

# EPICS QT Framework

## 3.4.2

Generated by Doxygen 1.6.1

Wed Apr 19 18:50:11 2017



# Contents

<b>1 QE framework - EPICS aware Qt Widgets and data access classes</b>	<b>1</b>
1.1 Documentation . . . . .	1
1.2 License . . . . .	2
1.3 Platforms . . . . .	2
1.4 Screenshots . . . . .	2
1.5 Downloads . . . . .	2
1.6 Installation . . . . .	2
1.7 Support . . . . .	3
1.8 Related Projects . . . . .	3
1.9 Credits: . . . . .	3
<b>2 GNU Lesser General Public License</b>	<b>5</b>
<b>3 ASgui screen shots</b>	<b>7</b>
<b>4 other applications using epicsqt widgets</b>	<b>13</b>
<b>5 Qt Designer</b>	<b>17</b>
<b>6 Qt Creator</b>	<b>19</b>
<b>7 Class Index</b>	<b>21</b>
7.1 Class Hierarchy . . . . .	21
<b>8 Class Index</b>	<b>27</b>
8.1 Class List . . . . .	27
<b>9 Class Documentation</b>	<b>33</b>
9.1 _CopyPaste Class Reference . . . . .	33

9.2	_Field Class Reference . . . . .	34
9.3	_Item Class Reference . . . . .	35
9.4	_QDialogItem Class Reference . . . . .	36
9.5	_QPushButtonGroup Class Reference . . . . .	37
9.6	_QTableWidgetFileBrowser Class Reference . . . . .	38
9.7	_QTableWidgetLog Class Reference . . . . .	39
9.8	_QTableWidgetScript Class Reference . . . . .	40
9.9	areaInfo Class Reference . . . . .	41
9.10	QEAnalogIndicator::Band Struct Reference . . . . .	42
9.11	QEAnalogIndicator::BandList Class Reference . . . . .	43
9.12	qcastatemachine::ConnectionQCaStateMachine Class Reference . . . . .	44
9.13	QEPeriodic::elementInfoStruct Struct Reference . . . . .	45
9.14	FFBuffer Class Reference . . . . .	46
9.15	FFThread Class Reference . . . . .	47
9.16	flipRotateMenu Class Reference . . . . .	48
9.17	fullScreenWindow Class Reference . . . . .	49
9.18	histogram Class Reference . . . . .	50
9.19	histogramScroll Class Reference . . . . .	51
9.20	historicImage Class Reference . . . . .	52
9.21	imageContextMenu Class Reference . . . . .	53
9.22	imageDisplayProperties Class Reference . . . . .	55
9.23	imageInfo Class Reference . . . . .	57
9.24	imageMarkup Class Reference . . . . .	58
9.25	imageMarkupLegendSetText Class Reference . . . . .	61
9.26	imageProcessor Class Reference . . . . .	62
9.26.1	Detailed Description . . . . .	65
9.26.2	Member Function Documentation . . . . .	65
9.26.2.1	getPixelValueFromData . . . . .	65
9.27	imageProperties Class Reference . . . . .	66
9.27.1	Detailed Description . . . . .	67
9.27.2	Member Enumeration Documentation . . . . .	68
9.27.2.1	rotationOptions . . . . .	68
9.27.3	Constructor & Destructor Documentation . . . . .	68
9.27.3.1	imageProperties . . . . .	68

