int { **FIFO** = 1 , **LIFO** = 2 , **HIGHESTVALUE** = 3 , **SMALLESTVALUE** = 4 }

### Public Member Functions

- **Queue** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- void **insertElement** (Waiting \*element)
- void **removeElement** (Waiting \*element)
- unsigned int **size** ()
- Waiting \* **first** ()
- Waiting \* **getAtRank** (unsigned int rank)
- void **setAttributeName** (std::string \_attributeName)
- std::string **getAttributeName** () const
- void **setOrderRule** (OrderRule \_orderRule)
- Queue::OrderRule **getOrderRule** () const
- double **sumAttributesFromWaiting** (Util::identification attributeID)
- double **getAttributeFromWaitingRank** (unsigned int rank, Util::identification attributeID)
- void **initBetweenReplications** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()
- virtual ParserChangesInformation \* **\_getParserChangesInformation** ()

### Additional Inherited Members

#### 4.49.1 Detailed Description

**Queue** module DESCRIPTION This data module may be utilized to change the ranking rule for a specified queue. The default ranking rule for all queues is First In, First Out unless otherwise specified in this module. There is an additional field that allows the queue to be defined as shared. TYPICAL USES Stack of work waiting for a resource at a **Process** module Holding area for documents waiting to be collated at a **Batch** module Prompt Description Name The name of the queue whose characteristics are being defined. This name must be unique. Type Ranking rule for the queue, which can be based on an attribute. Types include First In, First Out; Last In, First Out; Lowest **Attribute** Value (first); and Highest **Attribute** Value (first). A low attribute value would be 0 or 1, while a high value may be 200 or 300. **Attribute** Name **Attribute** that will be evaluated for the Lowest **Attribute** Value or Highest **Attribute** Value types. Entities with lowest or highest values of the attribute will be ranked first in the queue, with ties being broken using the First In, First Out rule. Shared Check box that determines whether a specific queue is used in multiple places within the simulation model. Shared queues can only be used for seizing resources (for example, with the **Seize** module from the Advanced **Process** panel). Report Statistics Specifies whether or not statistics will be collected automatically and stored in the report database for this queue.

Definition at line 91 of file Queue.h.

#### 4.49.2 Member Function Documentation

##### 4.49.2.1 **\_createInternalElements()** void Queue::\_createInternalElements ( ) [protected], [virtual]

This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal **StatisticsCollectors**, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist before checking such expression

Reimplemented from **ModelElement**.

Definition at line 133 of file Queue.cpp.

**4.49.2.2 `_getParserChangesInformation()`** `ParserChangesInformation * Queue::_getParserChangesInformation ( ) [protected], [virtual]`

This method returns all changes in the parser that are needed by plugins of this ModelElements. When connecting a new plugin, ParserChangesInformation are used to change parser source code, which is after compiled and dinamically linked to simulator kernel to reflect the changes

Reimplemented from [ModelElement](#).

Definition at line 151 of file Queue.cpp.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵2022\_GenESyS/plugins/elements/Queue.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵2022\_GenESyS/plugins/elements/Queue.cpp

## 4.50 Record Class Reference

```
#include <Record.h>
```

### Public Member Functions

- **Record** ([Model](#) \*model, std::string name="")
- void **setFilename** (std::string filename)
- std::string **getFilename** () const
- void **setExpression** (const std::string expression)
- std::string **getExpression** () const
- void **setExpressionName** (std::string expressionName)
- std::string **getExpressionName** () const
- [StatisticsCollector](#) \* **getCstatExpression** () const
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

## Additional Inherited Members

### 4.50.1 Detailed Description

**Record** module DESCRIPTION This module is used to collect statistics in the simulation model. Various types of observational statistics are available, including time between exits through the module, entity statistics (such as time or costing), general observations, and interval statistics (from some time stamp to the current simulation time). A count type of statistic is available as well. Tally and **Counter** sets can also be specified. TYPICAL USES Collect the number of jobs completed each hour Count how many orders have been late being fulfilled **Record** the time spent by priority customers in the main check-out line PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Type Type of observational (tally) or count statistic to be generated. Count will increase or decrease the value of the named statistic by the specified value. **Entity** Statistics will generate general entity statistics, such as time and costing/duration information. Time Interval will calculate and record the difference between a specified attribute's value and current simulation time. Time Between will track and record the time between entities entering the module. Expression will record the value of the specified expression. **Attribute** Name Name of the attribute whose value will be used for the interval statistics. Applies only when Type is Interval. Value Value that will be recorded to the observational statistic when Type is Expression or added to the counter when Type is Count. Tally Name This field defines the symbol name of the tally into which the observation is to be recorded. Applies only when Type is Time Interval, Time Between, or Expression. **Counter** This field defines the symbol name of the counter to Name increment/decrement. Applies only when Type is **Counter**. **Record** into **Set** Check box to specify whether or not a tally or counter set will be used. Tally **Set** Name Name of the tally set that will be used to record the observational-type statistic. Applies only when Type is Time Interval, Time Between, or Expression. **Counter Set** Name Name of the counter set that will be used to record the count-type statistic. Applies only when Type is Count. **Set** Index Index into the tally or counter set.

Definition at line 62 of file Record.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Record.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Record.cpp

## 4.51 Release Class Reference

```
#include <Release.h>
```

### Public Member Functions

- **Release** (**Model** \*model, std::string name="")
- virtual std::string **show** ()
- void **setPriority** (unsigned short \_priority)
- unsigned short **priority** () const
- **List**< SeizableItem \* > \* **getReleaseRequests** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static **ModelComponent** \* **LoadInstance** (**Model** \*model, std::map< std::string, std::string > \*fields)

## Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.51.1 Detailed Description

[Release](#) module DESCRIPTION The [Release](#) module is used to release units of a resource that an entity previously has seized. This module may be used to release individual resources or may be used to release resources within a set. For each resource to be released, the name and quantity to release are specified. When the entity enters the [Release](#) module, it gives up control of the specified resource(s). Any entities waiting in queues for those resources will gain control of the resources immediately. TYPICAL USES Finishing a customer order (release the operator) Completing a tax return (release the accountant) Leaving the hospital (release the doctor, nurse, hospital room) PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Type Type of resource for releasing, either specifying a particular resource, or selecting from a pool of resources (that is, a resource set). The resource name may also be specified by an expression or attribute value. [Resource](#) Name Name of the resource that will be released. [Set](#) Name Name of the resource set from which a member will be released. [Attribute](#) Name Name of the attribute that specifies the resource name to be released. Expression Name of the expression that specifies the name of the resource to be released. Quantity Number of resources of a given name or from a given set that will be released. For sets, this value specifies only the number of a selected resource that will be released (based on the resource's capacity), not the number of members to be released within the set. [Release](#) Rule Method of determining which resource within a set to release. Last Member Seized and First Member Seized will release the last/first member from within the set that was seized. Specific member indicates that a member number or attribute (with a member number value) will be used to specify the member to release. [Set](#) Index Member index of the resource set that the entity will release.

Definition at line 64 of file Release.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Release.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Release.cpp

## 4.52 Remove Class Reference

```
#include <Remove.h>
```

## Public Member Functions

- **Remove** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static ModelComponent \* **LoadInstance** (Model \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** (Entity \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.52.1 Detailed Description

**Remove** module DESCRIPTION The **Remove** module removes a single entity from a specified position in a queue and sends it to a designated module. When an entity arrives at a **Remove** module, it removes the entity from the specified queue and sends it to the connected module. The rank of the entity signifies the location of the entity within the queue. The entity that caused the removal proceeds to the next module specified and is processed before the removed entity. TYPICAL USES Removing an order from a queue that is due to be completed next Calling a patient from a waiting room for an examination Retrieving the next order to be processed from a pile of documents Prompt Description Name Unique module identifier displayed on the module shape. **Queue** Name Name of the queue from which the entity will be removed. Rank of **Entity** Rank of the entity to remove from within the queue.

Definition at line 37 of file Remove.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Remove.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Remove.cpp

## 4.53 Resource Class Reference

```
#include <Resource.h>
```

### Public Types

- enum class **ResourceState** : int {  
  **IDLE** = 1 , **BUSY** = 2 , **FAILED** = 3 , **INACTIVE** = 4 ,  
  **OTHER** = 5 }
- typedef std::function< void(**Resource** \*) > **ResourceEventHandler**
- typedef std::pair< std::pair< ResourceEventHandler, ModelComponent \* >, unsigned int > **Sorted**↵  
  **ResourceEventHandler**

## Public Member Functions

- **Resource** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- void **seize** (unsigned int quantity, double tnow)
- void **release** (unsigned int quantity, double tnow)
- void **setResourceState** (ResourceState \_resourceState)
- Resource::ResourceState **getResourceState** () const
- void **setCapacity** (unsigned int \_capacity)
- unsigned int **getCapacity** () const
- void **setCostBusyHour** (double \_costBusyHour)
- double **getCostBusyHour** () const
- void **setCostIdleHour** (double \_costIdleHour)
- double **getCostIdleHour** () const
- void **setCostPerUse** (double \_costPerUse)
- double **getCostPerUse** () const
- unsigned int **getNumberBusy** () const
- void **addReleaseResourceEventHandler** (ResourceEventHandler eventHandler, [ModelComponent](#) \*component, unsigned int priority)
- double **getLastTimeSeized** () const

