

QRodSystems

Generated by Doxygen 1.9.2



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

## Chapter 4

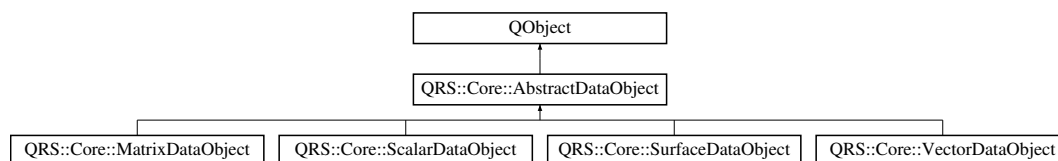
# Class Documentation

### 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 **numberItems** () 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:

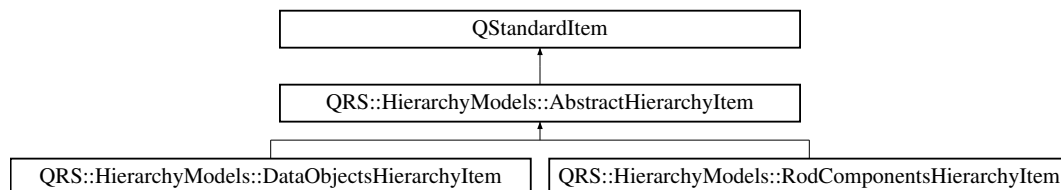
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractdataobject.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractdataobject.cpp](#)

## 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:

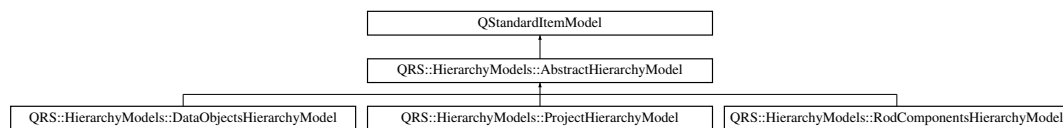
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[abstracthierarchyitem.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[abstracthierarchyitem.cpp](#)

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

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

```
#include <abstracthierarchymodel.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 &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.*

## 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:

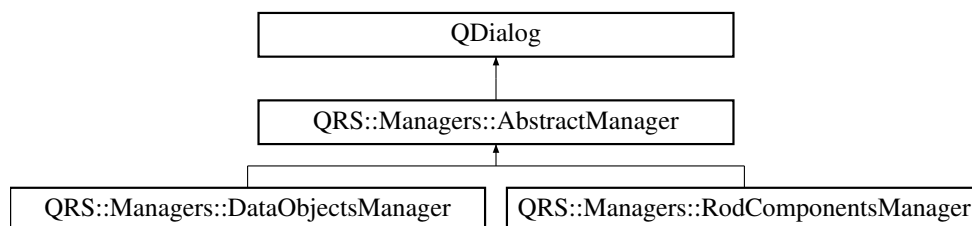
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[abstracthierarchymodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[abstracthierarchymodel.cpp](#)

## 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:

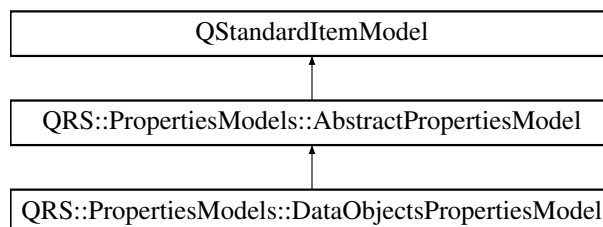
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[abstractmanager.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[abstractmanager.cpp](#)

## 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** ()  
*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

- 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:

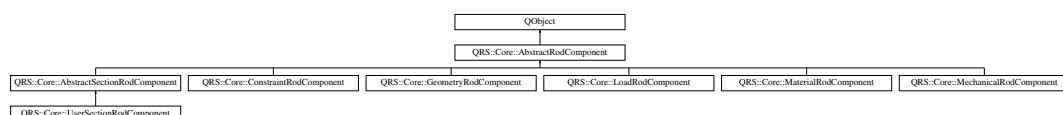
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/[abstractpropertiesmodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/[abstractpropertiesmodel.cpp](#)

## 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:

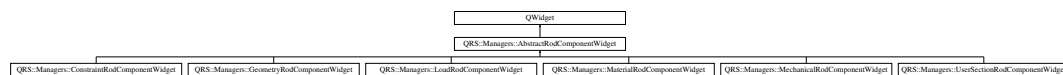
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/abstractrodcomponent.cpp](#)

## 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:

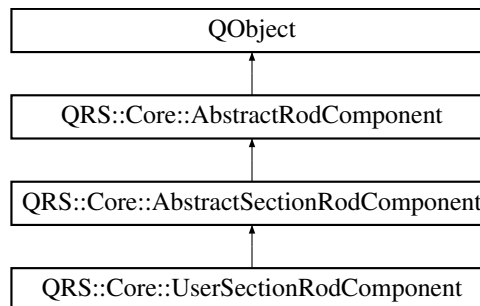
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractrodcomponentwidget.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/abstractrodcomponentwidget.cpp](#)

## 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 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

### Static Public Member Functions

- static quint32 **numberInstances** ()

## Protected Member Functions

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

## 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**

## 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:

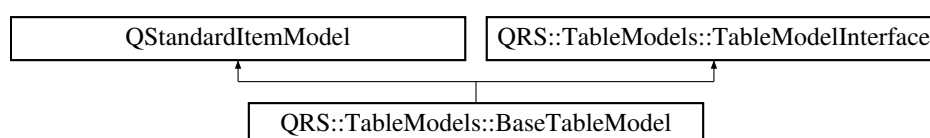
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.cpp](#)

## 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

## 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

### 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 (
    QItemSelectionModel * pSelectionModel ) [override], [virtual]
```

Insert a new item after selected one.

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

#### 4.10.2.2 insertLeadingItemAfterSelected()

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

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

#### 4.10.2.3 removeSelectedItem()

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

Remove an array under selection.

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

#### 4.10.2.4 removeSelectedLeadingItem()

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

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

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

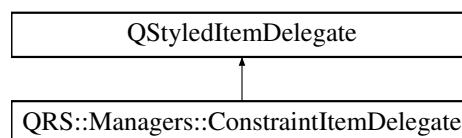
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.h
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/basetablemodel.cpp

## 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:

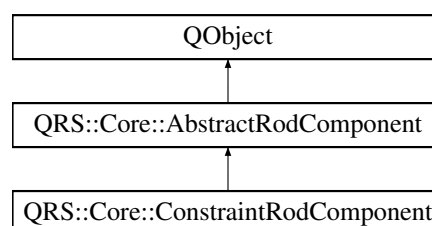
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[constraintitemdelegate.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[constraintitemdelegate.cpp](#)

## 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 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

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Private Attributes

- Constraints **mConstraints**

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

### 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:

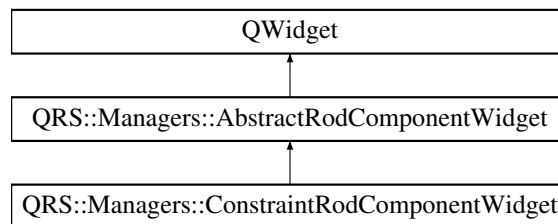
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[constraintrodcomponent.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[constraintrodcomponent.cpp](#)

### 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)

#### 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**
- QTableWidgetItem \* **mpTableConstraint**
- [ConstraintItemDelegate](#) \* **mplItemDelegate**
- ConstraintTypeNames **mTypeNames**
- ConstraintCoordinateSystemNames **mCoordinateSystemNames**

## Additional Inherited Members

### 4.13.1 Detailed Description

Widget to construct constraints of a rod.

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

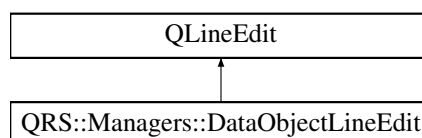
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintrodcomponentwidget.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/constraintrodcomponentwidget.cpp](#)

## 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:

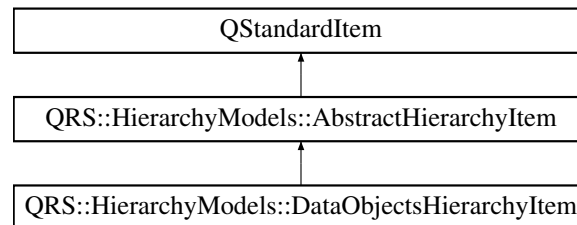
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[dataobjectlineedit.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[dataobjectlineedit.cpp](#)

## 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

### 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

#### 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:

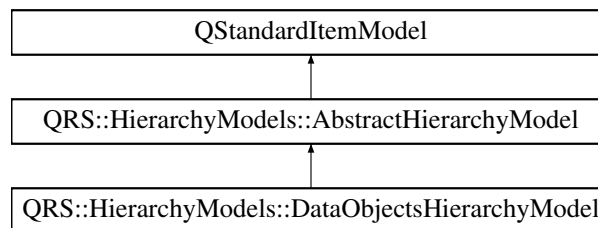
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchyitem.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchyitem.cpp](#)

## 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** ()

## Public Member Functions

- **DataObjectsHierarchyModel** (Core::DataObjects &dataObjects, [Core::HierarchyTree](#) &hierarchyData↔ Objects, QString const &mimeType, QTreeView \*pView=nullptr)  
*Update all the content.*
- 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.*

## Private Slots

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

## Private Member Functions

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

## Private Attributes

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

## Additional Inherited Members

### 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:

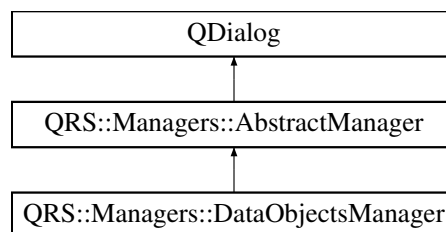
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/dataobjectshierarchymodel.cpp](#)

## 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.*

## Signals

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

## 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** ()

## 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

### 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:

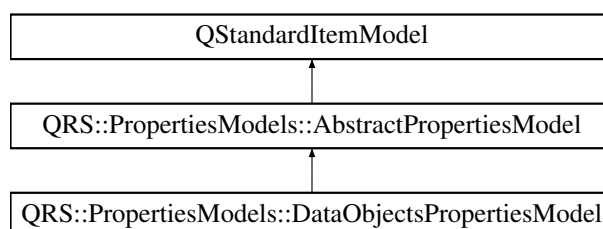
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectsmanager.h>
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/dataobjectsmanager.cpp>

## 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)

## Protected Slots

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

## Private Types

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

## Private Member Functions

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

## Additional Inherited Members

### 4.18.1 Detailed Description

Model to represent properties of selected data objects.

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

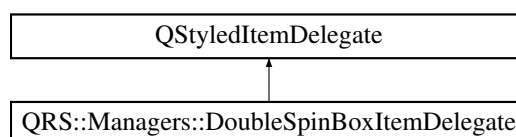
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/dataobjectspropertiesmodel.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/properties/dataobjectspropertiesmodel.cpp](#)

## 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:

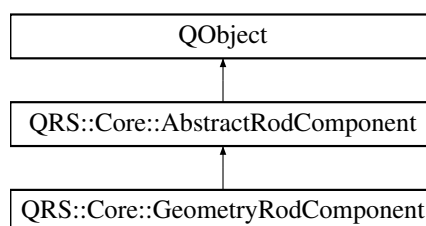
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[doublespinboxitemdelegate.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[doublespinboxitemdelegate.cpp](#)

## 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)

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Private Attributes

- QPointer< **VectorDataObject** const > **mpRadiusVector**
- QPointer< **MatrixDataObject** const > **mpRotationMatrix**

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

### 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:

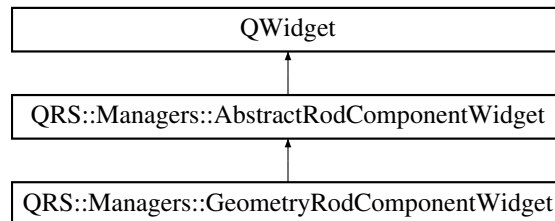
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/geometryrodcomponent.cpp](#)

## 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)

### 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

#### 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:

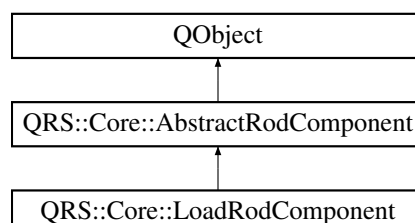
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[hierarchytree.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[hierarchytree.cpp](#)

## 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 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)

## Static Public Member Functions

- static quint32 **numberInstances** ()

## 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

### 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:

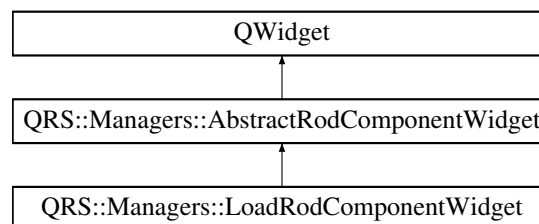
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/loadrodcomponent.cpp](#)

## 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)

## 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

### 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:

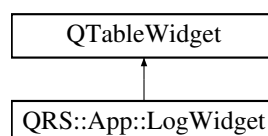
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.h
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/loadrodcomponentwidget.cpp

## 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:

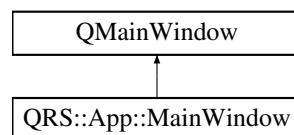
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[logwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[logwidget.cpp](#)

## 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

- **MainWidget** (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 properties of selected objects.*
- 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:

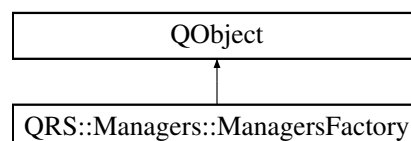
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[mainwindow.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[mainwindow.cpp](#)

## 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:

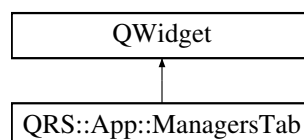
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[managersfactory.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[managersfactory.cpp](#)

## 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:

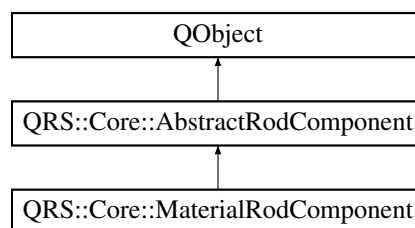
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[controltabs.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/central/[controltabs.cpp](#)

## 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)

## Static Public Member Functions

- static quint32 **numberInstances** ()

## 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

### 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:

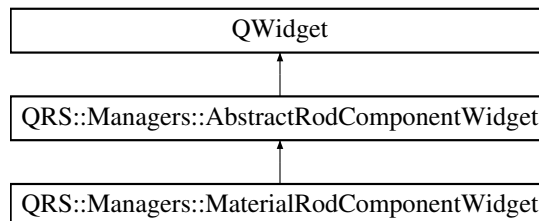
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/materialrodcomponent.cpp](#)

## 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)

### 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

#### 4.31.1 Detailed Description

Widget to construct a material rod component.