9.28	imagePropertiesCore Class Reference . . . . .	69
9.28.1	Member Function Documentation . . . . .	69
9.28.1.1	buildImageCore . . . . .	69
9.29	imageUpdateIndicator Class Reference . . . . .	70
9.30	loginWidget Class Reference . . . . .	71
9.31	markupCrosshair1 Class Reference . . . . .	72
9.32	markupCrosshair2 Class Reference . . . . .	73
9.33	markupDisplayMenu Class Reference . . . . .	74
9.34	markupEllipse Class Reference . . . . .	75
9.35	markupHLine Class Reference . . . . .	76
9.35.1	Member Function Documentation . . . . .	76
9.35.1.1	drawMarkup . . . . .	76
9.36	markupItem Class Reference . . . . .	77
9.37	markupLine Class Reference . . . . .	80
9.38	markupRegion Class Reference . . . . .	81
9.39	markupText Class Reference . . . . .	82
9.40	markupVLine Class Reference . . . . .	83
9.40.1	Member Function Documentation . . . . .	83
9.40.1.1	drawMarkup . . . . .	83
9.41	mpegSource Class Reference . . . . .	84
9.41.1	Member Function Documentation . . . . .	84
9.41.1.1	updateImage . . . . .	84
9.42	mpegSourceObject Class Reference . . . . .	85
9.43	QEStripChartToolBar::OwnTabWidget Class Reference . . . . .	86
9.44	PeriodicDialog Class Reference . . . . .	87
9.45	PeriodicElementSetupForm Class Reference . . . . .	88
9.46	PeriodicSetupDialog Class Reference . . . . .	89
9.47	playbackTimer Class Reference . . . . .	90
9.48	pointInfo Class Reference . . . . .	91
9.49	profilePlot Class Reference . . . . .	92
9.50	PushButtonSpecifications Struct Reference . . . . .	93
9.51	QAnalogSlider Class Reference . . . . .	94
9.51.1	Detailed Description . . . . .	96
9.51.2	Constructor & Destructor Documentation . . . . .	97

9.51.2.1	QAnalogSlider . . . . .	97
9.51.3	Property Documentation . . . . .	97
9.51.3.1	majorInterval . . . . .	97
9.51.3.2	minorInterval . . . . .	97
9.51.3.3	precision . . . . .	97
9.51.3.4	tracking . . . . .	97
9.52	QAnalogSliderManager Class Reference . . . . .	98
9.53	QBitStatus Class Reference . . . . .	99
9.54	QCaAlarmInfo Class Reference . . . . .	101
9.55	QCaConnectionInfo Class Reference . . . . .	102
9.56	QCaDataPoint Class Reference . . . . .	103
9.57	QCaDataPointList Class Reference . . . . .	104
9.58	QCaDateTime Class Reference . . . . .	105
9.58.1	Member Function Documentation . . . . .	105
9.58.1.1	addSeconds . . . . .	105
9.58.1.2	floating . . . . .	105
9.59	QCaEventFilter Class Reference . . . . .	106
9.60	QCaEventItem Class Reference . . . . .	107
9.61	QCaEventUpdate Class Reference . . . . .	108
9.62	QCaInstalledFiltersListItem Class Reference . . . . .	109
9.63	qcaobject::QCaObject Class Reference . . . . .	110
9.64	qcastatemachine::QCaStateMachine Class Reference . . . . .	113
9.65	QCaVariableNamePropertyManager Class Reference . . . . .	114
9.66	QEAbstractDynamicWidget Class Reference . . . . .	115
9.66.1	Detailed Description . . . . .	116
9.66.2	Property Documentation . . . . .	116
9.66.2.1	defaultDir . . . . .	116
9.67	QEAbstractWidget Class Reference . . . . .	117
9.67.1	Member Enumeration Documentation . . . . .	118
9.67.1.1	DisplayAlarmStateOptions . . . . .	118
9.67.1.2	UserLevels . . . . .	118
9.67.2	Member Function Documentation . . . . .	119
9.67.2.1	setManagedVisible . . . . .	119
9.67.3	Property Documentation . . . . .	119

9.67.3.1	allowDrop . . . . .	119
9.67.3.2	defaultStyle . . . . .	119
9.67.3.3	displayAlarmState . . . . .	119
9.67.3.4	displayAlarmStateOption . . . . .	119
9.67.3.5	int . . . . .	120
9.67.3.6	styleSheet . . . . .	120
9.67.3.7	userLevelEnabled . . . . .	120
9.67.3.8	userLevelEngineerStyle . . . . .	120
9.67.3.9	userLevelScientistStyle . . . . .	120
9.67.3.10	userLevelUserStyle . . . . .	121
9.67.3.11	userLevelVisibility . . . . .	121
9.67.3.12	variableAsToolTip . . . . .	121
9.67.3.13	visible . . . . .	121
9.68	QEAnalogIndicator Class Reference . . . . .	122
9.68.1	Detailed Description . . . . .	125
9.68.2	Member Enumeration Documentation . . . . .	126
9.68.2.1	Modes . . . . .	126
9.68.2.2	Orientations . . . . .	126
9.68.3	Property Documentation . . . . .	126
9.68.3.1	backgroundColour . . . . .	126
9.68.3.2	borderColour . . . . .	126
9.68.3.3	centreAngle . . . . .	126
9.68.3.4	fontColour . . . . .	126
9.68.3.5	foregroundColour . . . . .	126
9.68.3.6	logScale . . . . .	127
9.68.3.7	logScaleInterval . . . . .	127
9.68.3.8	majorInterval . . . . .	127
9.68.3.9	maximum . . . . .	127
9.68.3.10	minimum . . . . .	127
9.68.3.11	minorInterval . . . . .	127
9.68.3.12	mode . . . . .	127
9.68.3.13	orientation . . . . .	127
9.68.3.14	showScale . . . . .	127
9.68.3.15	showText . . . . .	127

