

QRodSystems

0.0.19

Generated by Doxygen 1.9.7



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>7</b>
3.1 File List	7
<b>4 Class Documentation</b>	<b>11</b>
4.1 QRS::Core::AbstractDataObject Class Reference	11
4.1.1 Detailed Description	12
4.1.2 Member Function Documentation	12
4.1.2.1 addItem()	12
4.1.2.2 clone()	13
4.1.2.3 deserialize()	13
4.1.2.4 getAvailableItemKey()	13
4.1.2.5 import()	13
4.1.2.6 serialize()	14
4.2 QRS::HierarchyModels::AbstractHierarchyItem Class Reference	14
4.2.1 Detailed Description	15
4.3 QRS::HierarchyModels::AbstractHierarchyModel Class Reference	15
4.3.1 Detailed Description	16
4.3.2 Member Function Documentation	16
4.3.2.1 clearContent()	16
4.3.2.2 updateContent()	16
4.3.2.3 updateContentExpanded()	17
4.4 QRS::Managers::AbstractManager Class Reference	17
4.4.1 Detailed Description	18
4.5 QRS::PropertiesModels::AbstractPropertiesModel Class Reference	18
4.5.1 Detailed Description	19
4.6 QRS::Core::AbstractRodComponent Class Reference	19
4.6.1 Detailed Description	21
4.6.2 Member Function Documentation	21
4.6.2.1 clone()	21
4.6.2.2 deserialize()	21
4.6.2.3 isDataComplete()	21
4.6.2.4 resolveReferences()	21
4.6.2.5 serialize()	22
4.7 QRS::Managers::AbstractRodComponentWidget Class Reference	22
4.7.1 Detailed Description	23
4.8 QRS::Core::AbstractSectionRodComponent Class Reference	23
4.8.1 Detailed Description	24

4.8.2 Member Function Documentation	24
4.8.2.1 deserialize()	24
4.8.2.2 resolveReferences()	24
4.8.2.3 serialize()	25
4.9 QRS::Core::Array< T > Class Template Reference	25
4.9.1 Detailed Description	26
4.10 QRS::TableModels::BaseTableModel Class Reference	26
4.10.1 Detailed Description	27
4.10.2 Member Function Documentation	27
4.10.2.1 insertItemAfterSelected()	27
4.10.2.2 insertLeadingItemAfterSelected()	28
4.10.2.3 removeSelectedItem()	28
4.10.2.4 removeSelectedLeadingItem()	28
4.11 QRS::Managers::ConstraintItemDelegate Class Reference	28
4.11.1 Detailed Description	29
4.12 QRS::Core::ConstraintRodComponent Class Reference	29
4.12.1 Detailed Description	30
4.12.2 Member Function Documentation	31
4.12.2.1 clone()	31
4.12.2.2 deserialize()	31
4.12.2.3 isDataComplete()	31
4.12.2.4 resolveReferences()	31
4.12.2.5 serialize()	32
4.13 QRS::Managers::ConstraintRodComponentWidget Class Reference	32
4.13.1 Detailed Description	33
4.14 QRS::Managers::DataObjectLineEdit Class Reference	33
4.14.1 Detailed Description	34
4.15 QRS::HierarchyModels::DataObjectsHierarchyItem Class Reference	35
4.15.1 Detailed Description	35
4.15.2 Member Function Documentation	36
4.15.2.1 type()	36
4.16 QRS::HierarchyModels::DataObjectsHierarchyModel Class Reference	36
4.16.1 Detailed Description	37
4.16.2 Member Function Documentation	37
4.16.2.1 clearContent()	38
4.16.2.2 updateContent()	38
4.17 QRS::Managers::DataObjectsManager Class Reference	38
4.17.1 Detailed Description	40
4.18 QRS::PropertiesModels::DataObjectsPropertiesModel Class Reference	40
4.18.1 Detailed Description	41
4.19 QRS::Managers::DoubleSpinBoxItemDelegate Class Reference	41
4.19.1 Detailed Description	42

4.20 QRS::Core::GeometryRodComponent Class Reference	42
4.20.1 Detailed Description	43
4.20.2 Member Function Documentation	43
4.20.2.1 clone()	44
4.20.2.2 deserialize()	44
4.20.2.3 isDataComplete()	44
4.20.2.4 resolveReferences()	44
4.20.2.5 serialize()	44
4.21 QRS::Managers::GeometryRodComponentWidget Class Reference	45
4.21.1 Detailed Description	45
4.22 QRS::Core::HierarchyNode Class Reference	46
4.22.1 Detailed Description	47
4.23 QRS::Core::HierarchyTree Class Reference	47
4.23.1 Detailed Description	48
4.24 QRS::Core::LoadRodComponent Class Reference	48
4.24.1 Detailed Description	50
4.24.2 Member Function Documentation	50
4.24.2.1 clone()	50
4.24.2.2 deserialize()	50
4.24.2.3 isDataComplete()	50
4.24.2.4 resolveReferences()	51
4.24.2.5 serialize()	51
4.25 QRS::Managers::LoadRodComponentWidget Class Reference	51
4.25.1 Detailed Description	52
4.26 QRS::App::LogWidget Class Reference	52
4.26.1 Detailed Description	53
4.27 QRS::App::MainWindow Class Reference	53
4.27.1 Detailed Description	55
4.28 QRS::Managers::ManagersFactory Class Reference	55
4.28.1 Detailed Description	56
4.29 QRS::App::ManagersTab Class Reference	56
4.29.1 Detailed Description	57
4.30 QRS::Core::MaterialRodComponent Class Reference	57
4.30.1 Detailed Description	58
4.30.2 Member Function Documentation	58
4.30.2.1 clone()	58
4.30.2.2 deserialize()	59
4.30.2.3 isDataComplete()	59
4.30.2.4 resolveReferences()	59
4.30.2.5 serialize()	59
4.31 QRS::Managers::MaterialRodComponentWidget Class Reference	60
4.31.1 Detailed Description	60

4.32 QRS::Core::MatrixDataObject Class Reference	61
4.32.1 Detailed Description	61
4.32.2 Member Function Documentation	61
4.32.2.1 addItem()	62
4.32.2.2 clone()	62
4.32.2.3 import()	62
4.33 QRS::TableModels::MatrixTableModel Class Reference	62
4.33.1 Detailed Description	63
4.33.2 Member Function Documentation	63
4.33.2.1 insertItemAfterSelected()	63
4.33.2.2 insertLeadingItemAfterSelected()	64
4.33.2.3 removeSelectedItem()	64
4.33.2.4 removeSelectedLeadingItem()	64
4.34 QRS::Core::MechanicalRodComponent Class Reference	64
4.34.1 Detailed Description	66
4.34.2 Member Function Documentation	66
4.34.2.1 clone()	66
4.34.2.2 deserialize()	66
4.34.2.3 isDataComplete()	66
4.34.2.4 resolveReferences()	67
4.34.2.5 serialize()	67
4.35 QRS::Managers::MechanicalRodComponentWidget Class Reference	67
4.35.1 Detailed Description	68
4.36 QRS::Core::Project Class Reference	68
4.36.1 Detailed Description	70
4.37 QRS::HierarchyModels::ProjectHierarchyModel Class Reference	71
4.37.1 Detailed Description	72
4.37.2 Member Function Documentation	72
4.37.2.1 clearContent()	72
4.37.2.2 updateContent()	72
4.38 QRS::HierarchyModels::RodComponentsHierarchyItem Class Reference	72
4.38.1 Detailed Description	73
4.38.2 Member Function Documentation	73
4.38.2.1 type()	73
4.39 QRS::HierarchyModels::RodComponentsHierarchyModel Class Reference	74
4.39.1 Detailed Description	75
4.39.2 Member Function Documentation	75
4.39.2.1 clearContent()	75
4.39.2.2 updateContent()	75
4.40 QRS::Managers::RodComponentsManager Class Reference	75
4.40.1 Detailed Description	77
4.41 QRS::Core::Array< T >::Row< U > Struct Template Reference	77

4.41.1 Detailed Description	78
4.42 QRS::Core::ScalarDataObject Class Reference	78
4.42.1 Detailed Description	79
4.42.2 Member Function Documentation	79
4.42.2.1 addItem()	79
4.42.2.2 clone()	79
4.42.2.3 import()	79
4.43 QRS::Core::SurfaceDataObject Class Reference	80
4.43.1 Detailed Description	81
4.43.2 Member Function Documentation	81
4.43.2.1 addItem()	81
4.43.2.2 clone()	81
4.43.2.3 deserialize()	81
4.43.2.4 import()	82
4.43.2.5 serialize()	82
4.44 QRS::TableModels::SurfaceTableModel Class Reference	82
4.44.1 Detailed Description	83
4.44.2 Member Function Documentation	83
4.44.2.1 insertItemAfterSelected()	83
4.44.2.2 insertLeadingItemAfterSelected()	83
4.44.2.3 removeSelectedItem()	84
4.44.2.4 removeSelectedLeadingItem()	84
4.45 QRS::TableModels::TableModelInterface Class Reference	84
4.45.1 Detailed Description	85
4.45.2 Member Function Documentation	85
4.45.2.1 insertItemAfterSelected()	85
4.45.2.2 insertLeadingItemAfterSelected()	85
4.45.2.3 removeSelectedItem()	85
4.45.2.4 removeSelectedLeadingItem()	86
4.46 QRS::Core::UserSectionRodComponent Class Reference	86
4.46.1 Detailed Description	87
4.46.2 Member Function Documentation	87
4.46.2.1 clone()	87
4.46.2.2 isDataComplete()	87
4.47 QRS::Managers::UserSectionRodComponentWidget Class Reference	87
4.47.1 Detailed Description	88
4.48 QRS::Core::VectorDataObject Class Reference	88
4.48.1 Detailed Description	89
4.48.2 Member Function Documentation	89
4.48.2.1 addItem()	89
4.48.2.2 clone()	90
4.48.2.3 import()	90

4.49 QRS::Graph::View3D Class Reference . . . . .	90
4.49.1 Detailed Description . . . . .	90
<b>5 File Documentation</b> . . . . .	<b>91</b>
5.1 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/controltabs.cpp File Reference . . . . .	91
5.1.1 Detailed Description . . . . .	91
5.2 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/controltabs.h File Reference . . . . .	91
5.2.1 Detailed Description . . . . .	92
5.3 controltabs.h . . . . .	92
5.4 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/logwidget.cpp File Reference . . . . .	92
5.4.1 Detailed Description . . . . .	93
5.5 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/logwidget.h File Reference . . . . .	93
5.5.1 Detailed Description . . . . .	93
5.6 logwidget.h . . . . .	94
5.7 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/mainwindow.cpp File Reference . . . . .	94
5.7.1 Detailed Description . . . . .	94
5.8 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/mainwindow.h File Reference . . . . .	95
5.8.1 Detailed Description . . . . .	95
5.9 mainwindow.h . . . . .	96
5.10 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/uiconstants.h File Reference . . . . .	97
5.10.1 Detailed Description . . . . .	97
5.11 uiconstants.h . . . . .	98
5.12 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractdataobject.cpp File Reference . . . . .	98
5.12.1 Detailed Description . . . . .	98
5.13 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractdataobject.h File Reference . . . . .	98
5.13.1 Detailed Description . . . . .	99
5.14 abstractdataobject.h . . . . .	99
5.15 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractrodcomponent.cpp File Reference . . . . .	100
5.15.1 Detailed Description . . . . .	100
5.16 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractrodcomponent.h File Reference . . . . .	101
5.16.1 Detailed Description . . . . .	101
5.17 abstractrodcomponent.h . . . . .	101
5.18 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractsectionrodcomponent.cpp File Reference . . . . .	102
5.18.1 Detailed Description . . . . .	102
5.19 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractsectionrodcomponent.h File Reference . . . . .	103
5.19.1 Detailed Description . . . . .	103
5.20 abstractsectionrodcomponent.h . . . . .	103
5.21 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/aliasdata.h File Reference . . . . .	104
5.21.1 Detailed Description . . . . .	104



5.22 aliasdata.h . . . . .	104
5.23 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/aliasdataset.h File Reference . . .	105
5.23.1 Detailed Description . . . . .	105
5.24 aliasdataset.h . . . . .	105
5.25 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.cpp File Reference . . . . .	105
5.25.1 Detailed Description . . . . .	106
5.26 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.h File Reference . . . . .	106
5.26.1 Detailed Description . . . . .	107
5.27 array.h . . . . .	107
5.28 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/constraintrodcomponent.cpp File Reference . . . . .	108
5.28.1 Detailed Description . . . . .	108
5.29 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/constraintrodcomponent.h File Reference . . . . .	108
5.29.1 Detailed Description . . . . .	109
5.30 constraintrodcomponent.h . . . . .	109
5.31 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.cpp File Reference . . . . .	110
5.31.1 Detailed Description . . . . .	110
5.32 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.h File Reference . . . . .	110
5.32.1 Detailed Description . . . . .	110
5.33 geometryrodcomponent.h . . . . .	111
5.34 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchyrodcomponent.cpp File Reference . . . . .	111
5.34.1 Detailed Description . . . . .	111
5.35 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchyrodcomponent.h File Reference . . . . .	112
5.35.1 Detailed Description . . . . .	112
5.36 hierarchyrodcomponent.h . . . . .	112
5.37 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchytree.cpp File Reference . . . . .	113
5.37.1 Detailed Description . . . . .	113
5.38 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchytree.h File Reference . . . . .	113
5.38.1 Detailed Description . . . . .	114
5.39 hierarchytree.h . . . . .	114
5.40 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.cpp File Reference . . . . .	115
5.40.1 Detailed Description . . . . .	115
5.41 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.h File Reference . . . . .	115
5.41.1 Detailed Description . . . . .	116
5.42 loadrodcomponent.h . . . . .	116
5.43 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.cpp File Reference . . . . .	117
5.43.1 Detailed Description . . . . .	117
5.44 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.h File Reference . . . . .	117
5.44.1 Detailed Description . . . . .	118

5.45 materialrodcomponent.h . . . . .	118
5.46 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.cpp File Reference . . . . .	118
5.46.1 Detailed Description . . . . .	119
5.47 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.h File Reference . . . . .	119
5.47.1 Detailed Description . . . . .	119
5.48 matrixdataobject.h . . . . .	120
5.49 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/mechanicalrodcomponent.cpp File Reference . . . . .	120
5.49.1 Detailed Description . . . . .	120
5.50 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/mechanicalrodcomponent.h File Reference . . . . .	120
5.50.1 Detailed Description . . . . .	121
5.51 mechanicalrodcomponent.h . . . . .	121
5.52 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/project-base.cpp File Reference . . . . .	122
5.52.1 Detailed Description . . . . .	123
5.53 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/project-io.cpp File Reference . . . . .	123
5.53.1 Detailed Description . . . . .	124
5.54 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/project.h File Reference . . . . .	124
5.54.1 Detailed Description . . . . .	124
5.55 project.h . . . . .	125
5.56 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/scalardataobject.cpp File Reference . . . . .	126
5.56.1 Detailed Description . . . . .	126
5.57 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/scalardataobject.h File Reference . . . . .	126
5.57.1 Detailed Description . . . . .	127
5.58 scalardataobject.h . . . . .	127
5.59 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/surfacedataobject.cpp File Reference . . . . .	127
5.59.1 Detailed Description . . . . .	127
5.60 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/surfacedataobject.h File Reference . . . . .	128
5.60.1 Detailed Description . . . . .	128
5.61 surfacedataobject.h . . . . .	128
5.62 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.cpp File Reference . . . . .	129
5.62.1 Detailed Description . . . . .	129
5.63 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.h File Reference . . . . .	129
5.63.1 Detailed Description . . . . .	129
5.64 usersectionrodcomponent.h . . . . .	130
5.65 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/utilities.cpp File Reference . . . . .	130
5.65.1 Detailed Description . . . . .	130
5.66 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/utilities.h File Reference . . . . .	131
5.66.1 Detailed Description . . . . .	131
5.67 utilities.h . . . . .	132
5.68 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.cpp File Reference . . . . .	132

5.68.1 Detailed Description	132
5.69 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.h File Reference	133
5.69.1 Detailed Description	133
5.70 vectordataobject.h	133
5.71 /home/qinterfly/Library/Projects/Current/QRodSystems/src/main/main.cpp File Reference	134
5.71.1 Detailed Description	134
5.72 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractmanager.cpp File Reference	134
5.72.1 Detailed Description	134
5.73 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractmanager.h File Reference	135
5.73.1 Detailed Description	135
5.74 abstractmanager.h	135
5.75 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractrodcomponentwidget.cpp File Reference	136
5.75.1 Detailed Description	136
5.76 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractrodcomponentwidget.h File Reference	136
5.76.1 Detailed Description	137
5.77 abstractrodcomponentwidget.h	137
5.78 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintitemdelegate.cpp File Reference	138
5.78.1 Detailed Description	138
5.79 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintitemdelegate.h File Reference	138
5.79.1 Detailed Description	139
5.80 constraintitemdelegate.h	139
5.81 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintrodcomponentwidget.cpp File Reference	139
5.81.1 Detailed Description	140
5.82 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintrodcomponentwidget.h File Reference	140
5.82.1 Detailed Description	140
5.83 constraintrodcomponentwidget.h	141
5.84 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectlineedit.cpp File Reference	141
5.84.1 Detailed Description	142
5.85 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectlineedit.h File Reference	142
5.85.1 Detailed Description	142
5.86 dataobjectlineedit.h	143
5.87 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectsmanager.cpp File Reference	143
5.87.1 Detailed Description	144

5.88	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectsmanager.h</a>	File Reference	144
5.88.1	Detailed Description		145
5.89	<a href="#">dataobjectsmanager.h</a>		145
5.90	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/doublespinboxitemdelegate.cpp</a>	File Reference	146
5.90.1	Detailed Description		146
5.91	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/doublespinboxitemdelegate.h</a>	File Reference	147
5.91.1	Detailed Description		147
5.92	<a href="#">doublespinboxitemdelegate.h</a>		147
5.93	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/geometryrodcomponentwidget.cpp</a>	File Reference	148
5.93.1	Detailed Description		148
5.94	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/geometryrodcomponentwidget.h</a>	File Reference	148
5.94.1	Detailed Description		148
5.95	<a href="#">geometryrodcomponentwidget.h</a>		149
5.96	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.cpp</a>	File Reference	149
5.96.1	Detailed Description		149
5.97	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.h</a>	File Reference	150
5.97.1	Detailed Description		150
5.98	<a href="#">loadrodcomponentwidget.h</a>		150
5.99	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/managersfactory.cpp</a>	File Reference	151
5.99.1	Detailed Description		151
5.100	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/managersfactory.h</a>	File Reference	152
5.100.1	Detailed Description		152
5.101	<a href="#">managersfactory.h</a>		152
5.102	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/materialrodcomponentwidget.cpp</a>	File Reference	153
5.102.1	Detailed Description		153
5.103	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/materialrodcomponentwidget.h</a>	File Reference	153
5.103.1	Detailed Description		154
5.104	<a href="#">materialrodcomponentwidget.h</a>		154
5.105	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/mechanicalrodcomponentwidget.cpp</a>	File Reference	155
5.105.1	Detailed Description		155
5.106	<a href="#">/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/mechanicalrodcomponentwidget.h</a>	File Reference	155
5.106.1	Detailed Description		155
5.107	<a href="#">mechanicalrodcomponentwidget.h</a>		156

5.108	/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/rodcomponentsmanager.cpp	
	File Reference	156
5.108.1	Detailed Description	157
5.109	/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/rodcomponentsmanager.h	
	File Reference	157
5.109.1	Detailed Description	158
5.110	rodcomponentsmanager.h	158
5.111	/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/usersectionrodcomponentwidget.cpp	
	File Reference	159
5.111.1	Detailed Description	159
5.112	/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/usersectionrodcomponentwidget.h	
	File Reference	160
5.112.1	Detailed Description	160
5.113	usersectionrodcomponentwidget.h	160
5.114	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchyitem.cpp	
	File Reference	161
5.114.1	Detailed Description	161
5.115	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchyitem.h	
	File Reference	161
5.115.1	Detailed Description	162
5.116	abstracthierarchyitem.h	162
5.117	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.cpp	
	File Reference	163
5.117.1	Detailed Description	163
5.118	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.h	
	File Reference	163
5.118.1	Detailed Description	164
5.119	abstracthierarchymodel.h	164
5.120	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchyitem.cpp	
	File Reference	165
5.120.1	Detailed Description	165
5.121	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchyitem.h	
	File Reference	165
5.121.1	Detailed Description	166
5.122	dataobjectshierarchyitem.h	166
5.123	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.cpp	
	File Reference	166
5.123.1	Detailed Description	167
5.124	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.h	
	File Reference	167
5.124.1	Detailed Description	167
5.125	dataobjectshierarchymodel.h	168
5.126	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.cpp	
	File Reference	168
5.126.1	Detailed Description	169

5.127 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.h	
File Reference	169
5.127.1 Detailed Description	169
5.128 projecthierarchymodel.h	170
5.129 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.cpp	
File Reference	170
5.129.1 Detailed Description	171
5.130 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.h	
File Reference	171
5.130.1 Detailed Description	171
5.131 rodcomponentshierarchyitem.h	172
5.132 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchymodel.cpp	
File Reference	172
5.132.1 Detailed Description	172
5.133 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchymodel.h	
File Reference	173
5.133.1 Detailed Description	173
5.134 rodcomponentshierarchymodel.h	173
5.135 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/abstractpropertiesmodel.cpp	
File Reference	174
5.135.1 Detailed Description	174
5.136 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/abstractpropertiesmodel.h	
File Reference	174
5.136.1 Detailed Description	175
5.137 abstractpropertiesmodel.h	175
5.138 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/dataobjectspropertiesmodel.cpp	
File Reference	176
5.138.1 Detailed Description	176
5.139 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/dataobjectspropertiesmodel.h	
File Reference	176
5.139.1 Detailed Description	177
5.140 dataobjectspropertiesmodel.h	177
5.141 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.cpp	File
Reference	177
5.141.1 Detailed Description	178
5.142 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.h	File
Reference	178
5.142.1 Detailed Description	178
5.143 basetablemodel.h	179
5.144 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/matrixtablemodel.cpp	File
Reference	179
5.144.1 Detailed Description	179
5.145 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/matrixtablemodel.h	File
Reference	180
5.145.1 Detailed Description	180
5.146 matrixtablemodel.h	180

5.147	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/surfacetablmodel.cpp	
	File Reference . . . . .	181
	5.147.1 Detailed Description . . . . .	181
5.148	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/surfacetablmodel.h File	
	Reference . . . . .	181
	5.148.1 Detailed Description . . . . .	182
5.149	surfacetablmodel.h . . . . .	182
5.150	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/tablemodelinterface.cpp	
	File Reference . . . . .	182
	5.150.1 Detailed Description . . . . .	183
5.151	/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/tablemodelinterface.h File	
	Reference . . . . .	183
	5.151.1 Detailed Description . . . . .	183
5.152	tablemodelinterface.h . . . . .	184
5.153	/home/qinterfly/Library/Projects/Current/QRodSystems/src/render/view3d.cpp File Reference . . .	184
	5.153.1 Detailed Description . . . . .	184
5.154	/home/qinterfly/Library/Projects/Current/QRodSystems/src/render/view3d.h File Reference . . .	185
	5.154.1 Detailed Description . . . . .	185
5.155	view3d.h . . . . .	185





# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

QRS::Core::Array< T > . . . . .	25
QRS::Core::HierarchyNode . . . . .	46
QRS::Core::HierarchyTree . . . . .	47
QDialog	
QRS::Managers::AbstractManager . . . . .	17
QRS::Managers::DataObjectsManager . . . . .	38
QRS::Managers::RodComponentsManager . . . . .	75
QLineEdit	
QRS::Managers::DataObjectLineEdit . . . . .	33
QMainWindow	
QRS::App::MainWindow . . . . .	53
QObject	
QRS::Core::AbstractDataObject . . . . .	11
QRS::Core::MatrixDataObject . . . . .	61
QRS::Core::ScalarDataObject . . . . .	78
QRS::Core::SurfaceDataObject . . . . .	80
QRS::Core::VectorDataObject . . . . .	88
QRS::Core::AbstractRodComponent . . . . .	19
QRS::Core::AbstractSectionRodComponent . . . . .	23
QRS::Core::UserSectionRodComponent . . . . .	86
QRS::Core::ConstraintRodComponent . . . . .	29
QRS::Core::GeometryRodComponent . . . . .	42
QRS::Core::LoadRodComponent . . . . .	48
QRS::Core::MaterialRodComponent . . . . .	57
QRS::Core::MechanicalRodComponent . . . . .	64
QRS::Core::Project . . . . .	68
QRS::Managers::ManagersFactory . . . . .	55
QOpenGLFunctions	
QRS::Graph::View3D . . . . .	90
QOpenGLWidget	
QRS::Graph::View3D . . . . .	90
QStandardItem	
QRS::HierarchyModels::AbstractHierarchyItem . . . . .	14
QRS::HierarchyModels::DataObjectsHierarchyItem . . . . .	35
QRS::HierarchyModels::RodComponentsHierarchyItem . . . . .	72

QStandardItemModel	
QRS::HierarchyModels::AbstractHierarchyModel	15
QRS::HierarchyModels::DataObjectsHierarchyModel	36
QRS::HierarchyModels::ProjectHierarchyModel	71
QRS::HierarchyModels::RodComponentsHierarchyModel	74
QRS::PropertiesModels::AbstractPropertiesModel	18
QRS::PropertiesModels::DataObjectsPropertiesModel	40
QRS::TableModels::BaseTableModel	26
QRS::TableModels::MatrixTableModel	62
QRS::TableModels::SurfaceTableModel	82
QStyledItemDelegate	
QRS::Managers::ConstraintItemDelegate	28
QRS::Managers::DoubleSpinBoxItemDelegate	41
QTableWidget	
QRS::App::LogWidget	52
QWidget	
QRS::App::ManagersTab	56
QRS::Managers::AbstractRodComponentWidget	22
QRS::Managers::ConstraintRodComponentWidget	32
QRS::Managers::GeometryRodComponentWidget	45
QRS::Managers::LoadRodComponentWidget	51
QRS::Managers::MaterialRodComponentWidget	60
QRS::Managers::MechanicalRodComponentWidget	67
QRS::Managers::UserSectionRodComponentWidget	87
QRS::Core::Array< T >::Row< U >	77
QRS::TableModels::TableModelInterface	84
QRS::TableModels::BaseTableModel	26
QRS::TableModels::MatrixTableModel	62
QRS::TableModels::SurfaceTableModel	82

## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">QRS::Core::AbstractDataObject</a>	11
Data object which is designed in the way to be represented in a table easily . . . . .	
<a href="#">QRS::HierarchyModels::AbstractHierarchyItem</a>	14
Item to represent a hierarchy of elements of the same type . . . . .	
<a href="#">QRS::HierarchyModels::AbstractHierarchyModel</a>	15
Hierarchy model which enables one to drag and drop elements of the same type . . . . .	
<a href="#">QRS::Managers::AbstractManager</a>	17
Abstract manager to create objects of different types . . . . .	
<a href="#">QRS::PropertiesModels::AbstractPropertiesModel</a>	18
Model to represent general properties . . . . .	
<a href="#">QRS::Core::AbstractRodComponent</a>	19
Component of the rod structure which characterizes one of its properties . . . . .	
<a href="#">QRS::Managers::AbstractRodComponentWidget</a>	22
Widget to construct rod components of different types . . . . .	
<a href="#">QRS::Core::AbstractSectionRodComponent</a>	23
General cross section of a rod . . . . .	
<a href="#">QRS::Core::Array&lt; T &gt;</a>	25
Numerical array class . . . . .	
<a href="#">QRS::TableModels::BaseTableModel</a>	26
Table model to represent either a scalar or vector data object . . . . .	
<a href="#">QRS::Managers::ConstraintItemDelegate</a>	28
Class to specify how options of a constraint can be edited . . . . .	
<a href="#">QRS::Core::ConstraintRodComponent</a>	29
Component to restrict movements of a rod . . . . .	
<a href="#">QRS::Managers::ConstraintRodComponentWidget</a>	32
Widget to construct constraints of a rod . . . . .	
<a href="#">QRS::Managers::DataObjectLineEdit</a>	33
Line edit widget to hold a pointer to a data object . . . . .	
<a href="#">QRS::HierarchyModels::DataObjectsHierarchyItem</a>	35
Item to represent a hierarchy of data objects . . . . .	
<a href="#">QRS::HierarchyModels::DataObjectsHierarchyModel</a>	36
Tree model to represent and modify a hierarchy of data objects . . . . .	
<a href="#">QRS::Managers::DataObjectsManager</a>	38
Manager to create objects of different types: scalars, vectors, matroces and surfaces . . . . .	
<a href="#">QRS::PropertiesModels::DataObjectsPropertiesModel</a>	40
Model to represent properties of selected data objects . . . . .	

<a href="#">QRS::Managers::DoubleSpinBoxItemDelegate</a>	
Class to specify how table values can be edited	41
<a href="#">QRS::Core::GeometryRodComponent</a>	
Geometrical configuration of a rod	42
<a href="#">QRS::Managers::GeometryRodComponentWidget</a>	
Widget to construct a geometrical rod component	45
<a href="#">QRS::Core::HierarchyNode</a>	
Hierarchy representative	46
<a href="#">QRS::Core::HierarchyTree</a>	
Hierarchy of data objects (n-array tree)	47
<a href="#">QRS::Core::LoadRodComponent</a>	
Load applied to a rod	48
<a href="#">QRS::Managers::LoadRodComponentWidget</a>	
Widget to construct a load applied to a rod	51
<a href="#">QRS::App::LogWidget</a>	
Log all the messages sent	52
<a href="#">QRS::App::MainWindow</a>	
The main window of the program	53
<a href="#">QRS::Managers::ManagersFactory</a>	
Factory to create managers which utilize and modify project data	55
<a href="#">QRS::App::ManagersTab</a>	
A toolbar consisted of object designers	56
<a href="#">QRS::Core::MaterialRodComponent</a>	
Material properties of a rod	57
<a href="#">QRS::Managers::MaterialRodComponentWidget</a>	
Widget to construct a material rod component	60
<a href="#">QRS::Core::MatrixDataObject</a>	
Matrix data object	61
<a href="#">QRS::TableModels::MatrixTableModel</a>	
Table model to represent a matrix data object	62
<a href="#">QRS::Core::MechanicalRodComponent</a>	
Stiffness and mass distributions of a rod	64
<a href="#">QRS::Managers::MechanicalRodComponentWidget</a>	
Widget to construct mechanical rod components consisted of stiffness and mass distributions	67
<a href="#">QRS::Core::Project</a>	
Project class to interact with a created system of rods	68
<a href="#">QRS::HierarchyModels::ProjectHierarchyModel</a>	
Project hierarchy representative	71
<a href="#">QRS::HierarchyModels::RodComponentsHierarchyItem</a>	
Item to represent a hierarchy of rod components	72
<a href="#">QRS::HierarchyModels::RodComponentsHierarchyModel</a>	
Tree model to represent and modify a hierarchy of rod components	74
<a href="#">QRS::Managers::RodComponentsManager</a>	
Manager to create rod components, such as a geometry, cross section and force	75
<a href="#">QRS::Core::Array&lt; T &gt;::Row&lt; U &gt;</a>	
Proxy class to acquire a row by index	77
<a href="#">QRS::Core::ScalarDataObject</a>	
Scalar data object	78
<a href="#">QRS::Core::SurfaceDataObject</a>	
Surface data object	80
<a href="#">QRS::TableModels::SurfaceTableModel</a>	
Table model to represent a surface data object	82
<a href="#">QRS::TableModels::TableModelInterface</a>	
User interface to add and remove items	84
<a href="#">QRS::Core::UserSectionRodComponent</a>	
Section which properties are defined by user	86
<a href="#">QRS::Managers::UserSectionRodComponentWidget</a>	
Widget to construct a user-defined section of a rod	87

<a href="#">QRS::Core::VectorDataObject</a>	
Vector data object . . . . .	88
<a href="#">QRS::Graph::View3D</a>	
A widget to represent the resulted rod system . . . . .	90



## Chapter 3

# File Index

### 3.1 File List

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

/home/qinterfly/Library/Projects/Current/QRodSystems/src/central/ <a href="#">controltabs.cpp</a>	
Implementation of the ControlTabs class . . . . .	91
/home/qinterfly/Library/Projects/Current/QRodSystems/src/central/ <a href="#">controltabs.h</a>	
Declaration of the ControlTabs class . . . . .	91
/home/qinterfly/Library/Projects/Current/QRodSystems/src/central/ <a href="#">logwidget.cpp</a>	
Implementation of the LogWidget class . . . . .	92
/home/qinterfly/Library/Projects/Current/QRodSystems/src/central/ <a href="#">logwidget.h</a>	
Declaration of the LogWidget class . . . . .	93
/home/qinterfly/Library/Projects/Current/QRodSystems/src/central/ <a href="#">mainwindow.cpp</a>	
Implementation of the MainWindow class . . . . .	94
/home/qinterfly/Library/Projects/Current/QRodSystems/src/central/ <a href="#">mainwindow.h</a>	
Declaration of the MainWindow class . . . . .	95
/home/qinterfly/Library/Projects/Current/QRodSystems/src/central/ <a href="#">uiconstants.h</a>	
Common graphical constants shared between several windows . . . . .	97
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">abstractdataobject.cpp</a>	
Implementation of the AbstractDataObject class . . . . .	98
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">abstractdataobject.h</a>	
Declaration of the AbstractDataObject class . . . . .	98
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">abstractrodcomponent.cpp</a>	
Definition of the AbstractRodComponent class . . . . .	100
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">abstractrodcomponent.h</a>	
Declaration of the AbstractRodComponent class . . . . .	101
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">abstractsectionrodcomponent.cpp</a>	
Definition of the AbstractSectionRodComponent class . . . . .	102
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">abstractsectionrodcomponent.h</a>	
Declaration of the AbstractSectionRodComponent class . . . . .	103
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">aliasdata.h</a>	
Specification of data types used in a project . . . . .	104
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">aliasdataset.h</a>	
Specification of types of datasets used in a project . . . . .	105
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">array.cpp</a>	
Implementation of the Array class . . . . .	105
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">array.h</a>	
Declaration of the Array class . . . . .	106
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/ <a href="#">constraintrodcomponent.cpp</a>	
Definition of the ConstraintRodComponent class . . . . .	108

/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/constraintrodcomponent.h	
Declaration of the ConstraintRodComponent class	108
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.cpp	
Definition of the GeometryRodComponent class	110
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.h	
Declaration of the GeometryRodComponent class	110
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchynode.cpp	
Implementation of the HierarchyNode class	111
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchynode.h	
Declaration of the HierarchyNode class	112
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchytree.cpp	
Implementation of the HierarchyTree class	113
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchytree.h	
Declaration of the HierarchyTree class	113
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.cpp	
Definition of the LoadRodComponent class	115
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.h	
Declaration of the LoadRodComponent class	115
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.cpp	
Definition of the MaterialRodComponent class	117
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.h	
Declaration of the MaterialRodComponent class	117
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.cpp	
Implementation of the MatrixDataObject class	118
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.h	
Declaration of the MatrixDataObject class	119
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/mechanicalrodcomponent.cpp	
Definition of the MechanicalRodComponent class	120
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/mechanicalrodcomponent.h	
Declaration of the MechanicalRodComponent class	120
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/project-base.cpp	
Implementation of the Project class	122
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/project-io.cpp	
Implementation of the Project class	123
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/project.h	
Declaration of the Project class	124
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/scalardataobject.cpp	
Implementation of the ScalarDataObject class	126
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/scalardataobject.h	
Declaration of the ScalarDataObject class	126
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/surfacedataobject.cpp	
Implementation of the SurfaceDataObject class	127
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/surfacedataobject.h	
Declaration of the SurfaceDataObject class	128
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.cpp	
Definition of the UserSectionRodComponent class	129
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.h	
Declaration of the UserSectionRodComponent class	129
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/utilities.cpp	
Implementation of utilities	130
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/utilities.h	
Declaration of utilities	131
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.cpp	
Implementation of the VectorDataObject class	132
/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.h	
Declaration of the VectorDataObject class	133
/home/qinterfly/Library/Projects/Current/QRodSystems/src/main/main.cpp	
The startup function	134



/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractmanager.cpp	
Definition of the AbstractManager class	134
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractmanager.h	
Declaration of the AbstractManager class	135
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractrodcomponentwidget.cpp	
Definition of the AbstractRodComponentWidget class	136
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractrodcomponentwidget.h	
Declaration of the AbstractRodComponentWidget class	136
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintitemdelegate.cpp	
Definition of the ComboBoxItemDelegate class	138
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintitemdelegate.h	
Declaration of the ComboBoxItemDelegate class	138
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintrodcomponentwidget.cpp	
Definition of the ConstraintRodComponentWidget class	139
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintrodcomponentwidget.h	
Declaration of the ConstraintRodComponentWidget class	140
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectlineedit.cpp	
Definition of the DataPointerLineEdit class	141
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectlineedit.h	
Declaration of the DataPointerLineEdit class	142
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectsmanager.cpp	
Implementation of the DataObjectsManager class	143
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectsmanager.h	
Declaration of the DataObjectsManager class	144
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/doublespinboxitemdelegate.cpp	
Definition of the DoubleSpinBoxItemDelegate class	146
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/doublespinboxitemdelegate.h	
Declaration of the DoubleSpinBoxItemDelegate class	147
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/geometryrodcomponentwidget.cpp	
Definiton of the GeometryComponentWidget class	148
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/geometryrodcomponentwidget.h	
Declaration of the GeometryComponentWidget class	148
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.cpp	
Definition of the LoadRodComponentWidget class	149
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.h	
Declaration of the LoadRodComponentWidget class	150
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/managersfactory.cpp	
Definition of the ManagersFactory class	151
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/managersfactory.h	
Declaration of the ManagersFactory class	152
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/materialrodcomponentwidget.cpp	
Definition of the MaterialRodComponentWidget class	153
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/materialrodcomponentwidget.h	
Declaration of the MaterialRodComponentWidget class	153
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/mechanicalrodcomponentwidget.cpp	
Definition of the MechanicalRodComponentWidget class	155
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/mechanicalrodcomponentwidget.h	
Declaration of the MechanicalRodComponentWidget class	155
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/rodcomponentsmanager.cpp	
Definition of the RodComponentsManager class	156
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/rodcomponentsmanager.h	
Declaration of the RodComponentsManager class	157
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/usersectionrodcomponentwidget.cpp	
Definition of the UserSectionRodComponentWidget class	159
/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/usersectionrodcomponentwidget.h	
Declaration of the UserSectionRodComponentWidget class	160
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchyitem.cpp	
Definition of the AbstractHierarchyItem class	161

/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchyitem.h	
Declaration of the AbstractHierarchyItem class . . . . .	161
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.cpp	
Definition of the AbstractHierarchyModel class . . . . .	163
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.h	
Declaration of the AbstractHierarchyModel class . . . . .	163
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchyitem.cpp	
Definition of the DataObjectsHierarchyItem class . . . . .	165
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchyitem.h	
Declaration of the DataObjectsHierarchyItem class . . . . .	165
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.cpp	
Definition of the DataObjectsHierarchyModel class . . . . .	166
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.h	
Declaration of the DataObjectsHierarchyModel class . . . . .	167
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.cpp	
Definition of the ProjectHierarchyModel class . . . . .	168
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.h	
Declaration of the ProjectHierarchyModel class . . . . .	169
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.cpp	
Definition of the RodComponentsHierarchyItem class . . . . .	170
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.h	
Declaration of the RodComponentsHierarchyItem class . . . . .	171
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchymodel.cpp	
Definition of the RodComponentsHierarchyModel class . . . . .	172
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchymodel.h	
Declaration of the RodComponentsHierarchyModel class . . . . .	173
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/abstractpropertiesmodel.cpp	
Definition of the AbstractPropertiesModel class . . . . .	174
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/abstractpropertiesmodel.h	
Declaration of the AbstractPropertiesModel class . . . . .	174
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/dataobjectspropertiesmodel.cpp	
Definition of the DataObjectsPropertiesModel class . . . . .	176
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/dataobjectspropertiesmodel.h	
Declaration of the DataObjectsPropertiesModel class . . . . .	176
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.cpp	
Implementation of the BaseTableModel class . . . . .	177
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.h	
Declaration of the BaseTableModel class . . . . .	178
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/matrixtablemodel.cpp	
Implementation of the MatrixTableModel class . . . . .	179
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/matrixtablemodel.h	
Declaration of the MatrixTableModel class . . . . .	180
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/surfacetablemodel.cpp	
Implementation of the SurfaceTableModel class . . . . .	181
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/surfacetablemodel.h	
Declaration of the SurfaceTableModel class . . . . .	181
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/tablemodelinterface.cpp	
Implementation of static functions of TableModelInterface . . . . .	182
/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/tablemodelinterface.h	
Declaration of the TableModelInterface . . . . .	183
/home/qinterfly/Library/Projects/Current/QRodSystems/src/render/view3d.cpp	
Implementation of the View3D class . . . . .	184
/home/qinterfly/Library/Projects/Current/QRodSystems/src/render/view3d.h	
Declaration of the View3D class . . . . .	185

## Chapter 4

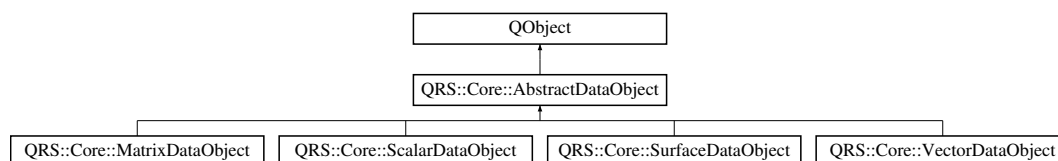
# Class Documentation

### 4.1 QRS::Core::AbstractDataObject Class Reference

Data object which is designed in the way to be represented in a table easily.

```
#include <abstractdataobject.h>
```

Inheritance diagram for QRS::Core::AbstractDataObject:



#### Public Types

- enum **ObjectType** { **kScalar** , **kVector** , **kMatrix** , **kSurface** }

#### Public Member Functions

- **AbstractDataObject** (ObjectType type, QString const &name)  
*Base constructor.*
- virtual [AbstractDataObject](#) \* **clone** () const =0
- virtual [DataItemType](#) & **addItem** (DataKeyType key)=0
- void **removeItem** (DataValueType key)  
*Remove an entity with the specified key.*
- bool **changeItemKey** (DataKeyType oldKey, DataKeyType newKey, DataHolder \*items=nullptr)  
*Modify a key existed.*
- DataValueType [getAvailableItemKey](#) (DataValueType key, DataHolder const \*items=nullptr) const
- bool **setArrayValue** (DataKeyType key, DataValueType newValue, IndexType iRow=0, IndexType iColumn=0)  
*Set an array value with the specified indices.*
- quint32 **numberOfItems** () const
- DataHolder const & **getItems** ()
- DataIDType **id** () const

- ObjectType **type** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const  
*Serialize an abstract data object.*
- virtual void **deserialize** (QDataStream &stream)  
*Partly deserialize an abstract data object.*
- virtual void **import** (QTextStream &stream)=0

### Static Public Member Functions

- static DataIDType **maxObjectID** ()
- static void **setMaxObjectID** (DataIDType iMaxObjectID)

### Protected Attributes

- const ObjectType **mkType**
- QString **mName**
- DataIDType **mID**
- DataHolder **mItems**

### Static Private Attributes

- static DataIDType **smMaxObjectID** = 0

### Friends

- QDataStream & **operator<<** (QDataStream &stream, [AbstractDataObject](#) const &obj)  
*Print a data object to a stream.*

## 4.1.1 Detailed Description

Data object which is designed in the way to be represented in a table easily.