## Static Public Member Functions

- template<typename Class >  
static ResourceEventHandler **SetResourceEventHandler** (void(Class::\*function)([Resource](#) \*), Class \*object)
- static PluginInformation \* **GetPluginInformation** ()
- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

## Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()
- virtual void **\_initBetweenReplications** ()

## Additional Inherited Members

### 4.53.1 Detailed Description

[Resource](#) module DESCRIPTION This data module defines the resources in the simulation system, including costing information and resource availability. Resources may have a fixed capacity that does not vary over the simulation run or may operate based on a schedule. [Resource](#) failures and states can also be specified in this module. TYPICAL USES Equipment (machinery, cash register, phone line) People (clerical, order processing, sales clerks, operators) PROMPTS Prompt Description Name The name of the resource whose characteristics are being defined. This name must be unique. Type Method for determining the capacity for a resource. Fixed Capacity will not change during the simulation run. Based on [Schedule](#) signifies that a [Schedule](#) module is used to specify the capacity and duration information for the resource. Capacity Number of resource units of a given name that are available to the system for processing. Applies only when Type is Fixed Capacity. [Schedule](#) Name Identifies the name of the schedule to be used by the resource. The schedule defines the capacity of a resource for a given period of time. Applies only when type is [Schedule](#). [Schedule](#) Rule Dictates when the actual capacity change is to occur



when a decrease in capacity is required for a busy resource unit. Applies only when Type is [Schedule](#). Busy/Hour Cost per hour of a resource that is processing an entity. The resource becomes busy when it is originally allocated to an entity and becomes idle when it is released. During the time when it is busy, cost will accumulate based on the busy/hour cost. The busy cost per hour is automatically converted to the appropriate base time unit specified within the Replication Parameters page of the Run > Setup menu item. Idle/Hour Cost per hour of a resource that is idle. The resource is idle while it is not processing an entity. During the time when it is idle, cost will accumulate based on the idle/hour cost. The idle cost per hour is automatically converted to the appropriate base time unit specified within the Replication Parameters page of the Run > Setup menu item. Per Use Cost of a resource on a usage basis, regardless of the time for which it is used. Each time the resource is allocated to an entity, it will incur a per-use cost. StateSet Name Name of states that the resource may be assigned during the simulation run. Initial State Initial state of a resource. If specified, the name must be defined within the repeat group of state names. This field is shown only when a StateSet Name is defined. Failures Lists all failures that will be associated with the resource. [Failure](#) Name—Name of the failure associated with the resource. [Failure](#) Rule—Behavior that should occur when a failure is to occur for a busy resource unit. Report Statistics Specifies whether or not statistics will be collected automatically and stored in the report database for this resource.

Definition at line 81 of file `Resource.h`.

### 4.53.2 Member Function Documentation

**4.53.2.1 `_createInternalElements()`** `void Resource::_createInternalElements ( ) [protected], [virtual]`

This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal `StatisticsCollectors`, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist before checking such expression

Reimplemented from [ModelElement](#).

Definition at line 192 of file `Resource.cpp`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/elements/Resource.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/elements/Resource.cpp`

## 4.54 `Sampler_if::RNG_Parameters` Struct Reference

```
#include <Sampler_if.h>
```

### 4.54.1 Detailed Description

class that encapsulates attributes required to generate random numbers, which depends on the generation method used.

Definition at line 26 of file `Sampler_if.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/kernel/statistics/Sampler_if.h`

## 4.55 Route Class Reference

```
#include <Route.h>
```

### Public Types

- enum class **DestinationType** : int { **Station** = 0 , **BySequence** = 1 }

### Public Member Functions

- **Route** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- void **setStation** (Station \*\_station)
- Station \* **getStation** () const
- void **setRouteTimeExpression** (std::string \_routeTimeExpression)
- std::string **getRouteTimeExpression** () const
- void **setRouteTimeUnit** (Util::TimeUnit \_routeTimeUnit)
- Util::TimeUnit **getRouteTimeUnit** () const
- void **setRouteDestinationType** (DestinationType \_routeDestinationType)
- Route::DestinationType **getRouteDestinationType** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_createInternalElements** ()

*This method is necessary only for those components that instantiate internal elements that must exist before simulation starts and even before model checking. That's the case of components that have internal StatisticsCollectors, since others components may refer to them as expressions (as in "TVAG(ThisCSTAT)") and therefore the element must exist when checking such expression.*

## Additional Inherited Members

### 4.55.1 Detailed Description

**Route** module DESCRIPTION The **Route** module transfers an entity to a specified station or the next station in the station visitation sequence defined for the entity. A delay time to transfer to the next station may be defined. When an entity enters the **Route** module, its Station attribute (Entity.Station) is set to the destination station. The entity is then sent to the destination station, using the route time specified. If the station destination is entered as By **Sequence**, the next station is determined by the entity's sequence and step within the set (defined by special-purpose attributes Entity.Sequence and Entity.Jobstep, respectively). TYPICAL USES Send a part to its next processing station based on its routing slip Send an account balance call to an account agent Send restaurant customers to a specific table PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart. **Route** Time Travel time from the entity's current location to the destination station. Units Time units for route-time parameters. Destination Type Method for determining the entity destination location. Selection of By **Sequence** requires that the entity has been assigned a sequence name and that the sequence itself has been defined. Station Name Name of the individual destination station. **Attribute** Name Name of the attribute that stores the station name to which entities will route. Expression Expression that is evaluated to the station name where entities will route.

Definition at line 53 of file Route.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Route.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Route.cpp

## 4.56 `Sampler_if` Class Reference

```
#include <Sampler_if.h>
```

### Classes

- struct **RNG\_Parameters**

### Public Member Functions

- virtual double **random** ()=0
- virtual double **sampleBeta** (double alpha, double beta, double infLimit, double supLimit)=0
- virtual double **sampleBeta** (double alpha, double beta)=0
- virtual double **sampleErlang** (double mean, int M)=0
- virtual double **sampleExponential** (double mean)=0
- virtual double **sampleGamma** (double mean, double alpha)=0
- virtual double **sampleGumbell** (double mode, double scale)=0
- virtual double **sampleLogNormal** (double mean, double stddev)=0
- virtual double **sampleNormal** (double mean, double stddev)=0
- virtual double **sampleTriangular** (double min, double mode, double max)=0
- virtual double **sampleUniform** (double min, double max)=0
- virtual double **sampleWeibull** (double alpha, double scale)=0
- virtual double **sampleBinomial** (int trials, double p)=0
- virtual double **sampleBernoulli** (double p)=0
- virtual double **sampleDiscrete** (double acumProb, double value,...)=0
- virtual double **sampleDiscrete** (double acumProb, double \*prob, double \*value, int size)=0
- virtual double **sampleGeometric** (double p)=0
- virtual void **setRNGparameters** (**RNG\_Parameters** \*param)=0
- virtual **RNG\_Parameters** \* **getRNGparameters** () const =0

### 4.56.1 Detailed Description

Interface that describes the methods to be implemented by classes that generate random values that follow a specific probability distribution.

Definition at line 20 of file `Sampler_if.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵  
2022_GenESyS/kernel/statistics/Sampler_if.h`

## 4.57 Schedule Class Reference

```
#include <Schedule.h>
```

### Public Member Functions

- **Schedule** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

### 4.57.1 Detailed Description

[Schedule](#) module DESCRIPTION This data module may be used in conjunction with the [Resource](#) module to define an operating schedule for a resource or with the [Create](#) module to define an arrival schedule. Additionally, a schedule may be used and referenced to factor time delays based on the simulation time. TYPICAL USES Work schedule for staff, including breaks Breakdown patterns for equipment Volume of customers arriving at a store Learning-curve factors for new workers PROMPTS [File](#) Read Time Specifies when to read the values from the file into the variable. If you select PreCheck, the values for the variable are read while the model is still in Edit mode (prior to the model being checked and compiled). If you select BeginSimulation, values are read when the model is compiled, prior to the first replication. If you select BeginReplication, values are read prior to each replication. Initial Values Lists the initial value or values of the variable. You can assign new values to the variable at different stages of the model by using the [Assign](#) module. Initial Value [Variable](#) value at the start of the simulation. Prompt Description Name The name of the schedule being defined. This name must be unique. Type Type of schedule being defined. This may be Capacity-related (for resource schedules), Arrival-related (for the [Create](#) module), or Other (miscellaneous time delays or factors) Time Units Time units used for the time-duration information. Scale

Factor Method of scaling the schedule for increases or decreases in Arrival/Other values. The specified Value fields will be multiplied by the scale factor to determine the new values. Not available for Capacity-type schedules. Durations Lists the value and duration pairs for the schedule. Values can be capacity, arrival, or other type values, while the duration is specified in time units. [Schedule](#) pairs will repeat after all durations have been completed, unless the last duration is left blank (infinite). [Schedule](#) data can be entered graphically using the graphical schedule editor or manually using the Value/ Duration fields. Value Represents either the capacity of a resource (if Type is Capacity), arrival rate (if Type is Arrival), or some other value (if Type is Other). Examples of Other may be a factor that is used in a delay expression to scale a delay time during various parts of the day. Duration Time duration for which a specified Value will be valid.