9.68.3.16 spanAngle . . . . .	127
9.68.3.17 value . . . . .	128
9.69 QEAnalogProgressBar Class Reference . . . . .	129
9.69.1 Member Enumeration Documentation . . . . .	132
9.69.1.1 ArrayActions . . . . .	132
9.69.1.2 DisplayAlarmStateOptions . . . . .	133
9.69.1.3 Formats . . . . .	133
9.69.1.4 Notations . . . . .	133
9.69.1.5 Separators . . . . .	133
9.69.1.6 UserLevels . . . . .	134
9.69.2 Constructor & Destructor Documentation . . . . .	134
9.69.2.1 QEAnalogProgressBar . . . . .	134
9.69.2.2 QEAnalogProgressBar . . . . .	134
9.69.3 Member Function Documentation . . . . .	134
9.69.3.1 dbConnectionChanged . . . . .	134
9.69.3.2 dbValueChanged . . . . .	134
9.69.3.3 setManagedVisible . . . . .	135
9.69.4 Property Documentation . . . . .	135
9.69.4.1 addUnits . . . . .	135
9.69.4.2 alarmSeverityDisplayStyle . . . . .	135
9.69.4.3 allowDrop . . . . .	135
9.69.4.4 arrayAction . . . . .	135
9.69.4.5 arrayIndex . . . . .	135
9.69.4.6 defaultStyle . . . . .	136
9.69.4.7 displayAlarmState . . . . .	136
9.69.4.8 displayAlarmStateOption . . . . .	136
9.69.4.9 format . . . . .	136
9.69.4.10 int . . . . .	136
9.69.4.11 leadingZero . . . . .	136
9.69.4.12 localEnumeration . . . . .	137
9.69.4.13 notation . . . . .	137
9.69.4.14 precision . . . . .	137
9.69.4.15 radix . . . . .	137
9.69.4.16 separator . . . . .	137

9.69.4.17 styleSheet . . . . .	138
9.69.4.18 trailingZeros . . . . .	138
9.69.4.19 useDbDisplayLimits . . . . .	138
9.69.4.20 useDbPrecision . . . . .	138
9.69.4.21 userLevelEnabled . . . . .	138
9.69.4.22 userLevelEngineerStyle . . . . .	138
9.69.4.23 userLevelScientistStyle . . . . .	138
9.69.4.24 userLevelUserStyle . . . . .	139
9.69.4.25 userLevelVisibility . . . . .	139
9.69.4.26 value . . . . .	139
9.69.4.27 variable . . . . .	139
9.69.4.28 variableAsToolTip . . . . .	139
9.69.4.29 variableSubstitutions . . . . .	139
9.69.4.30 visible . . . . .	140
9.70 QEAnalogSlider Class Reference . . . . .	141
9.70.1 Member Enumeration Documentation . . . . .	143
9.70.1.1 DisplayAlarmStateOptions . . . . .	143
9.70.1.2 UserLevels . . . . .	144
9.70.2 Constructor & Destructor Documentation . . . . .	144
9.70.2.1 QEAnalogSlider . . . . .	144
9.70.2.2 QEAnalogSlider . . . . .	144
9.70.3 Member Function Documentation . . . . .	144
9.70.3.1 dbValueChanged . . . . .	144
9.70.3.2 setManagedVisible . . . . .	144
9.70.4 Property Documentation . . . . .	145
9.70.4.1 allowDrop . . . . .	145
9.70.4.2 arrayIndex . . . . .	145
9.70.4.3 autoScale . . . . .	145
9.70.4.4 axisAlarmColours . . . . .	145
9.70.4.5 continuousWrite . . . . .	145
9.70.4.6 defaultStyle . . . . .	145
9.70.4.7 displayAlarmState . . . . .	145
9.70.4.8 displayAlarmStateOption . . . . .	146
9.70.4.9 int . . . . .	146