## 4.1.2 Member Function Documentation

### 4.1.2.1 addItem()

```
virtual DataItemType & QRS::Core::AbstractDataObject::addItem (
    DataKeyType key ) [pure virtual]
```

Implemented in [QRS::Core::MatrixDataObject](#), [QRS::Core::ScalarDataObject](#), [QRS::Core::SurfaceDataObject](#), and [QRS::Core::VectorDataObject](#).

#### 4.1.2.2 clone()

```
virtual AbstractDataObject * QRS::Core::AbstractDataObject::clone ( ) const [pure virtual]
```

Implemented in [QRS::Core::MatrixDataObject](#), [QRS::Core::ScalarDataObject](#), [QRS::Core::SurfaceDataObject](#), and [QRS::Core::VectorDataObject](#).

#### 4.1.2.3 deserialize()

```
void AbstractDataObject::deserialize (
    QDataStream & stream ) [virtual]
```

Partly deserialize an abstract data object.

It is assumed that a type and name have already been assigned. So, only an identifier and items need to be set.

Reimplemented in [QRS::Core::SurfaceDataObject](#).

#### 4.1.2.4 getAvailableItemKey()

```
DataValueType AbstractDataObject::getAvailableItemKey (
    DataValueType key,
    DataHolder const * items = nullptr ) const
```

Check if a given key is unique

##### Returns

Returns the input value of the key if it is unique, otherwise – a first available key

#### 4.1.2.5 import()

```
virtual void QRS::Core::AbstractDataObject::import (
    QTextStream & stream ) [pure virtual]
```

Implemented in [QRS::Core::MatrixDataObject](#), [QRS::Core::ScalarDataObject](#), [QRS::Core::SurfaceDataObject](#), and [QRS::Core::VectorDataObject](#).

#### 4.1.2.6 serialize()

```
void AbstractDataObject::serialize (
    QDataStream & stream ) const [virtual]
```

Serialize an abstract data object.

Reimplemented in [QRS::Core::SurfaceDataObject](#).

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

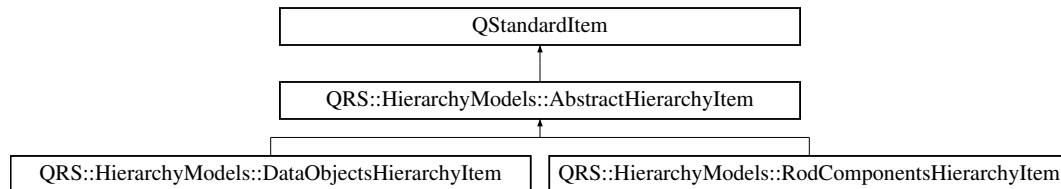
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[abstractdataobject.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[abstractdataobject.cpp](#)

## 4.2 QRS::HierarchyModels::AbstractHierarchyItem Class Reference

Item to represent a hierarchy of elements of the same type.

```
#include <abstracthierarchyitem.h>
```

Inheritance diagram for QRS::HierarchyModels::AbstractHierarchyItem:



### Public Types

- enum **ItemType** { **kDataObjects** = QStandardItem::UserType , **kRodComponents** }

### Public Member Functions

- AbstractHierarchyItem** (QIcon const &icon, QString const &text, [Core::HierarchyNode](#) \*pNode)
- void **writePointer** (QDataStream &out) const  
*Write the pointer to the current item to a stream.*
- virtual int **type** () const =0

### Static Public Member Functions

- static [AbstractHierarchyItem](#) \* **readPointer** (QDataStream &in)  
*Retrieve a pointer to an item from a stream.*

### Protected Attributes

- [Core::HierarchyNode](#) \* **mpNode** = nullptr

### Friends

- class **AbstractHierarchyModel**
- class **PropertiesModels::AbstractPropertiesModel**

### 4.2.1 Detailed Description

Item to represent a hierarchy of elements of the same type.

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

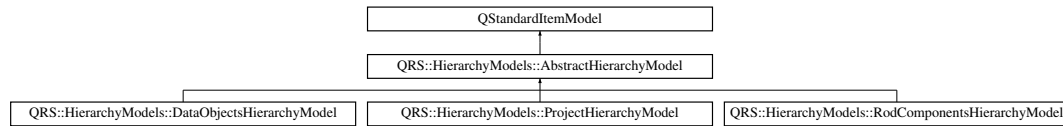
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[abstracthierarchyitem.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[abstracthierarchyitem.cpp](#)

## 4.3 QRS::HierarchyModels::AbstractHierarchyModel Class Reference

Hierarchy model which enables one to drag and drop elements of the same type.

```
#include <abstrachierarchymodel.h>
```

Inheritance diagram for QRS::HierarchyModels::AbstractHierarchyModel:



### Signals

- void **hierarchyChanged** ()  
*Emitted when hierarchical elements get renamed, moved or deleted.*

### Public Member Functions

- **AbstractHierarchyModel** (QString const &mimeType, QTreeView \*pView=nullptr)
- virtual void **updateContent** ()=0
- virtual void **clearContent** ()=0
- Qt::DropActions **supportedDragActions** () const override  
*Specify allowed drag actions.*
- Qt::DropActions **supportedDropActions** () const override  
*Specify allowed drop actions.*
- QStringList **mimeTypes** () const override  
*Retrieve the mime types.*
- QMimeData \* **mimeData** (const QModelIndexList &indicies) const override  
*Encode each item according to a given list of indicies.*
- bool **dropMimeData** (QMimeData const \*pMimeData, Qt::DropAction action, int row, int column, const QModelIndex &parent) override  
*Process the drop action.*

### Protected Attributes

- QString const **mkMimeType**

### Private Member Functions

- bool **processDropOnItem** (QDataStream &stream, int &numItems, QModelIndex const &indexParent)  
*Merge several items into one entity.*
- bool **processDropBetweenItems** (QDataStream &stream, int &numItems, QModelIndex const &indexParent, int row)  
*Change the order of items.*
- void **retrieveExpandedState** (NodesState &nodesState, QModelIndex const &indexParent, QTreeView const \*pView)  
*Retrieve information about whether each directory is expanded.*
- void **setExpandedState** (NodesState &nodesState, QModelIndex const &indexParent, QTreeView \*pView)  
*Set an expanded state of each directory.*
- void **updateContentExpanded** ()

### 4.3.1 Detailed Description

Hierarchy model which enables one to drag and drop elements of the same type.

### 4.3.2 Member Function Documentation

#### 4.3.2.1 clearContent()

```
virtual void QRS::HierarchyModels::AbstractHierarchyModel::clearContent ( ) [pure virtual]
```

Implemented in [QRS::HierarchyModels::DataObjectsHierarchyModel](#), [QRS::HierarchyModels::ProjectHierarchyModel](#), and [QRS::HierarchyModels::RodComponentsHierarchyModel](#).

#### 4.3.2.2 updateContent()

```
virtual void QRS::HierarchyModels::AbstractHierarchyModel::updateContent ( ) [pure virtual]
```

Implemented in [QRS::HierarchyModels::DataObjectsHierarchyModel](#), [QRS::HierarchyModels::ProjectHierarchyModel](#), and [QRS::HierarchyModels::RodComponentsHierarchyModel](#).

#### 4.3.2.3 updateContentExpanded()

```
void AbstractHierarchyModel::updateContentExpanded ( ) [private]
```

Since items are destroyed whenever the content is updated, an expanded state of each directory is saved and then set again.

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

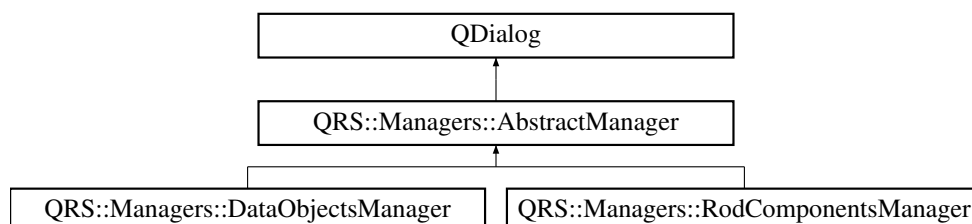
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.cpp](#)

## 4.4 QRS::Managers::AbstractManager Class Reference

Abstract manager to create objects of different types.

```
#include <abstractmanager.h>
```

Inheritance diagram for QRS::Managers::AbstractManager:





## Public Types

- enum **ManagerType** { **kDataObjects** , **kRodComponents** , **kRodConstructor** }

## Public Slots

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

## Signals

- void **closed** (QRS::Managers::AbstractManager::ManagerType type)

## Public Member Functions

- **AbstractManager** (QString &lastPath, QSettings &settings, ManagerType type, QString groupName, QWidget \*parent=nullptr)
- void **saveSettings** ()  
*Save settings to a file.*
- void **restoreSettings** ()  
*Restore settings from a file.*

## Protected Member Functions

- void **closeEvent** (QCloseEvent \*pEvent) override  
*Save settings and delete handling widgets before closing the window.*
- void **setToolBarShortcutHints** (QToolBar \*pToolBar)  
*Helper function to add a shortcut hint to all actions which a toolbar contains.*

## Protected Attributes

- ads::CDockManager \* **mpDockManager** = nullptr
- QString & **mLastPath**

## Private Attributes

- QSettings & **mSettings**
- ManagerType const **mkType**
- QString const **mkGroupName**

### 4.4.1 Detailed Description

Abstract manager to create objects of different types.

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

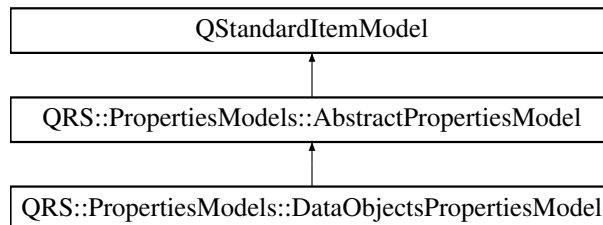
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[abstractmanager.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[abstractmanager.cpp](#)

## 4.5 QRS::PropertiesModels::AbstractPropertiesModel Class Reference

Model to represent general properties.

```
#include <abstractpropertiesmodel.h>
```

Inheritance diagram for QRS::PropertiesModels::AbstractPropertiesModel:



### Signals

- void **propertyChanged** ()

### Public Member Functions

- **AbstractPropertiesModel** (QTableView \*pView, QVector< [HierarchyModels::AbstractHierarchyItem](#) \* > items)

### Protected Slots

- virtual void **modifyProperty** (QStandardItem \*pChangedProperty)=0
- void **modifyDirectoryName** (QString const &name)

*Change names of selected directories.*

### Protected Member Functions

- void **setDirectoryAttributes** ()
- QList< QStandardItem \* > **preparePropertyRow** (int type, QString const &title, QVariant const &value, bool isValueEditable) const

*Prepare a row to insert into the table.*

### Protected Attributes

- QVector< [HierarchyModels::AbstractHierarchyItem](#) \* > **mItems**
- bool **mIsDirectory**
- QString const **mkEmptyProperty** = ""

### Private Types

- enum **PropertyDirectory** { **kName** , **kNumberChildren** }

### 4.5.1 Detailed Description

Model to represent general properties.

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

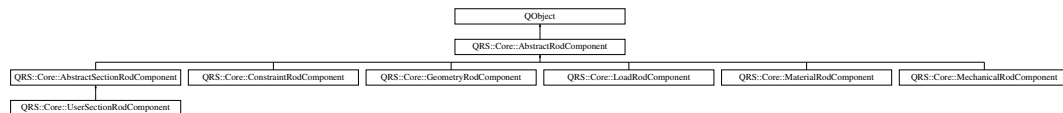
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/[abstractpropertiesmodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/[abstractpropertiesmodel.cpp](#)

## 4.6 QRS::Core::AbstractRodComponent Class Reference

Component of the rod structure which characterizes one of its properties.

```
#include <abstractrodcomponent.h>
```

Inheritance diagram for QRS::Core::AbstractRodComponent:



### Public Types

- enum **ComponentType** {  
**kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
**kConstraint** , **kMechanical** }

### Public Member Functions

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual [AbstractRodComponent](#) \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

### Static Public Member Functions

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

### Protected Member Functions

- void **writeDataObjectPointer** (QDataStream &stream, [AbstractDataObject](#) const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- [AbstractDataObject](#) const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- [AbstractDataObject](#) const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- [AbstractDataObject](#) const \* **substituteDataObject** (DataObjects const &dataObjects, [AbstractDataObject](#) const \*pDataObject) const  
*Substitute a data object with its updated version.*

### Protected Attributes

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

### Static Private Attributes

- static DataIDType **smMaxComponentID** = 0

### Friends

- QDataStream & **operator<<** (QDataStream &stream, [AbstractRodComponent](#) const &component)  
*Print a rod component to a stream.*

## 4.6.1 Detailed Description

Component of the rod structure which characterizes one of its properties.

## 4.6.2 Member Function Documentation

### 4.6.2.1 clone()

```
virtual AbstractRodComponent * QRS::Core::AbstractRodComponent::clone ( ) const [pure virtual]
```

Implemented in [QRS::Core::ConstraintRodComponent](#), [QRS::Core::GeometryRodComponent](#), [QRS::Core::LoadRodComponent](#), [QRS::Core::MaterialRodComponent](#), [QRS::Core::MechanicalRodComponent](#), and [QRS::Core::UserSectionRodComponent](#).

### 4.6.2.2 deserialize()

```
virtual void QRS::Core::AbstractRodComponent::deserialize (
    QDataStream & stream,
    DataObjects const & dataObjects ) [pure virtual]
```

Implemented in [QRS::Core::AbstractSectionRodComponent](#), [QRS::Core::ConstraintRodComponent](#), [QRS::Core::GeometryRodComponent](#), [QRS::Core::LoadRodComponent](#), [QRS::Core::MaterialRodComponent](#), and [QRS::Core::MechanicalRodComponent](#).

#### 4.6.2.3 isDataComplete()

```
virtual bool QRS::Core::AbstractRodComponent::isDataComplete ( ) const [pure virtual]
```

Implemented in [QRS::Core::GeometryRodComponent](#), [QRS::Core::LoadRodComponent](#), [QRS::Core::MaterialRodComponent](#), and [QRS::Core::UserSectionRodComponent](#).

#### 4.6.2.4 resolveReferences()

```
virtual void QRS::Core::AbstractRodComponent::resolveReferences (
    DataObjects const & dataObjects ) [pure virtual]
```

Implemented in [QRS::Core::AbstractSectionRodComponent](#), [QRS::Core::GeometryRodComponent](#), [QRS::Core::LoadRodComponent](#), [QRS::Core::MaterialRodComponent](#), and [QRS::Core::MechanicalRodComponent](#).

#### 4.6.2.5 serialize()

```
virtual void QRS::Core::AbstractRodComponent::serialize (
    QDataStream & stream ) const [pure virtual]
```

Implemented in [QRS::Core::AbstractSectionRodComponent](#), [QRS::Core::ConstraintRodComponent](#), [QRS::Core::GeometryRodComponent](#), [QRS::Core::LoadRodComponent](#), [QRS::Core::MaterialRodComponent](#), and [QRS::Core::MechanicalRodComponent](#).

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

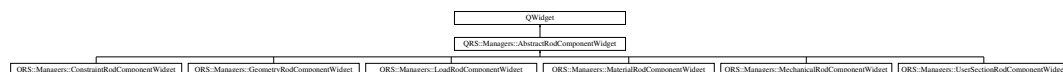
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractrodcomponent.cpp](#)

## 4.7 QRS::Managers::AbstractRodComponentWidget Class Reference

Widget to construct rod components of different types.

```
#include <abstractrodcomponentwidget.h>
```

Inheritance diagram for QRS::Managers::AbstractRodComponentWidget:



### Signals

- void **modified** ()
- void **editDataObjectRequested** (Core::DataIDType id)

### Public Member Functions

- **AbstractRodComponentWidget** (QString const &mimeType, QWidget \*parent=nullptr)

### Protected Member Functions

- void **setDataObjectEditConnections** ([DataObjectLineEdit](#) \*pEdit, DataObjectSetFun &setFun)  
*Specify connections of an editor which hold pointers to data objects of different types.*

### Protected Attributes

- QString const **mkMimeType**

## 4.7.1 Detailed Description

Widget to construct rod components of different types.

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

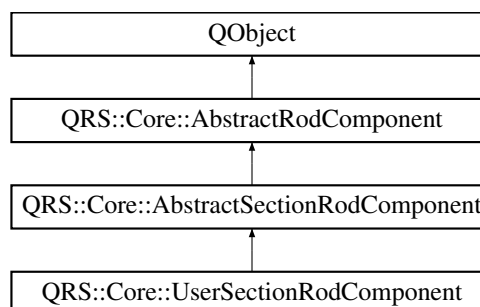
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[abstractrodcomponentwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[abstractrodcomponentwidget.cpp](#)

## 4.8 QRS::Core::AbstractSectionRodComponent Class Reference

General cross section of a rod.

```
#include <abstractsectionrodcomponent.h>
```

Inheritance diagram for QRS::Core::AbstractSectionRodComponent:



### Public Types

- enum **SectionType** { **kUserDefined** }

### Public Types inherited from [QRS::Core::AbstractRodComponent](#)

- enum **ComponentType** {  
  **kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
  **kConstraint** , **kMechanical** }

### Public Member Functions

- **AbstractSectionRodComponent** (SectionType sectionType, QString const &name)
- virtual **~AbstractSectionRodComponent** ()=0  
*Decrease a number of instances while being destroyed.*
- void **serialize** (QDataStream &stream) const override  
*Serialize a cross section.*
- void **deserialize** (QDataStream &stream, DataObjects const &dataObjects) override  
*Partly deserialize an abstract rod component.*
- void **resolveReferences** (DataObjects const &dataObjects) override  
*Resolve references of a cross-section.*
- SectionType **sectionType** () const

### Public Member Functions inherited from QRS::Core::AbstractRodComponent

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual **AbstractRodComponent** \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

### Static Public Member Functions

- static quint32 **numberInstances** ()

### Static Public Member Functions inherited from QRS::Core::AbstractRodComponent

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

### Protected Member Functions

- void **copyIntegratedProperties** (**AbstractSectionRodComponent** const \*pSection)  
*Copy integrated properties of a cross section.*

### Protected Member Functions inherited from QRS::Core::AbstractRodComponent

- void **writeDataObjectPointer** (QDataStream &stream, **AbstractDataObject** const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- **AbstractDataObject** const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- **AbstractDataObject** const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- **AbstractDataObject** const \* **substituteDataObject** (DataObjects const &dataObjects, **AbstractDataObject** const \*pDataObject) const  
*Substitute a data object with its updated version.*

### Protected Attributes

- SectionType const **mkSectionType**
- QPointer< [ScalarDataObject](#) const > **mpArea**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMomentTorsional**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMomentX**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMomentY**
- QPointer< [ScalarDataObject](#) const > **mpCenterCoordinateX**
- QPointer< [ScalarDataObject](#) const > **mpCenterCoordinateY**

### Protected Attributes inherited from [QRS::Core::AbstractRodComponent](#)

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

### Static Protected Attributes

- static quint32 **smNumInstances** = 0

## 4.8.1 Detailed Description

General cross section of a rod.

## 4.8.2 Member Function Documentation

### 4.8.2.1 deserialize()

```
void AbstractSectionRodComponent::deserialize (
    QDataStream & stream,
    DataObjects const & dataObjects ) [override], [virtual]
```

Partly deserialize an abstract rod component.

It is assumed that a type and name have already been assigned. So, only integrated properties need to be set.

Implements [QRS::Core::AbstractRodComponent](#).

### 4.8.2.2 resolveReferences()

```
void AbstractSectionRodComponent::resolveReferences (
    DataObjects const & dataObjects ) [override], [virtual]
```

Resolve references of a cross-section.

Implements [QRS::Core::AbstractRodComponent](#).



## 4.8.2.3 serialize()

```
void AbstractSectionRodComponent::serialize (
    QDataStream & stream ) const [override], [virtual]
```

Serialize a cross section.

Implements [QRS::Core::AbstractRodComponent](#).

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

- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[abstractsectionrodcomponent.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[abstractsectionrodcomponent.cpp](#)

## 4.9 QRS::Core::Array&lt; T &gt; Class Template Reference

Numerical array class.

```
#include <array.h>
```

## Classes

- struct [Row](#)  
*Proxy class to acquire a row by index.*

## Public Member Functions

- **Array** (IndexType numRows=0, IndexType numCols=0)
- **Array** ([Array](#)< T > const &another)  
*Copy constructor.*
- **Array** ([Array](#)< T > &&another)  
*Move constructor.*
- T \* **data** ()
- void **resize** (IndexType numRows, IndexType numCols)  
*Resize and copy previous values if possible.*
- void **removeColumn** (IndexType iRemoveColumn)  
*Remove a column by index.*
- void **swapColumns** (IndexType iFirstColumn, IndexType iSecondColumn)  
*Swap two columns.*
- IndexType **rows** () const
- IndexType **cols** () const
- IndexType **size** () const
- [Row](#)< T > **operator[]** (IndexType iRow)
- [Row](#)< T > **operator[]** (IndexType iRow) const
- [Array](#) & **operator=** ([Array](#)< T > const &another)  
*Assignment operator.*

### Private Attributes

- IndexType **mNumRows**  
*Number of rows.*
- IndexType **mNumCols**  
*Number of columns.*
- T \* **mpData** = nullptr  
*Pointer to the data stored.*

### Friends

- template<typename K >  
QDebug **operator**<< (QDebug stream, [Array](#)< K > &array)  
*Print all array values using the matrix format.*
- template<typename K >  
QDataStream & **operator**<< (QDataStream &stream, [Array](#)< K > const &array)  
*Write an array to a stream.*
- template<typename K >  
QDataStream & **operator**>> (QDataStream &stream, [Array](#)< K > &array)  
*Read an array from a stream.*

## 4.9.1 Detailed Description

**template<typename T>**  
**class QRS::Core::Array< T >**

Numerical array class.

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

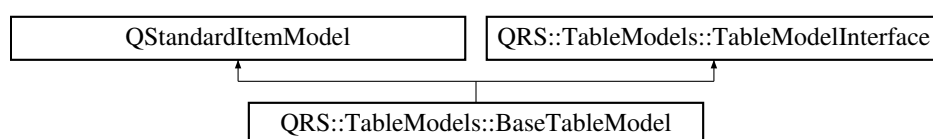
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.h>
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.cpp>

## 4.10 QRS::TableModels::BaseTableModel Class Reference

Table model to represent either a scalar or vector data object.

```
#include <basetablemodel.h>
```

Inheritance diagram for QRS::TableModels::BaseTableModel:



## Public Member Functions

- **BaseTableModel** (QWidget \*parent=nullptr)
- void **setDataObject** (Core::AbstractDataObject \*pDataObject)  
*Set a data object to represent.*
- bool **setData** (const QModelIndex &indexEdit, const QVariant &value, int role=Qt::EditRole) override  
*Set the data acquired from a delegate.*
- void **insertItemAfterSelected** (QItemSelectionModel \*pSelectionModel) override  
*Insert a new item after selected one.*
- void **insertLeadingItemAfterSelected** (QItemSelectionModel \*) override
- void **removeSelectedItem** (QItemSelectionModel \*pSelectionModel) override  
*Remove an array under selection.*
- void **removeSelectedLeadingItem** (QItemSelectionModel \*) override
- virtual void **insertItemAfterSelected** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **insertLeadingItemAfterSelected** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **removeSelectedItem** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **removeSelectedLeadingItem** (QItemSelectionModel \*pSelectionModel)=0

## Private Member Functions

- void **updateContent** ()  
*Represent all items which a data object contains.*
- void **clearContent** ()  
*Clear previously created items.*

## Private Attributes

- Core::AbstractDataObject \* **mpDataObject** = nullptr

## Additional Inherited Members

## Static Public Member Functions inherited from QRS::TableModels::TableModelInterface

- static QStandardItem \* **makeDoubleItem** (double value)  
*Helper function to make an item which holds a double value.*
- static QList< QStandardItem \* > **prepareRow** (Core::Array< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array.*
- static QList< QStandardItem \* > **prepareRow** (double const &key, Core::Array< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with an key.*
- static QList< QStandardItem \* > **prepareRow** (QString const &name, Core::Array< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with a name.*
- static QStandardItem \* **makeLabelItem** (QString const &name)  
*Helper function to create an item which holds a string and cannot be modified.*

### 4.10.1 Detailed Description

Table model to represent either a scalar or vector data object.

### 4.10.2 Member Function Documentation

#### 4.10.2.1 insertItemAfterSelected()

```
void BaseTableModel::insertItemAfterSelected (
    QTableWidgetItem * pSelectionModel ) [override], [virtual]
```

Insert a new item after selected one.

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.10.2.2 insertLeadingItemAfterSelected()

```
void QRS::TableModels::BaseTableModel::insertLeadingItemAfterSelected (
    QTableWidgetItem * ) [inline], [override], [virtual]
```

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.10.2.3 removeSelectedItem()

```
void BaseTableModel::removeSelectedItem (
    QTableWidgetItem * pSelectionModel ) [override], [virtual]
```

Remove an array under selection.

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.10.2.4 removeSelectedLeadingItem()

```
void QRS::TableModels::BaseTableModel::removeSelectedLeadingItem (
    QTableWidgetItem * ) [inline], [override], [virtual]
```

Implements [QRS::TableModels::TableModelInterface](#).

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

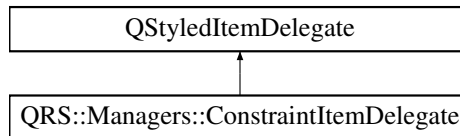
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.cpp](#)

## 4.11 QRS::Managers::ConstraintItemDelegate Class Reference

Class to specify how options of a constraint can be edited.

```
#include <constraintitemdelegate.h>
```

Inheritance diagram for QRS::Managers::ConstraintItemDelegate:



### Signals

- void **typeCreated** (int iRow) const
- void **typeChanged** (int iRow, Core::ConstraintRodComponent::ConstraintType oldType) const
- void **coordinateSystemChanged** (int iRow) const

### Public Member Functions

- **ConstraintItemDelegate** ([Core::ConstraintRodComponent](#) const &constraintRodComponent, ConstraintTypeNames const &types, ConstraintCoordinateSystemNames const &coordinateSystems, QObject \*parent=nullptr)
- QWidget \* **createEditor** (QWidget \*pCell, const QStyleOptionViewItem &option, const QModelIndex &index) const override  
*Create a comboBox to choose items.*
- void **setEditorData** (QWidget \*pEditor, const QModelIndex &index) const override  
*Specify data to show.*
- void **setModelData** (QWidget \*pEditor, QAbstractItemModel \*pModel, const QModelIndex &index) const override  
*Set data to a model.*
- void **updateEditorGeometry** (QWidget \*pEditor, const QStyleOptionViewItem &option, const QModelIndex &index) const override  
*Set a geometry to render.*

### Private Attributes

- [Core::ConstraintRodComponent](#) const & **mConstraintRodComponent**
- ConstraintTypeNames const & **mTypes**
- ConstraintCoordinateSystemNames const & **mCoordinateSystems**

#### 4.11.1 Detailed Description

Class to specify how options of a constraint can be edited.

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

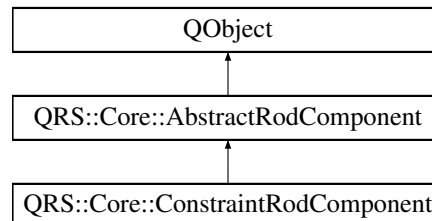
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[constraintitemdelegate.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[constraintitemdelegate.cpp](#)

## 4.12 QRS::Core::ConstraintRodComponent Class Reference

Component to restrict movements of a rod.

```
#include <constraintrodcomponent.h>
```

Inheritance diagram for QRS::Core::ConstraintRodComponent:



### Public Types

- enum **ConstraintType** {  
  **kDisplacementX** , **kDisplacementY** , **kDisplacementZ** , **kRotationX** ,  
  **kRotationY** , **kRotationZ** }
- enum **ConstraintCoordinateSystem** { **kGlobal** , **kLocal** }
- using **Constraints** = std::map< ConstraintType, ConstraintCoordinateSystem >

### Public Types inherited from [QRS::Core::AbstractRodComponent](#)

- enum **ComponentType** {  
  **kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
  **kConstraint** , **kMechanical** }

### Public Member Functions

- **ConstraintRodComponent** (QString const &name)
- **~ConstraintRodComponent** ()  
  *Decrease a number of instances while being destroyed.*
- [AbstractRodComponent](#) \* **clone** () const override  
  *Clone a constraint rod component.*
- bool **isDataComplete** () const override
- void **serialize** (QDataStream &stream) const override  
  *Serialize all properties of a constraint component.*
- void **deserialize** (QDataStream &stream, DataObjects const &dataObjects) override  
  *Deserialize a constraint component.*
- void **resolveReferences** (DataObjects const &) override
- bool **isConstraintExist** (ConstraintType type) const  
  *Check whether the constraint of the specified type exists.*
- void **setConstraint** (ConstraintType type, ConstraintCoordinateSystem coordinateSystem)  
  *Set a constraint.*
- bool **removeConstraint** (ConstraintType type)  
  *Remove the constraint of a given type.*
- Constraints const & **constraints** () const

## Public Member Functions inherited from QRS::Core::AbstractRodComponent

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual [AbstractRodComponent](#) \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Static Public Member Functions inherited from QRS::Core::AbstractRodComponent

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

## Private Attributes

- Constraints **mConstraints**

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

## Protected Member Functions inherited from QRS::Core::AbstractRodComponent

- void **writeDataObjectPointer** (QDataStream &stream, [AbstractDataObject](#) const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- [AbstractDataObject](#) const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- [AbstractDataObject](#) const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- [AbstractDataObject](#) const \* **substituteDataObject** (DataObjects const &dataObjects, [AbstractDataObject](#) const \*pDataObject) const  
*Substitute a data object with its updated version.*

## Protected Attributes inherited from QRS::Core::AbstractRodComponent

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

### 4.12.1 Detailed Description

Component to restrict movements of a rod.

### 4.12.2 Member Function Documentation

#### 4.12.2.1 clone()

```
AbstractRodComponent * ConstraintRodComponent::clone ( ) const [override], [virtual]
```

Clone a constraint rod component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.12.2.2 deserialize()

```
void ConstraintRodComponent::deserialize (
    QDataStream & stream,
    DataObjects const & dataObjects ) [override], [virtual]
```

Deserialize a constraint component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.12.2.3 isDataComplete()

```
bool QRS::Core::ConstraintRodComponent::isDataComplete ( ) const [inline], [override], [virtual]
```

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.12.2.4 resolveReferences()

```
void QRS::Core::ConstraintRodComponent::resolveReferences (
    DataObjects const & ) [inline], [override], [virtual]
```

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.12.2.5 serialize()

```
void ConstraintRodComponent::serialize (
    QDataStream & stream ) const [override], [virtual]
```

Serialize all properties of a constraint component.

Implements [QRS::Core::AbstractRodComponent](#).

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

- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/constraintrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/constraintrodcomponent.cpp](#)

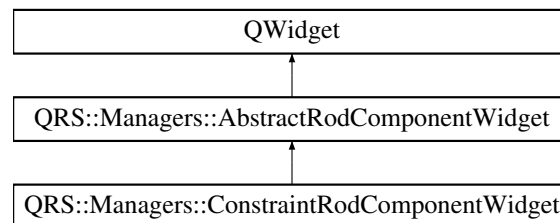


## 4.13 QRS::Managers::ConstraintRodComponentWidget Class Reference

Widget to consturct constraints of a rod.

```
#include <constraintrodcomponentwidget.h>
```

Inheritance diagram for QRS::Managers::ConstraintRodComponentWidget:



### Public Member Functions

- **ConstraintRodComponentWidget** ([Core::ConstraintRodComponent](#) &constraintRodComponent, QWidget \*parent=nullptr)

### Public Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- **AbstractRodComponentWidget** (QString const &mimeType, QWidget \*parent=nullptr)

### Private Slots

- void **setConstraintData** (int iRow)  
*Change a constraint property.*

### Private Member Functions

- void **createContent** ()  
*Create all the widgets.*
- QToolBar \* **createToolBar** ()  
*Create a toolbar to add and remove constraints.*
- void **createTableWidget** ()  
*Create a table to construct constraints.*
- void **addRow** ()  
*Add a row at the end of the table.*
- void **removeSelectedRows** ()  
*Remove selected rows from the table.*
- void **representConstraintData** ()  
*Represent existing constraints.*
- void **setTableHeight** ()  
*Set the height of the table to be enough to represent all rows.*
- void **specifyConstraintNames** ()  
*Specify names of constraints.*
- QVariant **getItemData** (int iRow, int iColumn)  
*Retrieve item data.*

### Private Attributes

- [Core::ConstraintRodComponent](#) & **mConstraintRodComponent**
- QWidget \* **mpTableConstraint**
- [ConstraintItemDelegate](#) \* **mplItemDelegate**
- ConstraintTypeNames **mTypeNames**
- ConstraintCoordinateSystemNames **mCoordinateSystemNames**

### Additional Inherited Members

### Signals inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- void **modified** ()
- void **editDataObjectRequested** (Core::DataIDType id)

### Protected Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- void **setDataObjectEditConnections** ([DataObjectLineEdit](#) \*pEdit, DataObjectSetFun &setFun)  
*Specify connections of an editor which hold pointers to data objects of different types.*

### Protected Attributes inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- QString const **mkMimeType**

## 4.13.1 Detailed Description

Widget to construct constraints of a rod.

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

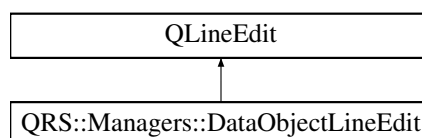
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[constraintrodcomponentwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[constraintrodcomponentwidget.cpp](#)

## 4.14 QRS::Managers::DataObjectLineEdit Class Reference

Line edit widget to hold a pointer to a data object.

```
#include <dataobjectlineedit.h>
```

Inheritance diagram for QRS::Managers::DataObjectLineEdit:



## Signals

- void **selected** ([Core::AbstractDataObject](#) const \*pDataObject)
- void **editRequested** (Core::DataIDType id)

## Public Member Functions

- **DataObjectLineEdit** ([Core::AbstractDataObject](#) const \*pDataObject, Core::AbstractDataObject::ObjectType type, QString const &mimeType, QWidget \*parent=nullptr)

## Private Slots

- void **showContextMenu** (const QPoint &point)  
*Show a menu to modify data.*
- void **reset** ()  
*Erase the address of the data object.*
- void **edit** ()  
*Try to edit a data object through managers.*

## Private Member Functions

- void **dragEnterEvent** (QDragEnterEvent \*pEvent) override  
*Check if the type of the dropped item is correct.*
- void **dropEvent** (QDropEvent \*pEvent) override  
*Process dropping of the approved item.*
- void **keyPressEvent** (QKeyEvent \*pEvent) override  
*Erase the data object address.*
- void **mouseDoubleClickEvent** (QMouseEvent \*pEvent) override  
*Start the editing session when a double click event occurs.*

## Private Attributes

- [Core::AbstractDataObject](#) const \* **mpDataObject**
- Core::AbstractDataObject::ObjectType **mType**
- QString const **mkMimeType**

### 4.14.1 Detailed Description

Line edit widget to hold a pointer to a data object.

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

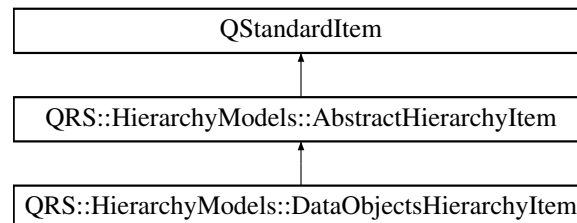
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[dataobjectlineedit.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[dataobjectlineedit.cpp](#)

## 4.15 QRS::HierarchyModels::DataObjectsHierarchyItem Class Reference

Item to represent a hierarchy of data objects.

```
#include <dataobjectshierarchyitem.h>
```

Inheritance diagram for QRS::HierarchyModels::DataObjectsHierarchyItem:



### Public Member Functions

- **DataObjectsHierarchyItem** (Core::DataObjects &dataObjects, [Core::HierarchyTree](#) &hierarchyDataObjects, QString const &text="Root", QIcon const &icon=QIcon())  
*Create the representative of the structure of data objects.*
- **DataObjectsHierarchyItem** ([Core::HierarchyNode](#) \*pNode, [Core::AbstractDataObject](#) \*pDataObject)  
*Construct an item to represent a data object.*
- **DataObjectsHierarchyItem** ([Core::HierarchyNode](#) \*pNode)  
*Construct an item to represent a directory.*
- int **type** () const override
- [Core::AbstractDataObject](#) const \* **getDataObject** () const

### Public Member Functions inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)

- **AbstractHierarchyItem** (QIcon const &icon, QString const &text, [Core::HierarchyNode](#) \*pNode)
- void **writePointer** (QDataStream &out) const  
*Write the pointer to the current item to a stream.*
- virtual int **type** () const =0

### Private Member Functions

- void **appendItems** (Core::DataObjects &dataObjects, [Core::HierarchyNode](#) \*pNode)  
*Create items based on the position in the tree structure.*

### Private Attributes

- [Core::AbstractDataObject](#) \* **mpDataObject** = nullptr

### Friends

- class **DataObjectsHierarchyModel**
- class **PropertiesModels::DataObjectsPropertiesModel**

### Additional Inherited Members

### Public Types inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)

- enum **ItemType** { **kDataObjects** = QStandardItem::UserType , **kRodComponents** }

### Static Public Member Functions inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)

- static [AbstractHierarchyItem](#) \* **readPointer** (QDataStream &in)  
*Retrieve a pointer to an item from a stream.*

### Protected Attributes inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)

- [Core::HierarchyNode](#) \* **mpNode** = nullptr

## 4.15.1 Detailed Description

Item to represent a hierarchy of data objects.