Definition at line 72 of file Schedule.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵2022\_GenESyS/plugins/elements/Schedule.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵2022\_GenESyS/plugins/elements/Schedule.cpp

## 4.58 Search Class Reference

```
#include <Search.h>
```

### Public Member Functions

- **Search** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.58.1 Detailed Description

**Search** module DESCRIPTION The **Search** module searches a queue, a group (batch), or an expression to find the entity rank (for entities in a queue or group) or the value of the global variable J that satisfies the specified search condition. When searching a queue or group, the value of the global system variable J is set to the rank of the first entity that satisfies **Search** Condition, or to 0 if **Search** Condition is not satisfied. When searching an expression, the global system variable J is set to the value of the first index value that satisfies the search condition or to zero if no value of J in the specified range satisfies the search condition. When an entity arrives at a **Search** module, the index J is set to the starting index and the search condition is then checked. If the search condition is satisfied, the search ends and the current value of J is retained. Otherwise, the value of J is increased or decreased and the condition is rechecked. This process repeats until the search condition is satisfied or the ending value is reached. If the condition is not met or there are no entities in the queue or group, J is set equal to 0. TYPICAL USES Looking for a particular order number in a queue Searching a group for a certain part type Determining which process to enter based on availability of resources (search an expression) Prompt Description Name Unique module identifier displayed on the module shape. Type Determination of what will be searched. **Search** options include entities in a queue, entities within a group (batch) or some expression(s). **Queue** Name Name of the queue that will be searched. Applies only when the Type is **Search** a **Queue**. Starting Value Starting rank in the queue or group or starting value for J in an expression. Ending Value Ending rank in the queue or group or ending value for J in an expression. **Search** Condition Condition containing the index J for searching expressions or containing an attribute name(s) for searching queues or batches.

Definition at line 55 of file Search.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵2022\_GenESyS/plugins/components/Search.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵2022\_GenESyS/plugins/components/Search.cpp

## 4.59 Seize Class Reference

```
#include <Seize.h>
```

### Public Member Functions

- **Seize** (**Model** \*model, std::string name="")
- virtual std::string **show** ()
- void **setPriority** (unsigned short \_priority)
- unsigned short **getPriority** () const
- void **setAllocationType** (unsigned int \_allocationType)
- unsigned int **getAllocationType** () const
- void **setQueue** (**Queue** \*queue)
- *Deprecated.*
- **List**< **SeizableItem** \* > \* **getSeizeRequests** () const
- void **setQueueableItem** (**QueueableItem** \*\_queueableItem)
- **QueueableItem** \* **getQueueableItem** () const
- void **setSaveAttribute** (std::string \_saveAttribute)
- std::string **getSaveAttribute** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.59.1 Detailed Description

**Seize** module DESCRIPTION The [Seize](#) module allocates units of one or more resources to an entity. The [Seize](#) module may be used to seize units of a particular resource, a member of a resource set, or a resource as defined by an alternative method, such as an attribute or expression. When an entity enters this module, it waits in a queue (if specified) until all specified resources are available simultaneously. Allocation type for resource usage is also specified. TYPICAL USES Beginning a customer order (seize the operator) Starting a tax return (seize the accountant) Being admitted to hospital (seize the hospital room, nurse, doctor) PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Allocation Determines to which category the resource usage cost will be allocated for an entity going through the [Seize](#) module. Priority Priority value of the entity waiting at this module for the resource(s) specified if one or more entities from other modules are waiting for the same resource(s). Type Type of resource for seizing, either specifying a particular resource, or selecting from a pool of resources (that is, a resource set). The name of the resource may also be specified as an attribute value or within an expression. [Resource](#) Name Name of the resource that will be seized. [Set](#) Name Name of the resource set from which a member will be seized. [Attribute](#) Name Name of the attribute that stores the resource name to be seized. Expression Expression that evaluates to a resource name to be seized. Quantity Number of resources of a given name or from a given set that will be seized. For sets, this value specifies only the number of a selected resource that will be seized (based on the resource's capacity), not the number of members to be seized within the set. Selection Rule Method of selecting among available resources in a set. Cyclical will cycle through available members (for example, 1-2-3-1-2-3). Random will randomly select a member. Preferred Order will always select the first available member (for example, 1, if available; then 2, if available; then 3). Specific Member requires an input attribute value to specify which member of the set (previously saved in the Save [Attribute](#) field). Largest Remaining Capacity and Smallest Number Busy are used for resources with multiple capacity. Save [Attribute](#) [Attribute](#) name used to store the index number into the set of the member that is chosen. This attribute can later be referenced with the Specific Member selection rule. [Set](#) Index Index value into the set that identifies the number into the set of the member requested. If an attribute name is used, the entity must have a value for the attribute before utilizing this option. [Resource](#) State State of the resource that will be assigned after the resource is seized. The resource state must be defined with the [Resource](#) module. [Queue](#) Type Determines the type of queue used to hold the entities while waiting to seize the resource(s). If [Queue](#) is selected, the queue name is specified. If [Set](#) is selected, the queue set and member in the set are specified. If Internal is selected, an internal queue is used to hold all waiting entities. [Attribute](#) and Expression are additional methods for defining the queue to be used. [Queue](#) Name This field is visible only if [Queue](#) Type is [Queue](#), and it defines the symbol name of the queue. [Set](#) Name This field is visible only if [Queue](#) Type is [Set](#), and it defines the queue set that contains the queue being referenced. [Set](#) Index This field is visible only if [Queue](#) Type is [Set](#), and it defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3. [Attribute](#) This field is visible only if [Queue](#) Type is [Attribute](#). The attribute entered in this field will be evaluated to indicate which queue is to be used. Expression This field is visible only if [Queue](#) Type is Expression. The expression entered in this field will be evaluated to indicate which queue is to be used.

Definition at line 122 of file Seize.h.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/components/Seize.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/components/Seize.cpp`

## 4.60 yy::genesyspp\_parser::semantic\_type Class Reference

```
#include <GenesysParser.h>
```

### Public Types

- typedef [semantic\\_type](#) [self\\_type](#)  
*Type of \*this.*

### Public Member Functions

- [semantic\\_type](#) () YY\_NOEXCEPT  
*Empty construction.*
- template<typename T >  
[semantic\\_type](#) (YY\_RVREF(T) t)  
*Construct and fill.*
- [~semantic\\_type](#) () YY\_NOEXCEPT  
*Destruction, allowed only if empty.*
- template<typename T >  
T & [emplace](#) ()  
*Instantiate an empty T in here.*
- template<typename T >  
T & [emplace](#) (const T &t)  
*Instantiate a T in here from t.*
- template<typename T >  
T & [build](#) ()
- template<typename T >  
T & [build](#) (const T &t)
- template<typename T >  
T & [as](#) () YY\_NOEXCEPT  
*Accessor to a built T.*
- template<typename T >  
const T & [as](#) () const YY\_NOEXCEPT  
*Const accessor to a built T (for printer).*
- template<typename T >  
void [swap](#) ([self\\_type](#) &that) YY\_NOEXCEPT
- template<typename T >  
void [move](#) ([self\\_type](#) &that)
- template<typename T >  
void [copy](#) (const [self\\_type](#) &that)  
*Copy the content of that to this.*
- template<typename T >  
void [destroy](#) ()  
*Destroy the stored T.*



### 4.60.1 Detailed Description

A buffer to store and retrieve objects.

Sort of a variant, but does not keep track of the nature of the stored data, since that knowledge is available via the current parser state.

Definition at line 231 of file GenesysParser.h.

### 4.60.2 Member Function Documentation

**4.60.2.1 build()** [1/2] `template<typename T >`  
`T& yy::genesyspp_parser::semantic_type::build ( )`

Instantiate an empty *T* in here. Obsolete, use `emplace`.

Definition at line 304 of file GenesysParser.h.

**4.60.2.2 build()** [2/2] `template<typename T >`  
`T& yy::genesyspp_parser::semantic_type::build (`  
`const T & t )`

Instantiate a *T* in here from *t*. Obsolete, use `emplace`.

Definition at line 313 of file GenesysParser.h.

**4.60.2.3 move()** `template<typename T >`  
`void yy::genesyspp_parser::semantic_type::move (`  
`self_type & that )`

Move the content of *that* to this.

Destroys *that*.

Definition at line 362 of file GenesysParser.h.

```

4.60.2.4 swap()  template<typename T >
void yy::genesyspp_parser::semantic_type::swap (
    self_type & that )

```

Swap the content with *that*, of same type.

Both variants must be built beforehand, because swapping the actual data requires reading it (with [as\(\)](#)), and this is not possible on unconstructed variants: it would require some dynamic testing, which should not be the variant's responsibility. Swapping between built and (possibly) non-built is done with [self\\_type::move](#) ().