9.70.4.10 styleSheet . . . . .	146
9.70.4.11 userLevelEnabled . . . . .	146
9.70.4.12 userLevelEngineerStyle . . . . .	146
9.70.4.13 userLevelScientistStyle . . . . .	147
9.70.4.14 userLevelUserStyle . . . . .	147
9.70.4.15 userLevelVisibility . . . . .	147
9.70.4.16 variable . . . . .	147
9.70.4.17 variableAsToolTip . . . . .	147
9.70.4.18 variableSubstitutions . . . . .	147
9.70.4.19 visible . . . . .	148
9.71 QEAnalogSliderManager Class Reference . . . . .	149
9.72 QEBitStatus Class Reference . . . . .	150
9.72.1 Member Enumeration Documentation . . . . .	152
9.72.1.1 DisplayAlarmStateOptions . . . . .	152
9.72.1.2 UserLevels . . . . .	152
9.72.2 Member Function Documentation . . . . .	152
9.72.2.1 dbConnectionChanged . . . . .	152
9.72.2.2 dbValueChanged . . . . .	152
9.72.2.3 setManagedVisible . . . . .	153
9.72.3 Property Documentation . . . . .	153
9.72.3.1 allowDrop . . . . .	153
9.72.3.2 arrayIndex . . . . .	153
9.72.3.3 defaultStyle . . . . .	153
9.72.3.4 displayAlarmState . . . . .	153
9.72.3.5 displayAlarmStateOption . . . . .	153
9.72.3.6 int . . . . .	154
9.72.3.7 styleSheet . . . . .	154
9.72.3.8 userLevelEnabled . . . . .	154
9.72.3.9 userLevelEngineerStyle . . . . .	154
9.72.3.10 userLevelScientistStyle . . . . .	154
9.72.3.11 userLevelUserStyle . . . . .	154
9.72.3.12 userLevelVisibility . . . . .	155
9.72.3.13 variable . . . . .	155
9.72.3.14 variableAsToolTip . . . . .	155

9.72.3.15	variableSubstitutions	155
9.72.3.16	visible	155
9.73	QEByteArray Class Reference	156
9.74	QECheckBox Class Reference	157
9.74.1	Member Enumeration Documentation	161
9.74.1.1	ArrayActions	161
9.74.1.2	CreationOptionNames	161
9.74.1.3	DisplayAlarmStateOptions	162
9.74.1.4	Formats	162
9.74.1.5	Notations	163
9.74.1.6	ProgramStartupOptionNames	163
9.74.1.7	Separators	163
9.74.1.8	UpdateOptions	163
9.74.1.9	UserLevels	164
9.74.2	Constructor & Destructor Documentation	164
9.74.2.1	QECheckBox	164
9.74.2.2	QECheckBox	164
9.74.3	Member Function Documentation	164
9.74.3.1	clicked	164
9.74.3.2	dbValueChanged	164
9.74.3.3	pressed	165
9.74.3.4	released	165
9.74.3.5	requestAction	165
9.74.3.6	setManagedVisible	165
9.74.4	Property Documentation	165
9.74.4.1	addUnits	165
9.74.4.2	alignment	165
9.74.4.3	allowDrop	165
9.74.4.4	arguments	166
9.74.4.5	arrayAction	166
9.74.4.6	arrayIndex	166
9.74.4.7	clickCheckedText	166
9.74.4.8	clickText	166
9.74.4.9	confirmAction	167