## 4.15.2 Member Function Documentation

### 4.15.2.1 type()

```
int QRS::HierarchyModels::DataObjectsHierarchyItem::type ( ) const [inline], [override],
[virtual]
```

Implements [QRS::HierarchyModels::AbstractHierarchyItem](#).

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

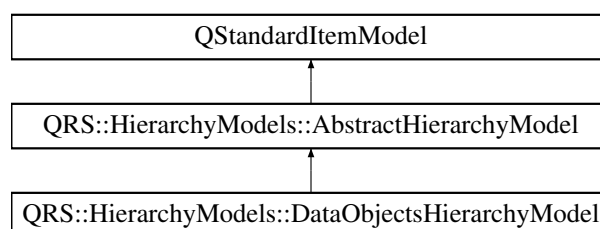
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[dataobjectshierarchyitem.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[dataobjectshierarchyitem.cpp](#)

## 4.16 QRS::HierarchyModels::DataObjectsHierarchyModel Class Reference

Tree model to represent and modify a hierarchy of data objects.

```
#include <dataobjectshierarchymodel.h>
```

Inheritance diagram for QRS::HierarchyModels::DataObjectsHierarchyModel:



## Public Slots

- void **retrieveSelectedItem** ()  
*Retrieve a selected data object.*
- void **removeSelectedItems** ()  
*Remove data objects under selection.*

## Signals

- void **selected** (Core::DataIDType id)
- void **selectionCleared** ()

## Signals inherited from [QRS::HierarchyModels::AbstractHierarchyModel](#)

- void **hierarchyChanged** ()  
*Emitted when hierarchical elements get renamed, moved or deleted.*

## Public Member Functions

- **DataObjectsHierarchyModel** (Core::DataObjects &dataObjects, [Core::HierarchyTree](#) &hierarchyData, Objects, QString const &mimeType, QTreeView \*pView=nullptr)
- void [updateContent](#) () override  
*Update all the content.*
- void [clearContent](#) () override  
*Clear all the items.*
- bool **isEmpty** () const  
*Check if there are data objects to represent.*
- void **selectItem** (int iRow)  
*Select an item by row index.*
- void **selectItemByID** (Core::DataIDType id)  
*Select an item by type and identifier.*

## Public Member Functions inherited from [QRS::HierarchyModels::AbstractHierarchyModel](#)

- **AbstractHierarchyModel** (QString const &mimeType, QTreeView \*pView=nullptr)
- virtual void [updateContent](#) ()=0
- virtual void [clearContent](#) ()=0
- Qt::DropActions **supportedDragActions** () const override  
*Specify allowed drag actions.*
- Qt::DropActions **supportedDropActions** () const override  
*Specify allowed drop actions.*
- QList **mimeTypes** () const override  
*Retrieve the mime types.*
- QMimeData \* **mimeData** (const QModelIndexList &indices) const override  
*Encode each item according to a given list of indices.*
- bool **dropMimeData** (QMimeData const \*pMimeData, Qt::DropAction action, int row, int column, const QModelIndex &parent) override  
*Process the drop action.*

### Private Slots

- void **renamelItem** (QStandardItem \*pStandardItem)  
*Rename a data object after editing.*

### Private Member Functions

- [DataObjectsHierarchyItem](#) \* **findItemByID** ([DataObjectsHierarchyItem](#) \*pltem, Core::DataIDType const &id)  
*Find an item by identifier.*
- void **selectItem** ([DataObjectsHierarchyItem](#) \*pltem)  
*Select a specified item.*

### Private Attributes

- Core::DataObjects & **mDataObjects**
- [Core::HierarchyTree](#) & **mHierarchyDataObjects**

### Additional Inherited Members

### Protected Attributes inherited from [QRS::HierarchyModels::AbstractHierarchyModel](#)

- QString const **mkMimeType**

## 4.16.1 Detailed Description

Tree model to represent and modify a hierarchy of data objects.

## 4.16.2 Member Function Documentation

### 4.16.2.1 clearContent()

```
void DataObjectsHierarchyModel::clearContent ( ) [override], [virtual]
```

Clear all the items.

Implements [QRS::HierarchyModels::AbstractHierarchyModel](#).

### 4.16.2.2 updateContent()

```
void DataObjectsHierarchyModel::updateContent ( ) [override], [virtual]
```

Update all the content.

Implements [QRS::HierarchyModels::AbstractHierarchyModel](#).

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

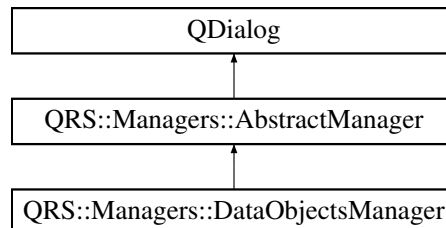
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.h>
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.cpp>

## 4.17 QRS::Managers::DataObjectsManager Class Reference

Manager to create objects of different types: scalars, vectors, matroces and surfaces.

```
#include <dataobjectsmanager.h>
```

Inheritance diagram for QRS::Managers::DataObjectsManager:



### Public Slots

- void **apply** () override  
*Apply all the changes made by user.*
- [Core::AbstractDataObject](#) \* **addScalar** ()  
*Add a scalar object.*
- [Core::AbstractDataObject](#) \* **addVector** ()  
*Add a vector object.*
- [Core::AbstractDataObject](#) \* **addMatrix** ()  
*Add a matrix object.*
- [Core::AbstractDataObject](#) \* **addSurface** ()  
*Add a surface object.*
- void **insertItemAfterSelected** ()  
*Insert a new array into the data object.*
- void **insertLeadingItemAfterSelected** ()  
*Insert a new leading item into the data object.*
- void **removeSelectedItem** ()  
*Remove a selected item.*
- void **removeSelectedLeadingItem** ()  
*Remove a selected leading item.*
- void **importDataObjects** ()  
*Import data objects from a file.*

### Public Slots inherited from [QRS::Managers::AbstractManager](#)

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

### Signals

- void **applied** ([Core::DataObjects](#) const &dataObjects, [Core::HierarchyTree](#) const &hierarchyDataObjects)



## Signals inherited from [QRS::Managers::AbstractManager](#)

- void **closed** (QRS::Managers::AbstractManager::ManagerType type)

## Public Member Functions

- **DataObjectsManager** (Core::DataObjects &&dataObjects, [Core::HierarchyTree](#) &&hierarchyDataObjects, QString &lastPath, QSettings &settings, QWidget \*parent=nullptr)
- void **selectDataObject** (int iRow)  
*Select a data object by row index.*
- void **selectDataObjectByID** (Core::DataIDType id)  
*Select a data object by identifier.*
- Core::DataObjects const & **getDataObjects** ()

## Public Member Functions inherited from [QRS::Managers::AbstractManager](#)

- **AbstractManager** (QString &lastPath, QSettings &settings, ManagerType type, QString groupName, QWidget \*parent=nullptr)
- void **saveSettings** ()  
*Save settings to a file.*
- void **restoreSettings** ()  
*Restore settings from a file.*

## Private Member Functions

- void **createContent** ()  
*Create all the widgets.*
- ads::CDockWidget \* **createDataTableWidget** ()  
*Create a tabbed widget to interact with data tables.*
- ads::CDockWidget \* **createHierarchyWidget** ()  
*Create an object to represent a hierarchy of data objects.*
- QLayout \* **createDialogControls** ()  
*Create dialog controls.*
- void **emplaceDataObject** ([Core::AbstractDataObject](#) \*pDataObject)  
*Helper function to insert data objects into the manager.*
- bool **isDataTableModifiable** ()  
*Helper function to check if it is possible to interact with data object content.*
- void **importDataObject** (QString const &path, QString const &fileName)  
*Import a data object from a file.*
- void **representDataObject** (Core::DataIDType id)  
*Represent a selected data object according to its type.*
- void **clearDataObjectRepresentation** ()  
*Clear a visual data of a data object.*

### Private Attributes

- QTreeView \* **mpTreeDataObjects**
- QTreeView \* **mpDataTable**
- Core::DataObjects **mDataObjects**
- [Core::HierarchyTree](#) **mHierarchyDataObjects**
- [TableModels::TableModelInterface](#) \* **mpTableModelInterface** = nullptr
- [TableModels::BaseTableModel](#) \* **mpBaseTableModel**
- [TableModels::MatrixTableModel](#) \* **mpMatrixTableModel**
- [TableModels::SurfaceTableModel](#) \* **mpSurfaceTableModel**
- [HierarchyModels::DataObjectsHierarchyModel](#) \* **mpTreeDataObjectsModel**

### Additional Inherited Members

### Public Types inherited from [QRS::Managers::AbstractManager](#)

- enum **ManagerType** { **kDataObjects** , **kRodComponents** , **kRodConstructor** }

### Protected Member Functions inherited from [QRS::Managers::AbstractManager](#)

- void **closeEvent** (QCloseEvent \*pEvent) override  
*Save settings and delete handling widgets before closing the window.*
- void **setToolBarShortcutHints** (QToolBar \*pToolBar)  
*Helper function to add a shortcut hint to all actions which a toolbar contains.*

### Protected Attributes inherited from [QRS::Managers::AbstractManager](#)

- ads::CDockManager \* **mpDockManager** = nullptr
- QString & **mLastPath**

#### 4.17.1 Detailed Description

Manager to create objects of different types: scalars, vectors, matroces and surfaces.

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

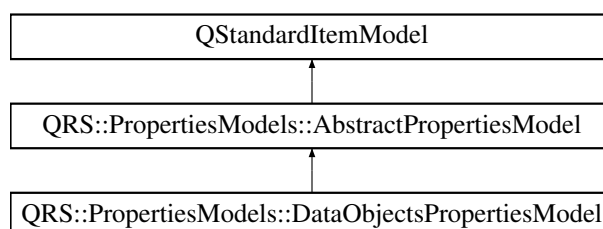
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[dataobjectsmanager.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[dataobjectsmanager.cpp](#)

## 4.18 QRS::PropertiesModels::DataObjectsPropertiesModel Class Reference

Model to represent properties of selected data objects.

```
#include <dataobjectspropertiesmodel.h>
```

Inheritance diagram for QRS::PropertiesModels::DataObjectsPropertiesModel:



## Public Member Functions

- **DataObjectsPropertiesModel** (QTableView \*pView, QVector< [HierarchyModels::AbstractHierarchyItem](#) \* > items)

## Public Member Functions inherited from [QRS::PropertiesModels::AbstractPropertiesModel](#)

- **AbstractPropertiesModel** (QTableView \*pView, QVector< [HierarchyModels::AbstractHierarchyItem](#) \* > items)

## Protected Slots

- void **modifyProperty** (QStandardItem \*pChangedProperty) override  
*Modify the selected property of all items.*

## Protected Slots inherited from [QRS::PropertiesModels::AbstractPropertiesModel](#)

- virtual void **modifyProperty** (QStandardItem \*pChangedProperty)=0
- void **modifyDirectoryName** (QString const &name)  
*Change names of selected directories.*

## Private Types

- enum **PropertyDataObject** {  
    **kName** , **kType** , **kNumberItems** , **kNumberEntities** ,  
    **kID** }

## Private Member Functions

- void **setObjectAttributes** ()  
*Set attributes of selected data objects.*

## Additional Inherited Members

## Signals inherited from [QRS::PropertiesModels::AbstractPropertiesModel](#)

- void **propertyChanged** ()

## Protected Member Functions inherited from [QRS::PropertiesModels::AbstractPropertiesModel](#)

- void **setDirectoryAttributes** ()  
*Set attributes of selected directories.*
- QList< QStandardItem \* > **preparePropertyRow** (int type, QString const &title, QVariant const &value, bool isValueEditable) const  
*Prepare a row to insert into the table.*

## Protected Attributes inherited from [QRS::PropertiesModels::AbstractPropertiesModel](#)

- QVector< [HierarchyModels::AbstractHierarchyItem](#) \* > **mItems**
- bool **mIsDirectory**
- QString const **mkEmptyProperty** = ""

### 4.18.1 Detailed Description

Model to represent properties of selected data objects.

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

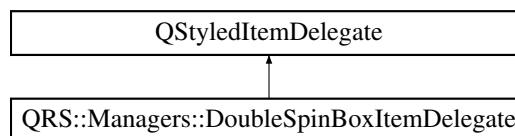
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/[dataobjectspropertiesmodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/[dataobjectspropertiesmodel.cpp](#)

## 4.19 QRS::Managers::DoubleSpinBoxItemDelegate Class Reference

Class to specify how table values can be edited.

```
#include <doublespinboxitemdelegate.h>
```

Inheritance diagram for QRS::Managers::DoubleSpinBoxItemDelegate:



### Public Member Functions

- **DoubleSpinBoxItemDelegate** (QObject \*parent=nullptr)
- QWidget \* **createEditor** (QWidget \*parent, const QStyleOptionViewItem &option, const QModelIndex &index) const override  
*Create a double value editor.*
- void **setEditorData** (QWidget \*pEditor, const QModelIndex &index) const override  
*Specify data to show.*
- void **setModelData** (QWidget \*pEditor, QAbstractItemModel \*pModel, const QModelIndex &index) const override  
*Set data to a model.*
- void **updateEditorGeometry** (QWidget \*pEditor, const QStyleOptionViewItem &option, const QModelIndex &index) const override  
*Set a geometry to render.*

### 4.19.1 Detailed Description

Class to specify how table values can be edited.

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

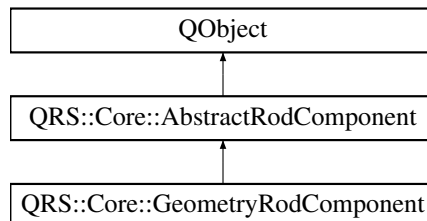
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[doublespinboxitemdelegate.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[doublespinboxitemdelegate.cpp](#)

## 4.20 QRS::Core::GeometryRodComponent Class Reference

Geometrical configuration of a rod.

```
#include <geometryrodcomponent.h>
```

Inheritance diagram for QRS::Core::GeometryRodComponent:



### Public Member Functions

- **GeometryRodComponent** (QString const &name)
- **~GeometryRodComponent** ()  
*Decrease a number of instances while being destroyed.*
- **AbstractRodComponent** \* **clone** () const override  
*Clone a geometrical rod component.*
- bool **isDataComplete** () const override  
*Check whether the component data is complete.*
- void **serialize** (QDataStream &stream) const override  
*Serialize all properties of a geometrical component.*
- void **deserialize** (QDataStream &stream, DataObjects const &dataObjects) override  
*Deserialize a geometrical component.*
- void **resolveReferences** (DataObjects const &dataObjects) override  
*Resolve references of a geometrical rod component.*
- **VectorDataObject** const \* **radiusVector** () const
- **MatrixDataObject** const \* **rotationMatrix** () const
- void **setRadiusVector** (**VectorDataObject** const \*pRadiusVector)
- void **setRotationMatrix** (**MatrixDataObject** const \*pRotationMatrix)

### Public Member Functions inherited from QRS::Core::AbstractRodComponent

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual **AbstractRodComponent** \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

### Static Public Member Functions

- static quint32 **numberInstances** ()

### Static Public Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

### Private Attributes

- QPointer< [VectorDataObject](#) const > **mpRadiusVector**
- QPointer< [MatrixDataObject](#) const > **mpRotationMatrix**

### Static Private Attributes

- static quint32 **smNumInstances** = 0

### Additional Inherited Members

### Public Types inherited from [QRS::Core::AbstractRodComponent](#)

- enum **ComponentType** {  
**kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
**kConstraint** , **kMechanical** }

### Protected Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- void **writeDataObjectPointer** (QDataStream &stream, [AbstractDataObject](#) const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- [AbstractDataObject](#) const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- [AbstractDataObject](#) const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- [AbstractDataObject](#) const \* **substituteDataObject** (DataObjects const &dataObjects, [AbstractDataObject](#) const \*pDataObject) const  
*Substitute a data object with its updated version.*

### Protected Attributes inherited from [QRS::Core::AbstractRodComponent](#)

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

## 4.20.1 Detailed Description

Geometrical configuration of a rod.

## 4.20.2 Member Function Documentation

### 4.20.2.1 clone()

```
AbstractRodComponent * GeometryRodComponent::clone ( ) const [override], [virtual]
```

Clone a geometrical rod component.

Implements [QRS::Core::AbstractRodComponent](#).

### 4.20.2.2 deserialize()

```
void GeometryRodComponent::deserialize (
    QDataStream & stream,
    DataObjects const & dataObjects ) [override], [virtual]
```

Deserialize a geometrical component.

Implements [QRS::Core::AbstractRodComponent](#).

### 4.20.2.3 isDataComplete()

```
bool GeometryRodComponent::isDataComplete ( ) const [override], [virtual]
```

Check whether the component data is complete.

Implements [QRS::Core::AbstractRodComponent](#).

### 4.20.2.4 resolveReferences()

```
void GeometryRodComponent::resolveReferences (
    DataObjects const & dataObjects ) [override], [virtual]
```

Resolve references of a geometrical rod component.

Implements [QRS::Core::AbstractRodComponent](#).

### 4.20.2.5 serialize()

```
void GeometryRodComponent::serialize (
    QDataStream & stream ) const [override], [virtual]
```

Serialize all properties of a geometrical component.

Implements [QRS::Core::AbstractRodComponent](#).

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

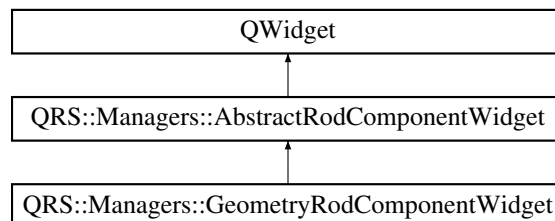
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.cpp](#)

## 4.21 QRS::Managers::GeometryRodComponentWidget Class Reference

Widget to construct a geometrical rod component.

```
#include <geometryrodcomponentwidget.h>
```

Inheritance diagram for QRS::Managers::GeometryRodComponentWidget:



### Public Member Functions

- **GeometryRodComponentWidget** ([Core::GeometryRodComponent](#) &geometryRodComponent, QString const &mimeType, QWidget \*parent=nullptr)

### Public Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- **AbstractRodComponentWidget** (QString const &mimeType, QWidget \*parent=nullptr)

### Private Member Functions

- void **createContent** ()  
*Create all the widgets.*
- template<typename T >  
void **setProperty** ([Core::AbstractDataObject](#) const \*pDataObject, auto setFun)  
*Set a property of a rod geometry.*

### Private Attributes

- [Core::GeometryRodComponent](#) & **mGeometryRodComponent**

### Additional Inherited Members

### Signals inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- void **modified** ()
- void **editDataObjectRequested** ([Core::DataIDType](#) id)



## Protected Member Functions inherited from QRS::Managers::AbstractRodComponentWidget

- void **setDataObjectEditConnections** ([DataObjectLineEdit](#) \*pEdit, DataObjectSetFun &setFun)  
*Specify connections of an editor which hold pointers to data objects of different types.*

## Protected Attributes inherited from QRS::Managers::AbstractRodComponentWidget

- QString const **mkMimeType**

### 4.21.1 Detailed Description

Widget to construct a geometrical rod component.

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

- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[geometryrodcomponentwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[geometryrodcomponentwidget.cpp](#)

## 4.22 QRS::Core::HierarchyNode Class Reference

Hierarchy representative.

```
#include <hierarchynode.h>
```

### Public Types

- enum **NodeType** { **kObject** , **kDirectory** }

### Public Member Functions

- **HierarchyNode** (NodeType type, QVariant value)  
*Node constructor.*
- void **appendChild** ([HierarchyNode](#) \*node)  
*Add a child node.*
- bool **hasParent** () const
- bool **hasChild** () const
- bool **hasNextSibling** () const
- [HierarchyNode](#) \* **parent** ()
- [HierarchyNode](#) \* **firstChild** ()
- [HierarchyNode](#) \* **nextSibling** ()
- NodeType **type** () const
- QVariant & **value** ()
- [HierarchyNode](#) \* **groupNodes** ([HierarchyNode](#) \*pChildNode)  
*Merge two nodes into one entity.*
- bool **setBefore** ([HierarchyNode](#) \*pSetNode)  
*Set a given node before the current one.*
- bool **setAfter** ([HierarchyNode](#) \*pSetNode)  
*Set a given node after the current one.*
- quint32 **numberChildren** () const  
*Retrieve a number of children of the current node.*

### Private Member Functions

- void **excludeNodeFromHierarchy** ()  
*Remove all links to the node.*
- bool **isSetAllowed** ([HierarchyNode](#) const \*pNode) const  
*Check whether it is possible to place a given item before or after the current one.*
- bool **isParentOf** ([HierarchyNode](#) const \*pNode) const  
*Check whether the current item contains a given node as a child.*
- quint32 **countNodes** ([HierarchyNode](#) \*pNode, quint32 &numNodes) const  
*Count all children and siblings of a given node.*

### Private Attributes

- [HierarchyNode](#) \* **mpParent** = nullptr
- [HierarchyNode](#) \* **mpFirstChild** = nullptr
- [HierarchyNode](#) \* **mpNextSibling** = nullptr
- [HierarchyNode](#) \* **mpPreviousSibling** = nullptr
- NodeType **mType**
- QVariant **mValue**

### Friends

- class **HierarchyTree**

## 4.22.1 Detailed Description

Hierarchy representative.

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

- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[hierarchynode.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[hierarchynode.cpp](#)

## 4.23 QRS::Core::HierarchyTree Class Reference

Hierarchy of data objects (n-array tree)

```
#include <hierarchytree.h>
```

## Public Member Functions

- **HierarchyTree** ()  
*Base tree constructor.*
- **HierarchyTree** ([HierarchyTree](#) &another)  
*Copy constructor.*
- **HierarchyTree** ([HierarchyTree](#) &&another)  
*Move constructor.*
- **HierarchyTree** ([HierarchyNode](#) \*pRootNode)  
*Take the user defined node as the root.*
- **HierarchyTree** (QDataStream &stream, int numNodes)  
*Read a tree from a stream.*
- [HierarchyTree](#) & **operator=** ([HierarchyTree](#) const &another)  
*Copy assignment operator.*
- [HierarchyTree](#) & **operator=** ([HierarchyTree](#) &&another)  
*Move assignment operator.*
- **~HierarchyTree** ()  
*Tree destructor.*
- void **clear** ()  
*Delete all nodes except the root node.*
- void **appendNode** ([HierarchyNode](#) \*pNode)  
*Append a node to the root node.*
- bool **removeNode** (HierarchyNode::NodeType type, QVariant const &value)  
*Remove a node by type and value.*
- void **removeNode** ([HierarchyNode](#) \*pNode)  
*Remove a node and all its subnodes.*
- void **changeNodeValue** (HierarchyNode::NodeType type, QVariant const &oldValue, QVariant const &newValue)  
*Change the value of a node.*
- [HierarchyNode](#) \* **root** ()
- [HierarchyTree](#) **clone** () const  
*Clone a tree.*
- [HierarchyNode](#) \* **findNode** ([HierarchyNode](#) \*pBaseNode, HierarchyNode::NodeType type, QVariant const &value) const  
*Find a node by type and value.*
- quint32 **size** () const  
*Get a number of nodes.*

## Private Member Functions

- [HierarchyNode](#) \* **copyNode** ([HierarchyNode](#) \*pBaseNode, quint32 relativeLevel) const  
*Copy a node.*
- void **removeNodeSiblings** ([HierarchyNode](#) \*pNode)  
*Remove all subnodes.*
- void **printNode** (quint32 level, [HierarchyNode](#) \*pNode, QDebug stream) const  
*Print a current node and all its subnodes.*
- void **writeNode** ([HierarchyNode](#) \*pNode, QDataStream &stream) const  
*Print a current node and all its subnodes.*

### Private Attributes

- [HierarchyNode](#) \* **mpRootNode** = nullptr

### Friends

- QDebug **operator**<< (QDebug stream, [HierarchyTree](#) &tree)  
*Print a tree structure.*
- QDataStream & **operator**<< (QDataStream &stream, [HierarchyTree](#) const &tree)  
*Write a tree structure to a stream.*

## 4.23.1 Detailed Description

Hierarchy of data objects (n-array tree)

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

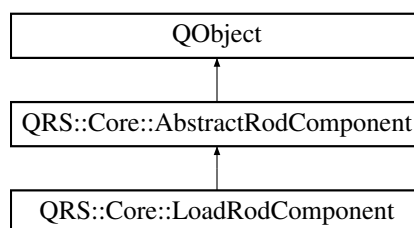
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[hierarchytree.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[hierarchytree.cpp](#)

## 4.24 QRS::Core::LoadRodComponent Class Reference

Load applied to a rod.

```
#include <loadrodcomponent.h>
```

Inheritance diagram for QRS::Core::LoadRodComponent:



### Public Types

- enum **LoadType** {  
**kNone** , **kForcedDisplacements** , **kForcedRotations** , **kPointForce** ,  
**kPointMoment** , **kPointMass** , **kPointInertiaMoment** , **kPointLinearDamper** ,  
**kPointRotationalDamper** , **kDistributedForce** , **kDistributedMoment** , **kAerodynamicFlow** ,  
**kAcceleration** , **kInnerLiquidFlow** , **kDisplacementDamping** , **kRotationDamping** }

### Public Types inherited from [QRS::Core::AbstractRodComponent](#)

- enum **ComponentType** {  
**kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
**kConstraint** , **kMechanical** }

## Public Member Functions

- **LoadRodComponent** (QString const &name)
- **~LoadRodComponent** ()  
*Decrease a number of instances while being destroyed.*
- **AbstractRodComponent** \* **clone** () const override  
*Clone a rod load.*
- bool **isDataComplete** () const override  
*Check whether the component data is complete.*
- void **serialize** (QDataStream &stream) const override  
*Serialize all properties of a rod load.*
- void **deserialize** (QDataStream &stream, DataObjects const &dataObjects) override  
*Deserialize a rod load.*
- void **resolveReferences** (DataObjects const &dataObjects) override  
*Resolve references of a rod load.*
- LoadType **loadType** () const
- **VectorDataObject** const \* **directionVector** () const
- **ScalarDataObject** const \* **longitudinalFunction** () const
- **ScalarDataObject** const \* **timeCoefficient** () const
- **VectorDataObject** const \* **timeRotationVector** () const
- DataValueType **multiplier** () const
- bool **isFollowing** () const
- void **setType** (LoadType type)
- void **setDirectionVector** (**VectorDataObject** const \*pDirectionVector)
- void **setLongitudinalFunction** (**ScalarDataObject** const \*pLongitudinalFunction)
- void **setTimeCoefficient** (**ScalarDataObject** const \*pTimeCoefficient)
- void **setTimeRotationVector** (**VectorDataObject** const \*pTimeRotationVector)
- void **setMultiplier** (DataValueType value)
- void **setFollowingState** (bool isFollowing)

## Public Member Functions inherited from QRS::Core::AbstractRodComponent

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual **AbstractRodComponent** \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Static Public Member Functions inherited from QRS::Core::AbstractRodComponent

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

### Private Attributes

- LoadType **mLoadType** = kNone
- QPointer< [VectorDataObject](#) const > **mpDirectionVector**
- QPointer< [ScalarDataObject](#) const > **mpLongitudinalFunction**
- QPointer< [ScalarDataObject](#) const > **mpTimeCoefficient**
- QPointer< [VectorDataObject](#) const > **mpTimeRotationVector**
- DataValueType **mMultiplier** = 1.0
- bool **mlsFollowing** = false

### Static Private Attributes

- static quint32 **smNumInstances** = 0

### Additional Inherited Members

### Protected Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- void **writeDataObjectPointer** (QDataStream &stream, [AbstractDataObject](#) const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- [AbstractDataObject](#) const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- [AbstractDataObject](#) const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- [AbstractDataObject](#) const \* **substituteDataObject** (DataObjects const &dataObjects, [AbstractDataObject](#) const \*pDataObject) const  
*Substitute a data object with its updated version.*

### Protected Attributes inherited from [QRS::Core::AbstractRodComponent](#)

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

## 4.24.1 Detailed Description

Load applied to a rod.

## 4.24.2 Member Function Documentation

### 4.24.2.1 clone()

```
AbstractRodComponent * LoadRodComponent::clone ( ) const [override], [virtual]
```

Clone a rod load.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.24.2.2 deserialize()

```
void LoadRodComponent::deserialize (
    QDataStream & stream,
    DataObjects const & dataObjects ) [override], [virtual]
```

Deserialize a rod load.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.24.2.3 isDataComplete()

```
bool LoadRodComponent::isDataComplete ( ) const [override], [virtual]
```

Check whether the component data is complete.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.24.2.4 resolveReferences()

```
void LoadRodComponent::resolveReferences (
    DataObjects const & dataObjects ) [override], [virtual]
```

Resolve references of a rod load.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.24.2.5 serialize()

```
void LoadRodComponent::serialize (
    QDataStream & stream ) const [override], [virtual]
```

Serialize all properties of a rod load.

Implements [QRS::Core::AbstractRodComponent](#).

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

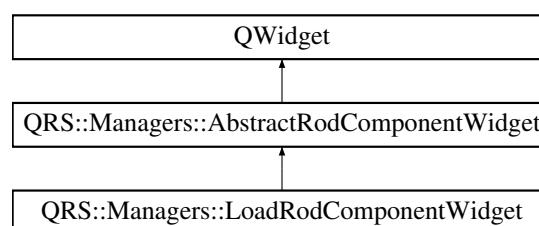
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.cpp](#)

## 4.25 QRS::Managers::LoadRodComponentWidget Class Reference

Widget to construct a load applied to a rod.

```
#include <loadrodcomponentwidget.h>
```

Inheritance diagram for QRS::Managers::LoadRodComponentWidget:



## Public Member Functions

- **LoadRodComponentWidget** ([Core::LoadRodComponent](#) &loadRodComponent, QString const &mimeType, QWidget \*parent=nullptr)

## Public Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- **AbstractRodComponentWidget** (QString const &mimeType, QWidget \*parent=nullptr)

## Private Member Functions

- void **createContent** ()  
*Create all the widgets.*
- QLayout \* **createBaseLayout** ()  
*Create a layout consisted of widgets to set loading parameters.*
- QWidget \* **createTimeGroup** ()  
*Create a group of widgets which depend on time.*
- QLayout \* **createLoadTypeLayout** ()  
*Create a layout consisted of widgets to set a load type and following state.*
- QComboBox \* **createLoadTypeComboBox** ()  
*Create a combobox to specify a type of load.*
- template<typename T >  
void **setProperty** ([Core::AbstractDataObject](#) const \*pDataObject, auto setFun)  
*Set a property of a rod load.*
- void **setLoadUnits** ([Core::LoadRodComponent::LoadType](#) type)  
*Set load units to show.*

## Private Attributes

- [Core::LoadRodComponent](#) & **mLoadRodComponent**
- QLabel \* **mpLoadRodUnits**

## Additional Inherited Members

## Signals inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- void **modified** ()
- void **editDataObjectRequested** ([Core::DataIDType](#) id)

## Protected Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- void **setDataObjectEditConnections** ([DataObjectLineEdit](#) \*pEdit, DataObjectSetFun &setFun)  
*Specify connections of an editor which hold pointers to data objects of different types.*



## Protected Attributes inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- QString const **mkMimeType**

### 4.25.1 Detailed Description

Widget to construct a load applied to a rod.

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

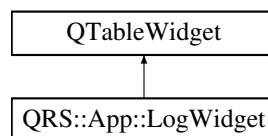
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.h>
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.cpp>

## 4.26 QRS::App::LogWidget Class Reference

Log all the messages sent.

```
#include <logwidget.h>
```

Inheritance diagram for QRS::App::LogWidget:



### Public Member Functions

- **LogWidget** (QWidget \*parent=nullptr)
- void **log** (QtMsgType messageType, const QString &message)  
*Represent a message sent.*

### 4.26.1 Detailed Description

Log all the messages sent.

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

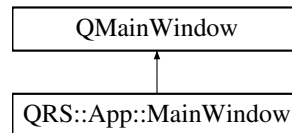
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/central/logwidget.h>
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/central/logwidget.cpp>

## 4.27 QRS::App::MainWindow Class Reference

The main window of the program.

```
#include <mainwindow.h>
```

Inheritance diagram for QRS::App::MainWindow:



### Public Member Functions

- **MainWindow** (QWidget \*parent=nullptr)
- void **openProject** (QString const &filePath)  
*Open the specific project.*
- bool **saveProject** ()  
*Save the current project.*

### Static Public Attributes

- static [LogWidget](#) \* **pLogger** = nullptr

### Private Slots

- void **createProject** ()  
*Create a project and substitute the current one with it.*
- void **openProjectDialog** ()  
*Open a project by using a dialog.*
- void **openRecentProject** ()  
*Open the project which was selected from the Recent Projects menu.*
- bool **saveAsProject** ()  
*Save the current project under a new name.*
- void **setModified** (bool flag)  
*Whenever a project has been modified.*
- void **representHierarchyProperties** (QVector< [HierarchyModels::AbstractHierarchyItem](#) \* > items)  
*Show information about the selected project items.*
- void **saveSettings** ()  
*Save the current window settings.*
- void **restoreSettings** ()  
*Restore window settings from a file.*
- void **createDataObjectsManager** ()  
*Show a manager for designing data objects.*
- void **createRodComponentsManager** ()  
*Show a manager to set rod components based on the created data objects.*
- void **createRodConstructorManager** ()  
*Show a manager to assemble a rod by using rod components.*
- void **aboutProgram** ()  
*Show information about a program.*

### Private Member Functions

- void **initializeWindow** ()  
*Set a state and geometry of [MainWindow](#).*
- void **createContent** ()  
*Create all the widgets and corresponding actions.*
- void **closeEvent** (QCloseEvent \*pEvent) override  
*Save project and settings before exit.*
- ads::CDockWidget \* **createProjectHierarchyWidget** ()  
*Create a widget to represent a project hierarchy.*
- ads::CDockWidget \* **createGLWidget** ()  
*Create an OpenGL widget to render rods.*
- ads::CDockWidget \* **createCodeWidget** ()  
*Create a widget enables to code.*
- ads::CDockWidget \* **createLogWidget** ()  
*Create a window for logging.*
- ads::CDockWidget \* **createPropertiesWidget** ()  
*Create a window to modify properies of selected obercts.*
- void **setProjectTitle** ()  
*Show information a name of a project.*
- void **retrieveRecentProjects** ()  
*Retrieve recent projects from the settings file.*
- void **addToRecentProjects** ()  
*Add the current project to the recent ones.*
- void **specifyMenuConnections** ()  
*Set signals and slots for menu actions.*
- void **specifyProjectConnections** ()  
*Set signals and slots for a project.*
- bool **saveProjectChangesDialog** ()  
*Save project changes.*
- bool **saveProjectHelper** (QString const &filePath)  
*Helper method to perform saving of the current project.*

### Private Attributes

- Ui::MainWindow \* **mpUi**
- ads::CDockManager \* **mpDockManager**
- QLabel \* **mpStatusLabel**
- QTableView \* **mpPropertiesWidget**
- [HierarchyModels::ProjectHierarchyModel](#) \* **mpProjectHierarchyModel** = nullptr
- [Managers::ManagersFactory](#) \* **mpManagersFactory** = nullptr
- [Core::Project](#) \* **mpProject**
- QSharedPointer< QSettings > **mpSettings**
- QString **mLastPath**
- QList< QString > **mPathRecentProjects**

## 4.27.1 Detailed Description

The main window of the program.

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

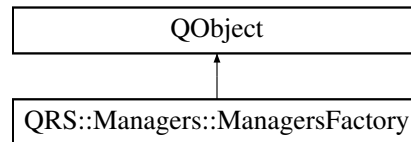
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[mainwindow.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[mainwindow.cpp](#)

## 4.28 QRS::Managers::ManagersFactory Class Reference

Factory to create managers which utilize and modify project data.

```
#include <managersfactory.h>
```

Inheritance diagram for QRS::Managers::ManagersFactory:



### Public Member Functions

- **ManagersFactory** ([Core::Project](#) &project, QString &lastPath, QSettings &settings, QWidget \*parent)
- bool **createManager** (AbstractManager::ManagerType type)  
*Create a manager according to a given type.*
- bool **deleteManager** (AbstractManager::ManagerType type)  
*Destroy a manager by given type.*
- [AbstractManager](#) \* **manager** (AbstractManager::ManagerType type)  
*Retrieve a manager of a given type.*

### Private Member Functions

- void **specifyConnections** ([DataObjectsManager](#) \*pManager)  
*Specify connections of the manager of data objects.*
- void **specifyConnections** ([RodComponentsManager](#) \*pManager)  
*Specify connections of the manager of rod components.*

### Private Attributes

- [Core::Project](#) & **mProject**
- QString & **mLastPath**
- QSettings & **mSettings**
- QWidget \* **mpParent**
- std::unordered\_map< AbstractManager::ManagerType, [AbstractManager](#) \* > **mManagers**

### 4.28.1 Detailed Description

Factory to create managers which utilize and modify project data.

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

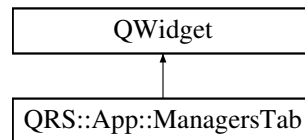
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[managersfactory.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[managersfactory.cpp](#)

## 4.29 QRS::App::ManagersTab Class Reference

A toolbar consisted of object designers.

```
#include <controltabs.h>
```

Inheritance diagram for QRS::App::ManagersTab:



### Signals

- void **actionDataObjectsTriggered** ()
- void **actionRodPropertiesTriggered** ()
- void **actionRodConstructorTriggered** ()

### Public Member Functions

- **ManagersTab** (QWidget \*parent=nullptr)  
*Managers tab constructor.*

### 4.29.1 Detailed Description

A toolbar consisted of object designers.

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

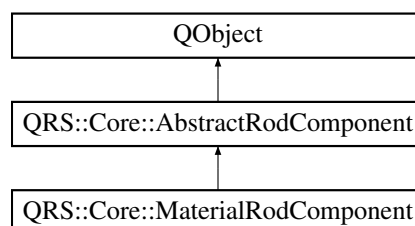
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[controltabs.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[controltabs.cpp](#)

## 4.30 QRS::Core::MaterialRodComponent Class Reference

Material properties of a rod.

```
#include <materialrodcomponent.h>
```

Inheritance diagram for QRS::Core::MaterialRodComponent:



## Public Member Functions

- **MaterialRodComponent** (QString const &name)
- **~MaterialRodComponent** ()  
*Decrease a number of instances while being destroyed.*
- **AbstractRodComponent** \* **clone** () const override  
*Clone a material rod component.*
- bool **isDataComplete** () const override  
*Check whether the component data is complete.*
- void **serialize** (QDataStream &stream) const override  
*Serialize all properties of a material component.*
- void **deserialize** (QDataStream &stream, DataObjects const &dataObjects) override  
*Deserialize a material component.*
- void **resolveReferences** (DataObjects const &dataObjects) override  
*Resolve references of a material rod component.*
- **ScalarDataObject** const \* **elasticModulus** () const
- **ScalarDataObject** const \* **shearModulus** () const
- **ScalarDataObject** const \* **poissonsRatio** () const
- **ScalarDataObject** const \* **density** () const
- void **setElasticModulus** (**ScalarDataObject** const \*pElasticModulus)
- void **setShearModulus** (**ScalarDataObject** const \*pShearModulus)
- void **setPoissonsRatio** (**ScalarDataObject** const \*pPoissonsRatio)
- void **setDensity** (**ScalarDataObject** const \*pDensity)

## Public Member Functions inherited from **QRS::Core::AbstractRodComponent**

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual **AbstractRodComponent** \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Static Public Member Functions inherited from **QRS::Core::AbstractRodComponent**

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

## Private Attributes

- QPointer< **ScalarDataObject** const > **mpElasticModulus**
- QPointer< **ScalarDataObject** const > **mpShearModulus**
- QPointer< **ScalarDataObject** const > **mpPoissonsRatio**
- QPointer< **ScalarDataObject** const > **mpDensity**

### Static Private Attributes

- static quint32 **smNumInstances** = 0

### Additional Inherited Members

### Public Types inherited from [QRS::Core::AbstractRodComponent](#)

- enum **ComponentType** {  
    **kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
    **kConstraint** , **kMechanical** }

### Protected Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- void **writeDataObjectPointer** (QDataStream &stream, [AbstractDataObject](#) const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- [AbstractDataObject](#) const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- [AbstractDataObject](#) const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- [AbstractDataObject](#) const \* **substituteDataObject** (DataObjects const &dataObjects, [AbstractDataObject](#) const \*pDataObject) const  
*Substitute a data object with its updated version.*

### Protected Attributes inherited from [QRS::Core::AbstractRodComponent](#)

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

## 4.30.1 Detailed Description

Material properties of a rod.