Definition at line 350 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)

## 4.61 Separate Class Reference

```
#include <Separate.h>
```

### Public Member Functions

- **Separate** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.61.1 Detailed Description

[Separate](#) module DESCRIPTION This module can be used to either copy an incoming entity into multiple entities or to split a previously batched entity. Rules for allocating costs and times to the duplicate are also specified. Rules for attribute assignment to member entities are specified as well. When splitting existing batches, the temporary representative entity that was formed is disposed and the original entities that formed the group are recovered. The entities proceed sequentially from the module in the same order in which they originally were added to the batch. When duplicating entities, the specified number of copies is made and sent from the module. The original incoming entity also leaves the module. TYPICAL USES Send individual entities to represent boxes removed from a container Send an order both to fulfillment and billing for parallel processing [Separate](#) a previously batched set of documents PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Type Method of separating the incoming entity. Duplicate Original will simply take the original entity and make some number of identical duplicates. Split Existing [Batch](#) requires that the incoming entity be a temporarily batched entity using the [Batch](#) module. The original entities from the batch will be split. Percent Cost to Duplicates Allocation of costs and times of the incoming entity to the outgoing duplicates. This value is specified as a percentage of the original entity's costs and times (between 0-100). The percentage specified will be split evenly between the duplicates, while the original entity will retain any remaining cost/time percentage. Visible only when Type is Duplicate Original.

of Duplicates Number of outgoing entities that will leave the module, in

addition to the original incoming entity. Applies only when Type is Duplicate Original. Member Attributes Method of determining how to assign the representative entity attribute values to the original entities. These options relate to six of the special-purpose attributes (Entity.Type, Entity.Picture, Entity.Sequence, Entity.Station, Entity.Jobstep, and Entity.HoldCostRate) and all user-defined attributes. Applies only when Type is Split Existing [Batch](#). [Attribute](#) Name Name of representative entity attribute(s) that are assigned to original entities of the group. Applies only when Member Attributes is Take Specific Representative Values.

Definition at line 65 of file Separate.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Separate.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/plugins/components/Separate.cpp

## 4.62 Sequence Class Reference

```
#include <Sequence.h>
```

### Public Member Functions

- **Sequence** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- [List](#)< SequenceStep \* > \* **getSteps** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.62.1 Detailed Description

**Sequence** module DESCRIPTION The **Sequence** module is used to define a sequence for entity flow through the model. A sequence consists of an ordered list of stations that an entity will visit. For each station in the visitation sequence, attributes and variables may be assigned values. Each station in the visitation sequence is referred to as a step (or jobstep) in the sequence. Three special-purpose attributes are provided for all entities. The **Sequence** attribute (Entity.Sequence) defines the sequence that an entity is to follow; a value of 0 indicates that the entity is not following any sequence. In order for an entity to follow a sequence, its **Sequence** attribute must be assigned a value (for example, in the **Assign** module). The **Jobstep** attribute (Entity.Jobstep) stores the entity's current step number in the sequence. This value is updated automatically each time an entity is transferred. You typically do not need to assign explicitly a value to **Jobstep** in the model. The **PlannedStation** attribute (Entity.PlannedStation) stores the number of the station associated with the next jobstep in the sequence. This attribute is not user-assignable. It is automatically updated whenever Entity.Sequence or Entity.JobStep changes, or whenever the entity enters a station. Jobstep names must be globally unique. TYPICAL USES Define a routing path for part processing Define a sequence of steps patients must take upon arrival at an emergency room

Definition at line 100 of file Sequence.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/elements/Sequence.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/elements/Sequence.cpp

## 4.63 Set Class Reference

```
#include <Set.h>
```

### Public Member Functions

- **Set** (**Model** \*model, std::string name="")
- virtual std::string **show** ()
- void **setSetOfType** (std::string \_setOfType)
- std::string **getSetOfType** () const
- **List**< **ModelElement** \* > \* **getElementSet** () const

### Static Public Member Functions

- static **ModelElement** \* **LoadInstance** (**Model** \*model, std::map< std::string, std::string > \*fields)
- static PluginInformation \* **GetPluginInformation** ()

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual ParserChangesInformation \* **\_getParserChangesInformation** ()

## Additional Inherited Members

### 4.63.1 Detailed Description

**Set** module DESCRIPTION This data module defines various types of sets, including resource, counter, tally, entity type, and entity picture. **Resource** sets can be used in the **Process** modules (and **Seize**, **Release**, **Enter**, and **Leave** of the Advanced **Process** and Advanced Transfer panels). **Counter** and Tally sets can be used in the **Record** module. **Queue** sets can be used with the **Seize**, **Hold**, **Access**, Request, **Leave**, and Allocate modules of the Advanced **Process** and Advanced Transfer panels. TYPICAL USES Machines that can perform the same operations in a manufacturing facility Supervisors, check-out clerks in a store Shipping clerks, receptionists in an office **Set** of pictures corresponding to a set of entity types PROMPTS Prompt Description Name The unique name of the set being defined. Type Type of set being defined. Members Repeat group that specifies the resource members with the set. The order of listing the members within the repeat group is important when using selection rules such as Preferred Order and Cyclical. **Resource** Name Name of the resource to include in the resource set. Applies only when Type is **Resource**. Tally Name Name of the tally within the tally set. Applies only when Type is Tally. **Counter** Name Name of the counter within the counter set. Applies only when Type is **Counter**. **Entity** Type Name of the entity type within the entity type set. Applies only when Type is **Entity**. Picture Name Name of the picture within the picture set. Applies only when Type is **Entity** Picture.

Definition at line 56 of file Set.h.

### 4.63.2 Member Function Documentation

**4.63.2.1** `_getParserChangesInformation()` `ParserChangesInformation * Set::_getParserChangesInformation ( ) [protected], [virtual]`

This method returns all changes in the parser that are needed by plugins of this ModelElements. When connecting a new plugin, ParserChangesInformation are used to change parser source code, which is after compiled and dynamically linked to simulator kernel to reflect the changes

Reimplemented from [ModelElement](#).

Definition at line 100 of file Set.cpp.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔2022_GenESyS/plugins/elements/Set.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↔2022_GenESyS/plugins/elements/Set.cpp`

## 4.64 Signal Class Reference

```
#include <Signal.h>
```

### Public Member Functions

- **Signal** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

## Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static ModelComponent \* **LoadInstance** (Model \*model, std::map< std::string, std::string > \*fields)

## Protected Member Functions

- virtual void **\_execute** (Entity \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.64.1 Detailed Description

**Signal** module DESCRIPTION The **Signal** module sends a signal value to each **Hold** module in the model set to Wait for **Signal** and releases the maximum specified number of entities. When an entity arrives at a **Signal** module, the signal is evaluated and the signal code is sent. At this time, entities at **Hold** modules that are waiting for the same signal are removed from their queues. The entity sending the signal continues processing until it encounters a delay, enters a queue, or is disposed. TYPICAL USES Analyzing traffic patterns at an intersection (signal when the light turns green) Signaling an operator to complete an order that was waiting for a component part PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. **Signal** Value Value of the signal to be sent to entities in **Hold** modules. Limit Maximum number of entities that are to be released from any **Hold** modules when the signal is received.

Definition at line 38 of file Signal.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Signal.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Signal.cpp

## 4.65 SimulationControl Class Reference

```
#include <SimulationControl.h>
```

### Public Member Functions

- **SimulationControl** (std::string type, std::string name, GetterMember getterMember, SetterMember setter↔  
Member, std::string parent="")
- std::string **show** ()
- void **setValue** (double value)

## Additional Inherited Members

### 4.65.1 Detailed Description

Represents any possible parameter or control for a simulation. Any element or event the model can declare one of its own attribute as a simulation control. It just have to create a [SimulationControl](#) object, passing the access to the methods that gets and sets the control value and including this [SimulationControl](#) in the corresponding list of the model

Definition at line 24 of file SimulationControl.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/SimulationControl.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/SimulationControl.cpp

## 4.66 SimulationEvent Class Reference

```
#include <OnEventManager.h>
```

### Public Member Functions

- void **setSimulatedTime** (double simulatedTime)
- double **getSimulatedTime** () const
- void **setStopRequested** (bool stopRequested)
- bool **isStopRequested** () const
- void **setPauseRequested** (bool pauseRequested)
- bool **isPauseRequested** () const
- void **setPaused** (bool Paused)
- bool **isPaused** () const
- void **setRunning** (bool Running)
- bool **isRunning** () const
- void **setCustomObject** (void \*customObject)
- void \* **getCustomObject** () const
- void **setCurrentReplicationNumber** (unsigned int currentReplicationNumber)
- unsigned int **getCurrentReplicationNumber** () const
- void **setCurrentEvent** ([Event](#) \*currentEvent)
- [Event](#) \* **getCurrentEvent** () const
- void **setCurrentInputNumber** (unsigned int currentInputNumber)
- unsigned int **getCurrentInputNumber** () const
- void **setCurrentComponent** ([ModelComponent](#) \*currentComponent)
- [ModelComponent](#) \* **getCurrentComponent** () const
- void **setCurrentEntity** ([Entity](#) \*currentEntity)
- [Entity](#) \* **getCurrentEntity** () const
- void **setEntityCreated** ([Entity](#) \*entityCreated)
- [Entity](#) \* **getEntityCreated** () const
- void **setDestinationComponent** ([ModelComponent](#) \*destinationComponent)
- [ModelComponent](#) \* **getDestinationComponent** () const

## Friends

- class **ModelSimulation**

### 4.66.1 Detailed Description

Stores an event that happened on a specific replication

Definition at line 30 of file OnEventManager.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/OnEventManager.h

## 4.67 SimulationReporterDefaultImpl1 Class Reference

```
#include <SimulationReporterDefaultImpl1.h>
```

### Public Member Functions

- **SimulationReporterDefaultImpl1** ([ModelSimulation](#) \*simulation, [Model](#) \*model, [List](#)< [ModelElement](#) \* > \*statsCountersSimulation)
- virtual void **showReplicationStatistics** ()
- virtual void **showSimulationStatistics** ()
- virtual void **showSimulationResponses** ()
- virtual void **showSimulationControls** ()

### 4.67.1 Detailed Description

Class that implements SimulationReporter\_if interface and is responsible for building and showing replication and simulation reports