9.74.4.10 confirmText . . . . .	167
9.74.4.11 creationOption . . . . .	167
9.74.4.12 customisationName . . . . .	167
9.74.4.13 defaultStyle . . . . .	167
9.74.4.14 disabledRecordPolicy . . . . .	167
9.74.4.15 displayAlarmState . . . . .	168
9.74.4.16 displayAlarmStateOption . . . . .	168
9.74.4.17 format . . . . .	168
9.74.4.18 guiFile . . . . .	168
9.74.4.19 int . . . . .	168
9.74.4.20 labelText . . . . .	168
9.74.4.21 leadingZero . . . . .	169
9.74.4.22 localEnumeration . . . . .	169
9.74.4.23 notation . . . . .	169
9.74.4.24 password . . . . .	170
9.74.4.25 pixmap0 . . . . .	170
9.74.4.26 pixmap1 . . . . .	170
9.74.4.27 pixmap2 . . . . .	170
9.74.4.28 pixmap3 . . . . .	170
9.74.4.29 pixmap4 . . . . .	170
9.74.4.30 pixmap5 . . . . .	170
9.74.4.31 pixmap6 . . . . .	170
9.74.4.32 pixmap7 . . . . .	170
9.74.4.33 precision . . . . .	171
9.74.4.34 pressText . . . . .	171
9.74.4.35 prioritySubstitutions . . . . .	171
9.74.4.36 program . . . . .	171
9.74.4.37 programStartupOption . . . . .	171
9.74.4.38 radix . . . . .	171
9.74.4.39 releaseText . . . . .	171
9.74.4.40 separator . . . . .	172
9.74.4.41 styleSheet . . . . .	172
9.74.4.42 subscribe . . . . .	172
9.74.4.43 trailingZeros . . . . .	172

9.74.4.44 updateOption . . . . .	172
9.74.4.45 useDbPrecision . . . . .	172
9.74.4.46 userLevelEnabled . . . . .	172
9.74.4.47 userLevelEngineerStyle . . . . .	172
9.74.4.48 userLevelScientistStyle . . . . .	173
9.74.4.49 userLevelUserStyle . . . . .	173
9.74.4.50 userLevelVisibility . . . . .	173
9.74.4.51 variable . . . . .	173
9.74.4.52 variableAsToolTip . . . . .	173
9.74.4.53 variableSubstitutions . . . . .	173
9.74.4.54 visible . . . . .	174
9.74.4.55 writeOnClick . . . . .	174
9.74.4.56 writeOnPress . . . . .	174
9.74.4.57 writeOnRelease . . . . .	174
9.75 QECheckBoxManager Class Reference . . . . .	175
9.76 QEComboBox Class Reference . . . . .	176
9.76.1 Member Enumeration Documentation . . . . .	178
9.76.1.1 DisplayAlarmStateOptions . . . . .	178
9.76.1.2 UserLevels . . . . .	178
9.76.2 Member Function Documentation . . . . .	179
9.76.2.1 dbValueChanged . . . . .	179
9.76.2.2 setManagedVisible . . . . .	179
9.76.3 Member Data Documentation . . . . .	179
9.76.3.1 useDbEnumerations . . . . .	179
9.76.3.2 writeOnChange . . . . .	179
9.76.4 Property Documentation . . . . .	179
9.76.4.1 allowDrop . . . . .	179
9.76.4.2 allowFocusUpdate . . . . .	179
9.76.4.3 arrayIndex . . . . .	179
9.76.4.4 defStyle . . . . .	180
9.76.4.5 displayAlarmState . . . . .	180
9.76.4.6 displayAlarmStateOption . . . . .	180
9.76.4.7 int . . . . .	180
9.76.4.8 localEnumeration . . . . .	180