## 4.30.2 Member Function Documentation

### 4.30.2.1 clone()

```
AbstractRodComponent * MaterialRodComponent::clone ( ) const [override], [virtual]
```

Clone a material rod component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.30.2.2 deserialize()

```
void MaterialRodComponent::deserialize (
    QDataStream & stream,
    DataObjects const & dataObjects ) [override], [virtual]
```

Deserialize a material component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.30.2.3 isDataComplete()

```
bool MaterialRodComponent::isDataComplete ( ) const [override], [virtual]
```

Check whether the component data is complete.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.30.2.4 resolveReferences()

```
void MaterialRodComponent::resolveReferences (
    DataObjects const & dataObjects ) [override], [virtual]
```

Resolve references of a material rod component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.30.2.5 serialize()

```
void MaterialRodComponent::serialize (
    QDataStream & stream ) const [override], [virtual]
```

Serialize all properties of a material component.

Implements [QRS::Core::AbstractRodComponent](#).

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

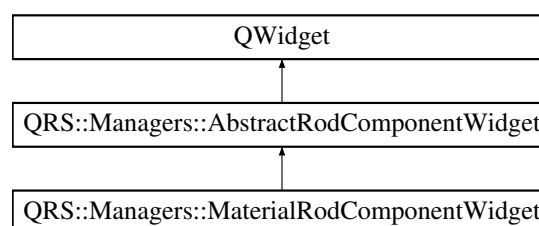
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.cpp](#)

## 4.31 QRS::Managers::MaterialRodComponentWidget Class Reference

Widget to construct a material rod component.

```
#include <materialrodcomponentwidget.h>
```

Inheritance diagram for QRS::Managers::MaterialRodComponentWidget:





## Public Member Functions

- **MaterialRodComponentWidget** ([Core::MaterialRodComponent](#) &materialRodComponent, QString const &mimeType, QWidget \*parent=nullptr)

## Public Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- **AbstractRodComponentWidget** (QString const &mimeType, QWidget \*parent=nullptr)

## Private Member Functions

- void **createContent** ()  
*Create all the widgets.*
- QWidget \* **createModuliGroup** ()  
*Create a group consisted of widgets to set physical moduli.*
- QLayout \* **createBaseLayout** ()  
*Create a layout consisted of widgets to set density and Poisson's ratio.*
- void **setProperty** ([Core::AbstractDataObject](#) const \*pDataObject, auto setFun)  
*Set a material property which takes a scalar data object.*

## Private Attributes

- [Core::MaterialRodComponent](#) & **mMaterialRodComponent**

## Additional Inherited Members

## Signals inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- void **modified** ()
- void **editDataObjectRequested** ([Core::DataIDType](#) id)

## Protected Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- void **setDataObjectEditConnections** ([DataObjectLineEdit](#) \*pEdit, DataObjectSetFun &setFun)  
*Specify connections of an editor which hold pointers to data objects of different types.*

## Protected Attributes inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- QString const **mkMimeType**

### 4.31.1 Detailed Description

Widget to construct a material rod component.

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

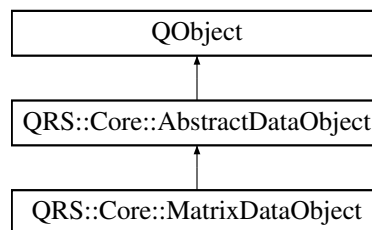
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/materialrodcomponentwidget.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/materialrodcomponentwidget.cpp](#)

## 4.32 QRS::Core::MatrixDataObject Class Reference

Matrix data object.

```
#include <matrixdataobject.h>
```

Inheritance diagram for QRS::Core::MatrixDataObject:



### Public Member Functions

- **MatrixDataObject** (QString const &name)  
*Construct a matrix data object.*
- **~MatrixDataObject** ()  
*Decrease a number of instances while being destroyed.*
- **AbstractDataObject \* clone** () const override  
*Clone a matrix data object.*
- **DataItemType & addItem** (DataValueType key) override  
*Insert a new item into [MatrixDataObject](#).*
- virtual void **import** (QTextStream &stream) override  
*Import a matrix data object from a file.*

### Public Member Functions inherited from [QRS::Core::AbstractDataObject](#)

- **AbstractDataObject** (ObjectType type, QString const &name)  
*Base constructor.*
- virtual **AbstractDataObject \* clone** () const =0
- virtual **DataItemType & addItem** (DataKeyType key)=0
- void **removeItem** (DataValueType key)  
*Remove an entity with the specified key.*
- bool **changeItemKey** (DataKeyType oldKey, DataKeyType newKey, DataHolder \*items=nullptr)  
*Modify a key existed.*

- DataValueType [getAvailableItemKey](#) (DataValueType key, DataHolder const \*items=nullptr) const
- bool **setArrayValue** (DataKeyType key, DataValueType newValue, IndexType iRow=0, IndexType iColumn=0)  
*Set an array value with the specified indices.*
- quint32 **numberOfItems** () const
- DataHolder const & **getItems** ()
- DataIDType **id** () const
- ObjectType **type** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void [serialize](#) (QDataStream &stream) const  
*Serialize an abstract data object.*
- virtual void [deserialize](#) (QDataStream &stream)  
*Partly deserialize an abstract data object.*
- virtual void [import](#) (QTextStream &stream)=0

#### Static Public Member Functions

- static quint32 **numberOfInstances** ()

#### Static Public Member Functions inherited from [QRS::Core::AbstractDataObject](#)

- static DataIDType **maxObjectID** ()
- static void **setMaxObjectID** (DataIDType iMaxObjectID)

#### Static Private Attributes

- static quint32 **smNumInstances** = 0

#### Additional Inherited Members

#### Public Types inherited from [QRS::Core::AbstractDataObject](#)

- enum **ObjectType** { kScalar , kVector , kMatrix , kSurface }

#### Protected Attributes inherited from [QRS::Core::AbstractDataObject](#)

- const ObjectType **mkType**
- QString **mName**
- DataIDType **mID**
- DataHolder **mItems**

### 4.32.1 Detailed Description

Matrix data object.

## 4.32.2 Member Function Documentation

### 4.32.2.1 addItem()

```
DataItemType & MatrixDataObject::addItem (
    DataValueType key ) [override], [virtual]
```

Insert a new item into [MatrixDataObject](#).

Implements [QRS::Core::AbstractDataObject](#).

### 4.32.2.2 clone()

```
AbstractDataObject * MatrixDataObject::clone ( ) const [override], [virtual]
```

Clone a matrix data object.

Implements [QRS::Core::AbstractDataObject](#).

### 4.32.2.3 import()

```
void MatrixDataObject::import (
    QTextStream & stream ) [override], [virtual]
```

Import a matrix data object from a file.

Implements [QRS::Core::AbstractDataObject](#).

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

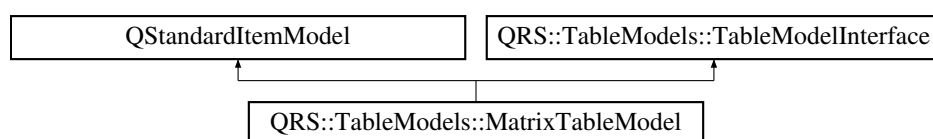
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.cpp](#)

## 4.33 QRS::TableModels::MatrixTableModel Class Reference

Table model to represent a matrix data object.

```
#include <matrixtablemodel.h>
```

Inheritance diagram for QRS::TableModels::MatrixTableModel:



## Public Member Functions

- **MatrixTableModel** (QWidget \*parent=nullptr)
- void **setDataObject** ([Core::AbstractDataObject](#) \*pDataObject)  
*Set a data object to represent.*
- bool **setData** (const QModelIndex &indexEdit, const QVariant &value, int role=Qt::EditRole) override  
*Set the data acquired from a delegate.*
- void **insertItemAfterSelected** (QItemSelectionModel \*pSelectionModel) override  
*Insert a new item after selected one.*
- void **insertLeadingItemAfterSelected** (QItemSelectionModel \*) override
- void **removeSelectedItem** (QItemSelectionModel \*pSelectionModel) override  
*Remove an array under selection.*
- void **removeSelectedLeadingItem** (QItemSelectionModel \*) override
  
- virtual void **insertItemAfterSelected** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **insertLeadingItemAfterSelected** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **removeSelectedItem** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **removeSelectedLeadingItem** (QItemSelectionModel \*pSelectionModel)=0

## Private Member Functions

- void **updateContent** ()  
*Represent all items which a vector data object contains.*
- void **clearContent** ()  
*Clear previously created items.*

## Private Attributes

- [Core::AbstractDataObject](#) \* **mpDataObject** = nullptr

## Additional Inherited Members

## Static Public Member Functions inherited from [QRS::TableModels::TableModelInterface](#)

- static QStandardItem \* **makeDoubleItem** (double value)  
*Helper function to make an item which holds a double value.*
- static QList< QStandardItem \* > **prepareRow** ([Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array.*
- static QList< QStandardItem \* > **prepareRow** (double const &key, [Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with an key.*
- static QList< QStandardItem \* > **prepareRow** (QString const &name, [Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with a name.*
- static QStandardItem \* **makeLabelItem** (QString const &name)  
*Helper function to create an item which holds a string and cannot be modified.*

### 4.33.1 Detailed Description

Table model to represent a matrix data object.

### 4.33.2 Member Function Documentation

#### 4.33.2.1 insertItemAfterSelected()

```
void MatrixTableModel::insertItemAfterSelected (
    QTableWidgetItem * pSelectionModel ) [override], [virtual]
```

Insert a new item after selected one.

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.33.2.2 insertLeadingItemAfterSelected()

```
void QRS::TableModels::MatrixTableModel::insertLeadingItemAfterSelected (
    QTableWidgetItem * ) [inline], [override], [virtual]
```

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.33.2.3 removeSelectedItem()

```
void MatrixTableModel::removeSelectedItem (
    QTableWidgetItem * pSelectionModel ) [override], [virtual]
```

Remove an array under selection.

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.33.2.4 removeSelectedLeadingItem()

```
void QRS::TableModels::MatrixTableModel::removeSelectedLeadingItem (
    QTableWidgetItem * ) [inline], [override], [virtual]
```

Implements [QRS::TableModels::TableModelInterface](#).

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

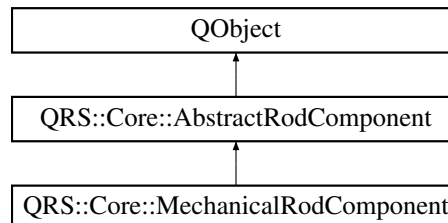
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[matrixtablemodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[matrixtablemodel.cpp](#)

## 4.34 QRS::Core::MechanicalRodComponent Class Reference

Stiffness and mass distributions of a rod.

```
#include <mechanicalrodcomponent.h>
```

Inheritance diagram for QRS::Core::MechanicalRodComponent:



### Public Member Functions

- **MechanicalRodComponent** (QString const &name)
- **~MechanicalRodComponent** ()  
*Decrease a number of instances while being destroyed.*
- **AbstractRodComponent** \* **clone** () const override  
*Clone a geometrical rod component.*
- bool **isDataComplete** () const override
- void **serialize** (QDataStream &stream) const override  
*Serialize all properties of a geometrical component.*
- void **deserialize** (QDataStream &stream, DataObjects const &dataObjects) override  
*Deserialize a geometrical component.*
- void **resolveReferences** (DataObjects const &dataObjects) override  
*Resolve references of a geometrical rod component.*
- **ScalarDataObject** const \* **tensionStiffness** () const
- **ScalarDataObject** const \* **torsionalStiffness** () const
- **ScalarDataObject** const \* **bendingStiffnessX** () const
- **ScalarDataObject** const \* **bendingStiffnessY** () const
- **ScalarDataObject** const \* **linearMassDensity** () const
- **ScalarDataObject** const \* **inertiaMassMomentX** () const
- **ScalarDataObject** const \* **inertiaMassMomentY** () const
- **ScalarDataObject** const \* **inertiaMassMomentZ** () const
- **ScalarDataObject** const \* **eccentricityX** () const
- **ScalarDataObject** const \* **eccentricityY** () const
- **ScalarDataObject** const \* **contactDiameter** () const
- void **setTensionStiffness** (**ScalarDataObject** const \*pTensionStiffness)
- void **setTorsionalStiffness** (**ScalarDataObject** const \*pTorsionalStiffness)
- void **setBendingStiffnessX** (**ScalarDataObject** const \*pBendingStiffnessX)
- void **setBendingStiffnessY** (**ScalarDataObject** const \*pBendingStiffnessY)
- void **setLinearMassDensity** (**ScalarDataObject** const \*pLinearMassDensity)
- void **setInertiaMassMomentX** (**ScalarDataObject** const \*pInertiaMassMomentX)
- void **setInertiaMassMomentY** (**ScalarDataObject** const \*pInertiaMassMomentY)
- void **setInertiaMassMomentZ** (**ScalarDataObject** const \*pInertiaMassMomentZ)
- void **setEccentricityX** (**ScalarDataObject** const \*pEccentricityX)
- void **setEccentricityY** (**ScalarDataObject** const \*pEccentricityY)
- void **setContactDiameter** (**ScalarDataObject** const \*pContactDiameter)

## Public Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual [AbstractRodComponent](#) \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Static Public Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

## Private Attributes

- QPointer< [ScalarDataObject](#) const > **mpTensionStiffness**
- QPointer< [ScalarDataObject](#) const > **mpTorsionalStiffness**
- QPointer< [ScalarDataObject](#) const > **mpBendingStiffnessX**
- QPointer< [ScalarDataObject](#) const > **mpBendingStiffnessY**
- QPointer< [ScalarDataObject](#) const > **mpLinearMassDensity**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMassMomentX**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMassMomentY**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMassMomentZ**
- QPointer< [ScalarDataObject](#) const > **mpEccentricityX**
- QPointer< [ScalarDataObject](#) const > **mpEccentricityY**
- QPointer< [ScalarDataObject](#) const > **mpContactDiameter**

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

## Public Types inherited from [QRS::Core::AbstractRodComponent](#)

- enum **ComponentType** {  
**kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
**kConstraint** , **kMechanical** }



## Protected Member Functions inherited from QRS::Core::AbstractRodComponent

- void **writeDataObjectPointer** (QDataStream &stream, [AbstractDataObject](#) const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- [AbstractDataObject](#) const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- [AbstractDataObject](#) const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- [AbstractDataObject](#) const \* **substituteDataObject** (DataObjects const &dataObjects, [AbstractDataObject](#) const \*pDataObject) const  
*Substitute a data object with its updated version.*

## Protected Attributes inherited from QRS::Core::AbstractRodComponent

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

### 4.34.1 Detailed Description

Stiffness and mass distributions of a rod.

### 4.34.2 Member Function Documentation

#### 4.34.2.1 clone()

```
AbstractRodComponent * MechanicalRodComponent::clone ( ) const [override], [virtual]
```

Clone a geometrical rod component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.34.2.2 deserialize()

```
void MechanicalRodComponent::deserialize (
    QDataStream & stream,
    DataObjects const & dataObjects ) [override], [virtual]
```

Deserialize a geometrical component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.34.2.3 isDataComplete()

```
bool QRS::Core::MechanicalRodComponent::isDataComplete ( ) const [inline], [override], [virtual]
```

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.34.2.4 resolveReferences()

```
void MechanicalRodComponent::resolveReferences (
    DataObjects const & dataObjects ) [override], [virtual]
```

Resolve references of a geometrical rod component.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.34.2.5 serialize()

```
void MechanicalRodComponent::serialize (
    QDataStream & stream ) const [override], [virtual]
```

Serialize all properties of a geometrical component.

Implements [QRS::Core::AbstractRodComponent](#).

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

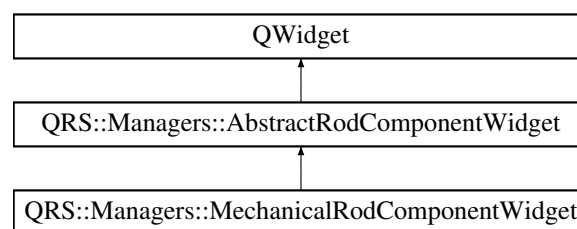
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[mechanicalrodcomponent.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[mechanicalrodcomponent.cpp](#)

### 4.35 QRS::Managers::MechanicalRodComponentWidget Class Reference

Widget to construct mechanical rod components consisted of stiffness and mass distributions.

```
#include <mechanicalrodcomponentwidget.h>
```

Inheritance diagram for QRS::Managers::MechanicalRodComponentWidget:



#### Public Member Functions

- **MechanicalRodComponentWidget** ([Core::MechanicalRodComponent](#) &mechanicalRodComponent, QString const &mimeType, QWidget \*parent=nullptr)

#### Public Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- **AbstractRodComponentWidget** (QString const &mimeType, QWidget \*parent=nullptr)

### Private Member Functions

- void **createContent** ()  
*Create all the widgets.*
- QWidget \* **createStiffnessGroup** ()  
*Create a group consisted of widgets to set stiffness distributions.*
- QWidget \* **createMassGroup** ()  
*Create a group consisted of widgets to set mass distributions.*
- QWidget \* **createEccentricityGroup** ()  
*Create a group consisted of widgets to set eccentricity distributions.*
- QLayout \* **createContactDiameterLayout** ()  
*Create a layout to set a contact diameter.*
- void **setProperty** (Core::AbstractDataObject const \*pDataObject, auto setFun)  
*Set a mechanical property which takes a scalar data object.*

### Private Attributes

- Core::MechanicalRodComponent & **mMechanicalRodComponent**

### Additional Inherited Members

### Signals inherited from QRS::Managers::AbstractRodComponentWidget

- void **modified** ()
- void **editDataObjectRequested** (Core::DataIDType id)

### Protected Member Functions inherited from QRS::Managers::AbstractRodComponentWidget

- void **setDataObjectEditConnections** (DataObjectLineEdit \*pEdit, DataObjectSetFun &setFun)  
*Specify connections of an editor which hold pointers to data objects of different types.*

### Protected Attributes inherited from QRS::Managers::AbstractRodComponentWidget

- QString const **mkMimeType**

## 4.35.1 Detailed Description

Widget to construct mechanical rod components consisted of stiffness and mass distributions.

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

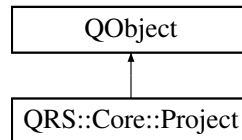
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/mechanicalrodcomponentwidget.h
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/mechanicalrodcomponentwidget.cpp

## 4.36 QRS::Core::Project Class Reference

[Project](#) class to interact with a created system of rods.

```
#include <project.h>
```

Inheritance diagram for QRS::Core::Project:



### Public Slots

- bool **save** (QString const &dir, QString const &fileName)  
*Save a project to a file.*
- void **setDataObjects** (QRS::Core::DataObjects const &dataObjects, [QRS::Core::HierarchyTree](#) const &hierarchyDataObjects)  
*Substitute current data objects with new ones.*
- void **setRodComponents** (QRS::Core::RodComponents const &rodComponents, [QRS::Core::HierarchyTree](#) const &hierarchyRodComponents)  
*Substitute current rod components with new ones.*

### Signals

- void **dataObjectsSubstituted** ()
- void **propertiesDataObjectsChanged** ()
- void **rodComponentsSubstituted** ()
- void **propertiesRodComponentsChanged** ()
- void **projectHierarchyChanged** ()

### Public Member Functions

- **Project** (QString const &name)  
*Construct a clean project with the user specified name.*
- **Project** (QString const &path, QString const &fileName)  
*Read a project from a file.*
- DataIDType **numberDataObjects** () const
- [AbstractDataObject](#) \* **addDataObject** (AbstractDataObject::ObjectType type)  
*Create a data object with the specified type.*
- DataObjects **cloneDataObjects** () const  
*Clone data objects.*
- [HierarchyTree](#) **cloneHierarchyDataObjects** () const
- DataIDType **numberRodComponents** () const
- [AbstractRodComponent](#) \* **addGeometry** ()  
*Create a geometrical rod component.*
- [AbstractRodComponent](#) \* **addCrossSection** (AbstractSectionRodComponent::SectionType sectionType)  
*Create a cross section.*

- [AbstractRodComponent](#) \* **addMaterial** ()  
*Add a material rod component.*
- [AbstractRodComponent](#) \* **addLoad** ()  
*Add a rod load.*
- [AbstractRodComponent](#) \* **addConstraint** ()  
*Add a rod constraint.*
- [AbstractRodComponent](#) \* **addMechanical** ()  
*Add a mechanical rod component.*
- RodComponents **cloneRodComponents** () const  
*Clone rod components.*
- [HierarchyTree](#) **cloneHierarchyRodComponents** () const
- QString const & **name** () const
- QString const & **filePath** () const
- void **importDataObjects** (QString const &path, QString const &fileName)  
*Import several data objects from a file.*

### Static Public Member Functions

- static QString const & **getFileExtension** ()

### Private Member Functions

- void **emplaceRodComponent** ([AbstractRodComponent](#) \*pRodComponent)  
*Emplace a rod component into a project.*

### Private Attributes

- quint32 **mID**  
*Unique project identifier.*
- QString **mName**  
*[Project](#) name.*
- QString **mFilePath**  
*Path to a file where a project is stored.*
- DataObjects **mDataObjects**  
*Data objects.*
- [HierarchyTree](#) **mHierarchyDataObjects**  
*Hierarchy of data objects.*
- RodComponents **mRodComponents**  
*Rod components.*
- [HierarchyTree](#) **mHierarchyRodComponents**  
*Hierarchy of rod components.*

### Static Private Attributes

- static const QString **skProjectExtension** = ".qrs"  
*File extensionn.*

## Friends

- class **QRS::HierarchyModels::ProjectHierarchyModel**
- class **QRS::Managers::ManagersFactory**

### 4.36.1 Detailed Description

[Project](#) class to interact with a created system of rods.

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

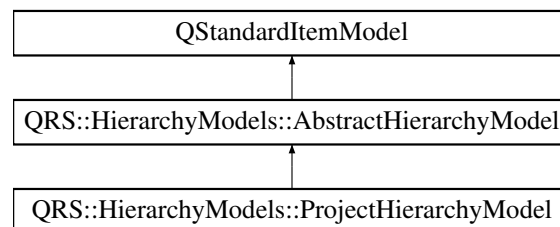
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[project.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[project-base.cpp](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[project-io.cpp](#)

## 4.37 QRS::HierarchyModels::ProjectHierarchyModel Class Reference

Project hierarchy representative.

```
#include <projecthierarchymodel.h>
```

Inheritance diagram for QRS::HierarchyModels::ProjectHierarchyModel:



## Public Slots

- void **validateItemSelection** ()  
*Check if an item selection is correct and if it is not – correct it.*

## Signals

- void **selectionValidated** (QVector< [QRS::HierarchyModels::AbstractHierarchyItem](#) \* > validatedItems)

## Signals inherited from [QRS::HierarchyModels::AbstractHierarchyModel](#)

- void **hierarchyChanged** ()  
*Emitted when hierarchical elements get renamed, moved or deleted.*

## Public Member Functions

- **ProjectHierarchyModel** (QString const &mimeType, QTreeView \*pView=nullptr)
- void **updateContent** () override  
*Update all the content.*
- void **clearContent** () override  
*Clear all the items.*
- void **setProject** (Core::Project \*pProject)  
*Set a project to represent.*

## Public Member Functions inherited from QRS::HierarchyModels::AbstractHierarchyModel

- **AbstractHierarchyModel** (QString const &mimeType, QTreeView \*pView=nullptr)
- virtual void **updateContent** ()=0
- virtual void **clearContent** ()=0
- Qt::DropActions **supportedDragActions** () const override  
*Specify allowed drag actions.*
- Qt::DropActions **supportedDropActions** () const override  
*Specify allowed drop actions.*
- QStringList **mimeType** () const override  
*Retrieve the mime types.*
- QMimeData \* **mimeData** (const QModelIndexList &indices) const override  
*Encode each item according to a given list of indices.*
- bool **dropMimeData** (QMimeData const \*pMimeData, Qt::DropAction action, int row, int column, const QModelIndex &parent) override  
*Process the drop action.*

## Private Member Functions

- DataObjectsHierarchyItem \* **retrieveDataObjectsItem** ()  
*Retrieve a representative of data objects.*
- RodComponentsHierarchyItem \* **retrieveRodComponentsItem** ()  
*Retrieve a representative of rod components.*

## Private Attributes

- Core::Project \* **mpProject** = nullptr

## Additional Inherited Members

## Protected Attributes inherited from QRS::HierarchyModels::AbstractHierarchyModel

- QString const **mkMimeType**

### 4.37.1 Detailed Description

Project hierarchy representative.

### 4.37.2 Member Function Documentation

#### 4.37.2.1 clearContent()

```
void ProjectHierarchyModel::clearContent ( ) [override], [virtual]
```

Clear all the items.

Implements [QRS::HierarchyModels::AbstractHierarchyModel](#).

#### 4.37.2.2 updateContent()

```
void ProjectHierarchyModel::updateContent ( ) [override], [virtual]
```

Update all the content.

Implements [QRS::HierarchyModels::AbstractHierarchyModel](#).

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

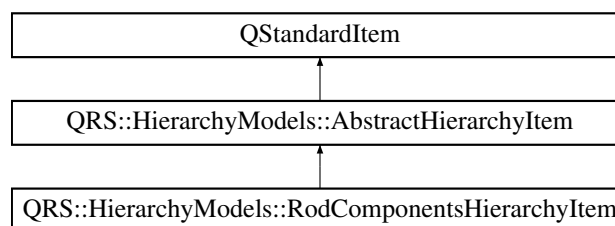
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.cpp](#)

## 4.38 QRS::HierarchyModels::RodComponentsHierarchyItem Class Reference

Item to represent a hierarchy of rod components.

```
#include <rodcomponentshierarchyitem.h>
```

Inheritance diagram for QRS::HierarchyModels::RodComponentsHierarchyItem:



### Public Member Functions

- **RodComponentsHierarchyItem** (Core::RodComponents &rodComponents, [Core::HierarchyTree](#) &hierarchy, Core::RodComponents, QString const &text="Root", QIcon const &icon=QIcon())  
*Create the representative of the structure of rod components.*
- **RodComponentsHierarchyItem** ([Core::HierarchyNode](#) \*pNode, [Core::AbstractRodComponent](#) \*pRodComponent)  
*Construct an item to represent a rod component.*
- **RodComponentsHierarchyItem** ([Core::HierarchyNode](#) \*pNode)  
*Construct an item to represent a directory.*
- int [type](#) () const override



**Public Member Functions inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)**

- **AbstractHierarchyItem** (QIcon const &icon, QString const &text, [Core::HierarchyNode](#) \*pNode)
- void **writePointer** (QDataStream &out) const  
*Write the pointer to the current item to a stream.*
- virtual int **type** () const =0

**Private Member Functions**

- void **appendItems** (Core::RodComponents &rodComponents, [Core::HierarchyNode](#) \*pNode)  
*Create items based on the position in the tree structure.*

**Private Attributes**

- [Core::AbstractRodComponent](#) \* **mpRodComponent** = nullptr

**Friends**

- class **RodComponentsHierarchyModel**

**Additional Inherited Members****Public Types inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)**

- enum **ItemType** { **kDataObjects** = QStandardItem::UserType , **kRodComponents** }

**Static Public Member Functions inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)**

- static [AbstractHierarchyItem](#) \* **readPointer** (QDataStream &in)  
*Retrieve a pointer to an item from a stream.*

**Protected Attributes inherited from [QRS::HierarchyModels::AbstractHierarchyItem](#)**

- [Core::HierarchyNode](#) \* **mpNode** = nullptr

**4.38.1 Detailed Description**

Item to represent a hierarchy of rod components.

## 4.38.2 Member Function Documentation

### 4.38.2.1 type()

```
int QRS::HierarchyModels::RodComponentsHierarchyItem::type ( ) const [inline], [override],
[virtual]
```

Implements [QRS::HierarchyModels::AbstractHierarchyItem](#).

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

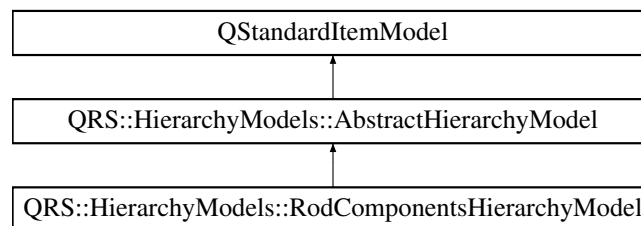
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.cpp](#)

## 4.39 QRS::HierarchyModels::RodComponentsHierarchyModel Class Reference

Tree model to represent and modify a hierarchy of rod components.

```
#include <rodcomponentshierarchymodel.h>
```

Inheritance diagram for QRS::HierarchyModels::RodComponentsHierarchyModel:



### Public Slots

- void **retrieveSelectedItem** ()  
*Retrieve a selected rod component.*
- void **removeSelectedItems** ()  
*Remove rod components under selection.*

### Signals

- void **selected** (Core::DataIDType id)
- void **selectionCleared** ()

### Signals inherited from [QRS::HierarchyModels::AbstractHierarchyModel](#)

- void **hierarchyChanged** ()  
*Emitted when hierarchical elements get renamed, moved or deleted.*

## Public Member Functions

- **RodComponentsHierarchyModel** (Core::RodComponents &rodComponents, [Core::HierarchyTree](#) &hierarchyRodComponents, QString const &mimeType, QTreeView \*pView=nullptr)
- void [updateContent](#) () override  
*Update all the content.*
- void [clearContent](#) () override  
*Clear all the items.*
- bool **isEmpty** () const  
*Check if there are data objects to represent.*
- void **selectItem** (int iRow)  
*Select an item by row index.*

## Public Member Functions inherited from [QRS::HierarchyModels::AbstractHierarchyModel](#)

- **AbstractHierarchyModel** (QString const &mimeType, QTreeView \*pView=nullptr)
- virtual void [updateContent](#) ()=0
- virtual void [clearContent](#) ()=0
- Qt::DropActions **supportedDragActions** () const override  
*Specify allowed drag actions.*
- Qt::DropActions **supportedDropActions** () const override  
*Specify allowed drop actions.*
- QStringList **mimeType** () const override  
*Retrieve the mime types.*
- QMimeData \* **mimeData** (const QModelIndexList &indices) const override  
*Encode each item according to a given list of indices.*
- bool **dropMimeData** (QMimeData const \*pMimeData, Qt::DropAction action, int row, int column, const QModelIndex &parent) override  
*Process the drop action.*

## Private Slots

- void **renameItem** (QStandardItem \*pStandardItem)  
*Rename a rod component after editing.*

## Private Attributes

- Core::RodComponents & **mRodComponents**
- [Core::HierarchyTree](#) & **mHierarchyRodComponents**

## Additional Inherited Members

## Protected Attributes inherited from [QRS::HierarchyModels::AbstractHierarchyModel](#)

- QString const **mkMimeType**

### 4.39.1 Detailed Description

Tree model to represent and modify a hierarchy of rod components.

### 4.39.2 Member Function Documentation

#### 4.39.2.1 clearContent()

```
void RodComponentsHierarchyModel::clearContent ( ) [override], [virtual]
```

Clear all the items.

Implements [QRS::HierarchyModels::AbstractHierarchyModel](#).

#### 4.39.2.2 updateContent()

```
void RodComponentsHierarchyModel::updateContent ( ) [override], [virtual]
```

Update all the content.

Implements [QRS::HierarchyModels::AbstractHierarchyModel](#).

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

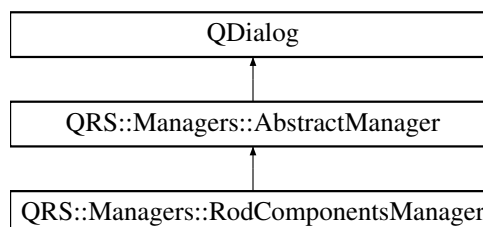
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchymodel.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchymodel.cpp](#)

## 4.40 QRS::Managers::RodComponentsManager Class Reference

Manager to create rod components, such as a geometry, cross section and force.

```
#include <rodcomponentsmanager.h>
```

Inheritance diagram for QRS::Managers::RodComponentsManager:



### Public Slots

- void **apply** () override  
*Apply all the changes made by user.*
- [Core::AbstractRodComponent](#) \* **addGeometry** ()  
*Add a geometrical component.*
- [Core::AbstractRodComponent](#) \* **addSection** ([Core::AbstractSectionRodComponent::SectionType](#) section↔  
Type)  
*Add a cross section.*
- [Core::AbstractRodComponent](#) \* **addMaterial** ()  
*Add a material component.*
- [Core::AbstractRodComponent](#) \* **addLoad** ()  
*Add a rod load.*
- [Core::AbstractRodComponent](#) \* **addConstraint** ()  
*Add a rod constraint.*
- [Core::AbstractRodComponent](#) \* **addMechanical** ()  
*Add a mechanical rod component.*
- void **resolveRodComponentsReferences** ()  
*Resolve references of rod components.*

### Public Slots inherited from [QRS::Managers::AbstractManager](#)

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

### Signals

- void **applied** ([Core::RodComponents](#) const &rodComponents, [Core::HierarchyTree](#) const &hierarchyRod↔  
Components)
- void **editDataObjectRequested** ([Core::DataIDType](#) id)

### Signals inherited from [QRS::Managers::AbstractManager](#)

- void **closed** ([QRS::Managers::AbstractManager::ManagerType](#) type)

### Public Member Functions

- **RodComponentsManager** ([Core::DataObjects](#) &dataObjects, [Core::HierarchyTree](#) &hierarchyData↔  
Objects, [Core::RodComponents](#) &&rodComponents, [Core::HierarchyTree](#) &&hierarchyRodComponents,  
QString &lastPath, QSettings &settings, QWidget \*parent=nullptr)
- void **selectRodComponent** (int iRow)  
*Select a rod component by row index.*
- void **updateDataObjects** ()  
*Update the representation of data objects.*

### Public Member Functions inherited from [QRS::Managers::AbstractManager](#)

- **AbstractManager** (QString &lastPath, QSettings &settings, ManagerType type, QString groupName, QWid-  
get \*parent=nullptr)
- void **saveSettings** ()  
*Save settings to a file.*
- void **restoreSettings** ()  
*Restore settings from a file.*

## Private Member Functions

- void **createContent** ()  
*Create all the widgets.*
- QLayout \* **createDialogControls** ()  
*Create dialog controls.*
- ads::CDockWidget \* **createHierarchyRodComponentsWidget** ()  
*Create a widget to show a hierarchy of rod components.*
- ads::CDockWidget \* **createConstructorDockWidget** ()  
*Create a dock widget to contain constructors of rod components.*
- ads::CDockWidget \* **createHierarchyDataObjectsWidget** ()  
*Create a widget to show a hierarchy of data objects.*
- void **emplaceRodComponent** (Core::AbstractRodComponent \*pRodComponent)  
*Helper function to insert a rod component into the manager.*
- void **representRodComponent** (Core::DataIDType id)  
*Represent a selected rod component according to its type.*
- void **clearRodComponentRepresentation** ()  
*Delete a widget to represent properties of a rod component.*
- QToolBar \* **createMainToolBar** ()  
*Create a menu to choose types of components to construct.*
- QWidget \* **makeGeometryToolBar** ()  
*Create a toolbar to create geometrical components.*
- QWidget \* **makeSectionsToolBar** ()  
*Create a toolbar to construct cross sections.*
- QWidget \* **makeBoundaryConditionsToolBar** ()  
*Create a toolbar to construct boundary conditions.*
- QWidget \* **makeLoadingToolBar** ()  
*Create a toolbar to construct loading.*
- QWidget \* **makeMaterialToolBar** ()  
*Create a toolbar to construct materials.*
- QWidget \* **makeMechanicalToolBar** ()  
*Create a toolbar to construct mechanical components.*
- QWidget \* **makeModificationToolBar** ()  
*Create a toolbar to modify rod components.*

## Private Attributes

- ads::CDockWidget \* **mpComponentDockWidget**
- QTreeView \* **mpTreeRodComponents**
- Core::DataObjects & **mDataObjects**
- Core::HierarchyTree & **mHierarchyDataObjects**
- Core::RodComponents & **mRodComponents**
- Core::HierarchyTree & **mHierarchyRodComponents**
- HierarchyModels::DataObjectsHierarchyModel \* **mpTreeDataObjectsModel**
- HierarchyModels::RodComponentsHierarchyModel \* **mpTreeRodComponentsModel**

## Additional Inherited Members

## Public Types inherited from QRS::Managers::AbstractManager

- enum **ManagerType** { kDataObjects , kRodComponents , kRodConstructor }

**Protected Member Functions inherited from [QRS::Managers::AbstractManager](#)**

- void **closeEvent** (QCloseEvent \*pEvent) override  
*Save settings and delete handling widgets before closing the window.*
- void **setToolBarShortcutHints** (QToolBar \*pToolBar)  
*Helper function to add a shortcut hint to all actions which a toolbar contains.*

**Protected Attributes inherited from [QRS::Managers::AbstractManager](#)**

- ads::CDockManager \* **mpDockManager** = nullptr
- QString & **mLastPath**

**4.40.1 Detailed Description**

Manager to create rod components, such as a geometry, cross section and force.

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

- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[rodcomponentsmanager.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[rodcomponentsmanager.cpp](#)

**4.41 QRS::Core::Array< T >::Row< U > Struct Template Reference**

Proxy class to acquire a row by index.

**Public Member Functions**

- **Row** (T \*pData)
- T & **operator[]** (IndexType iCol)
- T const & **operator[]** (IndexType iCol) const

**Public Attributes**

- T \* **pRow**

**4.41.1 Detailed Description**

```
template<typename T>
template<typename U>
struct QRS::Core::Array< T >::Row< U >
```

Proxy class to acquire a row by index.

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

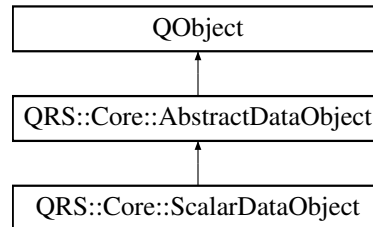
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[array.h](#)

## 4.42 QRS::Core::ScalarDataObject Class Reference

Scalar data object.

```
#include <scalardataobject.h>
```

Inheritance diagram for QRS::Core::ScalarDataObject:



### Public Member Functions

- **ScalarDataObject** (QString const &name)  
*Construct a scalar data object.*
- **~ScalarDataObject** ()  
*Decrease a number of instances while being destroyed.*
- **AbstractDataObject \* clone** () const override  
*Clone a scalar data object.*
- **DataItemType & addItem** (DataValueType key) override  
*Insert a new item into [ScalarDataObject](#).*
- virtual void **import** (QTextStream &stream) override  
*Import a scalar data object from a file.*

### Public Member Functions inherited from [QRS::Core::AbstractDataObject](#)

- **AbstractDataObject** (ObjectType type, QString const &name)  
*Base constructor.*
- virtual **AbstractDataObject \* clone** () const =0
- virtual **DataItemType & addItem** (DataKeyType key)=0
- void **removeItem** (DataValueType key)  
*Remove an entity with the specified key.*
- bool **changeItemKey** (DataKeyType oldKey, DataKeyType newKey, DataHolder \*items=nullptr)  
*Modify a key existed.*
- DataValueType **getAvailableItemKey** (DataValueType key, DataHolder const \*items=nullptr) const
- bool **setArrayValue** (DataKeyType key, DataValueType newValue, IndexType iRow=0, IndexType iColumn=0)  
*Set an array value with the specified indices.*
- quint32 **numberOfItems** () const
- DataHolder const & **getItems** ()
- DataIDType **id** () const
- ObjectType **type** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const  
*Serialize an abstract data object.*
- virtual void **deserialize** (QDataStream &stream)  
*Partly deserialize an abstract data object.*
- virtual void **import** (QTextStream &stream)=0



**Static Public Member Functions**

- static quint32 **numberInstances** ()

**Static Public Member Functions inherited from [QRS::Core::AbstractDataObject](#)**

- static DataIDType **maxObjectID** ()
- static void **setMaxObjectID** (DataIDType iMaxObjectID)

**Static Private Attributes**

- static quint32 **smNumInstances** = 0

**Additional Inherited Members****Public Types inherited from [QRS::Core::AbstractDataObject](#)**

- enum **ObjectType** { kScalar , kVector , kMatrix , kSurface }

**Protected Attributes inherited from [QRS::Core::AbstractDataObject](#)**

- const ObjectType **mkType**
- QString **mName**
- DataIDType **mID**
- DataHolder **mItems**

**4.42.1 Detailed Description**

Scalar data object.