Definition at line 26 of file SimulationReporterDefaultImpl1.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/SimulationReporterDefaultImpl1.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/SimulationReporterDefaultImpl1.cpp

## 4.68 SimulationResponse Class Reference

```
#include <SimulationResponse.h>
```



**Public Member Functions**

- **SimulationResponse** (std::string type, std::string name, GetterMember getterMember, std::string parent="")
- std::string **show** ()
- double **getValue** ()
- std::string **getName** () const
- void **setName** (std::string name)
- std::string **getType** () const
- std::string **getParent** () const

**Protected Attributes**

- std::string **\_type**
- std::string **\_parent**
- std::string **\_name**
- GetterMember **\_getterMemberFunction**

**4.68.1 Detailed Description**

Represents any possible response of a simulation. Any element or event the model can declare one of its own attribute as a simulation response. It just has to create a [SimulationResponse](#) object, passing the access to the method that gets the response value and including this [SimulationResponse](#) in the corresponding list of the model

Definition at line 25 of file SimulationResponse.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/SimulationResponse.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/SimulationResponse.cpp

**4.69 SimulationScenario Class Reference**

```
#include <SimulationScenario.h>
```

**Public Member Functions**

- bool **startSimulation** ([Simulator](#) \*sim, std::string \*errorMessage)
- std::list< std::pair< std::string, double > \* > \* **getResponseValues** () const
- std::list< std::pair< std::string, double > \* > \* **getControlValues** () const
- double **getResponseValue** (const std::string &responseName)
- void **setModelFilename** (std::string \_modelFilename)
- std::string **getModelFilename** () const
- void **setScenarioName** (std::string \_name)
- std::string **getScenarioName** () const
- void **setScenarioDescription** (std::string \_scenarioDescription)
- std::string **getScenarioDescription** () const
- std::list< std::string > \* **getSelectedControls** () const
- double **getControlValue** (const std::string &controlName)
- std::list< std::string > \* **getSelectedResponses** () const
- void **setSelectedControls** (std::list< std::string > \*selectedControls)
- void **setControl** (std::string name, double value) const

### 4.69.1 Detailed Description

Represents a scenario where a specific model (defined my ModelFilename) will be simulated. To each scenario will be associated a set of [SimulationControl](#) and [SimulationResponse](#), and their values are set to the scenario by the ProcessAnalyser.

Definition at line 26 of file SimulationScenario.h.

### 4.69.2 Member Function Documentation

**4.69.2.1 `getResponseValues()`** `std::list< std::pair< std::string, double > * > * SimulationScenario::getResponseValues ( ) const`

The final result of the simulationScenario

Definition at line 57 of file SimulationScenario.cpp.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_2022_GenESyS/kernel/simulator/SimulationScenario.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_2022_GenESyS/kernel/simulator/SimulationScenario.cpp`

## 4.70 Simulator Class Reference

```
#include <Simulator.h>
```

### Public Member Functions

- `std::string getVersion () const`
- `unsigned int getVersionNumber () const`
- `std::string getName () const`
- `LicenceManager * getLicenceManager () const`
- `PluginManager * getPlugins () const`
- `ModelManager * getModels () const`
- `TraceManager * getTracer () const`
- `ParserManager * getParser () const`
- `ExperimentManager * getExperimenter () const`

### 4.70.1 Detailed Description

The main class of the Genesys KERNEL simulation. It gives access to simulation models and tools. Simulation is the top level class and is supposed to be available to application as a dynamic linked library.

Definition at line 35 of file Simulator.h.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_2022_GenESyS/kernel/simulator/Simulator.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_2022_GenESyS/kernel/simulator/Simulator.cpp`

## 4.71 SinkModelComponent Class Reference

```
#include <SinkModelComponent.h>
```

### Public Member Functions

- **SinkModelComponent** ([Model](#) \*model, std::string componentType, std::string name="")

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.71.1 Detailed Description

This class is the basis for any component representing the end of a process flow, such as a [Dispose](#). It can remove entities from the system and collect statistics.

Definition at line 25 of file SinkModelComponent.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/SinkModelComponent.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/SinkModelComponent.cpp

## 4.72 yy::genesyspp\_parser::stack< T, S >::slice Class Reference

Present a slice of the top of a stack.

```
#include <GenesysParser.h>
```

### Public Member Functions

- **slice** (const stack &stack, index\_type range)
- const T & **operator[]** (index\_type i) const

### 4.72.1 Detailed Description

```
template<typename T, typename S = std::vector<T>>>
class yy::genesyspp_parser::stack< T, S >::slice
```

Definition at line 2824 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)

## 4.73 Solver\_if Class Reference

```
#include <solver_if.h>
```

### Public Member Functions

- virtual void **setPrecision** (double e)=0
- virtual double **getPrecision** ()=0
- virtual void **setMaxSteps** (double steps)=0
- virtual double **getMaxSteps** ()=0
- virtual double **integrate** (double min, double max, double(\*f)(double, double), double p2)=0
- virtual double **integrate** (double min, double max, double(\*f)(double, double, double), double p2, double p3)=0
- virtual double **integrate** (double min, double max, double(\*f)(double, double, double, double), double p2, double p3, double p4)=0
- virtual double **integrate** (double min, double max, double(\*f)(double, double, double, double, double), double p2, double p3, double p4, double p5)=0
- virtual double **derivate** (double initPoint, double initValue, double(\*f)(double, double), double p2)=0
- virtual double **derivate** (double initPoint, double initValue, double(\*f)(double, double, double), double p2, double p3)=0
- virtual double **derivate** (double initPoint, double initValue, double(\*f)(double, double, double, double), double p2, double p3, double p4)=0
- virtual double **derivate** (double initPoint, double initValue, double(\*f)(double, double, double, double, double), double p2, double p3, double p4, double p5)=0

### 4.73.1 Detailed Description

Interface used by classes that perform the numerical integration and derivation of functions with from one up to four parameters. It is mainly used for calculating the probability of theoretical distributions, from its probability distribution functions. p1 is the value where function is being evaluated and p2, ... are the function parameters

Definition at line 22 of file solver\_if.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/tools/solver\_if.h

## 4.74 SourceModelComponent Class Reference

```
#include <SourceModelComponent.h>
```

### Public Member Functions

- **SourceModelComponent** ([Model](#) \*model, std::string componentTypename, std::string name="")
- void **setFirstCreation** (double \_firstCreation)
- double **getFirstCreation** () const
- void **setEntityType** (EntityType \*\_entityType)
- EntityType \* **getEntityType** () const
- void **setTimeUnit** (Util::TimeUnit \_timeUnit)
- Util::TimeUnit **getTimeUnit** () const
- void **setTimeBetweenCreationsExpression** (std::string \_timeBetweenCreations)
- std::string **getTimeBetweenCreationsExpression** () const
- void **setMaxCreations** (unsigned long \_maxCreations)
- void **setMaxCreations** (std::string \_maxCreationsExpression)
- std::string **getMaxCreations** () const
- unsigned int **getEntitiesCreated** () const
- void **setEntitiesCreated** (unsigned int \_entitiesCreated)
- void **setEntitiesPerCreation** (unsigned int \_entitiesPerCreation)
- unsigned int **getEntitiesPerCreation** () const
- virtual std::string **show** ()

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual void **\_initBetweenReplications** ()
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Protected Attributes

- EntityType \* **\_entityType**
- const struct SourceModelComponent::DEFAULT\_VALUES **DEFAULT**
- double **\_firstCreation** = DEFAULT.firstCreation
- unsigned int **\_entitiesPerCreation** = DEFAULT.entitiesPerCreation
- std::string **\_maxCreationsExpression** = DEFAULT.maxCreationsExpression
- std::string **\_timeBetweenCreationsExpression** = DEFAULT.timeBetweenCreationsExpression
- Util::TimeUnit **\_timeBetweenCreationsTimeUnit** = DEFAULT.timeBetweenCreationsTimeUnit
- unsigned int **\_entitiesCreatedSoFar** = 0

### Additional Inherited Members

#### 4.74.1 Detailed Description

A source component implements the base for inserting entities into the model when its simulation is initialized. During the initialization, the new and empty future events list is populated by events of creating entities and sending them to the source components existing in the model

Definition at line 27 of file SourceModelComponent.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/SourceModelComponent.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/SourceModelComponent.cpp

## 4.75 Start Class Reference

```
#include <Start.h>
```

### Public Member Functions

- **Start** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.75.1 Detailed Description

[Start](#) module DESCRIPTION The [Start](#) module changes the status of a conveyor from inactive to active. The conveyor may have been deactivated from either the [Stop](#) module or by initially being set to inactive at the start of the simulation. The velocity of the conveyor may be changed permanently when the conveyor is started. TYPICAL USES [Start](#) a bottling conveyor after scheduled maintenance [Start](#) a baggage claim conveyor when bags have arrived PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart. Conveyor Name Name of the conveyor to start. Velocity Speed of the conveyor once it begins to operate. This value will change the speed of the conveyor permanently, until it is changed in another module. Units Velocity time units.