9.76.4.9	styleSheet . . . . .	180
9.76.4.10	subscribe . . . . .	180
9.76.4.11	userLevelEnabled . . . . .	181
9.76.4.12	userLevelEngineerStyle . . . . .	181
9.76.4.13	userLevelScientistStyle . . . . .	181
9.76.4.14	userLevelUserStyle . . . . .	181
9.76.4.15	userLevelVisibility . . . . .	181
9.76.4.16	variable . . . . .	182
9.76.4.17	variableAsToolTip . . . . .	182
9.76.4.18	variableSubstitutions . . . . .	182
9.76.4.19	visible . . . . .	182
9.77	QEConfiguredLayout Class Reference . . . . .	183
9.77.1	Member Enumeration Documentation . . . . .	185
9.77.1.1	DisplayAlarmStateOptions . . . . .	185
9.77.1.2	UserLevels . . . . .	185
9.77.2	Member Function Documentation . . . . .	186
9.77.2.1	setManagedVisible . . . . .	186
9.77.3	Property Documentation . . . . .	186
9.77.3.1	allowDrop . . . . .	186
9.77.3.2	defaultStyle . . . . .	186
9.77.3.3	displayAlarmState . . . . .	186
9.77.3.4	displayAlarmStateOption . . . . .	186
9.77.3.5	int . . . . .	186
9.77.3.6	styleSheet . . . . .	187
9.77.3.7	userLevelEnabled . . . . .	187
9.77.3.8	userLevelEngineerStyle . . . . .	187
9.77.3.9	userLevelScientistStyle . . . . .	187
9.77.3.10	userLevelUserStyle . . . . .	187
9.77.3.11	userLevelVisibility . . . . .	188
9.77.3.12	variableAsToolTip . . . . .	188
9.77.3.13	visible . . . . .	188
9.78	QEConfiguredLayoutManager Class Reference . . . . .	189
9.79	QEFfileBrowser Class Reference . . . . .	190
9.79.1	Detailed Description . . . . .	193

9.79.2 Member Enumeration Documentation . . . . .	193
9.79.2.1 DisplayAlarmStateOptions . . . . .	193
9.79.2.2 UserLevels . . . . .	193
9.79.3 Member Function Documentation . . . . .	193
9.79.3.1 selected . . . . .	193
9.79.3.2 setManagedVisible . . . . .	194
9.79.4 Member Data Documentation . . . . .	194
9.79.4.1 fileFilter . . . . .	194
9.79.5 Property Documentation . . . . .	194
9.79.5.1 allowDrop . . . . .	194
9.79.5.2 defaultStyle . . . . .	194
9.79.5.3 displayAlarmState . . . . .	194
9.79.5.4 displayAlarmStateOption . . . . .	194
9.79.5.5 int . . . . .	195
9.79.5.6 styleSheet . . . . .	195
9.79.5.7 userLevelEnabled . . . . .	195
9.79.5.8 userLevelEngineerStyle . . . . .	195
9.79.5.9 userLevelScientistStyle . . . . .	195
9.79.5.10 userLevelUserStyle . . . . .	195
9.79.5.11 userLevelVisibility . . . . .	196
9.79.5.12 variable . . . . .	196
9.79.5.13 variableAsToolTip . . . . .	196
9.79.5.14 variableSubstitutions . . . . .	196
9.79.5.15 visible . . . . .	196
9.80 QEFileImage Class Reference . . . . .	197
9.80.1 Detailed Description . . . . .	199
9.80.2 Member Enumeration Documentation . . . . .	199
9.80.2.1 DisplayAlarmStateOptions . . . . .	199
9.80.2.2 UserLevels . . . . .	199
9.80.3 Constructor & Destructor Documentation . . . . .	199
9.80.3.1 QEFileImage . . . . .	199
9.80.3.2 QEFileImage . . . . .	199
9.80.4 Member Function Documentation . . . . .	200
9.80.4.1 dbValueChanged . . . . .	200