**4.42.2 Member Function Documentation****4.42.2.1 addItem()**

```
DataItemType & ScalarDataObject::addItem (
    DataValueType key ) [override], [virtual]
```

Insert a new item into [ScalarDataObject](#).

Implements [QRS::Core::AbstractDataObject](#).

**4.42.2.2 clone()**

```
AbstractDataObject * ScalarDataObject::clone ( ) const [override], [virtual]
```

Clone a scalar data object.

Implements [QRS::Core::AbstractDataObject](#).

#### 4.42.2.3 import()

```
void ScalarDataObject::import (
    QTextStream & stream ) [override], [virtual]
```

Import a scalar data object from a file.

Implements [QRS::Core::AbstractDataObject](#).

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

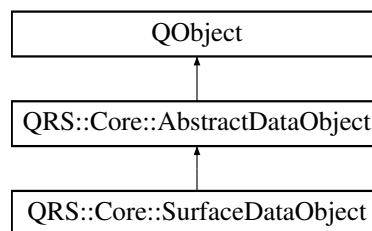
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[scalardataobject.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[scalardataobject.cpp](#)

### 4.43 QRS::Core::SurfaceDataObject Class Reference

Surface data object.

```
#include <surfacedataobject.h>
```

Inheritance diagram for QRS::Core::SurfaceDataObject:



#### Public Member Functions

- **SurfaceDataObject** (QString const &name)  
*Construct a surface data object.*
- **~SurfaceDataObject** ()  
*Decrease a number of instances while being destroyed.*
- [AbstractDataObject](#) \* **clone** () const override  
*Clone a surface data object.*
- [DataItemType](#) & **addItem** (DataValueType key) override  
*Insert a new item into [SurfaceDataObject](#).*
- DataKeyType **addLeadingItem** (DataValueType key)  
*Add a leading item.*
- void **removeLeadingItem** (DataValueType key)  
*Remove a leading item.*
- bool **changeLeadingItemKey** (DataKeyType oldKey, DataKeyType newKey)  
*Modify a leading item key.*
- quint32 **numberLeadingItems** () const
- DataHolder & **getLeadingItems** ()
- void **serialize** (QDataStream &stream) const override  
*Serialize additional data of a surface object.*
- virtual void **deserialize** (QDataStream &stream) override  
*Deserialize additional data of a surface object.*
- virtual void **import** (QTextStream &stream) override  
*Import a surface data object from a file.*

## Public Member Functions inherited from QRS::Core::AbstractDataObject

- **AbstractDataObject** (ObjectType type, QString const &name)  
*Base constructor.*
- virtual **AbstractDataObject** \* **clone** () const =0
- virtual **DataItemType** & **addItem** (DataKeyType key)=0
- void **removeItem** (DataValueType key)  
*Remove an entity with the specified key.*
- bool **changeItemKey** (DataKeyType oldKey, DataKeyType newKey, DataHolder \*items=nullptr)  
*Modify a key existed.*
- DataValueType **getAvailableItemKey** (DataValueType key, DataHolder const \*items=nullptr) const
- bool **setArrayValue** (DataKeyType key, DataValueType newValue, IndexType iRow=0, IndexType iColumn=0)  
*Set an array value with the specified indices.*
- quint32 **numberOfItems** () const
- DataHolder const & **getItems** ()
- DataIDType **id** () const
- ObjectType **type** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const  
*Serialize an abstract data object.*
- virtual void **deserialize** (QDataStream &stream)  
*Partly deserialize an abstract data object.*
- virtual void **import** (QTextStream &stream)=0

## Static Public Member Functions

- static quint32 **numberOfInstances** ()

## Static Public Member Functions inherited from QRS::Core::AbstractDataObject

- static DataIDType **maxObjectID** ()
- static void **setMaxObjectID** (DataIDType iMaxObjectID)

## Private Attributes

- DataHolder **mLeadingItems**

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

## Public Types inherited from QRS::Core::AbstractDataObject

- enum **ObjectType** { **kScalar** , **kVector** , **kMatrix** , **kSurface** }

## Protected Attributes inherited from [QRS::Core::AbstractDataObject](#)

- const ObjectType **mkType**
- QString **mName**
- DataIDType **mID**
- DataHolder **mItems**

### 4.43.1 Detailed Description

Surface data object.

### 4.43.2 Member Function Documentation

#### 4.43.2.1 addItem()

```
DataItemType & SurfaceDataObject::addItem (
    DataValueType key ) [override], [virtual]
```

Insert a new item into [SurfaceDataObject](#).

Implements [QRS::Core::AbstractDataObject](#).

#### 4.43.2.2 clone()

```
AbstractDataObject * SurfaceDataObject::clone ( ) const [override], [virtual]
```

Clone a surface data object.

Implements [QRS::Core::AbstractDataObject](#).

#### 4.43.2.3 deserialize()

```
void SurfaceDataObject::deserialize (
    QDataStream & stream ) [override], [virtual]
```

Deserialize additional data of a surface object.

Reimplemented from [QRS::Core::AbstractDataObject](#).

#### 4.43.2.4 import()

```
void SurfaceDataObject::import (
    QTextStream & stream ) [override], [virtual]
```

Import a surface data object from a file.

Implements [QRS::Core::AbstractDataObject](#).

## 4.43.2.5 serialize()

```
void SurfaceDataObject::serialize (
    QDataStream & stream ) const [override], [virtual]
```

Serialize additional data of a surface object.

Reimplemented from [QRS::Core::AbstractDataObject](#).

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

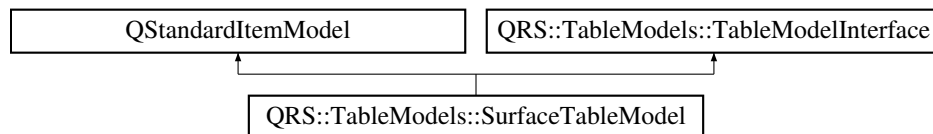
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[surfacedataobject.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[surfacedataobject.cpp](#)

## 4.44 QRS::TableModels::SurfaceTableModel Class Reference

Table model to represent a surface data object.

```
#include <surfacetablemodel.h>
```

Inheritance diagram for QRS::TableModels::SurfaceTableModel:



## Public Member Functions

- **SurfaceTableModel** (QWidget \*parent=nullptr)
- void **setDataObject** ([Core::SurfaceDataObject](#) \*pDataObject)  
*Set a surface data object to represent.*
- bool **setData** (const QModelIndex &indexEdit, const QVariant &value, int role=Qt::EditRole) override  
*Set the data acquired from a delegate.*
- void **insertItemAfterSelected** (QItemSelectionModel \*pSelectionModel) override  
*Insert a new item after selected one.*
- void **removeSelectedItem** (QItemSelectionModel \*pSelectionModel) override  
*Remove an array under selection.*
- void **insertLeadingItemAfterSelected** (QItemSelectionModel \*pSelectionModel) override  
*Add a new leading item after selected one.*
- void **removeSelectedLeadingItem** (QItemSelectionModel \*pSelectionModel) override  
*Remove a selected leading item.*
- virtual void **insertItemAfterSelected** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **insertLeadingItemAfterSelected** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **removeSelectedItem** (QItemSelectionModel \*pSelectionModel)=0
- virtual void **removeSelectedLeadingItem** (QItemSelectionModel \*pSelectionModel)=0

### Private Member Functions

- void **updateContent** ()  
*Represent all items which a data object contains.*
- void **clearContent** ()  
*Clear previously created items.*

### Private Attributes

- [Core::SurfaceDataObject](#) \* **mpDataObject** = nullptr

### Additional Inherited Members

### Static Public Member Functions inherited from [QRS::TableModels::TableModelInterface](#)

- static QStandardItem \* **makeDoubleItem** (double value)  
*Helper function to make an item which holds a double value.*
- static QList< QStandardItem \* > **prepareRow** ([Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array.*
- static QList< QStandardItem \* > **prepareRow** (double const &key, [Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with an key.*
- static QList< QStandardItem \* > **prepareRow** (QString const &name, [Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with a name.*
- static QStandardItem \* **makeLabelItem** (QString const &name)  
*Helper function to create an item which holds a string and cannot be modified.*

## 4.44.1 Detailed Description

Table model to represent a surface data object.

## 4.44.2 Member Function Documentation

### 4.44.2.1 insertItemAfterSelected()

```
void SurfaceTableModel::insertItemAfterSelected (
    QItemSelectionModel * pSelectionModel ) [override], [virtual]
```

Insert a new item after selected one.

Implements [QRS::TableModels::TableModelInterface](#).

### 4.44.2.2 insertLeadingItemAfterSelected()

```
void SurfaceTableModel::insertLeadingItemAfterSelected (
    QItemSelectionModel * pSelectionModel ) [override], [virtual]
```

Add a new leading item after selected one.

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.44.2.3 removeSelectedItem()

```
void SurfaceTableModel::removeSelectedItem (
    QTableWidgetItem * pSelectionModel ) [override], [virtual]
```

Remove an array under selection.

Implements [QRS::TableModels::TableModelInterface](#).

#### 4.44.2.4 removeSelectedLeadingItem()

```
void SurfaceTableModel::removeSelectedLeadingItem (
    QTableWidgetItem * pSelectionModel ) [override], [virtual]
```

Remove a selected leading item.

Implements [QRS::TableModels::TableModelInterface](#).

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

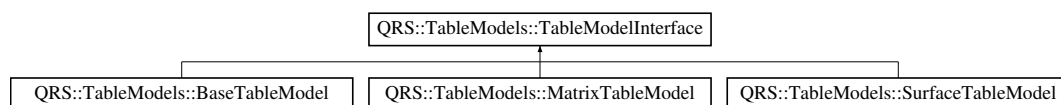
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[surfacetablemodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[surfacetablemodel.cpp](#)

## 4.45 QRS::TableModels::TableModelInterface Class Reference

User interface to add and remove items.

```
#include <tablemodelinterface.h>
```

Inheritance diagram for QRS::TableModels::TableModelInterface:



### Public Member Functions

- virtual void [insertItemAfterSelected](#) (QItemSelectionModel \*pSelectionModel)=0
- virtual void [insertLeadingItemAfterSelected](#) (QItemSelectionModel \*pSelectionModel)=0
- virtual void [removeSelectedItem](#) (QItemSelectionModel \*pSelectionModel)=0
- virtual void [removeSelectedLeadingItem](#) (QItemSelectionModel \*pSelectionModel)=0

## Static Public Member Functions

- static `QStandardItem * makeDoubleItem` (double value)  
*Helper function to make an item which holds a double value.*
- static `QList< QStandardItem * > prepareRow` ([Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array.*
- static `QList< QStandardItem * > prepareRow` (double const &key, [Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with an key.*
- static `QList< QStandardItem * > prepareRow` (QString const &name, [Core::Array](#)< double > const &array, quint32 iRow)  
*Helper function to copy a row from an array and associate it with a name.*
- static `QStandardItem * makeLabelItem` (QString const &name)  
*Helper function to create an item which holds a string and cannot be modified.*

### 4.45.1 Detailed Description

User interface to add and remove items.

### 4.45.2 Member Function Documentation

#### 4.45.2.1 insertItemAfterSelected()

```
virtual void QRS::TableModels::TableModelInterface::insertItemAfterSelected (
    QItemSelectionModel * pSelectionModel ) [pure virtual]
```

Implemented in [QRS::TableModels::BaseTableModel](#), [QRS::TableModels::MatrixTableModel](#), and [QRS::TableModels::SurfaceTableModel](#).

#### 4.45.2.2 insertLeadingItemAfterSelected()

```
virtual void QRS::TableModels::TableModelInterface::insertLeadingItemAfterSelected (
    QItemSelectionModel * pSelectionModel ) [pure virtual]
```

Implemented in [QRS::TableModels::SurfaceTableModel](#).

#### 4.45.2.3 removeSelectedItem()

```
virtual void QRS::TableModels::TableModelInterface::removeSelectedItem (
    QItemSelectionModel * pSelectionModel ) [pure virtual]
```

Implemented in [QRS::TableModels::BaseTableModel](#), [QRS::TableModels::MatrixTableModel](#), and [QRS::TableModels::SurfaceTableModel](#).



## 4.45.2.4 removeSelectedLeadingItem()

```
virtual void QRS::TableModels::TableModelInterface::removeSelectedLeadingItem (
    QTableWidgetItem * pSelectionModel ) [pure virtual]
```

Implemented in [QRS::TableModels::SurfaceTableModel](#).

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

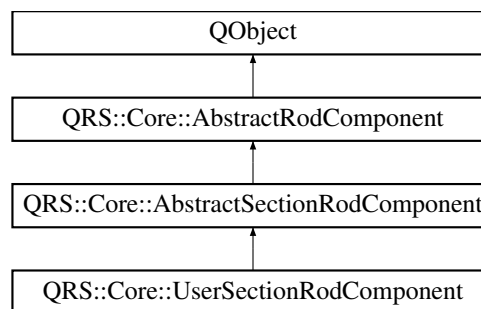
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[tablemodelinterface.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[tablemodelinterface.cpp](#)

## 4.46 QRS::Core::UserSectionRodComponent Class Reference

Section which properties are defined by user.

```
#include <usersectionrodcomponent.h>
```

Inheritance diagram for QRS::Core::UserSectionRodComponent:



## Public Member Functions

- **UserSectionRodComponent** (QString const &name)
- [AbstractRodComponent](#) \* **clone** () const override  
*Clone a user-defined cross section.*
- bool **isDataComplete** () const override  
*Check if specified data is complete.*
- [ScalarDataObject](#) const \* **area** () const
- [ScalarDataObject](#) const \* **inertiaMomentTorsional** () const
- [ScalarDataObject](#) const \* **inertiaMomentX** () const
- [ScalarDataObject](#) const \* **inertiaMomentY** () const
- [ScalarDataObject](#) const \* **centerCoordinateX** () const
- [ScalarDataObject](#) const \* **centerCoordinateY** () const
- void **setArea** ([ScalarDataObject](#) const \*pArea)
- void **setInertiaMomentTorsional** ([ScalarDataObject](#) const \*pInertiaMomentTorsional)
- void **setInertiaMomentX** ([ScalarDataObject](#) const \*pInertiaMomentX)
- void **setInertiaMomentY** ([ScalarDataObject](#) const \*pInertiaMomentY)
- void **setCenterCoordinateX** ([ScalarDataObject](#) const \*pCenterCoordinateX)
- void **setCenterCoordinateY** ([ScalarDataObject](#) const \*pCenterCoordinateY)

## Public Member Functions inherited from [QRS::Core::AbstractSectionRodComponent](#)

- **AbstractSectionRodComponent** (SectionType sectionType, QString const &name)
- virtual **~AbstractSectionRodComponent** ()=0  
*Decrease a number of instances while being destroyed.*
- void **serialize** (QDataStream &stream) const override  
*Serialize a cross section.*
- void **deserialize** (QDataStream &stream, DataObjects const &dataObjects) override  
*Partly deserialize an abstract rod component.*
- void **resolveReferences** (DataObjects const &dataObjects) override  
*Resolve references of a cross-section.*
- SectionType **sectionType** () const

## Public Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- **AbstractRodComponent** (ComponentType componentType, QString const &name)
- virtual **AbstractRodComponent** \* **clone** () const =0
- virtual bool **isDataComplete** () const =0
- DataIDType **id** () const
- ComponentType **componentType** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const =0
- virtual void **deserialize** (QDataStream &stream, DataObjects const &dataObjects)=0
- virtual void **resolveReferences** (DataObjects const &dataObjects)=0

## Additional Inherited Members

## Public Types inherited from [QRS::Core::AbstractSectionRodComponent](#)

- enum **SectionType** { **kUserDefined** }

## Public Types inherited from [QRS::Core::AbstractRodComponent](#)

- enum **ComponentType** {  
  **kGeometry** , **kSection** , **kMaterial** , **kLoad** ,  
  **kConstraint** , **kMechanical** }

## Static Public Member Functions inherited from [QRS::Core::AbstractSectionRodComponent](#)

- static quint32 **numberInstances** ()

## Static Public Member Functions inherited from [QRS::Core::AbstractRodComponent](#)

- static DataIDType **maxComponentID** ()
- static void **setMaxComponentID** (DataIDType iMaxComponentID)

## Protected Member Functions inherited from QRS::Core::AbstractSectionRodComponent

- void **copyIntegratedProperties** ([AbstractSectionRodComponent](#) const \*pSection)  
*Copy integrated properties of a cross section.*

## Protected Member Functions inherited from QRS::Core::AbstractRodComponent

- void **writeDataObjectPointer** (QDataStream &stream, [AbstractDataObject](#) const \*pDataObject) const  
*Helper function to write the identifier of a data object.*
- [AbstractDataObject](#) const \* **readDataObjectPointer** (QDataStream &stream, DataObjects const &dataObjects) const  
*Helper function to retrieve the pointer to the data object by its identifier.*
- [AbstractDataObject](#) const \* **getDataObject** (DataObjects const &dataObjects, DataIDType id) const  
*Retrieve a data object from a set by id.*
- [AbstractDataObject](#) const \* **substituteDataObject** (DataObjects const &dataObjects, [AbstractDataObject](#) const \*pDataObject) const  
*Substitute a data object with its updated version.*

## Protected Attributes inherited from QRS::Core::AbstractSectionRodComponent

- SectionType const **mkSectionType**
- QPointer< [ScalarDataObject](#) const > **mpArea**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMomentTorsional**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMomentX**
- QPointer< [ScalarDataObject](#) const > **mpInertiaMomentY**
- QPointer< [ScalarDataObject](#) const > **mpCenterCoordinateX**
- QPointer< [ScalarDataObject](#) const > **mpCenterCoordinateY**

## Protected Attributes inherited from QRS::Core::AbstractRodComponent

- ComponentType const **mkComponentType**
- QString **mName**
- DataIDType **mID**

## Static Protected Attributes inherited from QRS::Core::AbstractSectionRodComponent

- static quint32 **smNumInstances** = 0

### 4.46.1 Detailed Description

Section which properties are defined by user.

### 4.46.2 Member Function Documentation

#### 4.46.2.1 clone()

```
AbstractRodComponent * UserSectionRodComponent::clone ( ) const [override], [virtual]
```

Clone a user-defined cross section.

Implements [QRS::Core::AbstractRodComponent](#).

#### 4.46.2.2 isDataComplete()

```
bool UserSectionRodComponent::isDataComplete ( ) const [override], [virtual]
```

Check if specified data is complete.

Some of properties may be of zero values to achieve infinite stiffness

Implements [QRS::Core::AbstractRodComponent](#).

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

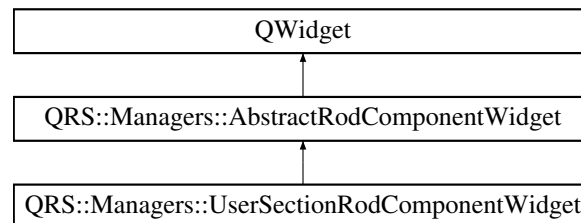
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.h
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.cpp

### 4.47 QRS::Managers::UserSectionRodComponentWidget Class Reference

Widget to construct a user-defined section of a rod.

```
#include <usersectionrodcomponentwidget.h>
```

Inheritance diagram for QRS::Managers::UserSectionRodComponentWidget:



#### Public Member Functions

- **UserSectionRodComponentWidget** ([Core::UserSectionRodComponent](#) &userSectionRodComponent, QString const &mimeType, QWidget \*parent=nullptr)

#### Public Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)

- **AbstractRodComponentWidget** (QString const &mimeType, QWidget \*parent=nullptr)

#### Private Member Functions

- void **createContent** ()  
*Create all the content.*
- QLayout \* **createAreaLayout** ()  
*Create an area layout.*
- QWidget \* **createInertiaMomentsGroup** ()  
*Create a group consisted of widgets to set moments of inertia.*
- QWidget \* **createCenterCoordinatesGroup** ()  
*Create a group consisted of widgets to set coordinates of the center.*
- void **setProperty** ([Core::AbstractDataObject](#) const \*pDataObject, auto setFun)  
*Set a section property which takes a scalar data object.*

**Private Attributes**

- [Core::UserSectionRodComponent](#) & **mUserSectionRodComponent**

**Additional Inherited Members****Signals inherited from [QRS::Managers::AbstractRodComponentWidget](#)**

- void **modified** ()
- void **editDataObjectRequested** (Core::DataIDType id)

**Protected Member Functions inherited from [QRS::Managers::AbstractRodComponentWidget](#)**

- void **setDataObjectEditConnections** ([DataObjectLineEdit](#) \*pEdit, DataObjectSetFun &setFun)  
*Specify connections of an editor which hold pointers to data objects of different types.*

**Protected Attributes inherited from [QRS::Managers::AbstractRodComponentWidget](#)**

- QString const **mkMimeType**

**4.47.1 Detailed Description**

Widget to construct a user-defined section of a rod.

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

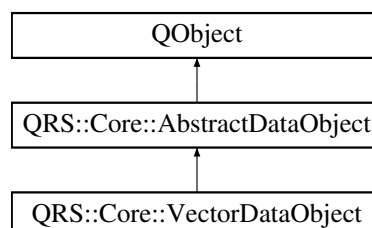
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[usersectionrodcomponentwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[usersectionrodcomponentwidget.cpp](#)

**4.48 QRS::Core::VectorDataObject Class Reference**

Vector data object.

```
#include <vectordataobject.h>
```

Inheritance diagram for QRS::Core::VectorDataObject:



## Public Member Functions

- **VectorDataObject** (QString const &name)  
*Construct a vector data object.*
- **~VectorDataObject** ()  
*Decrease a number of instances while being destroyed.*
- **AbstractDataObject** \* **clone** () const override  
*Clone a vector data object.*
- **DataItemType** & **addItem** (DataValueType key) override  
*Insert a new item into [VectorDataObject](#).*
- virtual void **import** (QTextStream &stream) override  
*Import a vector data object from a file.*

## Public Member Functions inherited from [QRS::Core::AbstractDataObject](#)

- **AbstractDataObject** (ObjectType type, QString const &name)  
*Base constructor.*
- virtual **AbstractDataObject** \* **clone** () const =0
- virtual **DataItemType** & **addItem** (DataKeyType key)=0
- void **removeItem** (DataValueType key)  
*Remove an entity with the specified key.*
- bool **changeItemKey** (DataKeyType oldKey, DataKeyType newKey, DataHolder \*items=nullptr)  
*Modify a key existed.*
- DataValueType **getAvailableItemKey** (DataValueType key, DataHolder const \*items=nullptr) const
- bool **setArrayValue** (DataKeyType key, DataValueType newValue, IndexType iRow=0, IndexType iColumn=0)  
*Set an array value with the specified indices.*
- quint32 **numberOfItems** () const
- DataHolder const & **getItems** ()
- DataIDType **id** () const
- ObjectType **type** () const
- QString const & **name** () const
- void **setName** (QString const &name)
- virtual void **serialize** (QDataStream &stream) const  
*Serialize an abstract data object.*
- virtual void **deserialize** (QDataStream &stream)  
*Partly deserialize an abstract data object.*
- virtual void **import** (QTextStream &stream)=0

## Static Public Member Functions

- static quint32 **numberOfInstances** ()

## Static Public Member Functions inherited from [QRS::Core::AbstractDataObject](#)

- static DataIDType **maxObjectID** ()
- static void **setMaxObjectID** (DataIDType iMaxObjectID)

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

### Public Types inherited from [QRS::Core::AbstractDataObject](#)

- enum **ObjectType** { kScalar , kVector , kMatrix , kSurface }

### Protected Attributes inherited from [QRS::Core::AbstractDataObject](#)

- const ObjectType **mkType**
- QString **mName**
- DataIDType **mID**
- DataHolder **mItems**

## 4.48.1 Detailed Description

Vector data object.

## 4.48.2 Member Function Documentation

### 4.48.2.1 addItem()

```
DataItemType & VectorDataObject::addItem (
    DataValueType key ) [override], [virtual]
```

Insert a new item into [VectorDataObject](#).

Implements [QRS::Core::AbstractDataObject](#).

### 4.48.2.2 clone()

```
AbstractDataObject * VectorDataObject::clone ( ) const [override], [virtual]
```

Clone a vector data object.

Implements [QRS::Core::AbstractDataObject](#).

### 4.48.2.3 import()

```
void VectorDataObject::import (
    QTextStream & stream ) [override], [virtual]
```

Import a vector data object from a file.

Implements [QRS::Core::AbstractDataObject](#).

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

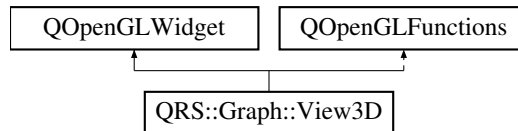
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.cpp](#)

## 4.49 QRS::Graph::View3D Class Reference

A widget to represent the resulted rod system.

```
#include <view3d.h>
```

Inheritance diagram for QRS::Graph::View3D:



### Public Member Functions

- **View3D** (QWidget \*parent=nullptr)

### Protected Member Functions

- void **initializeGL** () override  
*Initialize a graphical scene.*
- void **paintGL** () override  
*Render its content.*

### Private Attributes

- bool **mCore**

### 4.49.1 Detailed Description

A widget to represent the resulted rod system.

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

- /home/qinterfly/Library/Projects/Current/QRodSystems/src/render/[view3d.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/render/[view3d.cpp](#)



# Chapter 5

## File Documentation

### 5.1 [/home/qinterfly/Library/Projects/Current/QRod↵](#) Systems/src/central/controltabs.cpp File Reference

Implementation of the ControlTabs class.

```
#include <QLayout>
#include <QToolBar>
#include <QIcon>
#include "controltabs.h"
```

#### 5.1.1 Detailed Description

Implementation of the ControlTabs class.

Author

Pavel Lakiza

Date

March 2021

### 5.2 [/home/qinterfly/Library/Projects/Current/QRod↵](#) Systems/src/central/controltabs.h File Reference

Declaration of the ControlTabs class.

```
#include <QWidget>
```

## Classes

- class [QRS::App::ManagersTab](#)

*A toolbar consisted of object designers.*

### 5.2.1 Detailed Description

Declaration of the ControlTabs class.

#### Author

Pavel Lakiza

#### Date

March 2021

## 5.3 controlltabs.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef CONTROLTABS_H
00009 #define CONTROLTABS_H
00010
00011 #include <QWidget>
00012
00013 namespace QRS::App
00014 {
00015
00016     class ManagersTab : public QWidget
00017     {
00018     public:
00019         Q_OBJECT
00020
00021     public:
00022         explicit ManagersTab(QWidget* parent = nullptr);
00023         ~ManagersTab() = default;
00024
00025     signals:
00026         void actionDataObjectsTriggered();
00027         void actionRodPropertiesTriggered();
00028         void actionRodConstructorTriggered();
00029     };
00030
00031 }
00032
00033
00034 #endif // CONTROLTABS_H
```

## 5.4 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/central/logwidget.cpp File Reference

Implementation of the LogWidget class.

```
#include <QHeaderView>
#include <QTime>
#include <QTimer>
#include "logwidget.h"
```

## Enumerations

- enum **ColumnType** { **kTime** , **kType** , **kMessage** }

### 5.4.1 Detailed Description

Implementation of the LogWidget class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.5 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/logwidget.h File Reference

Declaration of the LogWidget class.

```
#include <QTableWidget>
```

## Classes

- class [QRS::App::LogWidget](#)  
*Log all the messages sent.*

### 5.5.1 Detailed Description

Declaration of the LogWidget class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.6 logwidget.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef LOGWIDGET_H
00009 #define LOGWIDGET_H
00010
00011 #include <QTableWidget>
00012
00013 namespace QRS::App
00014 {
00015
00017 class LogWidget : public QTableWidget
00018 {
00019 public:
00020     explicit LogWidget(QWidget* parent = nullptr);
00021     ~LogWidget() = default;
00022     void log(QtMsgType messageType, const QString& message);
00023 };
00024
00025 }
00026
00027
00028 #endif // LOGWIDGET_H

```

## 5.7 /home/qinterfly/Library/Projects/Current/QRod Systems/src/central/mainwindow.cpp File Reference

Implementation of the MainWindow class.

```

#include <QToolBar>
#include <QTreeView>
#include <QTableView>
#include <QHeaderView>
#include <QTextEdit>
#include <QVBoxLayout>
#include <QSettings>
#include <QMessageBox>
#include <QFileDialog>
#include <QLabel>
#include "DockManager.h"
#include "DockWidget.h"
#include "DockAreaWidget.h"
#include "ads_globals.h"
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "controltabs.h"
#include "logwidget.h"
#include "uiconstants.h"
#include "models/hierarchy/projecthierarchymodel.h"
#include "models/properties/dataobjectspropertiesmodel.h"
#include "managers/managersfactory.h"
#include "render/view3d.h"

```

### 5.7.1 Detailed Description

Implementation of the MainWindow class.

**Author**

Pavel Lakiza

**Date**

May 2021

## 5.8 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/mainwindow.h File Reference

Declaration of the MainWindow class.

```
#include <QMainWindow>
#include "logwidget.h"
#include "core/project.h"
#include "models/hierarchy/abstracthierarchyitem.h"
```

**Classes**

- class [QRS::App::MainWindow](#)  
*The main window of the program.*

**Functions**

- void **QRS::App::throwMessage** (QtMsgType type, const QMessageLogContext &, const QString &message)  
*Log all the messages.*

### 5.8.1 Detailed Description

Declaration of the MainWindow class.

**Author**

Pavel Lakiza

**Date**

May 2021

## 5.9 mainwindow.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef MAINWINDOW_H
00009 #define MAINWINDOW_H
00010
00011 #include <QMainWindow>
00012 #include "logwidget.h"
00013 #include "core/project.h"
00014 #include "models/hierarchy/abstracthierarchyitem.h"
00015
00016 QT_BEGIN_NAMESPACE
00017 namespace Ui
00018 {
00019     class MainWindow;
00020 }
00021 class QSettings;
00022 class QLabel;
00023 class QTableView;
00024 QT_END_NAMESPACE
00025
00026 namespace ads
00027 {
00028     class CDockManager;
00029     class CDockWidget;
00030 }
00031
00032 namespace QRS
00033 {
00034     namespace Managers
00035     {
00036         class ManagersFactory;
00037     }
00038 }
00039
00040 namespace HierarchyModels
00041 {
00042     class ProjectHierarchyModel;
00043 }
00044
00045 namespace App
00046 {
00047     class MainWindow : public QMainWindow
00048     {
00049     public:
00050         Q_OBJECT
00051
00052     public:
00053         MainWindow(QWidget* parent = nullptr);
00054         ~MainWindow();
00055         void openProject(QString const& filePath);
00056         bool saveProject();
00057
00058     private:
00059         // Content
00060         void initializeWindow();
00061         void createContent();
00062         void closeEvent(QCloseEvent* pEvent) override;
00063         ads::CDockWidget* createProjectHierarchyWidget();
00064         ads::CDockWidget* createGLWidget();
00065         ads::CDockWidget* createCodeWidget();
00066         ads::CDockWidget* createLogWidget();
00067         ads::CDockWidget* createPropertiesWidget();
00068         void setProjectTitle();
00069         void retrieveRecentProjects();
00070         void addToRecentProjects();
00071         // Signals & Slots
00072         void specifyMenuConnections();
00073         void specifyProjectConnections();
00074         // Project
00075         bool saveProjectChangesDialog();
00076         bool saveProjectHelper(QString const& filePath);
00077
00078     private slots:
00079         // Project
00080         void createProject();
00081         void openProjectDialog();
00082         void openRecentProject();
00083         bool saveAsProject();
00084         void setModified(bool flag);
00085         // Properties
00086         void representHierarchyProperties(QVector<HierarchyModels::AbstractHierarchyItem*> items);
00087         // Settings
00088         void saveSettings();
00089

```

```

00090     void restoreSettings();
00091     // Managers
00092     void createDataObjectsManager();
00093     void createRodComponentsManager();
00094     void createRodConstructorManager();
00095     // Help
00096     void aboutProgram();
00097
00098 private:
00099     // UI
00100     Ui::MainWindow* mpUi;
00101     ads::CDockManager* mpDockManager;
00102     QLabel* mpStatusLabel;
00103     QTableView* mpPropertiesWidget;
00104     // Models
00105     HierarchyModels::ProjectHierarchyModel* mpProjectHierarchyModel = nullptr;
00106     // Managers
00107     Managers::ManagersFactory* mpManagersFactory = nullptr;
00108     // Project data
00109     Core::Project* mpProject;
00110     // Settings
00111     QSharedPointer<QSettings> mpSettings;
00112     QString mLastPath;
00113     QList<QString> mPathRecentProjects;
00114
00115 public:
00116     static LogWidget* pLogger;
00117 };
00118
00119 inline void throwMessage(QtMsgType type, const QMessageLogContext& /*context*/, const QString&
    message)
00120 {
00121     {
00122         MainWindow::pLogger->log(type, message);
00123     }
00124 }
00125 }
00126 }
00127 }
00128
00129 #endif // MAINWINDOW_H

```

## 5.10 /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/uiconstants.h File Reference

Common graphical constants shared between several windows.

```
#include <QString>
```

### Variables

- const QString **QRS::UiConstants::Settings::skGeometry** = "geometry"
- const QString **QRS::UiConstants::Settings::skState** = "state"
- const QString **QRS::UiConstants::Settings::skDockingState** = "dockingState"

### 5.10.1 Detailed Description

Common graphical constants shared between several windows.

#### Author

Pavel Lakiza

#### Date

April 2021

## 5.11 uiconstants.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef UICONSTANTS_H
00009 #define UICONSTANTS_H
00010
00011 #include <QString>
00012
00013 namespace QRS::UiConstants
00014 {
00015
00016 namespace Settings
00017 {
00018     const QString skGeometry      = "geometry";
00019     const QString skState         = "state";
00020     const QString skDockingState = "dockingState";
00021 }
00022
00023 }
00024
00025 #endif // UICONSTANTS_H
```

## 5.12 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/abstractdataobject.cpp File Reference

Implementation of the AbstractDataObject class.

```
#include "abstractdataobject.h"
```

### 5.12.1 Detailed Description

Implementation of the AbstractDataObject class.

**Author**

Pavel Lakiza

**Date**

April 2021

## 5.13 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/abstractdataobject.h File Reference

Declaration of the AbstractDataObject class.

```
#include <QObject>
#include <QString>
#include <QDataStream>
#include <map>
#include "array.h"
#include "aliasdata.h"
```



**Classes**

- class [QRS::Core::AbstractDataObject](#)  
*Data object which is designed in the way to be represented in a table easily.*

**Typedefs**

- using [QRS::Core::DataItemType](#) = [Array](#)< DataValueType >
- using [QRS::Core::DataHolder](#) = std::map< DataKeyType, [DataItemType](#) >

**Functions**

- QDataStream & [QRS::Core::operator](#)<< (QDataStream &stream, [AbstractDataObject](#) const &obj)  
*Print a data object to a stream.*

**5.13.1 Detailed Description**

Declaration of the AbstractDataObject class.

**Author**

Pavel Lakiza

**Date**

July 2021

**5.14 abstractdataobject.h**

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef ABSTRACTDATAOBJECT_H
00009 #define ABSTRACTDATAOBJECT_H
00010
00011 #include <QObject>
00012 #include <QString>
00013 #include <QDataStream>
00014 #include <map>
00015 #include "array.h"
00016 #include "aliasdata.h"
00017
00018 namespace QRS::Core
00019 {
00020
00021 using DataItemType = Array<DataValueType>;
00022 using DataHolder = std::map<DataKeyType, DataItemType>;
00023
00025 class AbstractDataObject : public QObject
00026 {
00027 public:
00028     enum ObjectType
00029     {
00030         kScalar,
00031         kVector,
00032         kMatrix,
00033         kSurface
00034     };
00035     AbstractDataObject(ObjectType type, QString const& name);
00036     virtual ~AbstractDataObject() = 0;
00037     virtual AbstractDataObject* clone() const = 0;
00038     virtual DataItemType& addItem(DataKeyType key) = 0;

```

```

00039     void removeItem(DataValueType key);
00040     bool changeItemKey(DataKeyType oldKey, DataKeyType newKey, DataHolder* items = nullptr);
00041     DataValueType getAvailableItemKey(DataValueType key, DataHolder const* items = nullptr) const;
00042     bool setArrayValue(DataKeyType key, DataValueType newValue, IndexType iRow = 0, IndexType iColumn
= 0);
00043     quint32 numberItems() const { return mItems.size(); }
00044     DataHolder const& getItems() { return mItems; }
00045     DataIDType id() const { return mID; }
00046     ObjectType type() const { return mkType; }
00047     QString const& name() const { return mName; }
00048     void setName(QString const& name) { mName = name; }
00049     static DataIDType maxObjectID() { return smMaxObjectID; }
00050     static void setMaxObjectID(DataIDType iMaxObjectID) { smMaxObjectID = iMaxObjectID; }
00051     virtual void serialize(QDataStream& stream) const;
00052     virtual void deserialize(QDataStream& stream);
00053     friend QDataStream& operator<<(QDataStream& stream, AbstractDataObject const& obj);
00054     virtual void import(QTextStream& stream) = 0;
00055
00056 protected:
00057     const ObjectType mkType;
00058     QString mName;
00059     DataIDType mID;
00060     DataHolder mItems;
00061
00062 private:
00063     static DataIDType smMaxObjectID;
00064 };
00065
00067 inline QDataStream& operator<<(QDataStream& stream, AbstractDataObject const& obj)
00068 {
00069     obj.serialize(stream);
00070     return stream;
00071 }
00072
00073 }
00074
00075 #endif // ABSTRACTDATAOBJECT_H

```

## 5.15 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/abstractrodcomponent.cpp File Reference

Definition of the AbstractRodComponent class.

```

#include "abstractrodcomponent.h"
#include "abstractdataobject.h"

```

### 5.15.1 Detailed Description

Definition of the AbstractRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.16 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractrodcomponent.h File Reference

Declaration of the AbstractRodComponent class.