Definition at line 39 of file Start.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Start.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Start.cpp

## 4.76 Statistics\_if Class Reference

```
#include <Statistics_if.h>
```

### Public Member Functions

- virtual [Collector\\_if](#) \* **getCollector** ()=0
- virtual void **setCollector** ([Collector\\_if](#) \*collector)=0
- virtual unsigned int **numElements** ()=0
- virtual double **min** ()=0
- virtual double **max** ()=0
- virtual double **average** ()=0
- virtual double **variance** ()=0
- virtual double **stddeviation** ()=0
- virtual double **variationCoef** ()=0
- virtual double **halfWidthConfidenceInterval** ()=0
- virtual unsigned int **newSampleSize** (double halfWidth)=0
- virtual double **getConfidenceLevel** ()=0
- virtual void **setConfidenceLevel** (double confidencelevel)=0

#### 4.76.1 Detailed Description

Interface for statist synthesis of a stochastic variable collected by a [Collector\\_if](#). The statistics generated may be updated based only on the previous statistics and the single newest added value or they may be updated based on a datafile, depending on the Collector implementation.

Definition at line 23 of file `Statistics_if.h`.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵  
2022_GenESyS/kernel/statistics/Statistics_if.h`

## 4.77 StatisticsCollector Class Reference

```
#include <StatisticsCollector.h>
```

### Public Member Functions

- **StatisticsCollector** ([Model](#) \*model, std::string name="", [ModelElement](#) \*parent=nullptr, bool insertInto↵  
Model=true)
- virtual std::string **show** ()
- [ModelElement](#) \* **getParent** () const
- [Statistics\\_if](#) \* **getStatistics** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- void **\_addSimulationResponses** ()

## Additional Inherited Members

### 4.77.1 Detailed Description

The [StatisticsCollector](#) is the [ModelElement](#) responsible for collecting data from the model (using the Collector) and simultaneously keeping statistics updated (using the Statistics)

Definition at line 27 of file StatisticsCollector.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/StatisticsCollector.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/kernel/simulator/StatisticsCollector.cpp

## 4.78 Stop Class Reference

```
#include <Stop.h>
```

### Public Member Functions

- **Stop** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::string < std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

## Additional Inherited Members

### 4.78.1 Detailed Description

[Stop](#) module DESCRIPTION The [Stop](#) module sets the operational status of a conveyor to inactive. The conveyor may have been activated from either the [Start](#) module or by initially being set to active at the start of the simulation. When the entity enters the [Stop](#) module, the conveyor will stop immediately, regardless of the type of conveyor or the number of entities currently on the conveyor. TYPICAL USES [Stop](#) a baggage conveyor after a pre-determined amount of time [Stop](#) a conveyor for scheduled maintenance PROMPTS Prompt Description Name Unique name of the module that will be displayed in the flowchart. Conveyor Name Name of the conveyor to stop.

Definition at line 36 of file Stop.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Stop.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Stop.cpp



## 4.79 Store Class Reference

```
#include <Store.h>
```

### Public Member Functions

- **Store** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.79.1 Detailed Description

[Store](#) module DESCRIPTION The [Store](#) module adds an entity to storage. The [Unstore](#) module may then be used to remove the entity from the storage. When an entity arrives at the [Store](#) module, the storage specified is incremented, and the entity immediately moves to the next module in the model. Storages are useful for displaying entity animation while an entity undergoes processing in other modules. Additionally, statistics may be kept on the number of entities in storage. TYPICAL USES Animating a part through a number of delay operations (load, setup, process, unload) Tracking the number of customers within a grocery store (place in storage upon entry) PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Type Method of specifying the storage name as a Storage, [Set](#), [Attribute](#), or Expression. Storage Name Name of the storage to which the entity will be added. Applies only when the Type is Storage. [Set](#) Name Name of the storage set from which the storage is to be selected. Applies only when the Type is [Set](#). [Set](#) Index Index into the defined storage set that contains the desired storage name. Applies only when the Type is [Set](#). [Attribute](#) Name of the attribute whose value contains the storage. Applies only when the Type is [Attribute](#). Expression Expression that is evaluated to the storage into which the entity is placed. Applies only when the Type is Expression.

Definition at line 50 of file Store.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Store.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Store.cpp

## 4.80 Submodel Class Reference

```
#include <Submodel.h>
```

### Public Member Functions

- **Submodel** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.80.1 Detailed Description

This component ...

Definition at line 22 of file Submodel.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Submodel.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/plugins/components/Submodel.cpp

## 4.81 yy::genesyspp\_parser::symbol\_kind Struct Reference

Symbol kinds.

```
#include <GenesysParser.h>
```

## Public Types

- enum [symbol\\_kind\\_type](#) {  
YYNTOKENS = 90, S\_YYEMPTY = -2, S\_YYEOF = 0, S\_YYerror = 1,  
S\_YYUNDEF = 2, S\_NUMD = 3, S\_NUMH = 4, S\_CTEZERO = 5,  
S\_oLE = 6, S\_oGE = 7, S\_oEQ = 8, S\_oNE = 9,  
S\_oAND = 10, S\_oOR = 11, S\_oNAND = 12, S\_oXOR = 13,  
S\_oNOT = 14, S\_fSIN = 15, S\_fCOS = 16, S\_fROUND = 17,  
S\_fMOD = 18, S\_fTRUNC = 19, S\_fFRAC = 20, S\_fEXP = 21,  
S\_fSQRT = 22, S\_fLOG = 23, S\_fLN = 24, S\_fVAL = 25,  
S\_fEVAL = 26, S\_fLENG = 27, S\_fRND1 = 28, S\_fEXPO = 29,  
S\_fNORM = 30, S\_fUNIF = 31, S\_fWEIB = 32, S\_fLOGN = 33,  
S\_fGAMM = 34, S\_fERLA = 35, S\_fTRIA = 36, S\_fBETA = 37,  
S\_fDISC = 38, S\_ftNOW = 39, S\_ftFIN = 40, S\_fMAXREP = 41,  
S\_fNUMREP = 42, S\_fIDENT = 43, S\_cIF = 44, S\_cELSE = 45,  
S\_cFOR = 46, S\_cTO = 47, S\_cDO = 48, S\_ATTRIB = 49,  
S\_CSTAT = 50, S\_ftAVG = 51, S\_ILLEGAL = 52, S\_RESOURCE = 53,  
S\_fNR = 54, S\_fMR = 55, S\_fIRF = 56, S\_fRESSEIZES = 57,  
S\_fSTATE = 58, S\_fSETSUM = 59, S\_fRESUTIL = 60, S\_QUEUE = 61,  
S\_fNQ = 62, S\_fFIRSTINQ = 63, S\_fLASTINQ = 64, S\_fSAQUE = 65,  
S\_fAQUE = 66, S\_fENTATRANK = 67, S\_SET = 68, S\_fNUMSET = 69,  
S\_VARI = 70, S\_FORM = 71, S\_fNUMGR = 72, S\_fATRGR = 73,  
S\_LPAREN = 74, S\_RPAREN = 75, S\_LBRACKET = 76, S\_RBRACKET = 77,  
S\_PLUS = 78, S\_MINUS = 79, S\_STAR = 80, S\_POWER = 81,  
S\_SLASH = 82, S\_LESS = 83, S\_GREATER = 84, S\_ASSIGN = 85,  
S\_COMMA = 86, S\_NEG = 87, S\_88\_n = 88, S\_89\_USER = 89,  
S\_YYACCEPT = 90, S\_input = 91, S\_expression = 92, S\_numero = 93,  
S\_aritmetica = 94, S\_logica = 95, S\_relacional = 96, S\_comando = 97,  
S\_comandoIF = 98, S\_comandoFOR = 99, S\_function = 100, S\_kernelFunction = 101,  
S\_trigonFunction = 102, S\_mathFunction = 103, S\_probFunction = 104, S\_userFunction = 105,  
S\_listaparm = 106, S\_atributo = 107, S\_variavel = 108, S\_formula = 109,  
S\_atribuicao = 110, S\_pluginFunction = 111 }

## 4.81.1 Detailed Description

Definition at line 671 of file GenesysParser.h.

## 4.81.2 Member Enumeration Documentation

4.81.2.1 [symbol\\_kind\\_type](#) enum [yy::genesyspp\\_parser::symbol\\_kind::symbol\\_kind\\_type](#)

Enumerator

YYNTOKENS	Number of tokens.
-----------	-------------------

Definition at line 673 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)

## 4.82 yy::genesyspp\_parser::symbol\_type Struct Reference

"External" symbols: returned by the scanner.

```
#include <GenesysParser.h>
```

### Public Types

- typedef [basic\\_symbol](#)< [by\\_kind](#) > [super\\_type](#)  
*Superclass.*

### Public Member Functions

- [symbol\\_type](#) ()  
*Empty symbol.*
- [symbol\\_type](#) (int tok, const [location\\_type](#) &l)  
*Constructor for valueless symbols, and symbols from each type.*
- **symbol\_type** (int tok, const obj\_t &v, const [location\\_type](#) &l)

### Additional Inherited Members

#### 4.82.1 Detailed Description

Definition at line 1151 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)

## 4.83 yy::genesyspp\_parser::syntax\_error Struct Reference

Syntax errors thrown from user actions.

```
#include <GenesysParser.h>
```

### Public Member Functions

- **syntax\_error** (const [location\\_type](#) &l, const std::string &m)
- **syntax\_error** (const [syntax\\_error](#) &s)

### Public Attributes

- [location\\_type](#) **location**

### 4.83.1 Detailed Description

Definition at line 548 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/GenesysParser.cpp