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

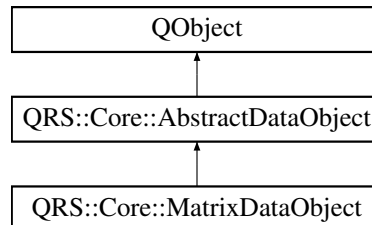
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[materialrodcomponentwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[materialrodcomponentwidget.cpp](#)

## 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.*
- **DataItem** & **addItem** (DataValueType key) override  
*Insert a new item into [MatrixDataObject](#).*
- virtual void **import** (QTextStream &stream) override  
*Import a matrix data object from a file.*

### Static Public Member Functions

- static quint32 **numberInstances** ()

### Static Private Attributes

- static quint32 **smNumInstances** = 0

### Additional Inherited Members

#### 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:

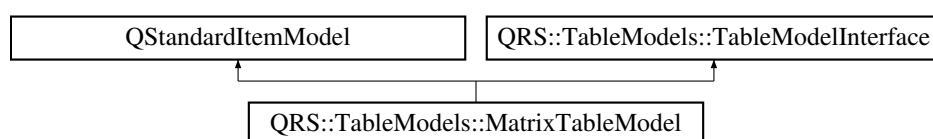
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.h>
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/core/matrixdataobject.cpp>

### 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

## 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

### 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 (
    QItemSelectionModel * pSelectionModel ) [override], [virtual]
```

Insert a new item after selected one.

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

#### 4.33.2.2 insertLeadingItemAfterSelected()

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

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

#### 4.33.2.3 removeSelectedItem()

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

Remove an array under selection.

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

#### 4.33.2.4 removeSelectedLeadingItem()

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

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

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

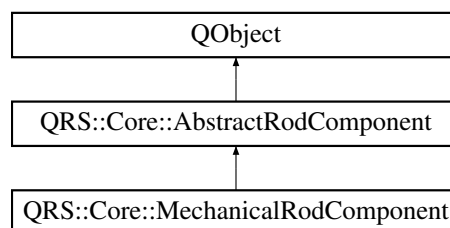
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[matrixtablemodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[matrixtablemodel.cpp](#)

## 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)

## Static Public Member Functions

- static quint32 **numberInstances** ()

## 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

### 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:

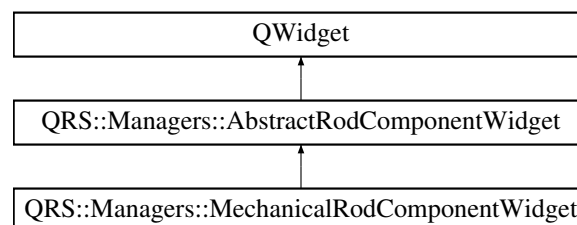
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/mechanicalrodcomponent.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/mechanicalrodcomponent.cpp](#)

## 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)

## 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

### 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:

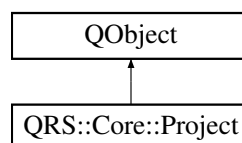
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[mechanicalrodcomponentwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[mechanicalrodcomponentwidget.cpp](#)

## 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:

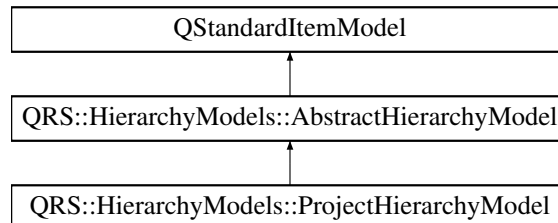
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[project.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[project-base.cpp](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[project-io.cpp](#)

## 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)

### 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.*

### 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

### 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:

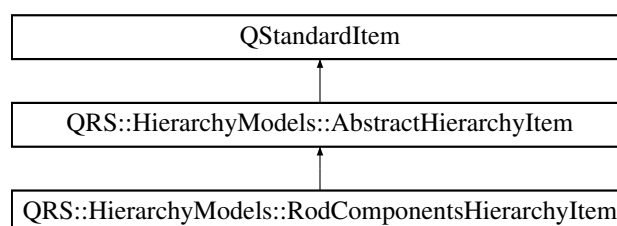
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/projecthierarchymodel.cpp](#)

## 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](#) &hierarchyRodComponents, 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

## 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

### 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:

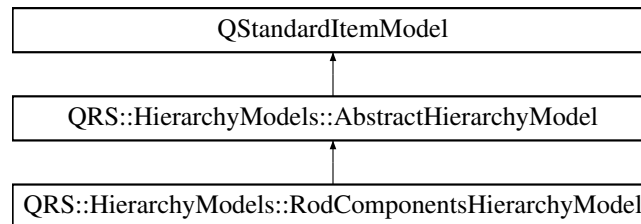
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.h
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/rodcomponentshierarchyitem.cpp

## 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** ()

### 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.*

### Private Slots

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

## Private Attributes

- Core::RodComponents & **mRodComponents**
- Core::HierarchyTree & **mHierarchyRodComponents**

## Additional Inherited Members

### 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:

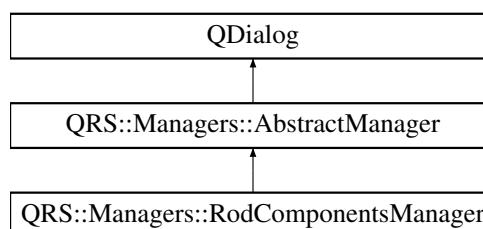
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[rodcomponentshierarchymodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/hierarchy/[rodcomponentshierarchymodel.cpp](#)

## 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.*

## Signals

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

## 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.*

## 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

### 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:

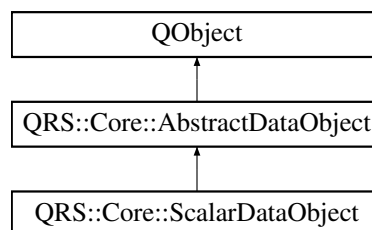
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/array.h

## 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.*

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

### 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:

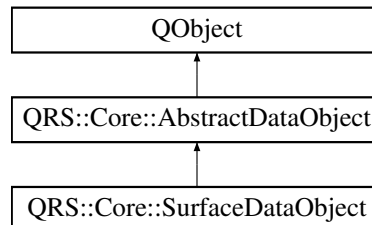
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[scalardataobject.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/[scalardataobject.cpp](#)

## 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.*

### Static Public Member Functions

- static quint32 **numberInstances** ()

### Private Attributes

- DataHolder **mLeadingItems**



## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

### 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:

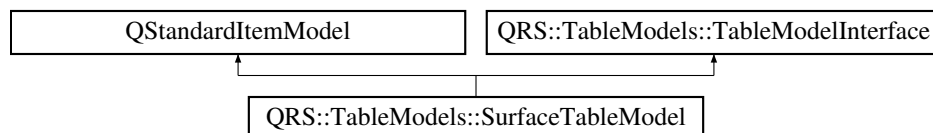
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/surfacedataobject.h
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/core/surfacedataobject.cpp

## 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.*

## 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

### 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 (
    QTableWidgetItem * pSelectionModel ) [override], [virtual]
```

Insert a new item after selected one.

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

#### 4.44.2.2 insertLeadingItemAfterSelected()

```
void SurfaceTableModel::insertLeadingItemAfterSelected (
    QTableWidgetItem * 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:

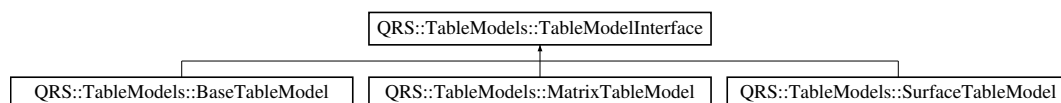
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[surfacetablemodel.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[surfacetablemodel.cpp](#)

## 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:

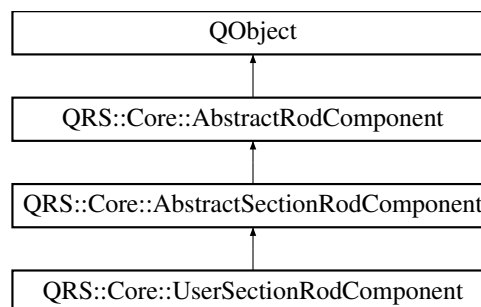
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[tablemodelinterface.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/models/table/[tablemodelinterface.cpp](#)

## 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)

## Additional Inherited Members

### 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:

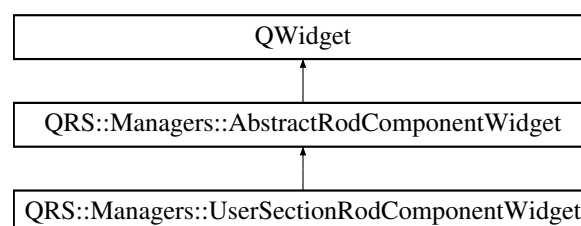
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.h>
- </home/qinterfly/Library/Projects/Current/QRodSystems/src/core/usersectionrodcomponent.cpp>

## 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)

## 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

### 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:

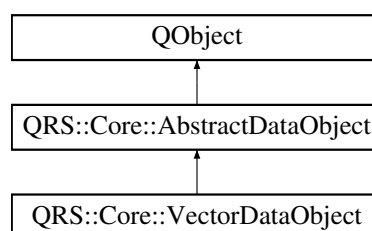
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[usersectionrodcomponentwidget.h](#)
- /home/qinterfly/Library/Projects/Current/QRodSystems/src/managers/[usersectionrodcomponentwidget.cpp](#)

## 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.*

## Static Public Member Functions

- static quint32 **numberInstances** ()

## Static Private Attributes

- static quint32 **smNumInstances** = 0

## Additional Inherited Members

### 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:

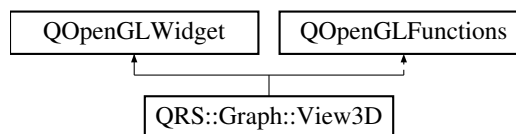
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.h](#)
- [/home/qinterfly/Library/Projects/Current/QRodSystems/src/core/vectordataobject.cpp](#)

## 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.](#)

```

1
2
3
4
5
6
7
8 #ifndef CONTROLTABS_H
9 #define CONTROLTABS_H
10
11 #include <QWidget>
12
13 namespace QRS::App
14 {
15
16
17 class ManagersTab : public QWidget
18 {
19     Q_OBJECT
20
21 public:
22     explicit ManagersTab(QWidget* parent = nullptr);
23     ~ManagersTab() = default;
24
25 signals:
26     void actionDataObjectsTriggered();
27     void actionRodPropertiesTriggered();
28     void actionRodConstructorTriggered();
29 };
30
31 }
32
33
34 #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.](#)

```

1
8 #ifndef LOGWIDGET_H
9 #define LOGWIDGET_H
10
11 #include <QTableWidget>
12
13 namespace QRS::App
14 {
15
16 class LogWidget : public QTableWidget
17 {
18 public:
19     explicit LogWidget(QWidget* parent = nullptr);
20     ~LogWidget() = default;
21     void log(QMsgType messageType, const QString& message);
22 };
23
24
25 }
26
27
28 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef MAINWINDOW_H
9 #define MAINWINDOW_H
10
11 #include <QMainWindow>
12 #include "logwidget.h"
13 #include "core/project.h"
14 #include "models/hierarchy/abstracthierarchyitem.h"
15
16 QT_BEGIN_NAMESPACE
17 namespace Ui
18 {
19     class MainWindow;
20 }
21 class QSettings;
22 class QLabel;
23 class QTableView;
24 QT_END_NAMESPACE
25
26 namespace ads
27 {
28     class CDockManager;
29     class CDockWidget;
30 }
31
32 namespace QRS
33 {
34
35     namespace Managers
36     {
37         class ManagersFactory;
38     }
39
40     namespace HierarchyModels
41     {
42         class ProjectHierarchyModel;
43     }
44
45     namespace App
46     {
47
48         class MainWindow : public QMainWindow
49         {
50         public:
51             Q_OBJECT
52
53             public:
54                 MainWindow(QWidget* parent = nullptr);
55                 ~MainWindow();
56                 void openProject(QString const& filePath);
57                 bool saveProject();
58
59             private:
60                 // Content
61                 void initializeWindow();
62                 void createContent();
63                 void closeEvent(QCloseEvent* pEvent) override;
64                 ads::CDockWidget* createProjectHierarchyWidget();
65                 ads::CDockWidget* createGLWidget();
66                 ads::CDockWidget* createCodeWidget();
67                 ads::CDockWidget* createLogWidget();
68                 ads::CDockWidget* createPropertiesWidget();
69                 void setProjectTitle();
70                 void retrieveRecentProjects();
71                 void addToRecentProjects();
72                 // Signals & Slots
73                 void specifyMenuConnections();
74                 void specifyProjectConnections();
75                 // Project
76                 bool saveProjectChangesDialog();
77                 bool saveProjectHelper(QString const& filePath);
78
79             private slots:
80                 // Project
81                 void createProject();
82                 void openProjectDialog();
83                 void openRecentProject();
84                 bool saveAsProject();
85                 void setModified(bool flag);
86                 // Properties
87                 void representHierarchyProperties(QVector<HierarchyModels::AbstractHierarchyItem*> items);
88                 // Settings
89                 void saveSettings();
90
91         };
92     }
93 }
94
95 #endif // MAINWINDOW_H