```
#include <QObject>
#include <QString>
#include <QDataStream>
#include "aliasdataset.h"
```

### Classes

- class [QRS::Core::AbstractRodComponent](#)  
*Component of the rod structure which characterizes one of its properties.*

### Functions

- QDataStream & [QRS::Core::operator<<](#) (QDataStream &stream, [AbstractRodComponent](#) const &component)  
*Print a rod component to a stream.*

### 5.16.1 Detailed Description

Declaration of the AbstractRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.17 abstractrodcomponent.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef ABSTRACTRODCOMPONENT_H
00009 #define ABSTRACTRODCOMPONENT_H
00010
00011 #include <QObject>
00012 #include <QString>
00013 #include <QDataStream>
00014 #include "aliasdataset.h"
00015
00016 namespace QRS::Core
00017 {
00018
00020 class AbstractRodComponent : public QObject
00021 {
00022 public:
00023     enum ComponentType
00024     {
00025         kGeometry,
```

```

00026         kSection,
00027         kMaterial,
00028         kLoad,
00029         kConstraint,
00030         kMechanical
00031     };
00032     AbstractRodComponent(ComponentType componentType, QString const& name);
00033     virtual ~AbstractRodComponent() = 0;
00034     virtual AbstractRodComponent* clone() const = 0;
00035     virtual bool isDataComplete() const = 0;
00036     DataIDType id() const { return mID; }
00037     ComponentType componentType() const { return mkComponentType; }
00038     QString const& name() const { return mName; }
00039     void setName(QString const& name) { mName = name; }
00040     static DataIDType maxComponentID() { return smMaxComponentID; }
00041     static void setMaxComponentID(DataIDType iMaxComponentID) { smMaxComponentID = iMaxComponentID; }
00042     virtual void serialize(QDataStream& stream) const = 0;
00043     virtual void deserialize(QDataStream& stream, DataObjects const& dataObjects) = 0;
00044     friend QDataStream& operator<<(QDataStream& stream, AbstractRodComponent const& component);
00045     virtual void resolveReferences(DataObjects const& dataObjects) = 0;
00046
00047 protected:
00048     void writeDataObjectPointer(QDataStream& stream, AbstractDataObject const* pDataObject) const;
00049     AbstractDataObject const* readDataObjectPointer(QDataStream& stream, DataObjects const&
dataObjects) const;
00050     AbstractDataObject const* getDataObject(DataObjects const& dataObjects, DataIDType id) const;
00051     AbstractDataObject const* substituteDataObject(DataObjects const& dataObjects, AbstractDataObject
const* pDataObject) const;
00052
00053 protected:
00054     ComponentType const mkComponentType;
00055     QString mName;
00056     DataIDType mID;
00057
00058 private:
00059     static DataIDType smMaxComponentID;
00060 };
00061
00063 inline QDataStream& operator<<(QDataStream& stream, AbstractRodComponent const& component)
00064 {
00065     component.serialize(stream);
00066     return stream;
00067 }
00068
00069 }
00070
00071 #endif // ABSTRACTRODCOMPONENT_H

```

## 5.18 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/core/abstractsectionrodcomponent.cpp File Reference

Definition of the AbstractSectionRodComponent class.

```

#include "abstractsectionrodcomponent.h"
#include "core/scalardataobject.h"

```

### 5.18.1 Detailed Description

Definition of the AbstractSectionRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.19 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractsectionrodcomponent.h File Reference

Declaration of the AbstractSectionRodComponent class.

```
#include <QPointer>
#include "abstractrodcomponent.h"
```

### Classes

- class [QRS::Core::AbstractSectionRodComponent](#)  
*General cross section of a rod.*

### 5.19.1 Detailed Description

Declaration of the AbstractSectionRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.20 abstractsectionrodcomponent.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef ABSTRACTSECTIONRODCOMPONENT_H
00009 #define ABSTRACTSECTIONRODCOMPONENT_H
00010
00011 #include <QPointer>
00012 #include "abstractrodcomponent.h"
00013
00014 namespace QRS::Core
00015 {
00016
00017 class ScalarDataObject;
00018
00020 class AbstractSectionRodComponent : public AbstractRodComponent
00021 {
00022 public:
00023     enum SectionType
00024     {
00025         kUserDefined
00026     };
00027     AbstractSectionRodComponent(SectionType sectionType, QString const& name);
00028     virtual ~AbstractSectionRodComponent() = 0;
00029     static quint32 numberInstances() { return smNumInstances; }
00030     void serialize(QDataStream& stream) const override;
00031     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
00032     void resolveReferences(DataObjects const& dataObjects) override;
00033     SectionType sectionType() const { return mkSectionType; }
00034
00035 protected:
00036     void copyIntegratedProperties(AbstractSectionRodComponent const* pSection);
00037
00038 protected:
00039     // Info
```

```

00040     SectionType const mkSectionType;
00041     static quint32 smNumInstances;
00042     // Area
00043     QPointer<ScalarDataObject const> mpArea;
00044     // Inertia moments
00045     QPointer<ScalarDataObject const> mpInertiaMomentTorsional;
00046     QPointer<ScalarDataObject const> mpInertiaMomentX;
00047     QPointer<ScalarDataObject const> mpInertiaMomentY;
00048     // Center coordinates
00049     QPointer<ScalarDataObject const> mpCenterCoordinateX;
00050     QPointer<ScalarDataObject const> mpCenterCoordinateY;
00051 };
00052
00053 }
00054
00055 #endif // ABSTRACTSECTIONRODCOMPONENT_H

```

## 5.21 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/core/aliasdata.h File Reference

Specification of data types used in a project.

```
#include <QtGlobal>
```

### Typedefs

- using **QRS::Core::DataValueType** = double
- using **QRS::Core::DataKeyType** = double
- using **QRS::Core::DataIDType** = quint64

### 5.21.1 Detailed Description

Specification of data types used in a project.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.22 aliasdata.h

[Go to the documentation of this file.](#)

```

00001
00002 #ifndef ALIASDATA_H
00003 #define ALIASDATA_H
00004
00005 #include <QtGlobal>
00006
00007 namespace QRS::Core
00008 {
00009
00010     using DataValueType = double;
00011     using DataKeyType = double;
00012     using DataIDType = quint64;
00013
00014 }
00015
00016 #endif // ALIASDATA_H

```

## 5.23 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/aliasdataset.h File Reference

Specification of types of datasets used in a project.

```
#include <unordered_map>
#include "aliasdata.h"
```

### Typedefs

- using **QRS::Core::DataObjects** = std::unordered\_map< DataIDType, [AbstractDataObject](#) \* >
- using **QRS::Core::RodComponents** = std::unordered\_map< DataIDType, [AbstractRodComponent](#) \* >

### 5.23.1 Detailed Description

Specification of types of datasets used in a project.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.24 aliasdataset.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef ALIASDATASET_H
00009 #define ALIASDATASET_H
00010
00011 #include <unordered_map>
00012 #include "aliasdata.h"
00013
00014 namespace QRS::Core
00015 {
00016
00017 class AbstractDataObject;
00018 class AbstractRodComponent;
00019
00020 using DataObjects = std::unordered_map<DataIDType, AbstractDataObject*>;
00021 using RodComponents = std::unordered_map<DataIDType, AbstractRodComponent*>;
00022
00023 }
00024
00025 #endif // ALIASDATASET_H
```

## 5.25 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.cpp File Reference

Implementation of the Array class.

```
#include "array.h"
```

### 5.25.1 Detailed Description

Implementation of the Array class.

Author

Pavel Lakiza

Date

March 2021

## 5.26 /home/qinterfly/Library/Projects/Current/QRod← Systems/src/core/array.h File Reference

Declaration of the Array class.

```
#include <QDebug>
```

### Classes

- class `QRS::Core::Array< T >`  
*Numerical array class.*
- struct `QRS::Core::Array< T >::Row< U >`  
*Proxy class to acquire a row by index.*

### Typedefs

- using `QRS::Core::IndexType` = quint32

### Functions

- template<typename K >  
`QDebug QRS::Core::operator<< (QDebug stream, Array< K > &array)`  
*Print all array values using the matrix format.*
- template<typename K >  
`QDataStream & QRS::Core::operator<< (QDataStream &stream, Array< K > const &array)`  
*Write an array to a stream.*
- template<typename K >  
`QDataStream & QRS::Core::operator>> (QDataStream &stream, Array< K > &array)`  
*Read an array from a stream.*



### 5.26.1 Detailed Description

Declaration of the Array class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.27 array.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef ARRAY_H
00009 #define ARRAY_H
00010
00011 #include <QDebug>
00012
00013 namespace QRS::Core
00014 {
00015
00016 using IndexType = quint32;
00017
00018 template<typename T>
00019 class Array
00020 {
00021 private:
00022     template <typename U> struct Row;
00023 public:
00024     Array(IndexType numRows = 0, IndexType numCols = 0);
00025     Array(Array<T> const& another);
00026     Array(Array<T>&& another);
00027     ~Array();
00028     T* data() { return mpData; }
00029     void resize(IndexType numRows, IndexType numCols);
00030     void removeColumn(IndexType iRemoveColumn);
00031     void swapColumns(IndexType iFirstColumn, IndexType iSecondColumn);
00032     IndexType rows() const { return mNumRows; };
00033     IndexType cols() const { return mNumCols; };
00034     IndexType size() const { return mNumRows * mNumCols; }
00035     Row<T> operator[] (IndexType iRow) { return Row<T>(&mpData[mNumCols * iRow]); };
00036     Row<T> operator[] (IndexType iRow) const { return Row<T>(&mpData[mNumCols * iRow]); };
00037     Array& operator=(Array<T> const& another);
00038     template<typename K> friend QDebug operator<<(QDebug stream, Array<K>& array);
00039     template<typename K> friend QDataStream& operator<<(QDataStream& stream, Array<K> const& array);
00040     template<typename K> friend QDataStream& operator>>(QDataStream& stream, Array<K>& array);
00041 private:
00042     IndexType mNumRows;
00043     IndexType mNumCols;
00044     T* mpData = nullptr;
00045     template <typename U>
00046     struct Row
00047     {
00048         Row() = delete;
00049         Row(T* pData) : pRow(pData) { };
00050         ~Row() { }
00051         T& operator[] (IndexType iCol) { return pRow[iCol]; }
00052         T const& operator[] (IndexType iCol) const { return pRow[iCol]; }
00053         T* pRow;
00054     };
00055 };
00056
00057 template<typename K>
00058 inline QDebug operator<<(QDebug stream, Array<K>& array)
00059 {
00060     IndexType const& nRows = array.mNumRows;
00061     IndexType const& nCols = array.mNumCols;
00062     stream = stream.noquote();
00063     stream << QString("Array size: %1 x %2").arg(QString::number(nRows), QString::number(nCols));

```

```

00072     stream << Qt::endl;
00073     for (IndexType iRow = 0; iRow != nRows; ++iRow)
00074     {
00075         for (IndexType jCol = 0; jCol != nCols; ++jCol)
00076             stream << QString::number(array[iRow][jCol]);
00077         stream << Qt::endl;
00078     }
00079     return stream;
00080 }
00081
00082 template<typename K>
00083 inline QDataStream& operator<<(QDataStream& stream, Array<K> const& array)
00084 {
00085     stream << array.mNumRows << array.mNumCols;
00086     IndexType const& size = array.size();
00087     for (IndexType i = 0; i != size; ++i)
00088         stream << array.mpData[i];
00089     return stream;
00090 }
00091
00092 template<typename K>
00093 inline QDataStream& operator>>(QDataStream& stream, Array<K>& array)
00094 {
00095     delete[] array.mpData;
00096     stream >> array.mNumRows >> array.mNumCols;
00097     IndexType const& size = array.size();
00098     array.mpData = new K[size];
00099     for (IndexType i = 0; i != size; ++i)
00100         stream >> array.mpData[i];
00101     return stream;
00102 }
00103
00104 }
00105
00106 }
00107
00108 #endif // ARRAY_H

```

## 5.28 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/core/constraintrodcomponent.cpp File Reference

Definition of the ConstraintRodComponent class.

```
#include "constraintrodcomponent.h"
```

### 5.28.1 Detailed Description

Definition of the ConstraintRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.29 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/core/constraintrodcomponent.h File Reference

Declaration of the ConstraintRodComponent class.

```
#include "abstractrodcomponent.h"
```

**Classes**

- class [QRS::Core::ConstraintRodComponent](#)  
*Component to restrict movements of a rod.*

**5.29.1 Detailed Description**

Declaration of the ConstraintRodComponent class.

**Author**

Pavel Lakiza

**Date**

July 2021

**5.30 constraintrodcomponent.h**

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef CONSTRAINTRODCOMPONENT_H
00009 #define CONSTRAINTRODCOMPONENT_H
00010
00011 #include "abstractrodcomponent.h"
00012
00013 namespace QRS::Core
00014 {
00015
00017 class ConstraintRodComponent : public AbstractRodComponent
00018 {
00019 public:
00020     enum ConstraintType
00021     {
00022         kDisplacementX, kDisplacementY, kDisplacementZ,
00023         kRotationX, kRotationY, kRotationZ
00024     };
00025     enum ConstraintCoordinateSystem
00026     {
00027         kGlobal,
00028         kLocal
00029     };
00030     using Constraints = std::map<ConstraintType, ConstraintCoordinateSystem>;
00031     ConstraintRodComponent(QString const& name);
00032     ~ConstraintRodComponent();
00033     AbstractRodComponent* clone() const override;
00034     bool isDataComplete() const override { return mConstraints.size() != 0; };
00035     static quint32 numberInstances() { return smNumInstances; }
00036     void serialize(QDataStream& stream) const override;
00037     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
00038     void resolveReferences(DataObjects const&) override {};
00039     bool isConstraintExist(ConstraintType type) const;
00040     void setConstraint(ConstraintType type, ConstraintCoordinateSystem coordinateSystem);
00041     bool removeConstraint(ConstraintType type);
00042     Constraints const& constraints() const { return mConstraints; }
00043
00044 private:
00045     static quint32 smNumInstances;
00046     Constraints mConstraints;
00047 };
00048
00049 }
00050
00051
00052 #endif // CONSTRAINTRODCOMPONENT_H

```

## 5.31 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/core/geometryrodcomponent.cpp File Reference

Definition of the GeometryRodComponent class.

```
#include "geometryrodcomponent.h"  
#include "vectordataobject.h"  
#include "matrixdataobject.h"
```

### 5.31.1 Detailed Description

Definition of the GeometryRodComponent class.

Author

Pavel Lakiza

Date

July 2021

## 5.32 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/core/geometryrodcomponent.h File Reference

Declaration of the GeometryRodComponent class.

```
#include <QPointer>  
#include "abstractrodcomponent.h"
```

### Classes

- class [QRS::Core::GeometryRodComponent](#)  
*Geometrical configuration of a rod.*

### 5.32.1 Detailed Description

Declaration of the GeometryRodComponent class.

Author

Pavel Lakiza

Date

July 2021

## 5.33 geometryrodcomponent.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef GEOMETRYRODCOMPONENT_H
00009 #define GEOMETRYRODCOMPONENT_H
00010
00011 #include <QPointer>
00012 #include "abstractrodcomponent.h"
00013
00014 namespace QRS::Core
00015 {
00016
00017 class VectorDataObject;
00018 class MatrixDataObject;
00019
00021 class GeometryRodComponent : public AbstractRodComponent
00022 {
00023 public:
00024     GeometryRodComponent(QString const& name);
00025     ~GeometryRodComponent();
00026     AbstractRodComponent* clone() const override;
00027     bool isDataComplete() const override;
00028     static quint32 numberInstances() { return smNumInstances; }
00029     void serialize(QDataStream& stream) const override;
00030     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
00031     void resolveReferences(DataObjects const& dataObjects) override;
00032     // Getters
00033     VectorDataObject const* radiusVector() const { return mpRadiusVector; }
00034     MatrixDataObject const* rotationMatrix() const { return mpRotationMatrix; }
00035     // Setters
00036     void setRadiusVector(VectorDataObject const* pRadiusVector) { mpRadiusVector = pRadiusVector; }
00037     void setRotationMatrix(MatrixDataObject const* pRotationMatrix) { mpRotationMatrix =
00038         pRotationMatrix; }
00039 private:
00040     static quint32 smNumInstances;
00041     QPointer<VectorDataObject const> mpRadiusVector;
00042     QPointer<MatrixDataObject const> mpRotationMatrix;
00043 };
00044
00045 }
00046
00047 #endif // GEOMETRYRODCOMPONENT_H

```

## 5.34 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/hierarchynode.cpp File Reference

Implementation of the HierarchyNode class.

```
#include "hierarchynode.h"
```

### 5.34.1 Detailed Description

Implementation of the HierarchyNode class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.35 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/core/hierarchynode.h File Reference

Declaration of the HierarchyNode class.

```
#include <QVariant>
#include <QDataStream>
```

### Classes

- class [QRS::Core::HierarchyNode](#)  
*Hierarchy representative.*

### 5.35.1 Detailed Description

Declaration of the HierarchyNode class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.36 hierarchynode.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef HIERARCHYNODE_H
00009 #define HIERARCHYNODE_H
00010
00011 #include <QVariant>
00012 #include <QDataStream>
00013
00014 namespace QRS::Core
00015 {
00016
00018 class HierarchyNode
00019 {
00020
00021 public:
00022     friend class HierarchyTree;
00023     enum NodeType
00024     {
00025         kObject,
00026         kDirectory
00027     };
00028     HierarchyNode(NodeType type, QVariant value);
00029     ~HierarchyNode() = default;
00030     void appendChild(HierarchyNode* node);
00031     bool hasParent() const { return mpParent; }
00032     bool hasChild() const { return mpFirstChild; }
00033     bool hasNextSibling() const { return mpNextSibling; }
00034     HierarchyNode* parent() { return mpParent; }
00035     HierarchyNode* firstChild() { return mpFirstChild; }
00036     HierarchyNode* nextSibling() { return mpNextSibling; }
00037     NodeType type() const { return mType; }
00038     QVariant& value() { return mValue; }
00039     HierarchyNode* groupNodes(HierarchyNode* pChildNode);
```

```

00040     bool setBefore(HierarchyNode* pSetNode);
00041     bool setAfter(HierarchyNode* pSetNode);
00042     quint32 numberChildren() const;
00043
00044 private:
00045     void excludeNodeFromHierarchy();
00046     bool isSetAllowed(HierarchyNode const* pNode) const;
00047     bool isParentOf(HierarchyNode const* pNode) const;
00048     quint32 countNodes(HierarchyNode* pNode, quint32& numNodes) const;
00049
00050 private:
00051     HierarchyNode* mpParent = nullptr;
00052     HierarchyNode* mpFirstChild = nullptr;
00053     HierarchyNode* mpNextSibling = nullptr;
00054     HierarchyNode* mpPreviousSibling = nullptr;
00055     NodeType mType;
00056     QVariant mValue;
00057 };
00058
00059 }
00060
00061 #endif // HIERARCHYNODE_H

```

## 5.37 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchytree.cpp File Reference

Implementation of the HierarchyTree class.

```
#include "hierarchytree.h"
```

### 5.37.1 Detailed Description

Implementation of the HierarchyTree class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.38 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/hierarchytree.h File Reference

Declaration of the HierarchyTree class.

```
#include <QDebug>
#include "hierarchynode.h"
```

#### Classes

- class [QRS::Core::HierarchyTree](#)  
*Hierarchy of data objects (n-array tree)*

## Functions

- QDebug **QRS::Core::operator<<** (QDebug stream, [HierarchyTree](#) &tree)  
*Print a tree structure.*
- QDataStream & **QRS::Core::operator<<** (QDataStream &stream, [HierarchyTree](#) const &tree)  
*Write a tree structure to a stream.*

### 5.38.1 Detailed Description

Declaration of the HierarchyTree class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.39 hierarchytree.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef HIERARCHYTREE_H
00009 #define HIERARCHYTREE_H
00010
00011 #include <QDebug>
00012 #include "hierarchytree.h"
00013
00014 namespace QRS::Core
00015 {
00016
00017     class HierarchyTree
00018     {
00019     public:
00020         HierarchyTree();
00021         HierarchyTree(HierarchyTree& another);
00022         HierarchyTree(HierarchyTree&& another);
00023         HierarchyTree(HierarchyNode* pRootNode);
00024         HierarchyTree(QDataStream& stream, int numNodes);
00025         HierarchyTree& operator=(HierarchyTree const& another);
00026         HierarchyTree& operator=(HierarchyTree&& another);
00027         ~HierarchyTree();
00028         void clear();
00029         void appendNode(HierarchyNode* pNode);
00030         bool removeNode(HierarchyNode::NodeType type, QVariant const& value);
00031         void removeNode(HierarchyNode* pNode);
00032         void changeNodeValue(HierarchyNode::NodeType type, QVariant const& oldValue, QVariant const&
00033 new value);
00034         HierarchyNode* root() { return mpRootNode; }
00035         HierarchyTree clone() const;
00036         HierarchyNode* findNode(HierarchyNode* pBaseNode, HierarchyNode::NodeType type, QVariant const&
00037 value) const;
00038         quint32 size() const;
00039         friend QDebug operator<<(QDebug stream, HierarchyTree& tree);
00040         friend QDataStream& operator<<(QDataStream& stream, HierarchyTree const& tree);
00041     private:
00042         HierarchyNode* copyNode(HierarchyNode* pBaseNode, quint32 relativeLevel) const;
00043         void removeNodeSiblings(HierarchyNode* pNode);
00044         void printNode(quint32 level, HierarchyNode* pNode, QDebug stream) const;
00045         void writeNode(HierarchyNode* pNode, QDataStream& stream) const;
00046     private:
00047         HierarchyNode* mpRootNode = nullptr;
00048     };
00049
00050
00052 inline QDebug operator<<(QDebug stream, HierarchyTree& tree)
```



```

00053 {
00054     tree.printNode(0, tree.mpRootNode, stream);
00055     return stream;
00056 }
00057
00059 inline QDataStream& operator<<(QDataStream& stream, HierarchyTree const& tree)
00060 {
00061     tree.writeNode(tree.mpRootNode, stream);
00062     return stream;
00063 }
00064
00065 }
00066
00067 #endif // HIERARCHYTREE_H

```

## 5.40 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.cpp File Reference

Definition of the LoadRodComponent class.

```

#include "loadrodcomponent.h"
#include "scalardataobject.h"
#include "vectordataobject.h"

```

### 5.40.1 Detailed Description

Definition of the LoadRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.41 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.h File Reference

Declaration of the LoadRodComponent class.

```

#include <QPointer>
#include "abstractrodcomponent.h"

```

#### Classes

- class [QRS::Core::LoadRodComponent](#)  
*Load applied to a rod.*

### 5.41.1 Detailed Description

Declaration of the LoadRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.42 loadrodcomponent.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef LOADRODCOMPONENT_H
00009 #define LOADRODCOMPONENT_H
00010
00011 #include <QPointer>
00012 #include "abstractrodcomponent.h"
00013
00014 namespace QRS::Core
00015 {
00016
00017 class ScalarDataObject;
00018 class VectorDataObject;
00019
00021 class LoadRodComponent : public AbstractRodComponent
00022 {
00023 public:
00024     enum LoadType
00025     {
00026         kNone,
00027         kForcedDisplacements, kForcedRotations,
00028         kPointForce, kPointMoment,
00029         kPointMass, kPointInertiaMoment,
00030         kPointLinearDamper, kPointRotationalDamper,
00031         kDistributedForce, kDistributedMoment,
00032         kAerodynamicFlow,
00033         kAcceleration,
00034         kInnerLiquidFlow,
00035         kDisplacementDamping, kRotationDamping
00036     };
00037     LoadRodComponent(QString const& name);
00038     ~LoadRodComponent();
00039     AbstractRodComponent* clone() const override;
00040     bool isDataComplete() const override;
00041     static quint32 numberInstances() { return smNumInstances; }
00042     void serialize(QDataStream& stream) const override;
00043     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
00044     void resolveReferences(DataObjects const& dataObjects) override;
00045     // Getters
00046     LoadType loadType() const { return mLoadType; }
00047     VectorDataObject const* directionVector() const { return mpDirectionVector; }
00048     ScalarDataObject const* longitudinalFunction() const { return mpLongitudinalFunction; }
00049     ScalarDataObject const* timeCoefficient() const { return mpTimeCoefficient; }
00050     VectorDataObject const* timeRotationVector() const { return mpTimeRotationVector; }
00051     DataValueType multiplier() const { return mMultiplier; }
00052     bool isFollowing() const { return mIsFollowing; }
00053     // Setters
00054     void setType(LoadType type) { mLoadType = type; }
00055     void setDirectionVector(VectorDataObject const* pDirectionVector) { mpDirectionVector =
pDirectionVector; }
00056     void setLongitudinalFunction(ScalarDataObject const* pLongitudinalFunction) {
mpLongitudinalFunction = pLongitudinalFunction; }
00057     void setTimeCoefficient(ScalarDataObject const* pTimeCoefficient) { mpTimeCoefficient =
pTimeCoefficient; }
00058     void setTimeRotationVector(VectorDataObject const* pTimeRotationVector) { mpTimeRotationVector =
pTimeRotationVector; }
00059     void setMultiplier(DataValueType value) { mMultiplier = value; }
00060     void setFollowingState(bool isFollowing) { mIsFollowing = isFollowing; }
00061
00062 private:

```

```

00063     static quint32 smNumInstances;
00064     LoadType mLoadType = kNone;
00065     QPointer<VectorDataObject const> mpDirectionVector;
00066     QPointer<ScalarDataObject const> mpLongitudinalFunction;
00067     QPointer<ScalarDataObject const> mpTimeCoefficient;
00068     QPointer<VectorDataObject const> mpTimeRotationVector;
00069     DataValueType mMultiplier = 1.0;
00070     bool mIsFollowing = false;
00071 };
00072
00073 }
00074
00075 #endif // LOADRODCOMPONENT_H

```

## 5.43 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.cpp File Reference

Definition of the MaterialRodComponent class.

```

#include "materialrodcomponent.h"
#include "scalardataobject.h"

```

### 5.43.1 Detailed Description

Definition of the MaterialRodComponent class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.44 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.h File Reference

Declaration of the MaterialRodComponent class.

```

#include <QPointer>
#include "abstractrodcomponent.h"

```

### Classes

- class [QRS::Core::MaterialRodComponent](#)  
*Material properties of a rod.*

### 5.44.1 Detailed Description

Declaration of the MaterialRodComponent class.

Author

Pavel Lakiza

Date

July 2021

## 5.45 materialrodcomponent.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef MATERIALRODCOMPONENT_H
00009 #define MATERIALRODCOMPONENT_H
00010
00011 #include <QPointer>
00012 #include "abstractrodcomponent.h"
00013
00014 namespace QRS::Core
00015 {
00016
00017 class ScalarDataObject;
00018
00020 class MaterialRodComponent : public AbstractRodComponent
00021 {
00022 public:
00023     MaterialRodComponent(QString const& name);
00024     ~MaterialRodComponent();
00025     AbstractRodComponent* clone() const override;
00026     bool isDataComplete() const override;
00027     static quint32 numberInstances() { return smNumInstances; }
00028     void serialize(QDataStream& stream) const override;
00029     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
00030     void resolveReferences(DataObjects const& dataObjects) override;
00031     // Getters
00032     ScalarDataObject const* elasticModulus() const { return mpElasticModulus; }
00033     ScalarDataObject const* shearModulus() const { return mpShearModulus; }
00034     ScalarDataObject const* poissonsRatio() const { return mpPoissonsRatio; }
00035     ScalarDataObject const* density() const { return mpDensity; }
00036     // Setters
00037     void setElasticModulus(ScalarDataObject const* pElasticModulus) { mpElasticModulus =
00038 pElasticModulus; }
00038     void setShearModulus(ScalarDataObject const* pShearModulus) { mpShearModulus = pShearModulus; }
00039     void setPoissonsRatio(ScalarDataObject const* pPoissonsRatio) { mpPoissonsRatio = pPoissonsRatio; }
00040 }
00040     void setDensity(ScalarDataObject const* pDensity) { mpDensity = pDensity; }
00041
00042 private:
00043     static quint32 smNumInstances;
00044     QPointer<ScalarDataObject const> mpElasticModulus;
00045     QPointer<ScalarDataObject const> mpShearModulus;
00046     QPointer<ScalarDataObject const> mpPoissonsRatio;
00047     QPointer<ScalarDataObject const> mpDensity;
00048 };
00049
00050 }
00051
00052 #endif // MATERIALRODCOMPONENT_H
```

## 5.46 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/matrixdataobject.cpp File Reference

Implementation of the MatrixDataObject class.

```
#include "matrixdataobject.h"
```

## Variables

- const IndexType **skNumElements** = 3

### 5.46.1 Detailed Description

Implementation of the MatrixDataObject class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.47 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.h File Reference

Declaration of the MatrixDataObject class.

```
#include "abstractdataobject.h"
```

## Classes

- class [QRS::Core::MatrixDataObject](#)  
*Matrix data object.*

### 5.47.1 Detailed Description

Declaration of the MatrixDataObject class.

#### Author

Pavel Lakiza

#### Date

April 2021

## 5.48 matrixdataobject.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef MATRIXDATAOBJECT_H
00009 #define MATRIXDATAOBJECT_H
00010
00011 #include "abstractdataobject.h"
00012
00013 namespace QRS::Core
00014 {
00015
00017 class MatrixDataObject : public AbstractDataObject
00018 {
00019 public:
00020     MatrixDataObject(QString const& name);
00021     ~MatrixDataObject();
00022     AbstractDataObject* clone() const override;
00023     DataItemType& addItem(DataValueType key) override;
00024     static quint32 numberInstances() { return smNumInstances; }
00025     virtual void import(QTextStream& stream) override;
00026
00027 private:
00028     static quint32 smNumInstances;
00029 };
00030
00031 }
00032
00033 #endif // MATRIXDATAOBJECT_H

```

## 5.49 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/mechanicalrodcomponent.cpp File Reference

Definition of the MechanicalRodComponent class.

```

#include "mechanicalrodcomponent.h"
#include "scalardataobject.h"

```

### 5.49.1 Detailed Description

Definition of the MechanicalRodComponent class.

**Author**

Pavel Lakiza

**Date**

July 2021

## 5.50 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/mechanicalrodcomponent.h File Reference

Declaration of the MechanicalRodComponent class.

```

#include <QPointer>
#include "abstractrodcomponent.h"

```

**Classes**

- class [QRS::Core::MechanicalRodComponent](#)  
*Stiffness and mass distributions of a rod.*

**5.50.1 Detailed Description**

Declaration of the MechanicalRodComponent class.

**Author**

Pavel Lakiza

**Date**

July 2021

**5.51 mechanicalrodcomponent.h**

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef MECHANICALRODCOMPONENT_H
00009 #define MECHANICALRODCOMPONENT_H
00010
00011 #include <QPointer>
00012 #include "abstractrodcomponent.h"
00013
00014 namespace QRS::Core
00015 {
00016
00017 class ScalarDataObject;
00018
00020 class MechanicalRodComponent : public AbstractRodComponent
00021 {
00022 public:
00023     MechanicalRodComponent(QString const& name);
00024     ~MechanicalRodComponent();
00025     AbstractRodComponent* clone() const override;
00026     bool isDataComplete() const override { return true; }
00027     static quint32 numberInstances() { return smNumInstances; }
00028     void serialize(QDataStream& stream) const override;
00029     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
00030     void resolveReferences(DataObjects const& dataObjects) override;
00031     // Getters
00032     // Stiffness distribution
00033     ScalarDataObject const* tensionStiffness() const { return mpTensionStiffness; }
00034     ScalarDataObject const* torsionalStiffness() const { return mpTorsionalStiffness; }
00035     ScalarDataObject const* bendingStiffnessX() const { return mpBendingStiffnessX; }
00036     ScalarDataObject const* bendingStiffnessY() const { return mpBendingStiffnessY; }
00037     // Mass distribution
00038     ScalarDataObject const* linearMassDensity() const { return mpLinearMassDensity; }
00039     ScalarDataObject const* inertiaMassMomentX() const { return mpInertiaMassMomentX; }
00040     ScalarDataObject const* inertiaMassMomentY() const { return mpInertiaMassMomentY; }
00041     ScalarDataObject const* inertiaMassMomentZ() const { return mpInertiaMassMomentZ; }
00042     // Eccentricity
00043     ScalarDataObject const* eccentricityX() const { return mpEccentricityX; }
00044     ScalarDataObject const* eccentricityY() const { return mpEccentricityY; }
00045     // Contact diameter
00046     ScalarDataObject const* contactDiameter() const { return mpContactDiameter; }
00047     // Setters
00048     // Stiffness distribution
00049     void setTensionStiffness(ScalarDataObject const* pTensionStiffness) { mpTensionStiffness =
pTensionStiffness; }
00050     void setTorsionalStiffness(ScalarDataObject const* pTorsionalStiffness) { mpTorsionalStiffness =
pTorsionalStiffness; }
00051     void setBendingStiffnessX(ScalarDataObject const* pBendingStiffnessX) { mpBendingStiffnessX =
pBendingStiffnessX; }
00052     void setBendingStiffnessY(ScalarDataObject const* pBendingStiffnessY) { mpBendingStiffnessY =
pBendingStiffnessY; }
00053     // Mass distribution

```

```

00054     void setLinearMassDensity(ScalarDataObject const* pLinearMassDensity) { mpLinearMassDensity =
pLinearMassDensity; }
00055     void setInertiaMassMomentX(ScalarDataObject const* pInertiaMassMomentX) { mpInertiaMassMomentX =
pInertiaMassMomentX; }
00056     void setInertiaMassMomentY(ScalarDataObject const* pInertiaMassMomentY) { mpInertiaMassMomentY =
pInertiaMassMomentY; }
00057     void setInertiaMassMomentZ(ScalarDataObject const* pInertiaMassMomentZ) { mpInertiaMassMomentZ =
pInertiaMassMomentZ; }
00058     // Eccentricity
00059     void setEccentricityX(ScalarDataObject const* pEccentricityX) { mpEccentricityX = pEccentricityX;
}
00060     void setEccentricityY(ScalarDataObject const* pEccentricityY) { mpEccentricityY = pEccentricityY;
}
00061     // Contact diameter
00062     void setContactDiameter(ScalarDataObject const* pContactDiameter) { mpContactDiameter =
pContactDiameter; }
00063
00064 private:
00065     static quint32 smNumInstances;
00066     // Stiffness distribution
00067     QPointer<ScalarDataObject const> mpTensionStiffness;
00068     QPointer<ScalarDataObject const> mpTorsionalStiffness;
00069     QPointer<ScalarDataObject const> mpBendingStiffnessX;
00070     QPointer<ScalarDataObject const> mpBendingStiffnessY;
00071     // Mass distribution
00072     QPointer<ScalarDataObject const> mpLinearMassDensity;
00073     QPointer<ScalarDataObject const> mpInertiaMassMomentX;
00074     QPointer<ScalarDataObject const> mpInertiaMassMomentY;
00075     QPointer<ScalarDataObject const> mpInertiaMassMomentZ;
00076     // Eccentricity
00077     QPointer<ScalarDataObject const> mpEccentricityX;
00078     QPointer<ScalarDataObject const> mpEccentricityY;
00079     // Contact diameter
00080     QPointer<ScalarDataObject const> mpContactDiameter;
00081 };
00082
00083 }
00084
00085 #endif // MECHANICALRODCOMPONENT_H

```

## 5.52 /home/qinterfly/Library/Projects/Current/QRod← Systems/src/core/project-base.cpp File Reference

Implementation of the Project class.

```

#include <QRandomGenerator>
#include "project.h"
#include "scalarsdataobject.h"
#include "vectordataobject.h"
#include "matrixdataobject.h"
#include "surfacedataobject.h"
#include "geometryrodcomponent.h"
#include "usersectionrodcomponent.h"
#include "materialrodcomponent.h"
#include "loadrodcomponent.h"
#include "constraintrodcomponent.h"
#include "mechanicalrodcomponent.h"

```

### Functions

- `template<typename T >`  
`void clearDataMap (std::unordered_map< DataIDType, T * > &dataMap)`  
*Helper function to clear a map consisted of data pointers.*
- `AbstractDataObject * createDataObject (AbstractDataObject::ObjectType type)`  
*Helper function to create DataObject instance by a type and name.*



### 5.52.1 Detailed Description

Implementation of the Project class.

#### Author

Pavel Lakiza

#### Date

June 2021

Implementation of the methods to operate with data objects, components and rods

## 5.53 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/project-io.cpp File Reference

Implementation of the Project class.

```
#include <QFileInfo>
#include <QDir>
#include <QDataStream>
#include <QDateTime>
#include "project.h"
#include "scalardataobject.h"
#include "vectordataobject.h"
#include "matrixdataobject.h"
#include "surfacedataobject.h"
#include "geometryrodcomponent.h"
#include "usersectionrodcomponent.h"
#include "materialrodcomponent.h"
#include "loadrodcomponent.h"
#include "constraintrodcomponent.h"
#include "mechanicalrodcomponent.h"
#include "utilities.h"
```

### Functions

- void **readDataObjects** (QDataStream &inputStream, DataObjects &dataObjects)  
*Helper function to read a set of data objects from a stream.*
- void **readRodComponents** (QDataStream &inputStream, DataObjects const &dataObjects, RodComponents &rodComponents)  
*Helper function to read rod components from a stream.*
- void **readHierarchyTree** (QDataStream &inputStream, [HierarchyTree](#) &hierarchy)  
*Helper function to read a hierarchial tree from a stream.*

### 5.53.1 Detailed Description

Implementation of the Project class.

Author

Pavel Lakiza

Date

June 2021

Implementation of the methods to operate with input/output streams

## 5.54 [/home/qinterfly/Library/Projects/Current/QRod](#)↵ Systems/src/core/project.h File Reference

Declaration of the Project class.

```
#include <QObject>
#include "aliasdataset.h"
#include "array.h"
#include "hierarchytree.h"
#include "abstractdataobject.h"
#include "abstractrodcomponent.h"
#include "abstractsectionrodcomponent.h"
```

### Classes

- class [QRS::Core::Project](#)  
*Project class to interact with a created system of rods.*

### 5.54.1 Detailed Description

Declaration of the Project class.

Author

Pavel Lakiza

Date

June 2021

## 5.55 project.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef PROJECT_H
00009 #define PROJECT_H
00010
00011 #include <QObject>
00012 #include "aliasdataset.h"
00013 #include "array.h"
00014 #include "hierarchytree.h"
00015 #include "abstractdataobject.h"
00016 #include "abstractrodcomponent.h"
00017 #include "abstractsectionrodcomponent.h"
00018
00019 QT_BEGIN_NAMESPACE
00020 class QString;
00021 QT_END_NAMESPACE
00022
00023 namespace QRS::HierarchyModels
00024 {
00025     class ProjectHierarchyModel;
00026 }
00027
00028 namespace QRS::Managers
00029 {
00030     class ManagersFactory;
00031 }
00032
00033 namespace QRS::Core
00034 {
00035
00037     class Project : public QObject
00038     {
00039     Q_OBJECT
00040
00041     friend class QRS::HierarchyModels::ProjectHierarchyModel;
00042     friend class QRS::Managers::ManagersFactory;
00043
00044     public:
00045         Project(QString const& name);
00046         Project(QString const& path, QString const& fileName);
00047         virtual ~Project();
00048         // Data objects
00049         DataIDType numberDataObjects() const { return mDataObjects.size(); }
00050         AbstractDataObject* addDataObject(AbstractDataObject::ObjectType type);
00051         DataObjects cloneDataObjects() const;
00052         HierarchyTree cloneHierarchyDataObjects() const { return mHierarchyDataObjects.clone(); }
00053         // Rod components
00054         DataIDType numberRodComponents() const { return mRodComponents.size(); }
00055         AbstractRodComponent* addGeometry();
00056         AbstractRodComponent* addCrossSection(AbstractSectionRodComponent::SectionType sectionType);
00057         AbstractRodComponent* addMaterial();
00058         AbstractRodComponent* addLoad();
00059         AbstractRodComponent* addConstraint();
00060         AbstractRodComponent* addMechanical();
00061         RodComponents cloneRodComponents() const;
00062         HierarchyTree cloneHierarchyRodComponents() const { return mHierarchyRodComponents.clone(); }
00063         // Getters and setters
00064         QString const& name() const { return mName; }
00065         QString const& filePath() const { return mFilePath; }
00066         static QString const& getFileNameExtension() { return skProjectExtension; }
00067         void importDataObjects(QString const& path, QString const& fileName);
00068
00069     signals:
00070         // Data objects
00071         void dataObjectsSubstituted();
00072         void propertiesDataObjectsChanged();
00073         // Rod components
00074         void rodComponentsSubstituted();
00075         void propertiesRodComponentsChanged();
00076         // Project hierarchy
00077         void projectHierarchyChanged();
00078
00079     public slots:
00080         bool save(QString const& dir, QString const& fileName);
00081         void setDataObjects(QRS::Core::DataObjects const& dataObjects, QRS::Core::HierarchyTree const&
            hierarchyDataObjects);
00082         void setRodComponents(QRS::Core::RodComponents const& rodComponents, QRS::Core::HierarchyTree
            const& hierarchyRodComponents);
00083
00084     private:
00085         void emplaceRodComponent(AbstractRodComponent* pRodComponent);
00086
00087     private:

```

```

00089     quint32 mID;
00091     QString mName;
00093     QString mFilePath;
00095     DataObjects mDataObjects;
00097     HierarchyTree mHierarchyDataObjects;
00099     RodComponents mRodComponents;
00101     HierarchyTree mHierarchyRodComponents;
00103     static const QString skProjectExtension;
00104 };
00105
00106 }
00107
00108 #endif // PROJECT_H

```

## 5.56 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/scalardataobject.cpp File Reference

Implementation of the ScalarDataObject class.

```
#include "scalardataobject.h"
```

### 5.56.1 Detailed Description

Implementation of the ScalarDataObject class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.57 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/scalardataobject.h File Reference

Declaration of the ScalarDataObject class.

```
#include "abstractdataobject.h"
```

### Classes

- class [QRS::Core::ScalarDataObject](#)  
*Scalar data object.*

### 5.57.1 Detailed Description

Declaration of the ScalarDataObject class.