## 4.84 yy::genesyspp\_parser::token Struct Reference

Token kinds.

```
#include <GenesysParser.h>
```

### Public Types

- enum **token\_kind\_type** {  
**YYEMPTY** = -2, **END** = 0, **YYerror** = 256, **YYUNDEF** = 257 ,  
**NUMD** = 258, **NUMH** = 259, **CTEZERO** = 260, **oLE** = 261 ,  
**oGE** = 262, **oEQ** = 263, **oNE** = 264, **oAND** = 265 ,  
**oOR** = 266, **oNAND** = 267, **oXOR** = 268, **oNOT** = 269 ,  
**fSIN** = 270, **fCOS** = 271, **fROUND** = 272, **fMOD** = 273 ,  
**fTRUNC** = 274, **fFRAC** = 275, **fEXP** = 276, **fSQRT** = 277 ,  
**fLOG** = 278, **fLN** = 279, **fVAL** = 280, **fEVAL** = 281 ,  
**fLENG** = 282, **fRND1** = 283, **fEXPO** = 284, **fNORM** = 285 ,  
**fUNIF** = 286, **fWEIB** = 287, **fLOGN** = 288, **fGAMM** = 289 ,  
**fERLA** = 290, **fTRIA** = 291, **fBETA** = 292, **fDISC** = 293 ,  
**fTNOW** = 294, **ftFIN** = 295, **fMAXREP** = 296, **fNUMREP** = 297 ,  
**fIDENT** = 298, **cIF** = 299, **cELSE** = 300, **cFOR** = 301 ,  
**cTO** = 302, **cDO** = 303, **ATTRIB** = 304, **CSTAT** = 305 ,  
**ftAVG** = 306, **ILLEGAL** = 307, **RESOURCE** = 308, **fNR** = 309 ,  
**fMR** = 310, **fIRF** = 311, **fRESSEIZES** = 312, **fSTATE** = 313 ,  
**fSETSUM** = 314, **fRESUTIL** = 315, **QUEUE** = 316, **fNQ** = 317 ,  
**fFIRSTINQ** = 318, **fLASTINQ** = 319, **fSAQUE** = 320, **fAQUE** = 321 ,  
**fENTATRANK** = 322, **SET** = 323, **fNUMSET** = 324, **VARI** = 325 ,  
**FORM** = 326, **fNUMGR** = 327, **fATRGR** = 328, **LPAREN** = 329 ,  
**RPAREN** = 330, **LBRACKET** = 331, **RBRACKET** = 332, **PLUS** = 333 ,  
**MINUS** = 334, **STAR** = 335, **POWER** = 336, **SLASH** = 337 ,  
**LESS** = 338, **GREATER** = 339, **ASSIGN** = 340, **COMMA** = 341 ,  
**NEG** = 342 }
- typedef token\_kind\_type [yytokentype](#)  
*Backward compatibility alias (Bison 3.6).*

### 4.84.1 Detailed Description

Definition at line 566 of file GenesysParser.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/parserBisonFlex/[GenesysParser.h](#)

## 4.85 TraceManager Class Reference

```
#include <TraceManager.h>
```

### Public Member Functions

- **TraceManager** ([Simulator](#) \*simulator)
- void **addTraceHandler** (traceListener traceListener)
- void **addTraceReportHandler** (traceListener traceReportListener)
- void **addTraceSimulationHandler** (traceSimulationListener traceSimulationListener)
- void **addTraceErrorHandler** (traceErrorListener traceErrorListener)
- template<typename Class >  
void **addTraceHandler** (Class \*object, void(Class::\*function)(TraceEvent))
- template<typename Class >  
void **addTraceErrorHandler** (Class \*object, void(Class::\*function)(TraceErrorEvent))
- template<typename Class >  
void **addTraceReportHandler** (Class \*object, void(Class::\*function)(TraceEvent))
- template<typename Class >  
void **addTraceSimulationHandler** (Class \*object, void(Class::\*function)(TraceSimulationEvent))
- void **trace** (Util::TraceLevel level, std::string text)
- void **traceError** (std::exception e, std::string text)
- void **traceReport** (Util::TraceLevel level, std::string text)
- void **traceSimulation** (Util::TraceLevel level, double time, [Entity](#) \*entity, [ModelComponent](#) \*component, std::string text)
- void **trace** (std::string text, Util::TraceLevel level=Util::TraceLevel::L8\_detailed)
- void **traceError** (std::string text, std::exception e)
- void **traceReport** (std::string text, Util::TraceLevel level=Util::TraceLevel::L2\_results)
- void **traceSimulation** (double time, [Entity](#) \*entity, [ModelComponent](#) \*component, std::string text, Util::TraceLevel level=Util::TraceLevel::L8\_detailed)
- [List](#)< std::string > \* **errorMessages** () const
- void **setTraceLevel** (Util::TraceLevel \_traceLevel)
- Util::TraceLevel **getTraceLevel** () const
- [Simulator](#) \* **getParentSimulator** () const

### 4.85.1 Detailed Description

The [TraceManager](#) is used to trace back model simulation information and track/debug the simulation. It works as the model simulation output (cout) and allows external methods to hook up such output as listeners.

Definition at line 106 of file TraceManager.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/TraceManager.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↵  
2022\_GenESyS/kernel/simulator/TraceManager.cpp

## 4.86 TraceSimulationProcess Class Reference

```
#include <TraceManager.h>
```

**Public Member Functions**

- **TraceSimulationProcess** (std::string text, Util::TraceLevel level=Util::TraceLevel::L8\_detailed)

**4.86.1 Detailed Description**

Events related to simulation "process" (usually process analyser), associated to entire replication or simulation events (begin/end/pause of replication/simulation)

Definition at line 85 of file TraceManager.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/kernel/simulator/TraceManager.h

**4.87 Traits< GenesysApplication\_if > Struct Reference**

```
#include <Traits.h>
```

**Public Types**

- typedef Smart\_OnEvent **Application**

**Static Public Attributes**

- static const Util::TraceLevel **traceLevel** = Util::TraceLevel::L6\_arrival

**4.87.1 Detailed Description**

Configure the Genesys Application to be compiled and executed

Definition at line 58 of file Traits.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/Traits.h

**4.88 Traits< HypothesisTester\_if > Struct Reference**

```
#include <Traits.h>
```

**Public Types**

- typedef HypothesisTesterDefaultImpl1 **Implementation**

### Static Public Attributes

- static constexpr unsigned int **ConfidenceLevel** = 95

#### 4.88.1 Detailed Description

Configure the Hypothesis Tester to be used

Definition at line 100 of file Traits.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/Traits.h

### 4.89 Traits< Solver\_if > Struct Reference

```
#include <Traits.h>
```

#### Public Types

- typedef SolverDefaultImpl1 **Implementation**

### Static Public Attributes

- static constexpr double **Precision** = 1e-5
- static constexpr unsigned int **MaxSteps** = 1e2

#### 4.89.1 Detailed Description

Configure the Solver to be used

Definition at line 91 of file Traits.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔  
2022\_GenESyS/Traits.h

### 4.90 Unstore Class Reference

```
#include <Unstore.h>
```

#### Public Member Functions

- **Unstore** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()



**Static Public Member Functions**

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

**Protected Member Functions**

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

**Additional Inherited Members****4.90.1 Detailed Description**

[Unstore](#) module DESCRIPTION The [Unstore](#) module removes an entity from storage. When an entity arrives at the [Unstore](#) module, the storage specified is decreased and the entity immediately moves to the next module in the model. TYPICAL USES Removing the entity from an animation location when processing is complete Tracking the number of customers within a grocery store (unstore upon exit) PROMPTS Prompt Description Name Unique module identifier displayed on the module shape. Type Method of specifying the storage name as a Storage, [Set](#), [Attribute](#), or Expression. Default will remove an entity from the last storage that it entered. Storage Name Name of the storage to which the entity will be added. Applies only when the Type is Storage. [Set](#) Name Name of the storage set from which the storage is to be selected. Applies only when the Type is [Set](#). [Set](#) Index Index into the defined storage set that contains the desired storage name. Applies only when the Type is [Set](#). [Attribute](#) Name of the attribute whose value contains the storage. Applies only when the Type is [Attribute](#). Expression Expression that is evaluated to the storage into which the entity is placed. Applies only when the Type is Expression.

Definition at line 45 of file Unstore.h.

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

- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Unstore.h
- /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_↔2022\_GenESyS/plugins/components/Unstore.cpp

**4.91 Variable Class Reference**

```
#include <Variable.h>
```

**Public Member Functions**

- **Variable** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- double **value** ()
- void **setValue** (double value)
- double **value** (std::string index)
- void **setValue** (std::string index, double value)
- double **initialValue** ()
- void **setInitialValue** (double value)
- double **initialValue** (std::string index)
- void **setInitialValue** (std::string index, double value)
- [List](#)< unsigned int > \* **dimensionSizes** () const

## Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelElement](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

## Protected Member Functions

- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)
- virtual void **\_initBetweenReplications** ()

## Additional Inherited Members

### 4.91.1 Detailed Description

[Variable](#) module DESCRIPTION This data module is used to define a variable's dimension and values. You can reference variables in other modules (for example, the [Decide](#) module), reassign new values to variables with the [Assign](#) module, and use variables in any expression. You can use an external data file to specify variable values, and you can specify the variable's initial values in the [Variable](#) module. If you use both methods, the values are read at different times, depending on the options you specify, including the [File](#) Read Time, the Clear Option, and the replication parameters you specify in the Run Setup dialog box. For more information, see the online Help. There are three methods for manually editing the Initial Values of a [Variable](#) module: Using the standard spreadsheet interface. In the module spreadsheet, right-click on the Initial Values cell and select the Edit via spreadsheet menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value. Using the module dialog box. In the module spreadsheet, right-click on any cell and select the Edit via dialog menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value. Using the two-dimensional (2-D) spreadsheet interface. In the module spreadsheet, click on the Initial Values cell.