```



```

90     void restoreSettings();
91     // Managers
92     void createDataObjectsManager();
93     void createRodComponentsManager();
94     void createRodConstructorManager();
95     // Help
96     void aboutProgram();
97
98 private:
99     // UI
100     Ui::MainWindow* mpUi;
101     ads::CDockManager* mpDockManager;
102     QLabel* mpStatusLabel;
103     QTableView* mpPropertiesWidget;
104     // Models
105     HierarchyModels::ProjectHierarchyModel* mpProjectHierarchyModel = nullptr;
106     // Managers
107     Managers::ManagersFactory* mpManagersFactory = nullptr;
108     // Project data
109     Core::Project* mpProject;
110     // Settings
111     QSharedPointer<QSettings> mpSettings;
112     QString mLastPath;
113     QList<QString> mPathRecentProjects;
114
115 public:
116     static LogWidget* pLogger;
117 };
118
119 inline void throwMessage(QtMsgType type, const QMessageLogContext& /*context*/, const QString& message)
120 {
121     MainWindow::pLogger->log(type, message);
122 }
123
124
125
126
127
128
129 #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.](#)

```

1
8 #ifndef UICONSTANTS_H
9 #define UICONSTANTS_H
10
11 #include <QString>
12
13 namespace QRS::UiConstants
14 {
15
16 namespace Settings
17 {
18     const QString skGeometry      = "geometry";
19     const QString skState         = "state";
20     const QString skDockingState = "dockingState";
21 }
22
23 }
24
25 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef ABSTRACTDATAOBJECT_H
9 #define ABSTRACTDATAOBJECT_H
10
11 #include <QObject>
12 #include <QString>
13 #include <QDataStream>
14 #include <map>
15 #include "array.h"
16 #include "aliasdata.h"
17
18 namespace QRS::Core
19 {
20
21 using DataItemType = Array<DataValueType>;
22 using DataHolder = std::map<DataKeyType, DataItemType>;
23
24 class AbstractDataObject : public QObject
25 {
26 public:
27     enum ObjectType
28     {
29         kScalar,
30         kVector,
31         kMatrix,
32         kSurface
33     };
34     AbstractDataObject(ObjectType type, QString const& name);
35     virtual ~AbstractDataObject() = 0;
36     virtual AbstractDataObject* clone() const = 0;

```

```

38     virtual DataItemType& addItem(DataKeyType key) = 0;
39     void removeItem(DataValueType key);
40     bool changeItemKey(DataKeyType oldKey, DataKeyType newKey, DataHolder* items = nullptr);
41     DataValueType getAvailableItemKey(DataValueType key, DataHolder const* items = nullptr) const;
42     bool setArrayValue(DataKeyType key, DataValueType newValue, IndexType iRow = 0, IndexType iColumn =
0);
43     quint32 numberItems() const { return mItems.size(); }
44     DataHolder const& getItems() { return mItems; }
45     DataIDType id() const { return mID; }
46     ObjectType type() const { return mkType; }
47     QString const& name() const { return mName; }
48     void setName(QString const& name) { mName = name; }
49     static DataIDType maxObjectID() { return smMaxObjectID; }
50     static void setMaxObjectID(DataIDType iMaxObjectID) { smMaxObjectID = iMaxObjectID; }
51     virtual void serialize(QDataStream& stream) const;
52     virtual void deserialize(QDataStream& stream);
53     friend QDataStream& operator<<(QDataStream& stream, AbstractDataObject const& obj);
54     virtual void import(QTextStream& stream) = 0;
55
56 protected:
57     const ObjectType mkType;
58     QString mName;
59     DataIDType mID;
60     DataHolder mItems;
61
62 private:
63     static DataIDType smMaxObjectID;
64 };
65
66 inline QDataStream& operator<<(QDataStream& stream, AbstractDataObject const& obj)
67 {
68     obj.serialize(stream);
69     return stream;
70 }
71
72
73 }
74
75 #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.](#)

```
1
8 #ifndef ABSTRACTRODCOMPONENT_H
9 #define ABSTRACTRODCOMPONENT_H
10
11 #include <QObject>
12 #include <QString>
13 #include <QDataStream>
14 #include "aliasdataset.h"
15
16 namespace QRS::Core
17 {
18
19     class AbstractRodComponent : public QObject
20     {
21     public:
22         enum ComponentType
23         {
24             kGeometry,
```

```

26         kSection,
27         kMaterial,
28         kLoad,
29         kConstraint,
30         kMechanical
31     };
32     AbstractRodComponent(ComponentType componentType, QString const& name);
33     virtual ~AbstractRodComponent() = 0;
34     virtual AbstractRodComponent* clone() const = 0;
35     virtual bool isDataComplete() const = 0;
36     DataIDType id() const { return mID; }
37     ComponentType componentType() const { return mkComponentType; }
38     QString const& name() const { return mName; }
39     void setName(QString const& name) { mName = name; }
40     static DataIDType maxComponentID() { return smMaxComponentID; }
41     static void setMaxComponentID(DataIDType iMaxComponentID) { smMaxComponentID = iMaxComponentID; }
42     virtual void serialize(QDataStream& stream) const = 0;
43     virtual void deserialize(QDataStream& stream, DataObjects const& dataObjects) = 0;
44     friend QDataStream& operator<<(QDataStream& stream, AbstractRodComponent const& component);
45     virtual void resolveReferences(DataObjects const& dataObjects) = 0;
46
47 protected:
48     void writeDataObjectPointer(QDataStream& stream, AbstractDataObject const* pDataObject) const;
49     AbstractDataObject const* readDataObjectPointer(QDataStream& stream, DataObjects const& dataObjects)
50     const;
51     AbstractDataObject const* getDataObject(DataObjects const& dataObjects, DataIDType id) const;
52     AbstractDataObject const* substituteDataObject(DataObjects const& dataObjects, AbstractDataObject
53     const* pDataObject) const;
54
55 protected:
56     ComponentType const mkComponentType;
57     QString mName;
58     DataIDType mID;
59
60 private:
61     static DataIDType smMaxComponentID;
62 };
63 inline QDataStream& operator<<(QDataStream& stream, AbstractRodComponent const& component)
64 {
65     component.serialize(stream);
66     return stream;
67 }
68
69 }
70
71 #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.](#)

```
1
2
3
4
5
6
7
8 #ifndef ABSTRACTSECTIONRODCOMPONENT_H
9 #define ABSTRACTSECTIONRODCOMPONENT_H
10
11 #include <QPointer>
12 #include "abstractrodcomponent.h"
13
14 namespace QRS::Core
15 {
16
17 class ScalarDataObject;
18
19 class AbstractSectionRodComponent : public AbstractRodComponent
20 {
21 public:
22     enum SectionType
23     {
24         kUserDefined
25     };
26     AbstractSectionRodComponent(SectionType sectionType, QString const& name);
27     virtual ~AbstractSectionRodComponent() = 0;
28     static quint32 numberInstances() { return smNumInstances; }
29     void serialize(QDataStream& stream) const override;
30     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
31     void resolveReferences(DataObjects const& dataObjects) override;
32     SectionType sectionType() const { return mkSectionType; }
33
34 protected:
35     void copyIntegratedProperties(AbstractSectionRodComponent const* pSection);
36
37 protected:
38     // Info
39 }
```

```

40     SectionType const mkSectionType;
41     static quint32 smNumInstances;
42     // Area
43     QPointer<ScalarDataObject const> mpArea;
44     // Inertia moments
45     QPointer<ScalarDataObject const> mpInertiaMomentTorsional;
46     QPointer<ScalarDataObject const> mpInertiaMomentX;
47     QPointer<ScalarDataObject const> mpInertiaMomentY;
48     // Center coordinates
49     QPointer<ScalarDataObject const> mpCenterCoordinateX;
50     QPointer<ScalarDataObject const> mpCenterCoordinateY;
51 };
52
53 }
54
55 #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.](#)

```

1
8 #ifndef ALIASDATA_H
9 #define ALIASDATA_H
10
11 #include <QtGlobal>
12
13 namespace QRS::Core
14 {
15
16     using DataValueType = double;
17     using DataKeyType = double;
18     using DataIDType = quint64;
19
20 }
21
22 #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.](#)

```
1
2
3 8 #ifndef ALIASDATASET_H
4 9 #define ALIASDATASET_H
5 10
6 11 #include <unordered_map>
7 12 #include "aliasdata.h"
8 13
9 14 namespace QRS::Core
10 15 {
11 16
12 17 class AbstractDataObject;
13 18 class AbstractRodComponent;
14 19
15 20 using DataObjects = std::unordered_map<DataIDType, AbstractDataObject*>;
16 21 using RodComponents = std::unordered_map<DataIDType, AbstractRodComponent*>;
17 22
18 23 }
19 24
20 25 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef ARRAY_H
9 #define ARRAY_H
10
11 #include <QDebug>
12
13 namespace QRS::Core
14 {
15
16 using IndexType = quint32;
17
18 template<typename T>
19 class Array
20 {
21 private:
22     template <typename U> struct Row;
23 public:
24     Array(IndexType numRows = 0, IndexType numCols = 0);
25     Array(Array<T> const& another);
26     Array(Array<T>&& another);
27     ~Array();
28     T* data() { return mpData; }
29     void resize(IndexType numRows, IndexType numCols);
30     void removeColumn(IndexType iRemoveColumn);
31     void swapColumns(IndexType iFirstColumn, IndexType iSecondColumn);
32     IndexType rows() const { return mNumRows; };
33     IndexType cols() const { return mNumCols; };
34     IndexType size() const { return mNumRows * mNumCols; }
35     Row<T> operator[] (IndexType iRow) { return Row<T>(&mpData[mNumCols * iRow]); };
36     Row<T> operator[] (IndexType iRow) const { return Row<T>(&mpData[mNumCols * iRow]); };
37     Array& operator=(Array<T> const& another);
38     template<typename K> friend QDebug operator<<(QDebug stream, Array<K>& array);
39     template<typename K> friend QDataStream& operator<<(QDataStream& stream, Array<K> const& array);
40     template<typename K> friend QDataStream& operator>>(QDataStream& stream, Array<K>& array);
41 private:
42     IndexType mNumRows;
43     IndexType mNumCols;
44     T* mpData = nullptr;
45     template <typename U>
46     struct Row
47     {
48         Row() = delete;
49         Row(T* pData) : pRow(pData) { };
50         ~Row() { }
51         T& operator[] (IndexType iCol) { return pRow[iCol]; }
52         T const& operator[] (IndexType iCol) const { return pRow[iCol]; }
53         T* pRow;
54     };
55 };
56
57 template<typename K>
58 inline QDebug operator<<(QDebug stream, Array<K>& array)
59 {
60     IndexType const& nRows = array.mNumRows;
61     IndexType const& nCols = array.mNumCols;
62     stream = stream.noquote();
63     stream << QString("Array size: %1 x %2").arg(QString::number(nRows), QString::number(nCols));
64 }

```