#### Author

Pavel Lakiza

#### Date

April 2021

## 5.58 scalardataobject.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef SCALARDATAOBJECT_H
00009 #define SCALARDATAOBJECT_H
00010
00011 #include "abstractdataobject.h"
00012
00013 namespace QRS::Core
00014 {
00015
00017 class ScalarDataObject : public AbstractDataObject
00018 {
00019 public:
00020     ScalarDataObject(QString const& name);
00021     ~ScalarDataObject();
00022     AbstractDataObject* clone() const override;
00023     DataItemType& addItem(DataValueType key) override;
00024     static quint32 numberInstances() { return smNumInstances; }
00025     virtual void import(QTextStream& stream) override;
00026
00027 private:
00028     static quint32 smNumInstances;
00029 };
00030
00031 }
00032
00033 #endif // SCALARDATAOBJECT_H
```

## 5.59 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/surfacedataobject.cpp File Reference

Implementation of the SurfaceDataObject class.

```
#include "surfacedataobject.h"
```

### 5.59.1 Detailed Description

Implementation of the SurfaceDataObject class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.60 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/core/surfacedataobject.h File Reference

Declaration of the SurfaceDataObject class.

```
#include "abstractdataobject.h"
```

### Classes

- class [QRS::Core::SurfaceDataObject](#)  
*Surface data object.*

### 5.60.1 Detailed Description

Declaration of the SurfaceDataObject class.

#### Author

Pavel Lakiza

#### Date

April 2021

## 5.61 surfacedataobject.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef SURFACEDATAOBJECT_H
00009 #define SURFACEDATAOBJECT_H
00010
00011 #include "abstractdataobject.h"
00012
00013 namespace QRS::Core
00014 {
00015
00017 class SurfaceDataObject : public AbstractDataObject
00018 {
00019 public:
00020     SurfaceDataObject(QString const& name);
00021     ~SurfaceDataObject();
00022     AbstractDataObject* clone() const override;
00023     DataItemType& addItem(DataValueType key) override;
00024     DataKeyType addLeadingItem(DataValueType key);
00025     void removeLeadingItem(DataValueType key);
00026     bool changeLeadingItemKey(DataKeyType oldKey, DataKeyType newKey);
00027     quint32 numberLeadingItems() const { return mLeadingItems.size(); }
00028     DataHolder& getLeadingItems() { return mLeadingItems; }
00029     static quint32 numberInstances() { return smNumInstances; }
00030     void serialize(QDataStream& stream) const override;
00031     virtual void deserialize(QDataStream& stream) override;
00032     virtual void import(QTextStream& stream) override;
00033
00034 private:
00035     static quint32 smNumInstances;
00036     DataHolder mLeadingItems;
00037 };
00038
00039 }
00040
00041 #endif // SURFACEDATAOBJECT_H
```

## 5.62 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/usersectionrodcomponent.cpp File Reference

Definition of the UserSectionRodComponent class.

```
#include "usersectionrodcomponent.h"
```

### 5.62.1 Detailed Description

Definition of the UserSectionRodComponent class.

Author

Pavel Lakiza

Date

June 2021

## 5.63 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/usersectionrodcomponent.h File Reference

Declaration of the UserSectionRodComponent class.

```
#include "abstractsectionrodcomponent.h"  
#include "core/scalardataobject.h"
```

### Classes

- class [QRS::Core::UserSectionRodComponent](#)  
*Section which properties are defined by user.*

### 5.63.1 Detailed Description

Declaration of the UserSectionRodComponent class.

Author

Pavel Lakiza

Date

June 2021

## 5.64 usersectionrodcomponent.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef USERSECTIONRODCOMPONENT_H
00009 #define USERSECTIONRODCOMPONENT_H
00010
00011 #include "abstractsectionrodcomponent.h"
00012 #include "core/scalardataobject.h"
00013
00014 namespace QRS::Core
00015 {
00016
00018 class UserSectionRodComponent : public AbstractSectionRodComponent
00019 {
00020 public:
00021     UserSectionRodComponent(QString const& name);
00022     AbstractRodComponent* clone() const override;
00023     bool isDataComplete() const override;
00024     // Getters
00025     // Area
00026     ScalarDataObject const* area() const { return mpArea; }
00027     // Inertia moments
00028     ScalarDataObject const* inertiaMomentTorsional() const { return mpInertiaMomentTorsional; }
00029     ScalarDataObject const* inertiaMomentX() const { return mpInertiaMomentX; }
00030     ScalarDataObject const* inertiaMomentY() const { return mpInertiaMomentY; }
00031     // Center coordinates
00032     ScalarDataObject const* centerCoordinateX() const { return mpCenterCoordinateX; }
00033     ScalarDataObject const* centerCoordinateY() const { return mpCenterCoordinateY; }
00034     // Setters
00035     // Area
00036     void setArea(ScalarDataObject const* pArea) { mpArea = pArea; }
00037     // Inertia moments
00038     void setInertiaMomentTorsional(ScalarDataObject const* pInertiaMomentTorsional) {
00039         mpInertiaMomentTorsional = pInertiaMomentTorsional; }
00040     void setInertiaMomentX(ScalarDataObject const* pInertiaMomentX) { mpInertiaMomentX =
00041         pInertiaMomentX; }
00042     void setInertiaMomentY(ScalarDataObject const* pInertiaMomentY) { mpInertiaMomentY =
00043         pInertiaMomentY; }
00044     // Center coordinates
00045     void setCenterCoordinateX(ScalarDataObject const* pCenterCoordinateX) { mpCenterCoordinateX =
00046         pCenterCoordinateX; }
00047     void setCenterCoordinateY(ScalarDataObject const* pCenterCoordinateY) { mpCenterCoordinateY =
00048         pCenterCoordinateY; }
00049 };
00050 #endif // USERSECTIONRODCOMPONENT_H

```

## 5.65 /home/qinterfly/Library/Projects/Current/QRod← Systems/src/core/utilities.cpp File Reference

Implementation of utilities.

```

#include <QDebug>
#include <QString>
#include <QFile>
#include <QDir>
#include <QPair>
#include "utilities.h"

```

### 5.65.1 Detailed Description

Implementation of utilities.



**Author**

Pavel Lakiza

**Date**

May 2021

## 5.66 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/utilities.h File Reference↵

Declaration of utilities.

```
#include <QSharedPointer>
#include "abstractdataobject.h"
```

**Functions**

- **QPair< Core::AbstractDataObject::ObjectType, QSharedPointer< QFile > > QRS::Utilities::File::get↵DataObjectFile** (QString const &path, QString const &fileName)  
*Retrieve a pair consisted of a data object file and its type.*
- **QString QRS::Utilities::File::loadFileContent** (QString const &path)  
*Load a style sheet.*

### 5.66.1 Detailed Description

Declaration of utilities.

**Author**

Pavel Lakiza

**Date**

May 2021

## 5.67 utilities.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef UTILITIES_H
00009 #define UTILITIES_H
00010
00011 #include <QSharedPointer>
00012 #include "abstractdataobject.h"
00013
00014 class QFile;
00015 class QString;
00016
00017 namespace QRS
00018 {
00019
00020 namespace Utilities
00021 {
00022
00023 namespace File
00024 {
00025
00026 QPair<Core::AbstractDataObject::ObjectType, QSharedPointer<QFile>> getDataObjectFile(QString const&
    path, QString const& fileName);
00027 QString loadFileContent(QString const& path);
00028
00029 }
00030
00031 }
00032
00033 }
00034
00035 #endif // UTILITIES_H
```

## 5.68 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/core/vectordataobject.cpp File Reference

Implementation of the VectorDataObject class.

```
#include "vectordataobject.h"
```

### Variables

- const IndexType **skNumElements** = 3

### 5.68.1 Detailed Description

Implementation of the VectorDataObject class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.69 /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.h File Reference

Declaration of the VectorDataObject class.

```
#include "abstractdataobject.h"
```

### Classes

- class [QRS::Core::VectorDataObject](#)  
*Vector data object.*

### 5.69.1 Detailed Description

Declaration of the VectorDataObject class.

#### Author

Pavel Lakiza

#### Date

April 2021

## 5.70 vectordataobject.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef VECTORDATAOBJECT_H
00009 #define VECTORDATAOBJECT_H
00010
00011 #include "abstractdataobject.h"
00012
00013 namespace QRS::Core
00014 {
00015
00017 class VectorDataObject : public AbstractDataObject
00018 {
00019 public:
00020     VectorDataObject(QString const& name);
00021     ~VectorDataObject();
00022     AbstractDataObject* clone() const override;
00023     DataItemType& addItem(DataValueType key) override;
00024     static quint32 numberInstances() { return smNumInstances; }
00025     virtual void import(QTextStream& stream) override;
00026
00027 private:
00028     static quint32 smNumInstances;
00029 };
00030
00031 }
00032
00033 #endif // VECTORDATAOBJECT_H
```

## 5.71 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/main/main.cpp File Reference

The startup function.

```
#include <QFile>
#include <QApplication>
#include <QFontDatabase>
#include "mainwindow.h"
#include "utilities.h"
```

### Functions

- int **main** (int argc, char \*argv[])  
*Entry point.*

#### 5.71.1 Detailed Description

The startup function.

##### Author

Pavel Lakiza

##### Date

May 2021

## 5.72 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/abstractmanager.cpp File Reference

Definition of the AbstractManager class.

```
#include <QMessageBox>
#include <QSettings>
#include <QToolBar>
#include "abstractmanager.h"
#include "central/uiconstants.h"
#include "DockManager.h"
```

#### 5.72.1 Detailed Description

Definition of the AbstractManager class.

##### Author

Pavel Lakiza

##### Date

May 2021

## 5.73 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractmanager.h File Reference

Declaration of the AbstractManager class.

```
#include <QDialog>
```

### Classes

- class [QRS::Managers::AbstractManager](#)  
*Abstract manager to create objects of different types.*

### 5.73.1 Detailed Description

Declaration of the AbstractManager class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.74 abstractmanager.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef ABSTRACTMANAGER_H
00003 #define ABSTRACTMANAGER_H
00004
00005 #include <QDialog>
00006
00007 QT_BEGIN_NAMESPACE
00008 class QSettings;
00009 class QToolBar;
00010 QT_END_NAMESPACE
00011
00012 namespace ads
00013 {
00014     class CDockManager;
00015 }
00016
00017 namespace QRS
00018 {
00019     namespace Managers
00020     {
00021         class AbstractManager : public QDialog
00022         {
00023             Q_OBJECT
00024
00025             public:
00026                 enum ManagerType
00027                 {
00028                     kDataObjects,
00029                     kRodComponents,
00030                     kRodConstructor
00031                 };
00032                 AbstractManager(QString& lastPath, QSettings& settings,
```

```

00042         ManagerType type, QString groupName, QWidget* parent = nullptr);
00043     virtual ~AbstractManager() = 0;
00044     void saveSettings();
00045     void restoreSettings();
00046
00047 signals:
00048     void closed(QRS::Managers::AbstractManager::ManagerType type);
00049
00050 public slots:
00051     virtual void apply() = 0;
00052
00053 protected:
00054     void closeEvent(QCloseEvent* pEvent) override;
00055     void setToolBarShortcutHints(QToolBar* pToolBar);
00056
00057 protected:
00058     // Dock manager
00059     ads::CDockManager* mpDockManager = nullptr;
00060     // Data
00061     QString& mLastPath;
00062
00063 private:
00064     QSettings& mSettings;
00065     ManagerType const mkType;
00066     QString const mkGroupName;
00067 };
00068
00069 }
00070
00071 }
00072
00073 #endif // ABSTRACTMANAGER_H

```

## 5.75 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/managers/abstractrodcomponentwidget.cpp File Reference

Definition of the AbstractRodComponentWidget class.

```

#include "abstractrodcomponentwidget.h"
#include "core/abstractdataobject.h"
#include "dataobjectlineedit.h"

```

### 5.75.1 Detailed Description

Definition of the AbstractRodComponentWidget class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.76 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/src/managers/abstractrodcomponentwidget.h File Reference

Declaration of the AbstractRodComponentWidget class.

```

#include <QWidget>
#include "core/aliasdata.h"

```

**Classes**

- class [QRS::Managers::AbstractRodComponentWidget](#)  
*Widget to construct rod components of different types.*

**Typedefs**

- using [QRS::Managers::DataObjectSetFun](#) = std::function< void([Core::AbstractDataObject](#) const \*)>

**5.76.1 Detailed Description**

Declaration of the AbstractRodComponentWidget class.

**Author**

Pavel Lakiza

**Date**

July 2021

**5.77 abstractrodcomponentwidget.h**

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef ABSTRACTRODCOMPONENTWIDGET_H
00009 #define ABSTRACTRODCOMPONENTWIDGET_H
00010
00011 #include <QWidget>
00012 #include "core/aliasdata.h"
00013
00014 namespace QRS
00015 {
00016
00017     namespace Core
00018     {
00019         class AbstractDataObject;
00020     }
00021
00022     namespace Managers
00023     {
00024         class DataObjectLineEdit;
00025
00026         using DataObjectSetFun = std::function<void(Core::AbstractDataObject const *)>;
00027
00028         class AbstractRodComponentWidget : public QWidget
00029         {
00030         public:
00031             Q_OBJECT
00032
00033             public:
00034                 AbstractRodComponentWidget(QString const& mimeType, QWidget* parent = nullptr);
00035                 virtual ~AbstractRodComponentWidget() = 0;
00036
00037             signals:
00038                 void modified();
00039                 void editDataObjectRequested(Core::DataIDType id);
00040
00041             protected:
00042                 void setDataObjectEditConnections(DataObjectLineEdit* pEdit, DataObjectSetFun& setFun);
00043
00044             protected:
00045                 QString const mkMimeType();
00046         };
00047     };
00048
00049 }
00050
00051 }
00052
00053 #endif // ABSTRACTRODCOMPONENTWIDGET_H
```

## 5.78 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/managers/constraintitemdelegate.cpp File Reference

Definition of the ComboBoxItemDelegate class.

```
#include <QComboBox>
#include "constraintitemdelegate.h"
```

### 5.78.1 Detailed Description

Definition of the ComboBoxItemDelegate class.

Author

Pavel Lakiza

Date

July 2021

## 5.79 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/managers/constraintitemdelegate.h File Reference

Declaration of the ComboBoxItemDelegate class.

```
#include <QStyledItemDelegate>
#include "core/constraintrodcomponent.h"
```

### Classes

- class [QRS::Managers::ConstraintItemDelegate](#)  
*Class to specify how options of a constraint can be edited.*

### Typedefs

- using **QRS::Managers::ConstraintTypeNames** = std::map< Core::ConstraintRodComponent::ConstraintType, QString >
- using **QRS::Managers::ConstraintCoordinateSystemNames** = std::map< Core::ConstraintRodComponent::ConstraintCoordinateSystem, QString >



### 5.79.1 Detailed Description

Declaration of the ComboBoxItemDelegate class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.80 constraintitemdelegate.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef CONSTRAINTITEMDELEGATE_H
00003 #define CONSTRAINTITEMDELEGATE_H
00004
00005 #include <QStyledItemDelegate>
00006 #include "core/constraintrodcomponent.h"
00007
00008 namespace QRS::Managers
00009 {
00010
00011 using ConstraintTypeNames = std::map<Core::ConstraintRodComponent::ConstraintType, QString>;
00012 using ConstraintCoordinateSystemNames =
00013     std::map<Core::ConstraintRodComponent::ConstraintCoordinateSystem, QString>;
00014
00015 class ConstraintItemDelegate : public QStyledItemDelegate
00016 {
00017     Q_OBJECT
00018
00019 public:
00020     ConstraintItemDelegate(Core::ConstraintRodComponent const& constraintRodComponent,
00021         ConstraintTypeNames const& types,
00022         ConstraintCoordinateSystemNames const& coordinateSystems, QObject* parent =
00023             nullptr);
00024     QWidget* createEditor(QWidget* pCell, const QStyleOptionViewItem& option, const QModelIndex&
00025         index) const override;
00026     void setEditorData(QWidget* pEditor, const QModelIndex& index) const override;
00027     void setModelData(QWidget* pEditor, QAbstractItemModel* pModel, const QModelIndex& index) const
00028         override;
00029     void updateEditorGeometry(QWidget* pEditor, const QStyleOptionViewItem& option, const QModelIndex&
00030         index) const override;
00031
00032 signals:
00033     void typeCreated(int iRow) const;
00034     void typeChanged(int iRow, Core::ConstraintRodComponent::ConstraintType oldType) const;
00035     void coordinateSystemChanged(int iRow) const;
00036
00037 private:
00038     Core::ConstraintRodComponent const& mConstraintRodComponent;
00039     ConstraintTypeNames const& mTypes;
00040     ConstraintCoordinateSystemNames const& mCoordinateSystems;
00041 };
00042
00043 #endif // CONSTRAINTITEMDELEGATE_H
```

## 5.81 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/constraintrodcomponentwidget.cpp File Reference

Definition of the ConstraintRodComponentWidget class.

```
#include <QVBoxLayout>
#include <QTableWidget>
#include <QHeaderView>
#include <QToolBar>
#include <set>
#include "constraintrodcomponentwidget.h"
#include "core/constraintrodcomponent.h"
```

### 5.81.1 Detailed Description

Definition of the ConstraintRodComponentWidget class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.82 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/constraintrodcomponentwidget.h File Reference

Declaration of the ConstraintRodComponentWidget class.

```
#include "abstractrodcomponentwidget.h"
#include "constraintitemdelegate.h"
```

### Classes

- class [QRS::Managers::ConstraintRodComponentWidget](#)  
*Widget to construrt constraints of a rod.*

### 5.82.1 Detailed Description

Declaration of the ConstraintRodComponentWidget class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.83 constraintrodcomponentwidget.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef CONSTRAINTRODCOMPONENTWIDGET_H
00009 #define CONSTRAINTRODCOMPONENTWIDGET_H
00010
00011 #include "abstractrodcomponentwidget.h"
00012 #include "constraintitemdelegate.h"
00013
00014 QT_BEGIN_NAMESPACE
00015 class QTableWidget;
00016 class QTableWidgetItem;
00017 class QToolBar;
00018 QT_END_NAMESPACE
00019
00020 namespace QRS
00021 {
00022
00023     namespace Managers
00024     {
00025
00026     class ConstraintRodComponentWidget : public AbstractRodComponentWidget
00027     {
00028     public:
00029         ConstraintRodComponentWidget(Core::ConstraintRodComponent& constraintRodComponent, QWidget* parent
00030         = nullptr);
00031         ~ConstraintRodComponentWidget();
00032
00033     private:
00034         // Creating
00035         void createContent();
00036         QToolBar* createToolBar();
00037         void createTableWidget();
00038         // Interaction
00039         void addRow();
00040         void removeSelectedRows();
00041         void representConstraintData();
00042         // Helpers
00043         void setTableHeight();
00044         void specifyConstraintNames();
00045         QVariant getItemData(int iRow, int iColumn);
00046
00047     private slots:
00048         void setConstraintData(int iRow);
00049
00050     private:
00051         Core::ConstraintRodComponent& mConstraintRodComponent;
00052         QTableWidget* mpTableConstraint;
00053         ConstraintItemDelegate* mpItemDelegate;
00054         ConstraintTypeNames mTypeNames;
00055         ConstraintCoordinateSystemNames mCoordinateSystemNames;
00056     };
00057
00058 }
00059
00060 }
00061
00062 #endif // CONSTRAINTRODCOMPONENTWIDGET_H

```

## 5.84 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/dataobjectlineedit.cpp File Reference

Definition of the DataPointerLineEdit class.

```

#include <QMimeType>
#include <QDragEnterEvent>
#include <QMenu>
#include "dataobjectlineedit.h"
#include "models/hierarchy/dataobjectshierarchyitem.h"

```

### 5.84.1 Detailed Description

Definition of the DataPointerLineEdit class.

Author

Pavel Lakiza

Date

June 2021

## 5.85 [/home/qinterfly/Library/Projects/Current/QRod↵](#) Systems/src/managers/dataobjectlineedit.h File Reference

Declaration of the DataPointerLineEdit class.

```
#include <QLineEdit>
#include "core/abstractdataobject.h"
```

### Classes

- class [QRS::Managers::DataObjectLineEdit](#)  
*Line edit widget to hold a pointer to a data object.*

### 5.85.1 Detailed Description

Declaration of the DataPointerLineEdit class.

Author

Pavel Lakiza

Date

June 2021

## 5.86 dataobjectlineedit.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef DATAOBJECTLINEEDIT_H
00009 #define DATAOBJECTLINEEDIT_H
00010
00011 #include <QLineEdit>
00012 #include "core/abstractdataobject.h"
00013
00014 namespace QRS
00015 {
00016
00017 namespace Managers
00018 {
00019
00021 class DataObjectLineEdit : public QLineEdit
00022 {
00023     Q_OBJECT
00024 public:
00025     DataObjectLineEdit(Core::AbstractDataObject const* pDataObject,
00026         Core::AbstractDataObject::ObjectType type,
00027         QString const& mimeType, QWidget* parent = nullptr);
00028     ~DataObjectLineEdit() = default;
00029 signals:
00030     void selected(Core::AbstractDataObject const* pDataObject);
00031     void editRequested(Core::DataIDType id);
00032
00033 private slots:
00034     void showContextMenu(const QPoint& point);
00035     void reset();
00036     void edit();
00037
00038 private:
00039     void dragEnterEvent(QDragEnterEvent* pEvent) override;
00040     void dropEvent(QDropEvent* pEvent) override;
00041     void keyPressEvent(QKeyEvent* pEvent) override;
00042     void mouseDoubleClickEvent(QMouseEvent* pEvent) override;
00043
00044 private:
00045     Core::AbstractDataObject const* mpDataObject;
00046     Core::AbstractDataObject::ObjectType mType;
00047     QString const mkMimeType;
00048 };
00049
00050 }
00051
00052 }
00053
00054 #endif // DATAOBJECTLINEEDIT_H

```

## 5.87 /home/qinterfly/Library/Projects/Current/QRod← Systems/src/managers/dataobjectsmanager.cpp File Reference

Implementation of the DataObjectsManager class.

```

#include <QTreeView>
#include <QSettings>
#include <QHBoxLayout>
#include <QToolBar>
#include <QListWidget>
#include <QTextEdit>
#include <QPushButton>
#include <QSpacerItem>
#include <QShortcut>
#include <QFileDialog>
#include "DockManager.h"
#include "DockWidget.h"
#include "DockAreaWidget.h"

```



### 5.88.1 Detailed Description

Declaration of the DataObjectsManager class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.89 dataobjectsmanager.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef DATAOBJECTSMANAGER_H
00009 #define DATAOBJECTSMANAGER_H
00010
00011 #include <unordered_map>
00012 #include "abstractmanager.h"
00013 #include "core/aliasdata.h"
00014 #include "core/aliasdataset.h"
00015 #include "core/hierarchytree.h"
00016
00017 QT_BEGIN_NAMESPACE
00018 class QTreeView;
00019 class QSettings;
00020 QT_END_NAMESPACE
00021
00022 namespace ads
00023 {
00024     class CDockManager;
00025     class CDockWidget;
00026 }
00027
00028 namespace QRS
00029 {
00030
00031     namespace TableModels
00032     {
00033         class TableModelInterface;
00034         class BaseTableModel;
00035         class MatrixTableModel;
00036         class SurfaceTableModel;
00037     }
00038
00039     namespace HierarchyModels
00040     {
00041         class DataObjectsHierarchyModel;
00042     }
00043
00044     namespace Managers
00045     {
00046
00048         class DataObjectsManager : public AbstractManager
00049         {
00050             Q_OBJECT
00051
00052         public:
00053             explicit DataObjectsManager(Core::DataObjects&& dataObjects, Core::HierarchyTree&&
                hierarchyDataObjects,
00054                                     QString& lastPath, QSettings& settings, QWidget* parent = nullptr);
00055             ~DataObjectsManager();
00056             void selectDataObject(int iRow);
00057             void selectDataObjectByID(Core::DataIDType id);
00058             Core::DataObjects const& getDataObjects() { return mDataObjects; };
00059
00060         signals:
00061             void applied(Core::DataObjects const& dataObjects, Core::HierarchyTree const&
                hierarchyDataObjects);
00062
00063         public slots:
00064             void apply() override;

```

```

00065     Core::AbstractDataObject* addScalar();
00066     Core::AbstractDataObject* addVector();
00067     Core::AbstractDataObject* addMatrix();
00068     Core::AbstractDataObject* addSurface();
00069     void insertItemAfterSelected();
00070     void insertLeadingItemAfterSelected();
00071     void removeSelectedItem();
00072     void removeSelectedLeadingItem();
00073     void importDataObjects();
00074
00075 private:
00076     // Content
00077     void createContent();
00078     ads::CDockWidget* createDataTableWidget();
00079     ads::CDockWidget* createHierarchyWidget();
00080     QLayout* createDialogControls();
00081     // Helpers
00082     void emplaceDataObject(Core::AbstractDataObject* pDataObject);
00083     bool isDataTableModifiable();
00084     void importDataObject(QString const& path, QString const& fileName);
00085     // Selection
00086     void representDataObject(Core::DataIDType id);
00087     void clearDataObjectRepresentation();
00088
00089 private:
00090     // Widgets
00091     QTreeView* mpTreeDataObjects;
00092     QTreeView* mpDataTable;
00093     // Data
00094     Core::DataObjects mDataObjects;
00095     Core::HierarchyTree mHierarchyDataObjects;
00096     // Models
00097     TableModels::TableModelInterface* mpTableModelInterface = nullptr;
00098     TableModels::BaseTableModel* mpBaseTableModel;
00099     TableModels::MatrixTableModel* mpMatrixTableModel;
00100     TableModels::SurfaceTableModel* mpSurfaceTableModel;
00101     HierarchyModels::DataObjectsHierarchyModel* mpTreeDataObjectsModel;
00102 };
00103
00104 }
00105
00106 }
00107
00108 #endif // DATAOBJECTSMANAGER_H

```

## 5.90 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/managers/doublespinboxitemdelegate.cpp File Reference

Definition of the DoubleSpinBoxItemDelegate class.

```

#include <QDoubleSpinBox>
#include "doublespinboxitemdelegate.h"

```

### 5.90.1 Detailed Description

Definition of the DoubleSpinBoxItemDelegate class.

#### Author

Pavel Lakiza

#### Date

July 2021



## 5.91 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/doublespinboxitemdelegate.h File Reference

Declaration of the DoubleSpinBoxItemDelegate class.

```
#include <QStyledItemDelegate>
```

### Classes

- class [QRS::Managers::DoubleSpinBoxItemDelegate](#)  
*Class to specify how table values can be edited.*

### 5.91.1 Detailed Description

Declaration of the DoubleSpinBoxItemDelegate class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.92 doublespinboxitemdelegate.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef DOUBLESPINBOXITEMDELEGATE_H
00009 #define DOUBLESPINBOXITEMDELEGATE_H
00010
00011 #include <QStyledItemDelegate>
00012
00013 namespace QRS::Managers
00014 {
00015
00017 class DoubleSpinBoxItemDelegate : public QStyledItemDelegate
00018 {
00019 public:
00020     DoubleSpinBoxItemDelegate(QObject* parent = nullptr);
00021     QWidget* createEditor(QWidget* parent, const QStyleOptionViewItem& option, const QModelIndex&
index) const override;
00022     void setEditorData(QWidget* pEditor, const QModelIndex& index) const override;
00023     void setModelData(QWidget* pEditor, QAbstractItemModel* pModel, const QModelIndex& index) const
override;
00024     void updateEditorGeometry(QWidget* pEditor, const QStyleOptionViewItem& option, const QModelIndex&
index) const override;
00025 };
00026
00027 }
00028
00029 #endif // DOUBLESPINBOXITEMDELEGATE_H
```

## 5.93 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/managers/geometryrodcomponentwidget.cpp File Reference

Definiton of the GeometryComponentWidget class.

```
#include <QGridLayout>
#include <QSpacerItem>
#include <QLabel>
#include "geometryrodcomponentwidget.h"
#include "dataobjectlineedit.h"
#include "core/geometryrodcomponent.h"
#include "core/vectordataobject.h"
#include "core/matrixdataobject.h"
```

### 5.93.1 Detailed Description

Definiton of the GeometryComponentWidget class.

Author

Pavel Lakiza

Date

July 2021

## 5.94 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/managers/geometryrodcomponentwidget.h File Reference

Declaration of the GeometryComponentWidget class.

```
#include "abstractrodcomponentwidget.h"
```

### Classes

- class [QRS::Managers::GeometryRodComponentWidget](#)  
*Widget to construct a geometrical rod component.*

### 5.94.1 Detailed Description

Declaration of the GeometryComponentWidget class.

Author

Pavel Lakiza

Date

July 2021

## 5.95 geometryrodcomponentwidget.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef GEOMETRYRODCOMPONENTWIDGET_H
00009 #define GEOMETRYRODCOMPONENTWIDGET_H
00010
00011 #include "abstractrodcomponentwidget.h"
00012
00013 namespace QRS
00014 {
00015
00016 namespace Core
00017 {
00018 class GeometryRodComponent;
00019 class AbstractDataObject;
00020 }
00021
00022 namespace Managers
00023 {
00024
00026 class GeometryRodComponentWidget : public AbstractRodComponentWidget
00027 {
00028 public:
00029     GeometryRodComponentWidget(Core::GeometryRodComponent& geometryRodComponent, QString const&
00030         mimeType, QWidget* parent = nullptr);
00031 private:
00032     void createContent();
00033     template<typename T>
00034     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
00035
00036 private:
00037     Core::GeometryRodComponent& mGeometryRodComponent;
00038 };
00039
00040 }
00041
00042 }
00043
00044 #endif // GEOMETRYRODCOMPONENTWIDGET_H
```

## 5.96 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/loadrodcomponentwidget.cpp File Reference

Definition of the LoadRodComponentWidget class.

```
#include <QVBoxLayout>
#include <QLabel>
#include <QComboBox>
#include <QGroupBox>
#include <QDoubleSpinBox>
#include <QCheckBox>
#include "loadrodcomponentwidget.h"
#include "dataobjectlineedit.h"
#include "core/scalardataobject.h"
#include "core/vectordataobject.h"
```

### 5.96.1 Detailed Description

Definition of the LoadRodComponentWidget class.

**Author**

Pavel Lakiza

**Date**

July 2021

## 5.97 [/home/qinterfly/Library/Projects/Current/QRod](#)↵ Systems/src/managers/loadrodcomponentwidget.h File Reference

Declaration of the LoadRodComponentWidget class.

```
#include "abstractrodcomponentwidget.h"  
#include "core/loadrodcomponent.h"
```

**Classes**

- class [QRS::Managers::LoadRodComponentWidget](#)  
*Widget to construct a load applied to a rod.*

### 5.97.1 Detailed Description

Declaration of the LoadRodComponentWidget class.

**Author**

Pavel Lakiza

**Date**

July 2021

## 5.98 loadrodcomponentwidget.h

[Go to the documentation of this file.](#)

```
00001  
00008 #ifndef LOADRODCOMPONENTWIDGET_H  
00009 #define LOADRODCOMPONENTWIDGET_H  
00010  
00011 #include "abstractrodcomponentwidget.h"  
00012 #include "core/loadrodcomponent.h"  
00013  
00014 QT_BEGIN_NAMESPACE  
00015 class QComboBox;  
00016 class QLabel;  
00017 QT_END_NAMESPACE  
00018  
00019 namespace QRS  
00020 {  
00021  
00022 namespace Core  
00023 {  
00024 class AbstractDataObject;
```

```

00025 }
00026
00027 namespace Managers
00028 {
00029
00031 class LoadRodComponentWidget : public AbstractRodComponentWidget
00032 {
00033 public:
00034     LoadRodComponentWidget(Core::LoadRodComponent& loadRodComponent, QString const& mimeType, QWidget*
parent = nullptr);
00035
00036 private:
00037     void createContent();
00038     QLayout* createBaseLayout();
00039     QWidget* createTimeGroup();
00040     QLayout* createLoadTypeLayout();
00041     QComboBox* createLoadTypeComboBox();
00042     template<typename T>
00043     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
00044     void setLoadUnits(Core::LoadRodComponent::LoadType type);
00045
00046 private:
00047     Core::LoadRodComponent& mLoadRodComponent;
00048     QLabel* mpLoadRodUnits;
00049 };
00050
00051 }
00052
00053 }
00054
00055 #endif // LOADRODCOMPONENTWIDGET_H

```

## 5.99 /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/managersfactory.cpp File Reference

Definition of the ManagersFactory class.

```

#include "managersfactory.h"
#include "core/project.h"
#include "managers/dataobjectsmanager.h"
#include "managers/rodcomponentsmanager.h"

```

### Functions

- void **moveToCenter** (QWidget \*pWidget)  
*Helper function to situate widgets at the center of their parent widgets.*

### 5.99.1 Detailed Description

Definition of the ManagersFactory class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.100 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/managers/managersfactory.h File Reference

Declaration of the ManagersFactory class.

```
#include <QObject>
#include "abstractmanager.h"
```

### Classes

- class [QRS::Managers::ManagersFactory](#)  
*Factory to create managers which utilize and modify project data.*

### 5.100.1 Detailed Description

Declaration of the ManagersFactory class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.101 managersfactory.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef MANAGERSFACTORY_H
00009 #define MANAGERSFACTORY_H
00010
00011 #include <QObject>
00012 #include "abstractmanager.h"
00013
00014 QT_BEGIN_NAMESPACE
00015 class QSettings;
00016 QT_END_NAMESPACE
00017
00018 namespace QRS
00019 {
00020
00021     namespace Core
00022     {
00023         class Project;
00024     }
00025
00026     namespace Managers
00027     {
00028
00029         class DataObjectsManager;
00030         class RodComponentsManager;
00031
00032         class ManagersFactory : public QObject
00033         {
00034             Q_OBJECT
00035
00036             public:
00037                 ManagersFactory(Core::Project& project, QString& lastPath, QSettings& settings, QWidget* parent);
00038                 ~ManagersFactory();
00039         }
00040     }
00041 }
```

```

00040     bool createManager(AbstractManager::ManagerType type);
00041     bool deleteManager(AbstractManager::ManagerType type);
00042     AbstractManager* manager(AbstractManager::ManagerType type);
00043
00044 private:
00045     void specifyConnections(DataObjectsManager* pManager);
00046     void specifyConnections(RodComponentsManager* pManager);
00047
00048 private:
00049     Core::Project& mProject;
00050     QString& mLastPath;
00051     QSettings& mSettings;
00052     QWidget* mpParent;
00053     std::unordered_map<AbstractManager::ManagerType, AbstractManager*> mManagers;
00054 };
00055
00056 }
00057
00058 }
00059
00060 #endif // MANAGERSFACTORY_H

```

## 5.102 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/materialrodcomponentwidget.cpp File Reference

Definition of the MaterialRodComponentWidget class.

```

#include <QGridLayout>
#include <QSpacerItem>
#include <QLabel>
#include <QGroupBox>
#include "materialrodcomponentwidget.h"
#include "dataobjectlineedit.h"
#include "core/materialrodcomponent.h"
#include "core/scalardataobject.h"

```

### 5.102.1 Detailed Description

Definition of the MaterialRodComponentWidget class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.103 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/materialrodcomponentwidget.h File Reference

Declaration of the MaterialRodComponentWidget class.

```

#include "abstractrodcomponentwidget.h"

```

## Classes

- class [QRS::Managers::MaterialRodComponentWidget](#)  
*Widget to construct a material rod component.*

### 5.103.1 Detailed Description

Declaration of the MaterialRodComponentWidget class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.104 materialrodcomponentwidget.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef MATERIALRODCOMPONENTWIDGET_H
00009 #define MATERIALRODCOMPONENTWIDGET_H
00010
00011 #include "abstractrodcomponentwidget.h"
00012
00013 namespace QRS
00014 {
00015
00016 namespace Core
00017 {
00018 class MaterialRodComponent;
00019 class AbstractDataObject;
00020 }
00021
00022 namespace Managers
00023 {
00024
00026 class MaterialRodComponentWidget : public AbstractRodComponentWidget
00027 {
00028 public:
00029     MaterialRodComponentWidget(Core::MaterialRodComponent& materialRodComponent, QString const&
mimeType, QWidget* parent = nullptr);
00030
00031 private:
00032     void createContent();
00033     QWidget* createModuliGroup();
00034     QLayout* createBaseLayout();
00035     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
00036
00037 private:
00038     Core::MaterialRodComponent& mMaterialRodComponent;
00039 };
00040
00041 }
00042
00043 }
00044
00045 #endif // MATERIALRODCOMPONENTWIDGET_H

```



## 5.105 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/managers/mechanicalrodcomponentwidget.cpp File Reference

Definition of the MechanicalRodComponentWidget class.

```
#include <QVBoxLayout>
#include <QGroupBox>
#include <QLabel>
#include "mechanicalrodcomponentwidget.h"
#include "dataobjectlineedit.h"
#include "core/mechanicalrodcomponent.h"
#include "core/scalardataobject.h"
```

### 5.105.1 Detailed Description

Definition of the MechanicalRodComponentWidget class.

Author

Pavel Lakiza

Date

July 2021

## 5.106 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/managers/mechanicalrodcomponentwidget.h File Reference

Declaration of the MechanicalRodComponentWidget class.

```
#include "abstractrodcomponentwidget.h"
```

Classes

- class [QRS::Managers::MechanicalRodComponentWidget](#)  
*Widget to construct mechanical rod components consisted of stiffness and mass distributions.*

### 5.106.1 Detailed Description

Declaration of the MechanicalRodComponentWidget class.

Author

Pavel Lakiza

Date

July 2021

## 5.107 mechanicalrodcomponentwidget.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef MECHANICALRODCOMPONENTWIDGET_H
00009 #define MECHANICALRODCOMPONENTWIDGET_H
00010
00011 #include "abstractrodcomponentwidget.h"
00012
00013 namespace QRS
00014 {
00015
00016 namespace Core
00017 {
00018 class AbstractDataObject;
00019 class MechanicalRodComponent;
00020 }
00021
00022 namespace Managers
00023 {
00024
00026 class MechanicalRodComponentWidget : public AbstractRodComponentWidget
00027 {
00028 public:
00029     MechanicalRodComponentWidget(Core::MechanicalRodComponent& mechanicalRodComponent, QString const&
mimeType, QWidget* parent = nullptr);
00030
00031 private:
00032     void createContent();
00033     QWidget* createStiffnessGroup();
00034     QWidget* createMassGroup();
00035     QWidget* createEccentricityGroup();
00036     QLayout* createContactDiameterLayout();
00037     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
00038
00039 private:
00040     Core::MechanicalRodComponent& mMechanicalRodComponent;
00041 };
00042
00043 }
00044
00045 }
00046
00047 #endif // MECHANICALRODCOMPONENTWIDGET_H

```

## 5.108 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/rodcomponentsmanager.cpp File Reference

Definition of the RodComponentsManager class.

```

#include <QVBoxLayout>
#include <QPushButton>
#include <QTreeView>
#include <QToolBar>
#include <QLabel>
#include "DockManager.h"
#include "DockWidget.h"
#include "DockAreaWidget.h"
#include "rodcomponentsmanager.h"
#include "core/vectordataobject.h"
#include "core/matrixdataobject.h"
#include "core/geometryrodcomponent.h"
#include "core/usersectionrodcomponent.h"
#include "core/materialrodcomponent.h"
#include "core/loadrodcomponent.h"
#include "core/constraintrodcomponent.h"

```

```
#include "core/mechanicalrodcomponent.h"
#include "managers/geometryrodcomponentwidget.h"
#include "managers/usersectionrodcomponentwidget.h"
#include "managers/materialrodcomponentwidget.h"
#include "managers/loadrodcomponentwidget.h"
#include "managers/constraintrodcomponentwidget.h"
#include "managers/mechanicalrodcomponentwidget.h"
#include "models/hierarchy/dataobjectshierarchymodel.h"
#include "models/hierarchy/rodcomponentshierarchymodel.h"
```

## Functions

- QWidget \* **addToolBarHeader** (QToolBar \*pToolBar, QString const &name)  
*Helper function to add the header to a toolbar.*
- [AbstractRodComponentWidget](#) \* **createRodComponentWidget** ([AbstractRodComponent](#) \*pRod↔  
Component, ads::CDockWidget \*pDockWidget)  
*Create an appropriate constructor of a rod component.*

## Variables

- QSize const **skToolBarIconSize** = QSize(27, 27)
- QString const **skDataObjectsMimeType** = "rodcomponentsmanager/dataobjectshierarchy"

### 5.108.1 Detailed Description

Definition of the RodComponentsManager class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.109 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/managers/rodcomponentsmanager.h File Reference

Declaration of the RodComponentsManager class.

```
#include "managers/abstractmanager.h"
#include "core/aliasdataset.h"
#include "core/hierarchytree.h"
#include "core/abstractsectionrodcomponent.h"
```

## Classes

- class [QRS::Managers::RodComponentsManager](#)  
*Manager to create rod components, such as a geometry, cross section and force.*

### 5.109.1 Detailed Description

Declaration of the RodComponentsManager class.