TYPICAL USES Number of documents processed per hour Serial number to assign to parts for unique identification Space available in a facility PROMPTS Prompt Description Name The unique name of the variable being defined. Rows Number of rows in a one- or two-dimensional variable. Columns Number of columns in a two-dimensional variable. Report Statistics Check box for determining whether or not statistics will be collected. This field is visible when the rows and columns are not specified (that is, for single variables). Data Type The data type of the values stored in the variable. Valid types are Real and String. The default type is Real. Clear Option Defines the time (if at all) when the value(s) of the variable is reset to the initial value(s) specified. Specifying Statistics resets this variable to its initial value(s) whenever statistics are cleared. Specifying System resets this variable to its initial value(s) whenever the system is cleared. Specifying None indicates that this variable is never reset to its initial value(s), except prior to the first replication. [File](#) Name Name of the file from which to read the variable's value or values. You can use any file access type supported by Arena except sequential text files and Lotus spreadsheet (.wks) files. If the file name you specify has not been created yet, Arena will create it, but you must edit the file to specify the file access type, path, and recordset (if required). Recordset Name of the recordset in the specified file from which to read values. This field is available only if you specify a [File](#) Name for a file that has been set up with a file access type, path, and recordset. Arena uses the Rows and Columns properties to determine the amount of data to read from the recordset. A recordset is required for all file types except .xml. The recordset size must be equal to or greater than the number of rows and columns specified for the variable. [File](#) Read Time Specifies when to read the values from the file into the variable. If you select PreCheck, the values for the variable are read while the model is still in Edit mode (prior to the model being checked and compiled). If you select BeginSimulation, values are read when the model is compiled, prior to the first replication. If you select BeginReplication, values are read prior to each replication. Initial Values Lists the initial value or values of the variable. You can assign new values to the variable at different stages of the model by using the [Assign](#) module. Initial Value [Variable](#) value at the start of the simulation.

Definition at line 90 of file Variable.h.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/elements/Variable.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/elements/Variable.cpp`

## 4.92 Write Class Reference

```
#include <Write.h>
```

### Public Types

- enum class **WriteToType** : int { **SCREEN** = 1 , **FILE** = 2 }

### Public Member Functions

- **Write** ([Model](#) \*model, std::string name="")
- virtual std::string **show** ()
- [List](#)< WriteText \* > \* **writeElements** () const
- void **setFilename** (std::string \_filename)
- std::string **filename** () const
- void **setWriteToType** (WriteToType \_writeToType)
- Write::WriteToType **writeToType** () const

### Static Public Member Functions

- static PluginInformation \* **GetPluginInformation** ()
- static [ModelComponent](#) \* **LoadInstance** ([Model](#) \*model, std::map< std::string, std::string > \*fields)

### Protected Member Functions

- virtual void **\_execute** ([Entity](#) \*entity)
- virtual void **\_initBetweenReplications** ()
- virtual bool **\_loadInstance** (std::map< std::string, std::string > \*fields)
- virtual std::map< std::string, std::string > \* **\_saveInstance** ()
- virtual bool **\_check** (std::string \*errorMessage)

### Additional Inherited Members

#### 4.92.1 Detailed Description

This component ...

Definition at line 80 of file Write.h.

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

- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/components/Write.h`
- `/home/rlcancian/Laboratory/Software_Lab/IA32_Architecture/GccProjects/RebornedGenESyS/2019_↵2022_GenESyS/plugins/components/Write.cpp`

## 5 File Documentation

### 5.1 /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/↵ RebornedGenESyS/2019\_2022\_GenESyS/kernel/simulator/parserBisonFlex/↵ GenesysParser.h File Reference

```
#include <string>
#include <cmath>
#include "obj_t.h"
#include "../util/Util.h"
#include "../Attribute.h"
#include "../../plugins/elements/Variable.h"
#include "../../plugins/elements/Queue.h"
#include "../../plugins/elements/Formula.h"
#include "../../plugins/elements/Resource.h"
#include "../../plugins/elements/Set.h"
#include <cassert>
#include <cstdlib>
#include <iostream>
#include <stdexcept>
#include <vector>
#include "location.hh"
#include <typeinfo>
```

#### Classes

- class [yy::genesyspp\\_parser](#)  
*A Bison parser.*
- class [yy::genesyspp\\_parser::semantic\\_type](#)
- struct [yy::genesyspp\\_parser::syntax\\_error](#)  
*Syntax errors thrown from user actions.*
- struct [yy::genesyspp\\_parser::token](#)  
*Token kinds.*
- struct [yy::genesyspp\\_parser::symbol\\_kind](#)  
*Symbol kinds.*
- struct [yy::genesyspp\\_parser::basic\\_symbol< Base >](#)
- struct [yy::genesyspp\\_parser::by\\_kind](#)  
*Type access provider for token (enum) based symbols.*
- struct [yy::genesyspp\\_parser::symbol\\_type](#)  
*"External" symbols: returned by the scanner.*
- class [yy::genesyspp\\_parser::stack< T, S >::slice](#)  
*Present a slice of the top of a stack.*

#### Macros

- `#define YY_CPLUSPLUS 199711L`
- `#define YY_MOVE`
- `#define YY_MOVE_OR_COPY copy`
- `#define YY_MOVE_REF(Type) Type&`
- `#define YY_RVREF(Type) const Type&`

- `#define YY_COPY(Type) const Type&`
- `#define YY_NOEXCEPT`
- `#define YY_NOTHROW throw ()`
- `#define YY_CONSTEXPR`
- `#define YY_ASSERT assert`
- `#define YY_ATTRIBUTE_PURE`
- `#define YY_ATTRIBUTE_UNUSED`
- `#define YY_USE(E) ((void) (E))`
- `#define YY_INITIAL_VALUE(Value) Value`
- `#define YY_IGNORE_MAYBE_UNINITIALIZED_BEGIN`
- `#define YY_IGNORE_MAYBE_UNINITIALIZED_END`
- `#define YY_IGNORE_USELESS_CAST_BEGIN`
- `#define YY_IGNORE_USELESS_CAST_END`
- `#define YY_CAST(Type, Val) ((Type) (Val))`
- `#define YY_REINTERPRET_CAST(Type, Val) ((Type) (Val))`
- `#define YYDEBUG 1`

### 5.1.1 Detailed Description

Define the `yy::parser` class.

## 5.2 /home/rlcancian/Laboratory/Software\_Lab/IA32\_Architecture/GccProjects/RebornedGenESyS/2019\_2022\_GenESyS/kernel/simulator/parserBisonFlex/location.hh File Reference

```
#include <iostream>
#include <string>
```

### Classes

- class `yy::position`  
*A point in a source file.*
- class `yy::location`  
*Two points in a source file.*

### Macros

- `#define YY_NULLPTR ((void*)0)`

## Functions

- position & [yy::operator+=](#) (position &res, position::counter\_type width)  
*Add width columns, in place.*
- position [yy::operator+](#) (position res, position::counter\_type width)  
*Add width columns.*
- position & [yy::operator-=](#) (position &res, position::counter\_type width)  
*Subtract width columns, in place.*
- position [yy::operator-](#) (position res, position::counter\_type width)  
*Subtract width columns.*
- template<typename YYChar >  
std::basic\_ostream< YYChar > & [yy::operator<<](#) (std::basic\_ostream< YYChar > &ostr, const position &pos)  
*Intercept output stream redirection.*
- location & [yy::operator+=](#) (location &res, const location &end)  
*Join two locations, in place.*
- location [yy::operator+](#) (location res, const location &end)  
*Join two locations.*
- location & [yy::operator+=](#) (location &res, location::counter\_type width)  
*Add width columns to the end position, in place.*
- location [yy::operator+](#) (location res, location::counter\_type width)  
*Add width columns to the end position.*
- location & [yy::operator-=](#) (location &res, location::counter\_type width)  
*Subtract width columns to the end position, in place.*
- location [yy::operator-](#) (location res, location::counter\_type width)  
*Subtract width columns to the end position.*
- template<typename YYChar >  
std::basic\_ostream< YYChar > & [yy::operator<<](#) (std::basic\_ostream< YYChar > &ostr, const location &loc)  
*Intercept output stream redirection.*

### 5.2.1 Detailed Description

Define the [yy::location](#) class.

### 5.2.2 Function Documentation

**5.2.2.1 operator<<()** [1/2] template<typename YYChar >  
std::basic\_ostream<YYChar>& [yy::operator<<](#) (  
std::basic\_ostream< YYChar > & ostr,  
const [location](#) & loc )

#### Parameters

<i>ostr</i>	the destination output stream
<i>loc</i>	a reference to the location to redirect

Avoid duplicate information.

Definition at line 272 of file location.hh.

**5.2.2.2 operator<<()** [2/2] `template<typename YYChar >`  
`std::basic_ostream<YYChar>& yy::operator<< (`  
    `std::basic_ostream< YYChar > & ostr,`  
    `const position & pos )`

#### Parameters

<i>ostr</i>	the destination output stream
<i>pos</i>	a reference to the position to redirect

Definition at line 146 of file location.hh.