```

72     stream « Qt::endl;
73     for (IndexType iRow = 0; iRow != nRows; ++iRow)
74     {
75         for (IndexType jCol = 0; jCol != nCols; ++jCol)
76             stream « QString::number(array[iRow][jCol]);
77         stream « Qt::endl;
78     }
79     return stream;
80 }
81
82 template<typename K>
83 inline QDataStream& operator<<(QDataStream& stream, Array<K> const& array)
84 {
85     stream « array.mNumRows « array.mNumCols;
86     IndexType const& size = array.size();
87     for (IndexType i = 0; i != size; ++i)
88         stream « array.mpData[i];
89     return stream;
90 }
91
92 template<typename K>
93 inline QDataStream& operator>>(QDataStream& stream, Array<K>& array)
94 {
95     delete[] array.mpData;
96     stream » array.mNumRows » array.mNumCols;
97     IndexType const& size = array.size();
98     array.mpData = new K[size];
99     for (IndexType i = 0; i != size; ++i)
100         stream » array.mpData[i];
101     return stream;
102 }
103
104 }
105
106 }
107
108 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef CONSTRAINTRODCOMPONENT_H
9 #define CONSTRAINTRODCOMPONENT_H
10
11 #include "abstractrodcomponent.h"
12
13 namespace QRS::Core
14 {
15
16
17 class ConstraintRodComponent : public AbstractRodComponent
18 {
19 public:
20     enum ConstraintType
21     {
22         kDisplacementX, kDisplacementY, kDisplacementZ,
23         kRotationX, kRotationY, kRotationZ
24     };
25     enum ConstraintCoordinateSystem
26     {
27         kGlobal,
28         kLocal
29     };
30     using Constraints = std::map<ConstraintType, ConstraintCoordinateSystem>;
31     ConstraintRodComponent(QString const& name);
32     ~ConstraintRodComponent();
33     AbstractRodComponent* clone() const override;
34     bool isDataComplete() const override { return mConstraints.size() != 0; };
35     static quint32 numberInstances() { return smNumInstances; }
36     void serialize(QDataStream& stream) const override;
37     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
38     void resolveReferences(DataObjects const&) override {};
39     bool isConstraintExist(ConstraintType type) const;
40     void setConstraint(ConstraintType type, ConstraintCoordinateSystem coordinateSystem);
41     bool removeConstraint(ConstraintType type);
42     Constraints const& constraints() const { return mConstraints; }
43
44 private:
45     static quint32 smNumInstances;
46     Constraints mConstraints;
47 };
48
49 }
50
51
52 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef GEOMETRYRODCOMPONENT_H
9 #define GEOMETRYRODCOMPONENT_H
10
11 #include <QPointer>
12 #include "abstractrodcomponent.h"
13
14 namespace QRS::Core
15 {
16
17 class VectorDataObject;
18 class MatrixDataObject;
19
20 class GeometryRodComponent : public AbstractRodComponent
21 {
22 public:
23     GeometryRodComponent(QString const& name);
24     ~GeometryRodComponent();
25     AbstractRodComponent* clone() const override;
26     bool isDataComplete() const override;
27     static quint32 numberInstances() { return smNumInstances; }
28     void serialize(QDataStream& stream) const override;
29     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
30     void resolveReferences(DataObjects const& dataObjects) override;
31     // Getters
32     VectorDataObject const* radiusVector() const { return mpRadiusVector; }
33     MatrixDataObject const* rotationMatrix() const { return mpRotationMatrix; }
34     // Setters
35     void setRadiusVector(VectorDataObject const* pRadiusVector) { mpRadiusVector = pRadiusVector; }
36     void setRotationMatrix(MatrixDataObject const* pRotationMatrix) { mpRotationMatrix = pRotationMatrix; }
37
38 private:
39     static quint32 smNumInstances;
40     QPointer<VectorDataObject const> mpRadiusVector;
41     QPointer<MatrixDataObject const> mpRotationMatrix;
42 };
43
44
45
46
47 #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.](#)

```
1
2
3
4
5
6
7
8 #ifndef HIERARCHYNODE_H
9 #define HIERARCHYNODE_H
10
11 #include <QVariant>
12 #include <QDataStream>
13
14 namespace QRS::Core
15 {
16
17     class HierarchyNode
18     {
19     public:
20         friend class HierarchyTree;
21         enum NodeType
22         {
23             kObject,
24             kDirectory
25         };
26         HierarchyNode(NodeType type, QVariant value);
27         ~HierarchyNode() = default;
28         void appendChild(HierarchyNode* node);
29         bool hasParent() const { return mpParent; }
30         bool hasChild() const { return mpFirstChild; }
31         bool hasNextSibling() const { return mpNextSibling; }
32         HierarchyNode* parent() { return mpParent; }
33         HierarchyNode* firstChild() { return mpFirstChild; }
34         HierarchyNode* nextSibling() { return mpNextSibling; }
35         NodeType type() const { return mType; }
36         QVariant& value() { return mValue; }
37         HierarchyNode* groupNodes(HierarchyNode* pChildNode);
```



```

40     bool setBefore(HierarchyNode* pSetNode);
41     bool setAfter(HierarchyNode* pSetNode);
42     quint32 numberChildren() const;
43
44 private:
45     void excludeNodeFromHierarchy();
46     bool isSetAllowed(HierarchyNode const* pNode) const;
47     bool isParentOf(HierarchyNode const* pNode) const;
48     quint32 countNodes(HierarchyNode* pNode, quint32& numNodes) const;
49
50 private:
51     HierarchyNode* mpParent = nullptr;
52     HierarchyNode* mpFirstChild = nullptr;
53     HierarchyNode* mpNextSibling = nullptr;
54     HierarchyNode* mpPreviousSibling = nullptr;
55     NodeType mType;
56     QVariant mValue;
57 };
58
59 }
60
61 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef HIERARCHYTREE_H
9 #define HIERARCHYTREE_H
10
11 #include <QDebug>
12 #include "hierarchytree.h"
13
14 namespace QRS::Core
15 {
16
17     class HierarchyTree
18     {
19     public:
20         HierarchyTree();
21         HierarchyTree(HierarchyTree& another);
22         HierarchyTree(HierarchyTree&& another);
23         HierarchyTree(HierarchyNode* pRootNode);
24         HierarchyTree(QDataStream& stream, int numNodes);
25         HierarchyTree& operator=(HierarchyTree const& another);
26         HierarchyTree& operator=(HierarchyTree&& another);
27         ~HierarchyTree();
28         void clear();
29         void appendNode(HierarchyNode* pNode);
30         bool removeNode(HierarchyNode::NodeType type, QVariant const& value);
31         void removeNode(HierarchyNode* pNode);
32         void changeNodeValue(HierarchyNode::NodeType type, QVariant const& oldValue, QVariant const&
33             newValue);
34         HierarchyNode* root() { return mpRootNode; }
35         HierarchyTree clone() const;
36         HierarchyNode* findNode(HierarchyNode* pBaseNode, HierarchyNode::NodeType type, QVariant const&
37             value) const;
38         quint32 size() const;
39         friend QDebug operator<<(QDebug stream, HierarchyTree& tree);
40         friend QDataStream& operator<<(QDataStream& stream, HierarchyTree const& tree);
41     private:
42         HierarchyNode* copyNode(HierarchyNode* pBaseNode, quint32 relativeLevel) const;
43         void removeNodeSiblings(HierarchyNode* pNode);
44         void printNode(quint32 level, HierarchyNode* pNode, QDebug stream) const;
45         void writeNode(HierarchyNode* pNode, QDataStream& stream) const;
46     private:
47         HierarchyNode* mpRootNode = nullptr;
48     };
49
50
51
52 inline QDebug operator<<(QDebug stream, HierarchyTree& tree)

```

```
53 {  
54     tree.printNode(0, tree.mpRootNode, stream);  
55     return stream;  
56 }  
57  
59 inline QDataStream& operator<<(QDataStream& stream, HierarchyTree const& tree)  
60 {  
61     tree.writeNode(tree.mpRootNode, stream);  
62     return stream;  
63 }  
64  
65 }  
66  
67 #endif // HIERARCHYTREE_H
```

## 5.40 /home/qinterfly/Library/Projects/Current/QRod↵ Systems/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/QRod↵ Systems/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.](#)

```

1
2 #ifndef LOADRODCOMPONENT_H
3 #define LOADRODCOMPONENT_H
4
5 #include <QPointer>
6 #include "abstractrodcomponent.h"
7
8 namespace QRS::Core
9 {
10
11 class ScalarDataObject;
12 class VectorDataObject;
13
14 class LoadRodComponent : public AbstractRodComponent
15 {
16 public:
17     enum LoadType
18     {
19         kNone,
20         kForcedDisplacements, kForcedRotations,
21         kPointForce, kPointMoment,
22         kPointMass, kPointInertiaMoment,
23         kPointLinearDamper, kPointRotationalDamper,
24         kDistributedForce, kDistributedMoment,
25         kAerodynamicFlow,
26         kAcceleration,
27         kInnerLiquidFlow,
28         kDisplacementDamping, kRotationDamping
29     };
30     LoadRodComponent(QString const& name);
31     ~LoadRodComponent();
32     AbstractRodComponent* clone() const override;
33     bool isDataComplete() const override;
34     static quint32 numberInstances() { return smNumInstances; }
35     void serialize(QDataStream& stream) const override;
36     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
37     void resolveReferences(DataObjects const& dataObjects) override;
38     // Getters
39     LoadType loadType() const { return mLoadType; }
40     VectorDataObject const* directionVector() const { return mpDirectionVector; }
41     ScalarDataObject const* longitudinalFunction() const { return mpLongitudinalFunction; }
42     ScalarDataObject const* timeCoefficient() const { return mpTimeCoefficient; }
43     VectorDataObject const* timeRotationVector() const { return mpTimeRotationVector; }
44     DataValueType multiplier() const { return mMultiplier; }
45     bool isFollowing() const { return mIsFollowing; }
46     // Setters
47     void setType(LoadType type) { mLoadType = type; }
48     void setDirectionVector(VectorDataObject const* pDirectionVector) { mpDirectionVector = pDirectionVector; }
49     void setLongitudinalFunction(ScalarDataObject const* pLongitudinalFunction) { mpLongitudinalFunction = pLongitudinalFunction; }
50     void setTimeCoefficient(ScalarDataObject const* pTimeCoefficient) { mpTimeCoefficient = pTimeCoefficient; }
51     void setTimeRotationVector(VectorDataObject const* pTimeRotationVector) { mpTimeRotationVector = pTimeRotationVector; }
52     void setMultiplier(DataValueType value) { mMultiplier = value; }
53     void setFollowingState(bool isFollowing) { mIsFollowing = isFollowing; }
54 private:

```

```

63     static quint32 smNumInstances;
64     LoadType mLoadType = kNone;
65     QPointer<VectorDataObject const> mpDirectionVector;
66     QPointer<ScalarDataObject const> mpLongitudinalFunction;
67     QPointer<ScalarDataObject const> mpTimeCoefficient;
68     QPointer<VectorDataObject const> mpTimeRotationVector;
69     DataValueType mMultiplier = 1.0;
70     bool mIsFollowing = false;
71 };
72
73 }
74
75 #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.](#)

```

1
2 #ifndef MATERIALRODCOMPONENT_H
3 #define MATERIALRODCOMPONENT_H
4
5 #include <QPointer>
6 #include "abstractrodcomponent.h"
7
8 namespace QRS::Core
9 {
10
11 class ScalarDataObject;
12
13 class MaterialRodComponent : public AbstractRodComponent
14 {
15 public:
16     MaterialRodComponent(QString const& name);
17     ~MaterialRodComponent();
18     AbstractRodComponent* clone() const override;
19     bool isDataComplete() const override;
20     static quint32 numberInstances() { return smNumInstances; }
21     void serialize(QDataStream& stream) const override;
22     void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
23     void resolveReferences(DataObjects const& dataObjects) override;
24     // Getters
25     ScalarDataObject const* elasticModulus() const { return mpElasticModulus; }
26     ScalarDataObject const* shearModulus() const { return mpShearModulus; }
27     ScalarDataObject const* poissonsRatio() const { return mpPoissonsRatio; }
28     ScalarDataObject const* density() const { return mpDensity; }
29     // Setters
30     void setElasticModulus(ScalarDataObject const* pElasticModulus) { mpElasticModulus = pElasticModulus; }
31     void setShearModulus(ScalarDataObject const* pShearModulus) { mpShearModulus = pShearModulus; }
32     void setPoissonsRatio(ScalarDataObject const* pPoissonsRatio) { mpPoissonsRatio = pPoissonsRatio; }
33     void setDensity(ScalarDataObject const* pDensity) { mpDensity = pDensity; }
34
35 private:
36     static quint32 smNumInstances;
37     QPointer<ScalarDataObject const> mpElasticModulus;
38     QPointer<ScalarDataObject const> mpShearModulus;
39     QPointer<ScalarDataObject const> mpPoissonsRatio;
40     QPointer<ScalarDataObject const> mpDensity;
41 };
42
43 #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.](#)

```

1
2 #ifndef MATRIXDATAOBJECT_H
3 #define MATRIXDATAOBJECT_H
4
5 #include "abstractdataobject.h"
6
7 namespace QRS::Core
8 {
9
10 class MatrixDataObject : public AbstractDataObject
11 {
12 public:
13     MatrixDataObject(QString const& name);
14     ~MatrixDataObject();
15     AbstractDataObject* clone() const override;
16     DataItemType& addItem(DataValueType key) override;
17     static quint32 numberInstances() { return smNumInstances; }
18     virtual void import(QTextStream& stream) override;
19
20 private:
21     static quint32 smNumInstances;
22 };
23
24 }
25
26 #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.](#)

```

1
2 #ifndef MECHANICALRODCOMPONENT_H
3 #define MECHANICALRODCOMPONENT_H
4
5 #include <QPointer>
6 #include "abstractrodcomponent.h"
7
8 namespace QRS::Core
9 {
10     class ScalarDataObject;
11
12     class MechanicalRodComponent : public AbstractRodComponent
13     {
14     public:
15         MechanicalRodComponent(QString const& name);
16         ~MechanicalRodComponent();
17         AbstractRodComponent* clone() const override;
18         bool isDataComplete() const override { return true; }
19         static quint32 numberInstances() { return smNumInstances; }
20         void serialize(QDataStream& stream) const override;
21         void deserialize(QDataStream& stream, DataObjects const& dataObjects) override;
22         void resolveReferences(DataObjects const& dataObjects) override;
23         // Getters
24         // Stiffness distribution
25         ScalarDataObject const* tensionStiffness() const { return mpTensionStiffness; }
26         ScalarDataObject const* torsionalStiffness() const { return mpTorsionalStiffness; }
27         ScalarDataObject const* bendingStiffnessX() const { return mpBendingStiffnessX; }
28         ScalarDataObject const* bendingStiffnessY() const { return mpBendingStiffnessY; }
29         // Mass distribution
30         ScalarDataObject const* linearMassDensity() const { return mpLinearMassDensity; }
31         ScalarDataObject const* inertiaMassMomentX() const { return mpInertiaMassMomentX; }
32         ScalarDataObject const* inertiaMassMomentY() const { return mpInertiaMassMomentY; }
33         ScalarDataObject const* inertiaMassMomentZ() const { return mpInertiaMassMomentZ; }
34         // Eccentricity
35         ScalarDataObject const* eccentricityX() const { return mpEccentricityX; }
36         ScalarDataObject const* eccentricityY() const { return mpEccentricityY; }
37         // Contact diameter
38         ScalarDataObject const* contactDiameter() const { return mpContactDiameter; }
39         // Setters
40         // Stiffness distribution
41         void setTensionStiffness(ScalarDataObject const* pTensionStiffness) { mpTensionStiffness = pTensionStiffness; }
42         void setTorsionalStiffness(ScalarDataObject const* pTorsionalStiffness) { mpTorsionalStiffness = pTorsionalStiffness; }
43         void setBendingStiffnessX(ScalarDataObject const* pBendingStiffnessX) { mpBendingStiffnessX = pBendingStiffnessX; }
44         void setBendingStiffnessY(ScalarDataObject const* pBendingStiffnessY) { mpBendingStiffnessY = pBendingStiffnessY; }
45         // Mass distribution

```