#### Author

Pavel Lakiza

#### Date

March 2021

## 5.110 rodcomponentsmanager.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef RODCOMPONENTSMANAGER_H
00009 #define RODCOMPONENTSMANAGER_H
00010
00011 #include "managers/abstractmanager.h"
00012 #include "core/aliasdataset.h"
00013 #include "core/hierarchytree.h"
00014 #include "core/abstractsectionrodcomponent.h"
00015
00016 QT_BEGIN_NAMESPACE
00017 class QTreeView;
00018 QT_END_NAMESPACE
00019
00020 namespace ads
00021 {
00022 class CDockWidget;
00023 }
00024
00025 namespace QRS
00026 {
00027
00028 namespace HierarchyModels
00029 {
00030 class DataObjectsHierarchyModel;
00031 class RodComponentsHierarchyModel;
00032 }
00033
00034 namespace Managers
00035 {
00036
00037 class RodComponentsManager : public AbstractManager
00038 {
00039     Q_OBJECT
00040
00041 public:
00042     RodComponentsManager(Core::DataObjects& dataObjects, Core::HierarchyTree& hierarchyDataObjects,
00043                         Core::RodComponents&& rodComponents, Core::HierarchyTree&&
00044                         hierarchyRodComponents,
00045                         QString& lastPath, QSettings& settings, QWidget* parent = nullptr);
00046     ~RodComponentsManager();
00047     void selectRodComponent(int iRow);
00048     void updateDataObjects();
00049
00050 signals:
00051     void applied(Core::RodComponents const& rodComponents, Core::HierarchyTree const&
00052                 hierarchyRodComponents);
00053     void editDataObjectRequested(Core::DataIDType id);
00054 public slots:
00055     void apply() override;
```

```

00056     Core::AbstractRodComponent* addGeometry();
00057     Core::AbstractRodComponent* addSection(Core::AbstractSectionRodComponent::SectionType
sectionType);
00058     Core::AbstractRodComponent* addMaterial();
00059     Core::AbstractRodComponent* addLoad();
00060     Core::AbstractRodComponent* addConstraint();
00061     Core::AbstractRodComponent* addMechanical();
00062     void resolveRodComponentsReferences();
00063
00064 private:
00065     // Content
00066     void createContent();
00067     QLayout* createDialogControls();
00068     ads::CDockWidget* createHierarchyRodComponentsWidget();
00069     ads::CDockWidget* createConstructorDockWidget();
00070     ads::CDockWidget* createHierarchyDataObjectsWidget();
00071     // Helpers
00072     void emplaceRodComponent(Core::AbstractRodComponent* pRodComponent);
00073     // Selection
00074     void representRodComponent(Core::DataIDType id);
00075     void clearRodComponentRepresentation();
00076     // Toolbars
00077     QToolBar* createMainToolBar();
00078     QWidget* makeGeometryToolBar();
00079     QWidget* makeSectionsToolBar();
00080     QWidget* makeBoundaryConditionsToolBar();
00081     QWidget* makeLoadingToolBar();
00082     QWidget* makeMaterialToolBar();
00083     QWidget* makeMechanicalToolBar();
00084     QWidget* makeModificationToolBar();
00085
00086 private:
00087     // Widgets
00088     ads::CDockWidget* mpComponentDockWidget;
00089     QTreeView* mpTreeRodComponents;
00090     // Data objects
00091     Core::DataObjects& mDataObjects;
00092     Core::HierarchyTree& mHierarchyDataObjects;
00093     // Rod components data
00094     Core::RodComponents mRodComponents;
00095     Core::HierarchyTree mHierarchyRodComponents;
00096     // Models
00097     HierarchyModels::DataObjectsHierarchyModel* mpTreeDataObjectsModel;
00098     HierarchyModels::RodComponentsHierarchyModel* mpTreeRodComponentsModel;
00099 };
00100
00101 }
00102
00103 }
00104
00105 #endif // RODCOMPONENTSMANAGER_H

```

## 5.111 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/managers/usersectionrodcomponentwidget.cpp File Reference

Definition of the UserSectionRodComponentWidget class.

```

#include <QVBoxLayout>
#include <QGroupBox>
#include <QLabel>
#include "usersectionrodcomponentwidget.h"
#include "core/usersectionrodcomponent.h"
#include "dataobjectlineedit.h"

```

### 5.111.1 Detailed Description

Definition of the UserSectionRodComponentWidget class.

**Author**

Pavel Lakiza

**Date**

July 2021

## 5.112 `/home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/managers/usersectionrodcomponentwidget.h` File Reference

Declaration of the UserSectionRodComponentWidget class.

```
#include "abstractrodcomponentwidget.h"
```

**Classes**

- class [QRS::Managers::UserSectionRodComponentWidget](#)  
*Widget to construct a user-defined section of a rod.*

### 5.112.1 Detailed Description

Declaration of the UserSectionRodComponentWidget class.

**Author**

Pavel Lakiza

**Date**

July 2021

## 5.113 `usersectionrodcomponentwidget.h`

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef USERSECTIONRODCOMPONENTWIDGET_H
00009 #define USERSECTIONRODCOMPONENTWIDGET_H
00010
00011 #include "abstractrodcomponentwidget.h"
00012
00013 namespace QRS
00014 {
00015
00016 namespace Core
00017 {
00018 class UserSectionRodComponent;
00019 class AbstractDataObject;
00020 }
00021
00022 namespace Managers
00023 {
```

```

00024
00026 class UserSectionRodComponentWidget : public AbstractRodComponentWidget
00027 {
00028 public:
00029     UserSectionRodComponentWidget(Core::UserSectionRodComponent& userSectionRodComponent,
00030                                   QString const& mimeType, QWidget* parent = nullptr);
00031 private:
00032     void createContent();
00033     QLayout* createAreaLayout();
00034     QWidget* createInertiaMomentsGroup();
00035     QWidget* createCenterCoordinatesGroup();
00036     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
00037 private:
00038     Core::UserSectionRodComponent& mUserSectionRodComponent;
00039 };
00040
00041
00042
00043 }
00044
00045 }
00046
00047 #endif // USERSECTIONRODCOMPONENTWIDGET_H

```

## 5.114 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/hierarchy/abstrachierarchyitem.cpp File Reference

Definition of the AbstractHierarchyItem class.

```

#include "abstrachierarchyitem.h"
#include "core/hierarchy/node.h"

```

### 5.114.1 Detailed Description

Definition of the AbstractHierarchyItem class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.115 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/hierarchy/abstrachierarchyitem.h File Reference

Declaration of the AbstractHierarchyItem class.

```

#include <QStandardItem>

```

## Classes

- class [QRS::HierarchyModels::AbstractHierarchyItem](#)  
*Item to represent a hierarchy of elements of the same type.*

### 5.115.1 Detailed Description

Declaration of the AbstractHierarchyItem class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.116 abstracthierarchyitem.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef ABSTRACTHIERARCHYITEM_H
00009 #define ABSTRACTHIERARCHYITEM_H
00010
00011 #include <QStandardItem>
00012
00013 namespace QRS
00014 {
00015
00016     namespace Core
00017     {
00018         class HierarchyNode;
00019         class HierarchyTree;
00020     }
00021
00022     namespace PropertiesModels
00023     {
00024         class AbstractPropertiesModel;
00025     }
00026
00027     namespace HierarchyModels
00028     {
00029
00031         class AbstractHierarchyItem : public QStandardItem
00032         {
00033             friend class AbstractHierarchyModel;
00034             friend class PropertiesModels::AbstractPropertiesModel;
00035
00036         public:
00037             enum ItemType
00038             {
00039                 kDataObjects = QStandardItem::UserType,
00040                 kRodComponents
00041             };
00042             AbstractHierarchyItem(QIcon const& icon, QString const& text, Core::HierarchyNode* pNode);
00043             virtual ~AbstractHierarchyItem() = 0;
00044             void writePointer(QDataStream& out) const;
00045             static AbstractHierarchyItem* readPointer(QDataStream& in);
00046             virtual int type() const = 0;
00047
00048         protected:
00049             Core::HierarchyNode* mpNode = nullptr;
00050     };
00051
00052 }
00053
00054 }
00055
00056 #endif // ABSTRACTHIERARCHYITEM_H

```



## 5.117 [↔](#) /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.cpp File Reference

Definition of the AbstractHierarchyModel class.

```
#include <QTreeView>
#include <QMimeData>
#include <unordered_map>
#include "abstracthierarchymodel.h"
#include "core/hierarchy/hierarchynode.h"
```

### 5.117.1 Detailed Description

Definition of the AbstractHierarchyModel class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.118 [↔](#) /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/abstracthierarchymodel.h File Reference

Declaration of the AbstractHierarchyModel class.

```
#include <QStandardItemModel>
#include "abstracthierarchyitem.h"
```

#### Classes

- class [QRS::HierarchyModels::AbstractHierarchyModel](#)  
*Hierarchy model which enables one to drag and drop elements of the same type.*

#### Typedefs

- using [QRS::HierarchyModels::NodesState](#) = std::unordered\_map< [Core::HierarchyNode](#) \*, bool >

### 5.118.1 Detailed Description

Declaration of the AbstractHierarchyModel class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.119 abstracthierarchymodel.h

[Go to the documentation of this file.](#)

```

00001
00002 #ifndef ABSTRACTHIERARCHYMODEL_H
00003 #define ABSTRACTHIERARCHYMODEL_H
00004
00005 #include <QStandardItemModel>
00006 #include "abstracthierarchyitem.h"
00007
00008 QT_BEGIN_NAMESPACE
00009 class QTreeView;
00010 QT_END_NAMESPACE
00011
00012 namespace QRS
00013 {
00014
00015     namespace Core
00016     {
00017         class HierarchyNode;
00018     }
00019
00020     namespace HierarchyModels
00021     {
00022         using NodesState = std::unordered_map<Core::HierarchyNode*, bool>;
00023
00024         class AbstractHierarchyModel : public QStandardItemModel
00025         {
00026             Q_OBJECT
00027
00028             public:
00029                 AbstractHierarchyModel(QString const& mimeType, QTreeView* pView = nullptr);
00030                 virtual ~AbstractHierarchyModel() = 0;
00031                 virtual void updateContent() = 0;
00032                 virtual void clearContent() = 0;
00033                 Qt::DropActions supportedDragActions() const override;
00034                 Qt::DropActions supportedDropActions() const override;
00035                 QStringList mimeTypeTypes() const override;
00036                 QMimeData* mimeTypeData(const QModelIndexList& indices) const override;
00037                 bool dropMimeData(QMimeData const* pMimeData, Qt::DropAction action, int row, int column, const
00038                     QModelIndex& parent) override;
00039
00040             signals:
00041                 void hierarchyChanged();
00042
00043             private:
00044                 bool processDropOnItem(QDataStream& stream, int& numItems, QModelIndex const& indexParent);
00045                 bool processDropBetweenItems(QDataStream& stream, int& numItems, QModelIndex const& indexParent,
00046                     int row);
00047                 void retrieveExpandedState(NodesState& nodesState, QModelIndex const& indexParent, QTreeView
00048                     const* pView);
00049                 void setExpandedState(NodesState& nodesState, QModelIndex const& indexParent, QTreeView* pView);
00050                 void updateContentExpanded();
00051
00052             protected:
00053                 QString const mkMimeType();
00054         };
00055     }
00056 }
00057
00058 #endif // ABSTRACTHIERARCHYMODEL_H

```

## 5.120 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/models/hierarchy/dataobjectshierarchyitem.cpp File Reference

Definition of the DataObjectsHierarchyItem class.

```
#include "dataobjectshierarchyitem.h"  
#include "core/abstractdataobject.h"  
#include "core/hierarchytree.h"
```

### Functions

- QIcon **getDataObjectIcon** (AbstractDataObject::ObjectType type)  
*Helper function to assign an appropriate data object icon.*

### 5.120.1 Detailed Description

Definition of the DataObjectsHierarchyItem class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.121 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/models/hierarchy/dataobjectshierarchyitem.h File Reference

Declaration of the DataObjectsHierarchyItem class.

```
#include "models/hierarchy/abstrachierarchyitem.h"  
#include "core/aliasdataset.h"
```

### Classes

- class [QRS::HierarchyModels::DataObjectsHierarchyItem](#)  
*Item to represent a hierarchy of data objects.*

### 5.121.1 Detailed Description

Declaration of the DataObjectsHierarchyItem class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.122 dataobjectshierarchyitem.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef DATAOBJECTSHIERARCHYITEM_H
00009 #define DATAOBJECTSHIERARCHYITEM_H
00010
00011 #include "models/hierarchy/abstracthierarchyitem.h"
00012 #include "core/aliasdataset.h"
00013
00014 namespace QRS
00015 {
00016
00017 namespace PropertiesModels
00018 {
00019 class DataObjectsPropertiesModel;
00020 }
00021
00022 namespace HierarchyModels
00023 {
00024
00026 class DataObjectsHierarchyItem : public AbstractHierarchyItem
00027 {
00028     friend class DataObjectsHierarchyModel;
00029     friend class PropertiesModels::DataObjectsPropertiesModel;
00030
00031 public:
00032     DataObjectsHierarchyItem(Core::DataObjects& dataObjects, Core::HierarchyTree&
        hierarchyDataObjects,
00033                             QString const& text = "Root", QIcon const& icon = QIcon());
00034     DataObjectsHierarchyItem(Core::HierarchyNode* pNode, Core::AbstractDataObject* pDataObject);
00035     DataObjectsHierarchyItem(Core::HierarchyNode* pNode);
00036     int type() const override { return AbstractHierarchyItem::ItemType::kDataObjects; }
00037     Core::AbstractDataObject const* getDataObject() const { return mpDataObject; }
00038
00039 private:
00040     void appendItems(Core::DataObjects& dataObjects, Core::HierarchyNode* pNode);
00041
00042 private:
00043     Core::AbstractDataObject* mpDataObject = nullptr;
00044 };
00045
00046 }
00047
00048 }
00049
00050 #endif // DATAOBJECTSHIERARCHYITEM_H
```

## 5.123 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/hierarchy/dataobjectshierarchyitem.cpp File Reference

Definition of the DataObjectsHierarchyModel class.

## 5.124

**/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.h**  
**File Reference**

181

```
#include <QTreeView>
#include <QMimeData>
#include "dataobjectshierarchymodel.h"
#include "dataobjectshierarchyitem.h"
#include "core/abstractdataobject.h"
#include "core/hierarchytree.h"
```

### 5.123.1 Detailed Description

Definition of the DataObjectsHierarchyModel class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.124 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/hierarchy/dataobjectshierarchymodel.h File Reference

Declaration of the DataObjectsHierarchyModel class.

```
#include "models/hierarchy/abstrachierarchymodel.h"
#include "core/aliasdataset.h"
```

#### Classes

- class [QRS::HierarchyModels::DataObjectsHierarchyModel](#)  
*Tree model to represent and modify a hierarchy of data objects.*

### 5.124.1 Detailed Description

Declaration of the DataObjectsHierarchyModel class.

#### Author

Pavel Lakiza

#### Date

July 2021

## 5.125 dataobjectshierarchymodel.h

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef DATAOBJECTSHIERARCHYMODEL_H
00009 #define DATAOBJECTSHIERARCHYMODEL_H
00010
00011 #include "models/hierarchy/abstracthierarchymodel.h"
00012 #include "core/aliasdataset.h"
00013
00014 namespace QRS
00015 {
00016
00017 namespace Core
00018 {
00019 class HierarchyTree;
00020 }
00021
00022 namespace HierarchyModels
00023 {
00024
00025 class DataObjectsHierarchyItem;
00027 class DataObjectsHierarchyModel : public AbstractHierarchyModel
00028 {
00029     Q_OBJECT
00030
00031 public:
00032     DataObjectsHierarchyModel(Core::DataObjects& dataObjects, Core::HierarchyTree&
00033     hierarchyDataObjects,
00034                               QString const& mimeType, QTreeView* pView = nullptr);
00035     ~DataObjectsHierarchyModel() = default;
00036     void updateContent() override;
00037     void clearContent() override;
00038     bool isEmpty() const;
00039     void selectItem(int iRow);
00040     void selectItemByID(Core::DataIDType id);
00041 signals:
00042     void selected(Core::DataIDType id);
00043     void selectionCleared();
00044
00045 public slots:
00046     void retrieveSelectedItem();
00047     void removeSelectedItem();
00048
00049 private slots:
00050     void renameItem(QStandardItem* pStandardItem);
00051
00052 private:
00053     DataObjectsHierarchyItem* findItemByID(DataObjectsHierarchyItem* pItem, Core::DataIDType const&
00054     id);
00055     void selectItem(DataObjectsHierarchyItem* pItem);
00056
00057 private:
00058     Core::DataObjects& mDataObjects;
00059     Core::HierarchyTree& mHierarchyDataObjects;
00060 };
00061
00062 }
00063
00064 #endif // DATAOBJECTSHIERARCHYMODEL_H

```

## 5.126 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/hierarchy/projecthierarchymodel.cpp File Reference

Definition of the ProjectHierarchyModel class.

```

#include <QTreeView>
#include "projecthierarchymodel.h"
#include "dataobjectshierarchyitem.h"
#include "rodcomponentshierarchyitem.h"

```

**5.126.1 Detailed Description**

---

Definition of the ProjectHierarchyModel class.

**Author**

Pavel Lakiza

**Date**

May 2021

**5.127 /home/qinterfly/Library/Projects/Current/QRod↵  
Systems/src/models/hierarchy/projecthierarchymodel.h File  
Reference**

Declaration of the ProjectHierarchyModel class.

```
#include "models/hierarchy/abstrachhierarchymodel.h"  
#include "core/aliasdata.h"  
#include "core/project.h"
```

**Classes**

- class [QRS::HierarchyModels::ProjectHierarchyModel](#)  
*Project hierarchy representative.*

**5.127.1 Detailed Description**

Declaration of the ProjectHierarchyModel class.

**Author**

Pavel Lakiza

**Date**

May 2021

## 5.128 projecthierarchymodel.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef PROJECTHIERARCHYMODEL_H
00003 #define PROJECTHIERARCHYMODEL_H
00004
00005 #include "models/hierarchy/abstracthierarchymodel.h"
00006 #include "core/aliasdata.h"
00007 #include "core/project.h"
00008
00009 namespace QRS::HierarchyModels
00010 {
00011
00012 class DataObjectsHierarchyItem;
00013 class RodComponentsHierarchyItem;
00014
00015 class ProjectHierarchyModel : public AbstractHierarchyModel
00016 {
00017     Q_OBJECT
00018
00019 public:
00020     ProjectHierarchyModel(QString const& mimeType, QTreeView* pView = nullptr);
00021     void updateContent() override;
00022     void clearContent() override;
00023     void setProject(Core::Project* pProject);
00024
00025 signals:
00026     void selectionValidated(QVector<QRS::HierarchyModels::AbstractHierarchyItem*> validatedItems);
00027
00028 public slots:
00029     void validateItemSelection();
00030
00031 private:
00032     DataObjectsHierarchyItem* retrieveDataObjectsItem();
00033     RodComponentsHierarchyItem* retrieveRodComponentsItem();
00034
00035 private:
00036     Core::Project* mpProject = nullptr;
00037 };
00038
00039 #endif // PROJECTHIERARCHYMODEL_H
```

## 5.129 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/hierarchy/rodcomponentshierarchyitem.cpp File Reference

Definition of the RodComponentsHierarchyItem class.

```
#include "rodcomponentshierarchyitem.h"
#include "core/abstractrodcomponent.h"
#include "core/abstractsectionrodcomponent.h"
#include "core/hierarchytree.h"
```

### Functions

- QIcon **getRodComponentIcon** ([AbstractRodComponent](#) const \*pRodComponent)  
*Helper function to assign an appropriate rod component icon.*



### 5.129.1 Detailed Description

Definition of the RodComponentsHierarchyItem class.

**Author**

Pavel Lakiza

**Date**

June 2021

## 5.130 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/models/hierarchy/rodcomponentshierarchyitem.h File Reference

Declaration of the RodComponentsHierarchyItem class.

```
#include "models/hierarchy/abstrachierarchyitem.h"  
#include "core/aliasdataset.h"
```

### Classes

- class [QRS::HierarchyModels::RodComponentsHierarchyItem](#)  
*Item to represent a hierarchy of rod components.*

### 5.130.1 Detailed Description

Declaration of the RodComponentsHierarchyItem class.

**Author**

Pavel Lakiza

**Date**

June 2021

## 5.131 rodcomponentshierarchyitem.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef RODCOMPONENTSHIERARCHYITEM_H
00009 #define RODCOMPONENTSHIERARCHYITEM_H
00010
00011 #include "models/hierarchy/abstracthierarchyitem.h"
00012 #include "core/aliasdataset.h"
00013
00014 namespace QRS
00015 {
00016
00017 namespace HierarchyModels
00018 {
00019
00021 class RodComponentsHierarchyItem : public AbstractHierarchyItem
00022 {
00023     friend class RodComponentsHierarchyModel;
00024
00025 public:
00026     RodComponentsHierarchyItem(Core::RodComponents& rodComponents, Core::HierarchyTree&
hierarchyRodComponents,
00027                               QString const& text = "Root", QIcon const& icon = QIcon());
00028     RodComponentsHierarchyItem(Core::HierarchyNode* pNode, Core::AbstractRodComponent* pRodComponent);
00029     RodComponentsHierarchyItem(Core::HierarchyNode* pNode);
00030     int type() const override { return AbstractHierarchyItem::ItemType::kRodComponents; }
00031
00032 private:
00033     void appendItems(Core::RodComponents& rodComponents, Core::HierarchyNode* pNode);
00034
00035 private:
00036     Core::AbstractRodComponent* mpRodComponent = nullptr;
00037 };
00038
00039 }
00040
00041 }
00042
00043 #endif // RODCOMPONENTSHIERARCHYITEM_H
```

## 5.132 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/hierarchy/rodcomponentshierarchymodel.cpp File Reference

Definition of the RodComponentsHierarchyModel class.

```
#include <QTreeView>
#include <QMimeType>
#include "rodcomponentshierarchymodel.h"
#include "rodcomponentshierarchyitem.h"
#include "core/abstractrodcomponent.h"
#include "core/hierarchytree.h"
```

### 5.132.1 Detailed Description

Definition of the RodComponentsHierarchyModel class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.133 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/models/hierarchy/rodcomponentshierarchymodel.h File Reference

Declaration of the RodComponentsHierarchyModel class.

```
#include "models/hierarchy/abstracthierarchymodel.h"
#include "core/aliasdataset.h"
```

### Classes

- class [QRS::HierarchyModels::RodComponentsHierarchyModel](#)  
*Tree model to represent and modify a hierarchy of rod components.*

### 5.133.1 Detailed Description

Declaration of the RodComponentsHierarchyModel class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.134 rodcomponentshierarchymodel.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef RODCOMPONENTSHIERARCHYMODEL_H
00009 #define RODCOMPONENTSHIERARCHYMODEL_H
00010
00011 #include "models/hierarchy/abstracthierarchymodel.h"
00012 #include "core/aliasdataset.h"
00013
00014 namespace QRS
00015 {
00016
00017 namespace HierarchyModels
00018 {
00019
00021 class RodComponentsHierarchyModel : public AbstractHierarchyModel
00022 {
00023     Q_OBJECT
00024
00025 public:
00026     RodComponentsHierarchyModel(Core::RodComponents& rodComponents, Core::HierarchyTree&
        hierarchyRodComponents,
00027                                 QString const& mimeType, QTreeView* pView = nullptr);
00028     ~RodComponentsHierarchyModel() = default;
00029     void updateContent() override;
00030     void clearContent() override;
00031     bool isEmpty() const;
00032     void selectItem(int iRow);
00033
00034 signals:
00035     void selected(Core::DataIDType id);
00036     void selectionCleared();
```

```

00037
00038 public slots:
00039     void retrieveSelectedItem();
00040     void removeSelectedItems();
00041
00042 private slots:
00043     void renameItem(QStandardItem* pStandardItem);
00044
00045 private:
00046     Core::RodComponents& mRodComponents;
00047     Core::HierarchyTree& mHierarchyRodComponents;
00048 };
00049
00050 }
00051
00052 }
00053
00054 #endif // RODCOMPONENTSHIERARCHYMODEL_H

```

### 5.135 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/models/properties/abstractpropertiesmodel.cpp File Reference

Definition of the AbstractPropertiesModel class.

```

#include <QTableView>
#include "abstractpropertiesmodel.h"
#include "hierarchy/abstrachierarchyitem.h"
#include "core/hierarchynode.h"

```

#### 5.135.1 Detailed Description

Defintion of the AbstractPropertiesModel class.

##### Author

Pavel Lakiza

##### Date

July 2021

### 5.136 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/models/properties/abstractpropertiesmodel.h File Reference

Declaration of the AbstractPropertiesModel class.

```

#include <QStandardItemModel>

```

**Classes**

- class [QRS::PropertiesModels::AbstractPropertiesModel](#)  
*Model to represent general properties.*

**5.136.1 Detailed Description**

Declaration of the AbstractPropertiesModel class.

**Author**

Pavel Lakiza

**Date**

July 2021

**5.137 abstractpropertiesmodel.h**

[Go to the documentation of this file.](#)

```

00001
00008 #ifndef ABSTRACTPROPERTIESMODEL_H
00009 #define ABSTRACTPROPERTIESMODEL_H
00010
00011 #include <QStandardItemModel>
00012
00013 QT_BEGIN_NAMESPACE
00014 class QTableView;
00015 QT_END_NAMESPACE
00016
00017 namespace QRS
00018 {
00019
00020 namespace HierarchyModels
00021 {
00022 class AbstractHierarchyItem;
00023 }
00024
00025 namespace PropertiesModels
00026 {
00027
00029 class AbstractPropertiesModel : public QStandardItemModel
00030 {
00031     Q_OBJECT
00032
00033 public:
00034     AbstractPropertiesModel(QTableView* pView, QVector<HierarchyModels::AbstractHierarchyItem*>
items);
00035     virtual ~AbstractPropertiesModel() = 0;
00036
00037 signals:
00038     void propertyChanged();
00039
00040 protected slots:
00041     virtual void modifyProperty(QStandardItem* pChangedProperty) = 0;
00042     void modifyDirectoryName(QString const& name);
00043
00044 protected:
00045     void setDirectoryAttributes();
00046     QList<QStandardItem*> preparePropertyRow(int type, QString const& title, QVariant const& value,
bool isValueEditable) const;
00047
00048 protected:
00049     QVector<HierarchyModels::AbstractHierarchyItem*> mItems;
00050     bool mIsDirectory;
00051     QString const mkEmptyProperty = "";
00052
00053 private:
00054     enum PropertyDirectory
00055     {

```

```

00056         kName,
00057         kNumberChildren
00058     };
00059 };
00060
00061 }
00062
00063 }
00064
00065 #endif // ABSTRACTPROPERTIESMODEL_H

```

## 5.138 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/models/properties/dataobjectspropertiesmodel.cpp File Reference

Definition of the DataObjectsPropertiesModel class.

```

#include <QTableView>
#include "dataobjectspropertiesmodel.h"
#include "core/abstractdataobject.h"
#include "core/surfacedataobject.h"
#include "core/hierarchynode.h"
#include "models/hierarchy/abstracthierarchymodel.h"
#include "models/hierarchy/dataobjectshierarchyitem.h"

```

### 5.138.1 Detailed Description

Definition of the DataObjectsPropertiesModel class.

#### Author

Pavel Lakiza

#### Date

May 2021

## 5.139 [/home/qinterfly/Library/Projects/Current/QRod](#) Systems/src/models/properties/dataobjectspropertiesmodel.h File Reference

Declaration of the DataObjectsPropertiesModel class.

```

#include "abstractpropertiesmodel.h"

```

#### Classes

- class [QRS::PropertiesModels::DataObjectsPropertiesModel](#)  
*Model to represent properties of selected data objects.*

### 5.139.1 Detailed Description

Declaration of the DataObjectsPropertiesModel class.

Author

Pavel Lakiza

Date

July 2021

## 5.140 dataobjectspropertiesmodel.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef DATAOBJECTSPROPERTIESMODEL_H
00009 #define DATAOBJECTSPROPERTIESMODEL_H
00010
00011 #include "abstractpropertiesmodel.h"
00012
00013 QT_BEGIN_NAMESPACE
00014 class QTableView;
00015 QT_END_NAMESPACE
00016
00017 namespace QRS
00018 {
00019
00020 namespace HierarchyModels
00021 {
00022 class AbstractHierarchyItem;
00023 }
00024
00025 namespace PropertiesModels
00026 {
00027
00029 class DataObjectsPropertiesModel : public AbstractPropertiesModel
00030 {
00031     Q_OBJECT
00032
00033 public:
00034     DataObjectsPropertiesModel(QTableView* pView, QVector<HierarchyModels::AbstractHierarchyItem>
items);
00035
00036 protected slots:
00037     void modifyProperty(QStandardItem* pChangedProperty) override;
00038
00039 private:
00040     enum PropertyDataObject
00041     {
00042         kName,
00043         kType,
00044         kNumberItems,
00045         kNumberEntities,
00046         kID
00047     };
00048     void setObjectAttributes();
00049 };
00050
00051 }
00052
00053 }
00054
00055 #endif // DATAOBJECTSPROPERTIESMODEL_H
```

## 5.141 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/table/basetablemodel.cpp File Reference

Implementation of the BaseTableModel class.

```
#include <QTreeView>
#include "basetablemodel.h"
#include "core/abstractdataobject.h"
```

### 5.141.1 Detailed Description

Implementation of the BaseTableModel class.

Author

Pavel Lakiza

Date

June 2021

## 5.142 [/home/qinterfly/Library/Projects/Current/QRod↩](#) Systems/src/models/table/basetablemodel.h File Reference

Declaration of the BaseTableModel class.

```
#include <QStandardItemModel>
#include "tablemodelinterface.h"
```

### Classes

- class [QRS::TableModels::BaseTableModel](#)  
*Table model to represent either a scalar or vector data object.*

### 5.142.1 Detailed Description

Declaration of the BaseTableModel class.

Author

Pavel Lakiza

Date

March 2021



## 5.143 basetablemodel.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef BASETABLEMODEL_H
00009 #define BASETABLEMODEL_H
00010
00011 #include <QStandardItemModel>
00012 #include "tablemodelinterface.h"
00013
00014 namespace QRS
00015 {
00016
00017 namespace Core
00018 {
00019     class AbstractDataObject;
00020 }
00021
00022 namespace TableModels
00023 {
00024
00026     class BaseTableModel : public QStandardItemModel, public TableModelInterface
00027     {
00028         Q_OBJECT
00029
00030     public:
00031         BaseTableModel(QWidget* parent = nullptr);
00032         ~BaseTableModel() = default;
00033         void setDataObject(Core::AbstractDataObject* pDataObject);
00034         bool setData(const QModelIndex& indexEdit, const QVariant& value, int role = Qt::EditRole)
00035             override;
00036         void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
00037         void insertLeadingItemAfterSelected(QItemSelectionModel* /*pSelectionModel*/) override { };
00038         void removeSelectedItem(QItemSelectionModel* pSelectionModel) override;
00039         void removeSelectedLeadingItem(QItemSelectionModel* /*pSelectionModel*/) override { };
00040
00041     private:
00042         void updateContent();
00043         void clearContent();
00044
00045     private:
00046         Core::AbstractDataObject* mpDataObject = nullptr;
00047     };
00048 }
00049
00050 }
00051
00052 #endif // BASETABLEMODEL_H
```

## 5.144 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/table/matrixtablemodel.cpp File Reference

Implementation of the MatrixTableModel class.

```
#include <QTreeView>
#include "matrixtablemodel.h"
#include "core/abstractdataobject.h"
```

### 5.144.1 Detailed Description

Implementation of the MatrixTableModel class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.145 [/home/qinterfly/Library/Projects/Current/QRod](#)↔ Systems/src/models/table/matrixtablemodel.h File Reference

Declaration of the MatrixTableModel class.

```
#include <QStandardItemModel>
#include "tablemodelinterface.h"
```

### Classes

- class [QRS::TableModels::MatrixTableModel](#)  
*Table model to represent a matrix data object.*

### 5.145.1 Detailed Description

Declaration of the MatrixTableModel class.

#### Author

Pavel Lakiza

#### Date

March 2021

## 5.146 matrixtablemodel.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef MATRIXTABLEMODEL_H
00009 #define MATRIXTABLEMODEL_H
00010
00011 #include <QStandardItemModel>
00012 #include "tablemodelinterface.h"
00013
00014 namespace QRS
00015 {
00016
00017 namespace Core
00018 {
00019 class AbstractDataObject;
00020 }
00021
00022 namespace TableModels
00023 {
00024
00026 class MatrixTableModel : public QStandardItemModel, public TableModelInterface
00027 {
00028     Q_OBJECT
00029
00030 public:
00031     MatrixTableModel(QWidget* parent = nullptr);
00032     ~MatrixTableModel() = default;
00033     void setDataObject(Core::AbstractDataObject* pDataObject);
00034     bool setData(const QModelIndex& indexEdit, const QVariant& value, int role = Qt::EditRole)
00035         override;
00036     void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
00037     void insertLeadingItemAfterSelected(QItemSelectionModel* /*pSelectionModel*/) override { };
00038     void removeSelectedItem(QItemSelectionModel* pSelectionModel) override;
00039     void removeSelectedLeadingItem(QItemSelectionModel* /*pSelectionModel*/) override { };
00040 }
```

```

00039
00040 private:
00041     void updateContent();
00042     void clearContent();
00043
00044 private:
00045     Core::AbstractDataObject* mpDataObject = nullptr;
00046 };
00047
00048 }
00049
00050 }
00051
00052 #endif // MATRIXTABLEMODEL_H

```

## 5.147 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/surfacetablemodel.cpp File Reference

Implementation of the SurfaceTableModel class.

```

#include <QTreeView>
#include "surfacetablemodel.h"
#include "core/surfacedataobject.h"

```

### 5.147.1 Detailed Description

Implementation of the SurfaceTableModel class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.148 /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/surfacetablemodel.h File Reference

Declaration of the SurfaceTableModel class.

```

#include <QStandardItemModel>
#include "tablemodelinterface.h"

```

#### Classes

- class [QRS::TableModels::SurfaceTableModel](#)  
*Table model to represent a surface data object.*

### 5.148.1 Detailed Description

Declaration of the SurfaceTableModel class.

Author

Pavel Lakiza

Date

March 2021

## 5.149 surfacetablemodel.h

[Go to the documentation of this file.](#)

```
00001
00002 #ifndef SURFACETABLEMODEL_H
00003 #define SURFACETABLEMODEL_H
00004
00005 #include <QStandardItemModel>
00006 #include "tablemodelinterface.h"
00007
00008 namespace QRS
00009 {
00010
00011 namespace Core
00012 {
00013     class SurfaceDataObject;
00014 }
00015
00016 namespace TableModels
00017 {
00018     class SurfaceTableModel : public QStandardItemModel, public TableModelInterface
00019     {
00020     public:
00021         Q_OBJECT
00022         SurfaceTableModel(QWidget* parent = nullptr);
00023         ~SurfaceTableModel() = default;
00024         void setDataObject(Core::SurfaceDataObject* pDataObject);
00025         bool setData(const QModelIndex& indexEdit, const QVariant& value, int role = Qt::EditRole)
00026             override;
00027         void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
00028         void removeSelectedItem(QItemSelectionModel* pSelectionModel) override;
00029         void insertLeadingItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
00030         void removeSelectedLeadingItem(QItemSelectionModel* pSelectionModel) override;
00031     private:
00032         void updateContent();
00033         void clearContent();
00034     private:
00035         Core::SurfaceDataObject* mpDataObject = nullptr;
00036     };
00037 }
00038
00039 #endif // SURFACETABLEMODEL_H
```

## 5.150 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/table/tablemodelinterface.cpp File Reference

Implementation of static functions of TableModelInterface.

```
#include <QStandardItem>
#include "tablemodelinterface.h"
#include "core/array.h"
```

### 5.150.1 Detailed Description

Implementation of static functions of TableModelInterface.

Author

Pavel Lakiza

Date

June 2021

## 5.151 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/models/table/tablemodelinterface.h File Reference

Declaration of the TableModelInterface.

```
#include <QItemSelection>
```

### Classes

- class [QRS::TableModels::TableModelInterface](#)  
*User interface to add and remove items.*

### 5.151.1 Detailed Description

Declaration of the TableModelInterface.

Author

Pavel Lakiza

Date

June 2021

## 5.152 tablemodelinterface.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef TABLEMODELINTERFACE_H
00009 #define TABLEMODELINTERFACE_H
00010
00011 #include <QItemSelection>
00012
00013 QT_BEGIN_NAMESPACE
00014 class QStandardItem;
00015 QT_END_NAMESPACE
00016
00017 namespace QRS
00018 {
00019
00020 namespace Core
00021 {
00022     template <typename T>
00023     class Array;
00024 }
00025
00026 namespace TableModels
00027 {
00028
00029     static const short kNumShowPrecision = 9;
00030
00031     class TableModelInterface
00032     {
00033     public:
00034         virtual void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) = 0;
00035         virtual void insertLeadingItemAfterSelected(QItemSelectionModel* pSelectionModel) = 0;
00036         virtual void removeSelectedItem(QItemSelectionModel* pSelectionModel) = 0;
00037         virtual void removeSelectedLeadingItem(QItemSelectionModel* pSelectionModel) = 0;
00038         virtual ~TableModelInterface() { };
00039         static QStandardItem* makeDoubleItem(double value);
00040         static QList<QStandardItem*> prepareRow(Core::Array<double> const& array, quint32 iRow);
00041         static QList<QStandardItem*> prepareRow(double const& key, Core::Array<double> const& array,
00042             quint32 iRow);
00043         static QList<QStandardItem*> prepareRow(QString const& name, Core::Array<double> const& array,
00044             quint32 iRow);
00045         static QStandardItem* makeLabelItem(QString const& name);
00046     };
00047 }
00048
00049 }
00050
00051 #endif // TABLEMODELINTERFACE_H
```

## 5.153 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/src/render/view3d.cpp File Reference

Implementation of the View3D class.

```
#include <QOpenGLContext>
#include <QPainter>
#include "view3d.h"
```

### 5.153.1 Detailed Description

Implementation of the View3D class.

#### Author

Pavel Lakiza

#### Date

March 2021

## 5.154 /home/qinterfly/Library/Projects/Current/QRodSystems/src/render/view3d.h File Reference

Declaration of the View3D class.

```
#include <QOpenGLWidget>
#include <QOpenGLFunctions>
```

### Classes

- class [QRS::Graph::View3D](#)  
A widget to represent the resulted rod system.

### 5.154.1 Detailed Description

Declaration of the View3D class.

#### Author

Pavel Lakiza

#### Date

March 2021

## 5.155 view3d.h

[Go to the documentation of this file.](#)

```
00001
00008 #ifndef VIEW3D_H
00009 #define VIEW3D_H
00010
00011 #include <QOpenGLWidget>
00012 #include <QOpenGLFunctions>
00013
00014 namespace QRS::Graph
00015 {
00016
00018 class View3D : public QOpenGLWidget, protected QOpenGLFunctions
00019 {
00020     Q_OBJECT
00021
00022 public:
00023     View3D(QWidget* parent = nullptr);
00024     ~View3D() = default;
00025
00026 protected:
00027     void initializeGL() override;
00028     void paintGL() override;
00029
00030 private:
00031     bool mCore;
00032 };
00033
00034 }
00035
00036 #endif // VIEW3D_H
```