```

54     void setLinearMassDensity(ScalarDataObject const* pLinearMassDensity) { mpLinearMassDensity =
pLinearMassDensity; }
55     void setInertiaMassMomentX(ScalarDataObject const* pInertiaMassMomentX) { mpInertiaMassMomentX =
pInertiaMassMomentX; }
56     void setInertiaMassMomentY(ScalarDataObject const* pInertiaMassMomentY) { mpInertiaMassMomentY =
pInertiaMassMomentY; }
57     void setInertiaMassMomentZ(ScalarDataObject const* pInertiaMassMomentZ) { mpInertiaMassMomentZ =
pInertiaMassMomentZ; }
58     // Eccentricity
59     void setEccentricityX(ScalarDataObject const* pEccentricityX) { mpEccentricityX = pEccentricityX; }
60     void setEccentricityY(ScalarDataObject const* pEccentricityY) { mpEccentricityY = pEccentricityY; }
61     // Contact diameter
62     void setContactDiameter(ScalarDataObject const* pContactDiameter) { mpContactDiameter =
pContactDiameter; }
63
64 private:
65     static quint32 smNumInstances;
66     // Stiffness distribution
67     QPointer<ScalarDataObject const> mpTensionStiffness;
68     QPointer<ScalarDataObject const> mpTorsionalStiffness;
69     QPointer<ScalarDataObject const> mpBendingStiffnessX;
70     QPointer<ScalarDataObject const> mpBendingStiffnessY;
71     // Mass distribution
72     QPointer<ScalarDataObject const> mpLinearMassDensity;
73     QPointer<ScalarDataObject const> mpInertiaMassMomentX;
74     QPointer<ScalarDataObject const> mpInertiaMassMomentY;
75     QPointer<ScalarDataObject const> mpInertiaMassMomentZ;
76     // Eccentricity
77     QPointer<ScalarDataObject const> mpEccentricityX;
78     QPointer<ScalarDataObject const> mpEccentricityY;
79     // Contact diameter
80     QPointer<ScalarDataObject const> mpContactDiameter;
81 };
82
83 }
84
85 #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 "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"

```

### 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 "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"
#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.](#)

```

1
2
3
4
5
6
7
8 #ifndef PROJECT_H
9 #define PROJECT_H
10
11 #include <QObject>
12 #include "aliasdataset.h"
13 #include "array.h"
14 #include "hierarchytree.h"
15 #include "abstractdataobject.h"
16 #include "abstractrodcomponent.h"
17 #include "abstractsectionrodcomponent.h"
18
19 QT_BEGIN_NAMESPACE
20 class QString;
21 QT_END_NAMESPACE
22
23 namespace QRS::HierarchyModels
24 {
25     class ProjectHierarchyModel;
26 }
27
28 namespace QRS::Managers
29 {
30     class ManagersFactory;
31 }
32
33 namespace QRS::Core
34 {
35
36     class Project : public QObject
37     {
38     public:
39         Q_OBJECT
40
41         friend class QRS::HierarchyModels::ProjectHierarchyModel;
42         friend class QRS::Managers::ManagersFactory;
43
44     public:
45         Project(QString const& name);
46         Project(QString const& path, QString const& fileName);
47         virtual ~Project();
48         // Data objects
49         DataIDType numberDataObjects() const { return mDataObjects.size(); }
50         AbstractDataObject* addDataObject(AbstractDataObject::ObjectType type);
51         DataObjects cloneDataObjects() const;
52         HierarchyTree cloneHierarchyDataObjects() const { return mHierarchyDataObjects.clone(); }
53         // Rod components
54         DataIDType numberRodComponents() const { return mRodComponents.size(); }
55         AbstractRodComponent* addGeometry();
56         AbstractRodComponent* addCrossSection(AbstractSectionRodComponent::SectionType sectionType);
57         AbstractRodComponent* addMaterial();
58         AbstractRodComponent* addLoad();
59         AbstractRodComponent* addConstraint();
60         AbstractRodComponent* addMechanical();
61         RodComponents cloneRodComponents() const;
62         HierarchyTree cloneHierarchyRodComponents() const { return mHierarchyRodComponents.clone(); }
63         // Getters and setters
64         QString const& name() const { return mName; }
65         QString const& filePath() const { return mFilePath; }
66         static QString const& getExtension() { return skProjectExtension; }
67         void importDataObjects(QString const& path, QString const& fileName);
68
69     signals:
70         // Data objects
71         void dataObjectsSubstituted();
72         void propertiesDataObjectsChanged();
73         // Rod components
74         void rodComponentsSubstituted();
75         void propertiesRodComponentsChanged();
76         // Project hierarchy
77         void projectHierarchyChanged();
78
79     public slots:
80         bool save(QString const& dir, QString const& fileName);
81         void setDataObjects(QRS::Core::DataObjects const& dataObjects, QRS::Core::HierarchyTree const&
            hierarchyDataObjects);
82         void setRodComponents(QRS::Core::RodComponents const& rodComponents, QRS::Core::HierarchyTree const&
            hierarchyRodComponents);
83
84     private:
85         void emplaceRodComponent(AbstractRodComponent* pRodComponent);
86
87     private:

```

```

89     quint32 mID;
90     QString mName;
91     QString mFilePath;
92     DataObjects mDataObjects;
93     HierarchyTree mHierarchyDataObjects;
94     RodComponents mRodComponents;
95     HierarchyTree mHierarchyRodComponents;
96     static const QString skProjectExtension;
97 };
98
99 }
100
101 #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.](#)

```

1
2 #ifndef SCALARDATAOBJECT_H
3 #define SCALARDATAOBJECT_H
4
5 #include "abstractdataobject.h"
6
7 namespace QRS::Core
8 {
9
10 class ScalarDataObject : public AbstractDataObject
11 {
12 public:
13     ScalarDataObject(QString const& name);
14     ~ScalarDataObject();
15     AbstractDataObject* clone() const override;
16     DataItemTypes& addItem(DataValueType key) override;
17     static quint32 numberInstances() { return smNumInstances; }
18     virtual void import(QTextStream& stream) override;
19
20 private:
21     static quint32 smNumInstances;
22 };
23
24 #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.](#)

```
1
2
3
4
5
6
7
8 #ifndef SURFACEDATAOBJECT_H
9 #define SURFACEDATAOBJECT_H
10
11 #include "abstractdataobject.h"
12
13 namespace QRS::Core
14 {
15
16
17 class SurfaceDataObject : public AbstractDataObject
18 {
19 public:
20     SurfaceDataObject(QString const& name);
21     ~SurfaceDataObject();
22     AbstractDataObject* clone() const override;
23     DataItemType& addItem(DataValueType key) override;
24     DataKeyType addLeadingItem(DataValueType key);
25     void removeLeadingItem(DataValueType key);
26     bool changeLeadingItemKey(DataKeyType oldKey, DataKeyType newKey);
27     quint32 numberLeadingItems() const { return mLeadingItems.size(); }
28     DataHolder& getLeadingItems() { return mLeadingItems; }
29     static quint32 numberInstances() { return smNumInstances; }
30     void serialize(QDataStream& stream) const override;
31     virtual void deserialize(QDataStream& stream) override;
32     virtual void import(QTextStream& stream) override;
33
34 private:
35     static quint32 smNumInstances;
36     DataHolder mLeadingItems;
37 };
38
39 }
40
41 #endif // SURFACEDATAOBJECT_H
```



## 5.62 /home/qinterfly/Library/Projects/Current/QRodSystems/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/QRodSystems/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.](#)

```

1
2
3
4
5
6
7
8 #ifndef USERSECTIONRODCOMPONENT_H
9 #define USERSECTIONRODCOMPONENT_H
10
11 #include "abstractsectionrodcomponent.h"
12 #include "core/scalardataobject.h"
13
14 namespace QRS::Core
15 {
16
17
18 class UserSectionRodComponent : public AbstractSectionRodComponent
19 {
20 public:
21     UserSectionRodComponent(QString const& name);
22     AbstractRodComponent* clone() const override;
23     bool isDataComplete() const override;
24     // Getters
25     // Area
26     ScalarDataObject const* area() const { return mpArea; }
27     // Inertia moments
28     ScalarDataObject const* inertiaMomentTorsional() const { return mpInertiaMomentTorsional; }
29     ScalarDataObject const* inertiaMomentX() const { return mpInertiaMomentX; }
30     ScalarDataObject const* inertiaMomentY() const { return mpInertiaMomentY; }
31     // Center coordinates
32     ScalarDataObject const* centerCoordinateX() const { return mpCenterCoordinateX; }
33     ScalarDataObject const* centerCoordinateY() const { return mpCenterCoordinateY; }
34     // Setters
35     // Area
36     void setArea(ScalarDataObject const* pArea) { mpArea = pArea; }
37     // Inertia moments
38     void setInertiaMomentTorsional(ScalarDataObject const* pInertiaMomentTorsional) {
39         mpInertiaMomentTorsional = pInertiaMomentTorsional; }
40     void setInertiaMomentX(ScalarDataObject const* pInertiaMomentX) { mpInertiaMomentX = pInertiaMomentX; }
41     void setInertiaMomentY(ScalarDataObject const* pInertiaMomentY) { mpInertiaMomentY = pInertiaMomentY; }
42     // Center coordinates
43     void setCenterCoordinateX(ScalarDataObject const* pCenterCoordinateX) { mpCenterCoordinateX =
44         pCenterCoordinateX; }
45     void setCenterCoordinateY(ScalarDataObject const* pCenterCoordinateY) { mpCenterCoordinateY =
46         pCenterCoordinateY; }
47
48 };
49 #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.](#)

```

1
8 #ifndef UTILITIES_H
9 #define UTILITIES_H
10
11 #include <QSharedPointer>
12 #include "abstractdataobject.h"
13
14 class QFile;
15 class QString;
16
17 namespace QRS
18 {
19
20 namespace Utilities
21 {
22
23 namespace File
24 {
25
26 QPair<Core::AbstractDataObject::ObjectType, QSharedPointer<QFile>> getDataObjectFile(QString const& path,
27     QString const& fileName);
28
29 QString loadFileContent(QString const& path);
30
31 }
32
33 }
34
35 #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.](#)

```
1
2
3
4
5
6
7
8 #ifndef VECTORDATAOBJECT_H
9 #define VECTORDATAOBJECT_H
10
11 #include "abstractdataobject.h"
12
13 namespace QRS::Core
14 {
15
16
17 class VectorDataObject : public AbstractDataObject
18 {
19 public:
20     VectorDataObject(QString const& name);
21     ~VectorDataObject();
22     AbstractDataObject* clone() const override;
23     DataItemType& addItem(DataValueType key) override;
24     static quint32 numberInstances() { return smNumInstances; }
25     virtual void import(QTextStream& stream) override;
26
27 private:
28     static quint32 smNumInstances;
29 };
30
31 }
32
33 #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.](#)

```
1
2
3 #ifndef ABSTRACTMANAGER_H
4 #define ABSTRACTMANAGER_H
5
6 #include <QDialog>
7
8 QT_BEGIN_NAMESPACE
9 class QSettings;
10 class QToolBar;
11 QT_END_NAMESPACE
12
13 namespace ads
14 {
15     class CDockManager;
16 }
17
18 namespace QRS
19 {
20     namespace Managers
21     {
22     {
23         class AbstractManager : public QDialog
24         {
25             Q_OBJECT
26
27         public:
28             enum ManagerType
29             {
30                 kDataObjects,
31                 kRodComponents,
32                 kRodConstructor
33             };
34         };
35     }
36 }
```

```

41     AbstractManager(QString& lastPath, QSettings& settings,
42                     ManagerType type, QString groupName, QWidget* parent = nullptr);
43     virtual ~AbstractManager() = 0;
44     void saveSettings();
45     void restoreSettings();
46
47 signals:
48     void closed(QRS::Managers::AbstractManager::ManagerType type);
49
50 public slots:
51     virtual void apply() = 0;
52
53 protected:
54     void closeEvent(QCloseEvent* pEvent) override;
55     void setToolBarShortcutHints(QToolBar* pToolBar);
56
57 protected:
58     // Dock manager
59     ads::CDockManager* mpDockManager = nullptr;
60     // Data
61     QString& mLastPath;
62
63 private:
64     QSettings& mSettings;
65     ManagerType const mkType;
66     QString const mkGroupName;
67 };
68
69 }
70
71 }
72
73 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef ABSTRACTRODCOMPONENTWIDGET_H
9 #define ABSTRACTRODCOMPONENTWIDGET_H
10
11 #include <QWidget>
12 #include "core/aliasdata.h"
13
14 namespace QRS
15 {
16
17     namespace Core
18     {
19         class AbstractDataObject;
20     }
21
22     namespace Managers
23     {
24
25         class DataObjectLineEdit;
26
27         using DataObjectSetFun = std::function<void(Core::AbstractDataObject const*)>;
28
29         class AbstractRodComponentWidget : public QWidget
30         {
31         public:
32             Q_OBJECT
33
34         public:
35             AbstractRodComponentWidget(QString const& mimeType, QWidget* parent = nullptr);
36             virtual ~AbstractRodComponentWidget() = 0;
37
38         signals:
39             void modified();
40             void editDataObjectRequested(Core::DataIDType id);
41
42         protected:
43             void setDataObjectEditConnections(DataObjectLineEdit* pEdit, DataObjectSetFun& setFun);
44
45         protected:
46             QString const mkMimeType();
47     };
48
49 }
50
51 }
52
53 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef CONSTRAINTITEMDELEGATE_H
9 #define CONSTRAINTITEMDELEGATE_H
10
11 #include <QStyledItemDelegate>
12 #include "core/constraintrodcomponent.h"
13
14 namespace QRS::Managers
15 {
16
17 using ConstraintTypeNames = std::map<Core::ConstraintRodComponent::ConstraintType, QString>;
18 using ConstraintCoordinateSystemNames =
19     std::map<Core::ConstraintRodComponent::ConstraintCoordinateSystem, QString>;
20
21 class ConstraintItemDelegate : public QStyledItemDelegate
22 {
23     Q_OBJECT
24
25 public:
26     ConstraintItemDelegate(Core::ConstraintRodComponent const& constraintRodComponent,
27         ConstraintTypeNames const& types,
28         ConstraintCoordinateSystemNames const& coordinateSystems, QObject* parent =
29             nullptr);
30     QWidget* createEditor(QWidget* pCell, const QStyleOptionViewItem& option, const QModelIndex& index)
31         const override;
32     void setEditorData(QWidget* pEditor, const QModelIndex& index) const override;
33     void setModelData(QWidget* pEditor, QAbstractItemModel* pModel, const QModelIndex& index) const
34         override;
35     void updateEditorGeometry(QWidget* pEditor, const QStyleOptionViewItem& option, const QModelIndex&
36         index) const override;
37
38 signals:
39     void typeCreated(int iRow) const;
40     void typeChanged(int iRow, Core::ConstraintRodComponent::ConstraintType oldType) const;
41     void coordinateSystemChanged(int iRow) const;
42
43 private:
44     Core::ConstraintRodComponent const& mConstraintRodComponent;
45     ConstraintTypeNames const& mTypes;
46     ConstraintCoordinateSystemNames const& mCoordinateSystems;
47 };
48
49 }
50
51 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef CONSTRAINTRODCOMPONENTWIDGET_H
9 #define CONSTRAINTRODCOMPONENTWIDGET_H
10
11 #include "abstractrodcomponentwidget.h"
12 #include "constraintitemdelegate.h"
13
14 QT_BEGIN_NAMESPACE
15 class QTableWidget;
16 class QTableWidgetItem;
17 class QToolBar;
18 QT_END_NAMESPACE
19
20 namespace QRS
21 {
22
23 namespace Managers
24 {
25
26
27 class ConstraintRodComponentWidget : public AbstractRodComponentWidget
28 {
29 public:
30     ConstraintRodComponentWidget(Core::ConstraintRodComponent& constraintRodComponent, QWidget* parent =
        nullptr);
31     ~ConstraintRodComponentWidget();
32
33 private:
34     // Creating
35     void createContent();
36     QToolBar* createToolBar();
37     void createTableWidget();
38     // Interaction
39     void addRow();
40     void removeSelectedRows();
41     void representConstraintData();
42     // Helpers
43     void setTableHeight();
44     void specifyConstraintNames();
45     QVariant getItemData(int iRow, int iColumn);
46
47 private slots:
48     void setConstraintData(int iRow);
49
50 private:
51     Core::ConstraintRodComponent& mConstraintRodComponent;
52     QTableWidget* mpTableConstraint;
53     ConstraintItemDelegate* mpItemDelegate;
54     ConstraintTypeNames mTypeNames;
55     ConstraintCoordinateSystemNames mCoordinateSystemNames;
56 };
57
58 }
59
60 }
61
62 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef DATAOBJECTLINEEDIT_H
9 #define DATAOBJECTLINEEDIT_H
10
11 #include <QLineEdit>
12 #include "core/abstractdataobject.h"
13
14 namespace QRS
15 {
16
17 namespace Managers
18 {
19
20
21 class DataObjectLineEdit : public QLineEdit
22 {
23     Q_OBJECT
24 public:
25     DataObjectLineEdit(Core::AbstractDataObject const* pDataObject, Core::AbstractDataObject::ObjectType
        type,
26                     QString const& mimeType, QWidget* parent = nullptr);
27     ~DataObjectLineEdit() = default;
28
29 signals:
30     void selected(Core::AbstractDataObject const* pDataObject);
31     void editRequested(Core::DataIDType id);
32
33 private slots:
34     void showContextMenu(const QPoint& point);
35     void reset();
36     void edit();
37
38 private:
39     void dragEnterEvent(QDragEnterEvent* pEvent) override;
40     void dropEvent(QDropEvent* pEvent) override;
41     void keyPressEvent(QKeyEvent* pEvent) override;
42     void mouseDoubleClickEvent(QMouseEvent* pEvent) override;
43
44 private:
45     Core::AbstractDataObject const* mpDataObject;
46     Core::AbstractDataObject::ObjectType mType;
47     QString const mkMimeType;
48 };
49
50 }
51
52 }
53
54 #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"

```

```
#include "dataobjectsmanager.h"
#include "central/uiconstants.h"
#include "core/scalardataobject.h"
#include "core/vectordataobject.h"
#include "core/matrixdataobject.h"
#include "core/surfacedataobject.h"
#include "core/utilities.h"
#include "models/table/basetablemodel.h"
#include "models/table/matrixtablemodel.h"
#include "models/table/surfacetablemodel.h"
#include "models/hierarchy/dataobjectshierarchy.h"
#include "doublespinboxitemdelegate.h"
```

## Functions

- void **setToolBarShortcutHints** (QToolBar \*pToolBar)
- QIcon **getDataObjectIcon** (AbstractDataObject::ObjectType type)

*Helper function to assign an appropriate data object icon.*

### 5.87.1 Detailed Description

Implementation of the DataObjectsManager class.

#### Author

Pavel Lakiza

#### Date

June 2021

## 5.88 /home/qinterfly/Library/Projects/Current/QRod↔ Systems/src/managers/dataobjectsmanager.h File Reference

Declaration of the DataObjectsManager class.

```
#include <unordered_map>
#include "abstractmanager.h"
#include "core/aliasdata.h"
#include "core/aliasdataset.h"
#include "core/hierarchytree.h"
```

## Classes

- class [QRS::Managers::DataObjectsManager](#)

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



### 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.](#)

```

1
2 #ifndef DATAOBJECTSMANAGER_H
3 #define DATAOBJECTSMANAGER_H
4
5 #include <unordered_map>
6 #include "abstractmanager.h"
7 #include "core/aliasdata.h"
8 #include "core/aliasdataset.h"
9 #include "core/hierarchytree.h"
10
11 QT_BEGIN_NAMESPACE
12 class QTreeView;
13 class QSettings;
14 QT_END_NAMESPACE
15
16 namespace ads
17 {
18     class CDockManager;
19     class CDockWidget;
20 }
21
22 namespace QRS
23 {
24     namespace TableModels
25     {
26         class TableModelInterface;
27         class BaseTableModel;
28         class MatrixTableModel;
29         class SurfaceTableModel;
30     }
31
32     namespace HierarchyModels
33     {
34         class DataObjectsHierarchyModel;
35     }
36 }
37
38 namespace Managers
39 {
40     class DataObjectsManager : public AbstractManager
41     {
42     public:
43         explicit DataObjectsManager(Core::DataObjects&& dataObjects, Core::HierarchyTree&&
44             hierarchyDataObjects,
45             QString& lastPath, QSettings& settings, QWidget* parent = nullptr);
46         ~DataObjectsManager();
47         void selectDataObject(int iRow);
48         void selectDataObjectByID(Core::DataIDType id);
49         Core::DataObjects const& getDataObjects() { return mDataObjects; };
50
51     signals:
52         void applied(Core::DataObjects const& dataObjects, Core::HierarchyTree const& hierarchyDataObjects);
53
54     public slots:
55         void apply() override;
56         Core::AbstractDataObject* addScalar();
57     };
58 }
59
60 #endif

```

```

66     Core::AbstractDataObject* addVector();
67     Core::AbstractDataObject* addMatrix();
68     Core::AbstractDataObject* addSurface();
69     void insertItemAfterSelected();
70     void insertLeadingItemAfterSelected();
71     void removeSelectedItem();
72     void removeSelectedLeadingItem();
73     void importDataObjects();
74
75 private:
76     // Content
77     void createContent();
78     ads::CDockWidget* createDataTableWidget();
79     ads::CDockWidget* createHierarchyWidget();
80     QLayout* createDialogControls();
81     // Helpers
82     void emplaceDataObject(Core::AbstractDataObject* pDataObject);
83     bool isDataTableModifiable();
84     void importDataObject(QString const& path, QString const& fileName);
85     // Selection
86     void representDataObject(Core::DataIDType id);
87     void clearDataObjectRepresentation();
88
89 private:
90     // Widgets
91     QTreeView* mpTreeDataObjects;
92     QTreeView* mpDataTable;
93     // Data
94     Core::DataObjects mDataObjects;
95     Core::HierarchyTree mHierarchyDataObjects;
96     // Models
97     TableModels::TableModelInterface* mpTableModelInterface = nullptr;
98     TableModels::BaseTableModel* mpBaseTableModel;
99     TableModels::MatrixTableModel* mpMatrixTableModel;
100    TableModels::SurfaceTableModel* mpSurfaceTableModel;
101    HierarchyModels::DataObjectsHierarchyModel* mpTreeDataObjectsModel;
102 };
103
104 }
105
106 }
107
108 #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.](#)

```
1
2
3
4
5
6
7
8 #ifndef DOUBLESPINBOXITEMDELEGATE_H
9 #define DOUBLESPINBOXITEMDELEGATE_H
10
11 #include <QStyledItemDelegate>
12
13 namespace QRS::Managers
14 {
15
16
17 class DoubleSpinBoxItemDelegate : public QStyledItemDelegate
18 {
19 public:
20     DoubleSpinBoxItemDelegate(QObject* parent = nullptr);
21     QWidget* createEditor(QWidget* parent, const QStyleOptionViewItem& option, const QModelIndex& index)
22         const override;
23     void setEditorData(QWidget* pEditor, const QModelIndex& index) const override;
24     void setModelData(QWidget* pEditor, QAbstractItemModel* pModel, const QModelIndex& index) const
25         override;
26     void updateEditorGeometry(QWidget* pEditor, const QStyleOptionViewItem& option, const QModelIndex&
27         index) const override;
28
29 };
30
31
32
33 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef GEOMETRYRODCOMPONENTWIDGET_H
9 #define GEOMETRYRODCOMPONENTWIDGET_H
10
11 #include "abstractrodcomponentwidget.h"
12
13 namespace QRS
14 {
15
16 namespace Core
17 {
18 class GeometryRodComponent;
19 class AbstractDataObject;
20 }
21
22 namespace Managers
23 {
24
25
26 class GeometryRodComponentWidget : public AbstractRodComponentWidget
27 {
28 public:
29     GeometryRodComponentWidget(Core::GeometryRodComponent& geometryRodComponent, QString const& mimeType,
30                               QWidget* parent = nullptr);
31 private:
32     void createContent();
33     template<typename T>
34     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
35
36 private:
37     Core::GeometryRodComponent& mGeometryRodComponent;
38 };
39
40 }
41
42 }
43
44 #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.](#)

```
1  
8 #ifndef LOADRODCOMPONENTWIDGET_H  
9 #define LOADRODCOMPONENTWIDGET_H  
10  
11 #include "abstractrodcomponentwidget.h"  
12 #include "core/loadrodcomponent.h"  
13  
14 QT_BEGIN_NAMESPACE  
15 class QComboBox;  
16 class QLabel;  
17 QT_END_NAMESPACE  
18  
19 namespace QRS  
20 {  
21  
22 namespace Core  
23 {  
24 class AbstractDataObject;
```

```

25 }
26
27 namespace Managers
28 {
29
31 class LoadRodComponentWidget : public AbstractRodComponentWidget
32 {
33 public:
34     LoadRodComponentWidget(Core::LoadRodComponent& loadRodComponent, QString const& mimeType, QWidget*
        parent = nullptr);
35
36 private:
37     void createContent();
38     QLayout* createBaseLayout();
39     QWidget* createTimeGroup();
40     QLayout* createLoadTypeLayout();
41     QComboBox* createLoadTypeComboBox();
42     template<typename T>
43     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
44     void setLoadUnits(Core::LoadRodComponent::LoadType type);
45
46 private:
47     Core::LoadRodComponent& mLoadRodComponent;
48     QLabel* mpLoadRodUnits;
49 };
50
51 }
52
53 }
54
55 #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.](#)

```
1
2
3 8 #ifndef MANAGERSFACTORY_H
4 9 #define MANAGERSFACTORY_H
5 10
6 11 #include <QObject>
7 12 #include "abstractmanager.h"
8 13
9 14 QT_BEGIN_NAMESPACE
10 15 class QSettings;
11 16 QT_END_NAMESPACE
12 17
13 18 namespace QRS
14 19 {
15 20
16 21 namespace Core
17 22 {
18 23 class Project;
19 24 }
20 25
21 26 namespace Managers
22 27 {
23 28
24 29 class DataObjectsManager;
25 30 class RodComponentsManager;
26 31
27 32 class ManagersFactory : public QObject
28 33 {
29 34     Q_OBJECT
30 35
31 36 public:
32 37     ManagersFactory(Core::Project& project, QString& lastPath, QSettings& settings, QWidget* parent);
33 38     ~ManagersFactory();
34 39
```



---

```

40     bool createManager(AbstractManager::ManagerType type);
41     bool deleteManager(AbstractManager::ManagerType type);
42     AbstractManager* manager(AbstractManager::ManagerType type);
43
44 private:
45     void specifyConnections(DataObjectsManager* pManager);
46     void specifyConnections(RodComponentsManager* pManager);
47
48 private:
49     Core::Project& mProject;
50     QString& mLastPath;
51     QSettings& mSettings;
52     QWidget* mpParent;
53     std::unordered_map<AbstractManager::ManagerType, AbstractManager*> mManagers;
54 };
55
56 }
57
58 }
59
60 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef MATERIALRODCOMPONENTWIDGET_H
9 #define MATERIALRODCOMPONENTWIDGET_H
10
11 #include "abstractrodcomponentwidget.h"
12
13 namespace QRS
14 {
15
16 namespace Core
17 {
18 class MaterialRodComponent;
19 class AbstractDataObject;
20 }
21
22 namespace Managers
23 {
24
25
26 class MaterialRodComponentWidget : public AbstractRodComponentWidget
27 {
28 public:
29     MaterialRodComponentWidget(Core::MaterialRodComponent& materialRodComponent, QString const& mimeType,
30                               QWidget* parent = nullptr);
31 private:
32     void createContent();
33     QWidget* createModuliGroup();
34     QLayout* createBaseLayout();
35     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
36
37 private:
38     Core::MaterialRodComponent& mMaterialRodComponent;
39 };
40
41 }
42
43 }
44
45 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef MECHANICALRODCOMPONENTWIDGET_H
9 #define MECHANICALRODCOMPONENTWIDGET_H
10
11 #include "abstractrodcomponentwidget.h"
12
13 namespace QRS
14 {
15
16 namespace Core
17 {
18 class AbstractDataObject;
19 class MechanicalRodComponent;
20 }
21
22 namespace Managers
23 {
24
25
26 class MechanicalRodComponentWidget : public AbstractRodComponentWidget
27 {
28 public:
29     MechanicalRodComponentWidget(Core::MechanicalRodComponent& mechanicalRodComponent, QString const&
        mimeType, QWidget* parent = nullptr);
30
31 private:
32     void createContent();
33     QWidget* createStiffnessGroup();
34     QWidget* createMassGroup();
35     QWidget* createEccentricityGroup();
36     QLayout* createContactDiameterLayout();
37     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
38
39 private:
40     Core::MechanicalRodComponent& mMechanicalRodComponent;
41 };
42
43 }
44
45 }
46
47 #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](#) \*pRodComponent, 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/QRodSystems/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.](#)

```

1
2
3
4
5
6
7
8 #ifndef RODCOMPONENTSMANAGER_H
9 #define RODCOMPONENTSMANAGER_H
10
11 #include "managers/abstractmanager.h"
12 #include "core/aliasdataset.h"
13 #include "core/hierarchytree.h"
14 #include "core/abstractsectionrodcomponent.h"
15
16 QT_BEGIN_NAMESPACE
17 class QTreeView;
18 QT_END_NAMESPACE
19
20 namespace ads
21 {
22 class CDockWidget;
23 }
24
25 namespace QRS
26 {
27
28 namespace HierarchyModels
29 {
30 class DataObjectsHierarchyModel;
31 class RodComponentsHierarchyModel;
32 }
33
34 namespace Managers
35 {
36
37 class RodComponentsManager : public AbstractManager
38 {
39     Q_OBJECT
40
41 public:
42     RodComponentsManager(Core::DataObjects& dataObjects, Core::HierarchyTree& hierarchyDataObjects,
43                         Core::RodComponents&& rodComponents, Core::HierarchyTree&&
44                         hierarchyRodComponents,
45                         QString& lastPath, QSettings& settings, QWidget* parent = nullptr);
46     ~RodComponentsManager();
47     void selectRodComponent(int iRow);
48     void updateDataObjects();
49
50 signals:
51     void applied(Core::RodComponents const& rodComponents, Core::HierarchyTree const&
52                 hierarchyRodComponents);
53     void editDataObjectRequested(Core::DataIDType id);
54 public slots:
55     void apply() override;

```

```

56     Core::AbstractRodComponent* addGeometry();
57     Core::AbstractRodComponent* addSection(Core::AbstractSectionRodComponent::SectionType sectionType);
58     Core::AbstractRodComponent* addMaterial();
59     Core::AbstractRodComponent* addLoad();
60     Core::AbstractRodComponent* addConstraint();
61     Core::AbstractRodComponent* addMechanical();
62     void resolveRodComponentsReferences();
63
64 private:
65     // Content
66     void createContent();
67     QLayout* createDialogControls();
68     ads::CDockWidget* createHierarchyRodComponentsWidget();
69     ads::CDockWidget* createConstructorDockWidget();
70     ads::CDockWidget* createHierarchyDataObjectsWidget();
71     // Helpers
72     void emplaceRodComponent(Core::AbstractRodComponent* pRodComponent);
73     // Selection
74     void representRodComponent(Core::DataIDType id);
75     void clearRodComponentRepresentation();
76     // Toolbars
77     QToolBar* createMainToolBar();
78     QWidget* makeGeometryToolBar();
79     QWidget* makeSectionsToolBar();
80     QWidget* makeBoundaryConditionsToolBar();
81     QWidget* makeLoadingToolBar();
82     QWidget* makeMaterialToolBar();
83     QWidget* makeMechanicalToolBar();
84     QWidget* makeModificationToolBar();
85
86 private:
87     // Widgets
88     ads::CDockWidget* mpComponentDockWidget;
89     QTreeView* mpTreeRodComponents;
90     // Data objects
91     Core::DataObjects& mDataObjects;
92     Core::HierarchyTree& mHierarchyDataObjects;
93     // Rod components data
94     Core::RodComponents mRodComponents;
95     Core::HierarchyTree mHierarchyRodComponents;
96     // Models
97     HierarchyModels::DataObjectsHierarchyModel* mpTreeDataObjectsModel;
98     HierarchyModels::RodComponentsHierarchyModel* mpTreeRodComponentsModel;
99 };
100
101 }
102
103 }
104
105 #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.](#)

```
1
2
3 8 #ifndef USERSECTIONRODCOMPONENTWIDGET_H
4 9 #define USERSECTIONRODCOMPONENTWIDGET_H
5 10
6 11 #include "abstractrodcomponentwidget.h"
7 12
8 13 namespace QRS
9 14 {
10 15
11 16 namespace Core
12 17 {
13 18 class UserSectionRodComponent;
14 19 class AbstractDataObject;
15 20 }
16 21
17 22 namespace Managers
18 23 {
```



```

24
26 class UserSectionRodComponentWidget : public AbstractRodComponentWidget
27 {
28 public:
29     UserSectionRodComponentWidget(Core::UserSectionRodComponent& userSectionRodComponent,
30                                   QString const& mimeType, QWidget* parent = nullptr);
31
32 private:
33     void createContent();
34     QLayout* createAreaLayout();
35     QWidget* createInertiaMomentsGroup();
36     QWidget* createCenterCoordinatesGroup();
37     void setProperty(Core::AbstractDataObject const* pDataObject, auto setFun);
38
39 private:
40     Core::UserSectionRodComponent& mUserSectionRodComponent;
41 };
42
43 }
44
45 }
46
47 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef ABSTRACTHIERARCHYITEM_H
9 #define ABSTRACTHIERARCHYITEM_H
10
11 #include <QStandardItem>
12
13 namespace QRS
14 {
15
16 namespace Core
17 {
18 class HierarchyNode;
19 class HierarchyTree;
20 }
21
22 namespace PropertiesModels
23 {
24 class AbstractPropertiesModel;
25 }
26
27 namespace HierarchyModels
28 {
29
30 class AbstractHierarchyItem : public QStandardItem
31 {
32 {
33     friend class AbstractHierarchyModel;
34     friend class PropertiesModels::AbstractPropertiesModel;
35
36 public:
37     enum ItemType
38     {
39         kDataObjects = QStandardItem::UserType,
40         kRodComponents
41     };
42     AbstractHierarchyItem(QIcon const& icon, QString const& text, Core::HierarchyNode* pNode);
43     virtual ~AbstractHierarchyItem() = 0;
44     void writePointer(QDataStream& out) const;
45     static AbstractHierarchyItem* readPointer(QDataStream& in);
46     virtual int type() const = 0;
47
48 protected:
49     Core::HierarchyNode* mpNode = nullptr;
50 };
51
52 }
53
54 }
55
56 #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.](#)

```

1
2 #ifndef ABSTRACTHIERARCHYMODEL_H
3 #define ABSTRACTHIERARCHYMODEL_H
4
5 #include <QStandardItemModel>
6 #include "abstracthierarchyitem.h"
7
8 QT_BEGIN_NAMESPACE
9 class QTreeView;
10 QT_END_NAMESPACE
11
12 namespace QRS
13 {
14     namespace Core
15     {
16         class HierarchyNode;
17     }
18     namespace HierarchyModels
19     {
20         using NodesState = std::unordered_map<Core::HierarchyNode*, bool>;
21         class AbstractHierarchyModel : public QStandardItemModel
22         {
23             Q_OBJECT
24         public:
25             AbstractHierarchyModel(QString const& mimeType, QTreeView* pView = nullptr);
26             virtual ~AbstractHierarchyModel() = 0;
27             virtual void updateContent() = 0;
28             virtual void clearContent() = 0;
29             Qt::DropActions supportedDragActions() const override;
30             Qt::DropActions supportedDropActions() const override;
31             QStringList mimeTypees() const override;
32             QMimeData* mimeTypeData(const QModelIndexList& indices) const override;
33             bool dropMimeData(QMimeData const* pMimeData, Qt::DropAction action, int row, int column, const
34                 QModelIndex& parent) override;
35
36         signals:
37             void hierarchyChanged();
38
39         private:
40             bool processDropOnItem(QDataStream& stream, int& numItems, QModelIndex const& indexParent);
41             bool processDropBetweenItems(QDataStream& stream, int& numItems, QModelIndex const& indexParent, int
42                 row);
43             void retrieveExpandedState(NodesState& nodesState, QModelIndex const& indexParent, QTreeView const*
44                 pView);
45             void setExpandedState(NodesState& nodesState, QModelIndex const& indexParent, QTreeView* pView);
46             void updateContentExpanded();
47
48         protected:
49             QString const mkMimeType();
50     };
51 }
52
53 #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.](#)

```

1
2 #ifndef DATAOBJECTSHIERARCHYITEM_H
3 #define DATAOBJECTSHIERARCHYITEM_H
4
5 #include "models/hierarchy/abstracthierarchyitem.h"
6 #include "core/aliasdataset.h"
7
8 namespace QRS
9 {
10
11 namespace PropertiesModels
12 {
13     class DataObjectsPropertiesModel;
14 }
15
16 namespace HierarchyModels
17 {
18     class DataObjectsHierarchyItem : public AbstractHierarchyItem
19     {
20     public:
21         friend class DataObjectsHierarchyModel;
22         friend class PropertiesModels::DataObjectsPropertiesModel;
23
24     public:
25         DataObjectsHierarchyItem(Core::DataObjects& dataObjects, Core::HierarchyTree& hierarchyDataObjects,
26                                 QString const& text = "Root", QIcon const& icon = QIcon());
27         DataObjectsHierarchyItem(Core::HierarchyNode* pNode, Core::AbstractDataObject* pDataObject);
28         DataObjectsHierarchyItem(Core::HierarchyNode* pNode);
29         int type() const override { return AbstractHierarchyItem::ItemType::kDataObjects; }
30         Core::AbstractDataObject const* getDataObject() const { return mpDataObject; }
31
32     private:
33         void appendItems(Core::DataObjects& dataObjects, Core::HierarchyNode* pNode);
34
35     private:
36         Core::AbstractDataObject* mpDataObject = nullptr;
37     };
38 }
39
40 #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**

167

```
#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.](#)

```

1
2
3
4
5
6
7
8 #ifndef DATAOBJECTSHIERARCHYMODEL_H
9 #define DATAOBJECTSHIERARCHYMODEL_H
10
11 #include "models/hierarchy/abstracthierarchymodel.h"
12 #include "core/aliasdataset.h"
13
14 namespace QRS
15 {
16
17     namespace Core
18     {
19         class HierarchyTree;
20     }
21
22     namespace HierarchyModels
23     {
24
25         class DataObjectsHierarchyItem;
26         class DataObjectsHierarchyModel : public AbstractHierarchyModel
27         {
28         public:
29             Q_OBJECT
30
31             public:
32                 DataObjectsHierarchyModel(Core::DataObjects& dataObjects, Core::HierarchyTree& hierarchyDataObjects,
33                     QString const& mimeType, QTreeView* pView = nullptr);
34                 ~DataObjectsHierarchyModel() = default;
35                 void updateContent() override;
36                 void clearContent() override;
37                 bool isEmpty() const;
38                 void selectItem(int iRow);
39                 void selectItemByID(Core::DataIDType id);
40
41             signals:
42                 void selected(Core::DataIDType id);
43                 void selectionCleared();
44
45             public slots:
46                 void retrieveSelectedItem();
47                 void removeSelectedItem();
48
49             private slots:
50                 void renameItem(QStandardItem* pStandardItem);
51
52             private:
53                 DataObjectsHierarchyItem* findItemByID(DataObjectsHierarchyItem* pItem, Core::DataIDType const& id);
54                 void selectItem(DataObjectsHierarchyItem* pItem);
55
56             private:
57                 Core::DataObjects& mDataObjects;
58                 Core::HierarchyTree& mHierarchyDataObjects;
59         };
60
61     }
62 }
63
64
65 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef PROJECTHIERARCHYMODEL_H
9 #define PROJECTHIERARCHYMODEL_H
10
11 #include "models/hierarchy/abstracthierarchymodel.h"
12 #include "core/aliasdata.h"
13 #include "core/project.h"
14
15 namespace QRS::HierarchyModels
16 {
17
18 class DataObjectsHierarchyItem;
19 class RodComponentsHierarchyItem;
20
21
22 class ProjectHierarchyModel : public AbstractHierarchyModel
23 {
24     Q_OBJECT
25
26 public:
27     ProjectHierarchyModel(QString const& mimeType, QTreeView* pView = nullptr);
28     void updateContent() override;
29     void clearContent() override;
30     void setProject(Core::Project* pProject);
31
32 signals:
33     void selectionValidated(QVector<QRS::HierarchyModels::AbstractHierarchyItem*> validatedItems);
34
35 public slots:
36     void validateItemSelection();
37
38 private:
39     DataObjectsHierarchyItem* retrieveDataObjectsItem();
40     RodComponentsHierarchyItem* retrieveRodComponentsItem();
41
42 private:
43     Core::Project* mpProject = nullptr;
44 };
45
46 }
47
48 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef RODCOMPONENTSHIERARCHYITEM_H
9 #define RODCOMPONENTSHIERARCHYITEM_H
10
11 #include "models/hierarchy/abstracthierarchyitem.h"
12 #include "core/aliasdataset.h"
13
14 namespace QRS
15 {
16
17 namespace HierarchyModels
18 {
19
20
21 class RodComponentsHierarchyItem : public AbstractHierarchyItem
22 {
23     friend class RodComponentsHierarchyModel;
24
25 public:
26     RodComponentsHierarchyItem(Core::RodComponents& rodComponents, Core::HierarchyTree&
        hierarchyRodComponents,
27                               QString const& text = "Root", QIcon const& icon = QIcon());
28     RodComponentsHierarchyItem(Core::HierarchyNode* pNode, Core::AbstractRodComponent* pRodComponent);
29     RodComponentsHierarchyItem(Core::HierarchyNode* pNode);
30     int type() const override { return AbstractHierarchyItem::ItemType::kRodComponents; }
31
32 private:
33     void appendItems(Core::RodComponents& rodComponents, Core::HierarchyNode* pNode);
34
35 private:
36     Core::AbstractRodComponent* mpRodComponent = nullptr;
37 };
38
39 }
40
41 }
42
43 #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.](#)

```
1  
8 #ifndef RODCOMPONENTSHIERARCHYMODEL_H  
9 #define RODCOMPONENTSHIERARCHYMODEL_H  
10  
11 #include "models/hierarchy/abstracthierarchymodel.h"  
12 #include "core/aliasdataset.h"  
13  
14 namespace QRS  
15 {  
16  
17 namespace HierarchyModels  
18 {  
19  
21 class RodComponentsHierarchyModel : public AbstractHierarchyModel  
22 {  
23     Q_OBJECT  
24  
25 public:  
26     RodComponentsHierarchyModel(Core::RodComponents& rodComponents, Core::HierarchyTree&  
        hierarchyRodComponents,  
27         QString const& mimeType, QTreeView* pView = nullptr);  
28     ~RodComponentsHierarchyModel() = default;  
29     void updateContent() override;  
30     void clearContent() override;  
31     bool isEmpty() const;  
32     void selectItem(int iRow);  
33  
34 signals:  
35     void selected(Core::DataIDType id);  
36     void selectionCleared();
```

```

37
38 public slots:
39     void retrieveSelectedItem();
40     void removeSelectedItems();
41
42 private slots:
43     void renameItem(QStandardItem* pStandardItem);
44
45 private:
46     Core::RodComponents& mRodComponents;
47     Core::HierarchyTree& mHierarchyRodComponents;
48 };
49
50 }
51
52 }
53
54 #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

Definition 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.](#)

```

1
2
3
4
5
6
7
8 #ifndef ABSTRACTPROPERTIESMODEL_H
9 #define ABSTRACTPROPERTIESMODEL_H
10
11 #include <QStandardItemModel>
12
13 QT_BEGIN_NAMESPACE
14 class QTableView;
15 QT_END_NAMESPACE
16
17 namespace QRS
18 {
19
20     namespace HierarchyModels
21     {
22     class AbstractHierarchyItem;
23     }
24
25     namespace PropertiesModels
26     {
27
28     class AbstractPropertiesModel : public QStandardItemModel
29     {
30     public:
31         Q_OBJECT
32
33     public:
34         AbstractPropertiesModel(QTableView* pView, QVector<HierarchyModels::AbstractHierarchyItem*> items);
35         virtual ~AbstractPropertiesModel() = 0;
36
37     signals:
38         void propertyChanged();
39
40     protected slots:
41         virtual void modifyProperty(QStandardItem* pChangedProperty) = 0;
42         void modifyDirectoryName(QString const& name);
43
44     protected:
45         void setDirectoryAttributes();
46         QList<QStandardItem*> preparePropertyRow(int type, QString const& title, QVariant const& value, bool
            isValueEditable) const;
47
48     protected:
49         QVector<HierarchyModels::AbstractHierarchyItem*> mItems;
50         bool mIsDirectory;
51         QString const mkEmptyProperty = "";
52
53     private:
54         enum PropertyDirectory
55         {
56             kName,
```

```

57         kNumberChildren
58     };
59 };
60
61 }
62
63 }
64
65 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef DATAOBJECTSPROPERTIESMODEL_H
9 #define DATAOBJECTSPROPERTIESMODEL_H
10
11 #include "abstractpropertiesmodel.h"
12
13 QT_BEGIN_NAMESPACE
14 class QTableView;
15 QT_END_NAMESPACE
16
17 namespace QRS
18 {
19
20 namespace HierarchyModels
21 {
22 class AbstractHierarchyItem;
23 }
24
25 namespace PropertiesModels
26 {
27
28 class DataObjectsPropertiesModel : public AbstractPropertiesModel
29 {
30     Q_OBJECT
31
32 public:
33     DataObjectsPropertiesModel(QTableView* pView, QVector<HierarchyModels::AbstractHierarchyItem*>
34         items);
35
36 protected slots:
37     void modifyProperty(QStandardItem* pChangedProperty) override;
38
39 private:
40     enum PropertyDataObject
41     {
42         kName,
43         kType,
44         kNumberItems,
45         kNumberEntities,
46         kID
47     };
48     void setObjectAttributes();
49 };
50
51 }
52
53 }
54
55 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef BASETABLEMODEL_H
9 #define BASETABLEMODEL_H
10
11 #include <QStandardItemModel>
12 #include "tablemodelinterface.h"
13
14 namespace QRS
15 {
16
17 namespace Core
18 {
19 class AbstractDataObject;
20 }
21
22 namespace TableModels
23 {
24
25 class BaseTableModel : public QStandardItemModel, public TableModelInterface
26 {
27     Q_OBJECT
28
29 public:
30     BaseTableModel(QWidget* parent = nullptr);
31     ~BaseTableModel() = default;
32     void setDataObject(Core::AbstractDataObject* pDataObject);
33     bool setData(const QModelIndex& indexEdit, const QVariant& value, int role = Qt::EditRole) override;
34     void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
35     void insertLeadingItemAfterSelected(QItemSelectionModel* /*pSelectionModel*/) override { };
36     void removeSelectedItem(QItemSelectionModel* pSelectionModel) override;
37     void removeSelectedLeadingItem(QItemSelectionModel* /*pSelectionModel*/) override { };
38
39 private:
40     void updateContent();
41     void clearContent();
42
43 private:
44     Core::AbstractDataObject* mpDataObject = nullptr;
45 };
46
47
48
49
50
51
52 #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.](#)

```
1
2
3
4
5
6
7
8 #ifndef MATRIXTABLEMODEL_H
9 #define MATRIXTABLEMODEL_H
10
11 #include <QStandardItemModel>
12 #include "tablemodelinterface.h"
13
14 namespace QRS
15 {
16
17 namespace Core
18 {
19 class AbstractDataObject;
20 }
21
22 namespace TableModels
23 {
24
25 class MatrixTableModel : public QStandardItemModel, public TableModelInterface
26 {
27     Q_OBJECT
28
29 public:
30     MatrixTableModel(QWidget* parent = nullptr);
31     ~MatrixTableModel() = default;
32     void setDataObject(Core::AbstractDataObject* pDataObject);
33     bool setData(const QModelIndex& indexEdit, const QVariant& value, int role = Qt::EditRole) override;
34     void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
35     void insertLeadingItemAfterSelected(QItemSelectionModel* /*pSelectionModel*/) override { };
36     void removeSelectedItem(QItemSelectionModel* pSelectionModel) override;
37     void removeSelectedLeadingItem(QItemSelectionModel* /*pSelectionModel*/) override { };
38
39 }
```

```
40 private:
41     void updateContent();
42     void clearContent();
43
44 private:
45     Core::AbstractDataObject* mpDataObject = nullptr;
46 };
47
48 }
49
50 }
51
52 #endif // MATRIXTABLEMODEL_H
```

## 5.147 /home/qinterfly/Library/Projects/Current/QRod↩ Systems/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/QRod↩ Systems/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.](#)

```

1
2
3
4
5
6
7
8 #ifndef SURFACETABLEMODEL_H
9 #define SURFACETABLEMODEL_H
10
11 #include <QStandardItemModel>
12 #include "tablemodelinterface.h"
13
14 namespace QRS
15 {
16
17 namespace Core
18 {
19 class SurfaceDataObject;
20 }
21
22 namespace TableModels
23 {
24
25
26 class SurfaceTableModel : public QStandardItemModel, public TableModelInterface
27 {
28     Q_OBJECT
29
30 public:
31     SurfaceTableModel(QWidget* parent = nullptr);
32     ~SurfaceTableModel() = default;
33     void setDataObject(Core::SurfaceDataObject* pDataObject);
34     bool setData(const QModelIndex& indexEdit, const QVariant& value, int role = Qt::EditRole) override;
35     void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
36     void removeSelectedItem(QItemSelectionModel* pSelectionModel) override;
37     void insertLeadingItemAfterSelected(QItemSelectionModel* pSelectionModel) override;
38     void removeSelectedItem(QItemSelectionModel* pSelectionModel) override;
39
40 private:
41     void updateContent();
42     void clearContent();
43
44 private:
45     Core::SurfaceDataObject* mpDataObject = nullptr;
46 };
47
48 }
49
50 }
51
52 #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.](#)

```

1
2
3
4
5
6
7
8 #ifndef TABLEMODELINTERFACE_H
9 #define TABLEMODELINTERFACE_H
10
11 #include <QItemSelection>
12
13 QT_BEGIN_NAMESPACE
14 class QStandardItem;
15 QT_END_NAMESPACE
16
17 namespace QRS
18 {
19
20 namespace Core
21 {
22     template <typename T>
23     class Array;
24 }
25
26 namespace TableModels
27 {
28
29     static const short kNumShowPrecision = 9;
30
31
32     class TableModelInterface
33     {
34     public:
35         virtual void insertItemAfterSelected(QItemSelectionModel* pSelectionModel) = 0;
36         virtual void insertLeadingItemAfterSelected(QItemSelectionModel* pSelectionModel) = 0;
37         virtual void removeSelectedItem(QItemSelectionModel* pSelectionModel) = 0;
38         virtual void removeSelectedLeadingItem(QItemSelectionModel* pSelectionModel) = 0;
39         virtual ~TableModelInterface() { };
40         static QStandardItem* makeDoubleItem(double value);
41         static QList<QStandardItem*> prepareRow(Core::Array<double> const& array, quint32 iRow);
42         static QList<QStandardItem*> prepareRow(double const& key, Core::Array<double> const& array, quint32
iRow);
43         static QList<QStandardItem*> prepareRow(QString const& name, Core::Array<double> const& array,
quint32 iRow);
44         static QStandardItem* makeLabelItem(QString const& name);
45     };
46
47 }
48
49 }
50
51 #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.](#)

```
1
2
3
4
5
6
7
8 #ifndef VIEW3D_H
9 #define VIEW3D_H
10
11 #include <QOpenGLWidget>
12 #include <QOpenGLFunctions>
13
14 namespace QRS::Graph
15 {
16
17     class View3D : public QOpenGLWidget, protected QOpenGLFunctions
18     {
19     public:
20         Q_OBJECT
21
22     public:
23         View3D(QWidget* parent = nullptr);
24         ~View3D() = default;
25
26     protected:
27         void initializeGL() override;
28         void paintGL() override;
29
30     private:
31         bool mCore;
32     };
33
34 }
35
36 #endif // VIEW3D_H
```