9.80.4.2	setManagedVisible . . . . .	200
9.80.5	Property Documentation . . . . .	200
9.80.5.1	allowDrop . . . . .	200
9.80.5.2	defaultStyle . . . . .	200
9.80.5.3	displayAlarmState . . . . .	200
9.80.5.4	displayAlarmStateOption . . . . .	200
9.80.5.5	int . . . . .	201
9.80.5.6	styleSheet . . . . .	201
9.80.5.7	userLevelEnabled . . . . .	201
9.80.5.8	userLevelEngineerStyle . . . . .	201
9.80.5.9	userLevelScientistStyle . . . . .	201
9.80.5.10	userLevelUserStyle . . . . .	201
9.80.5.11	userLevelVisibility . . . . .	202
9.80.5.12	variable . . . . .	202
9.80.5.13	variableAsToolTip . . . . .	202
9.80.5.14	variableSubstitutions . . . . .	202
9.80.5.15	visible . . . . .	202
9.81	QEFileImageManager Class Reference . . . . .	203
9.82	QEFloating Class Reference . . . . .	204
9.83	QEFloatingArray Class Reference . . . . .	205
9.83.1	Detailed Description . . . . .	205
9.84	QEFloatingFormatting Class Reference . . . . .	206
9.85	QEForm Class Reference . . . . .	207
9.85.1	Member Data Documentation . . . . .	209
9.85.1.1	handleGuiLaunchRequests . . . . .	209
9.85.1.2	resizeContents . . . . .	209
9.85.2	Property Documentation . . . . .	210
9.85.2.1	allowDrop . . . . .	210
9.85.2.2	displayAlarmStateOption . . . . .	210
9.85.2.3	int . . . . .	210
9.85.2.4	messageFormFilter . . . . .	210
9.85.2.5	messageSourceFilter . . . . .	211
9.85.2.6	uiFile . . . . .	211
9.85.2.7	variableAsToolTip . . . . .	211

9.85.2.8	variableSubstitutions	211
9.86	QEFormGrid Class Reference	212
9.86.1	Detailed Description	213
9.86.2	Member Enumeration Documentation	213
9.86.2.1	GridOrders	213
9.86.3	Constructor & Destructor Documentation	214
9.86.3.1	QEFormGrid	214
9.86.4	Property Documentation	214
9.86.4.1	allowDrop	214
9.86.4.2	colMacroPrefix	214
9.86.4.3	colNumberOffset	214
9.86.4.4	colNumberWidth	214
9.86.4.5	colStrings	214
9.86.4.6	columns	214
9.86.4.7	displayAlarmStateOption	214
9.86.4.8	gridOrder	215
9.86.4.9	margin	215
9.86.4.10	number	215
9.86.4.11	rowMacroPrefix	215
9.86.4.12	rowNumberOffset	215
9.86.4.13	rowNumberWidth	215
9.86.4.14	rowStrings	215
9.86.4.15	slotMacroPrefix	215
9.86.4.16	slotNumberOffset	215
9.86.4.17	slotNumberWidth	215
9.86.4.18	slotStrings	216
9.86.4.19	spacing	216
9.86.4.20	uiFile	216
9.86.4.21	variableAsToolTip	216
9.87	QEFormGridManager Class Reference	217
9.88	QEFrame Class Reference	218
9.88.1	Member Enumeration Documentation	220
9.88.1.1	DisplayAlarmStateOptions	220
9.88.1.2	UserLevels	220

9.88.2 Member Function Documentation . . . . .	220
9.88.2.1 setManagedVisible . . . . .	220
9.88.3 Property Documentation . . . . .	221
9.88.3.1 allowDrop . . . . .	221
9.88.3.2 defaultStyle . . . . .	221
9.88.3.3 displayAlarmState . . . . .	221
9.88.3.4 displayAlarmStateOption . . . . .	221
9.88.3.5 int . . . . .	221
9.88.3.6 pixmap . . . . .	221
9.88.3.7 pixmap0 . . . . .	222
9.88.3.8 pixmap1 . . . . .	222
9.88.3.9 pixmap2 . . . . .	222
9.88.3.10 pixmap3 . . . . .	222
9.88.3.11 pixmap4 . . . . .	222
9.88.3.12 pixmap5 . . . . .	222
9.88.3.13 pixmap6 . . . . .	222
9.88.3.14 pixmap7 . . . . .	222
9.88.3.15 scaledContents . . . . .	222
9.88.3.16 styleSheet . . . . .	223
9.88.3.17 userLevelEnabled . . . . .	223
9.88.3.18 userLevelEngineerStyle . . . . .	223
9.88.3.19 userLevelScientistStyle . . . . .	223
9.88.3.20 userLevelUserStyle . . . . .	223
9.88.3.21 userLevelVisibility . . . . .	223
9.88.3.22 variableAsToolTip . . . . .	224
9.88.3.23 visible . . . . .	224
9.89 QEGeneralEdit Class Reference . . . . .	225
9.89.1 Detailed Description . . . . .	226
9.89.2 Constructor & Destructor Documentation . . . . .	226
9.89.2.1 QEGeneralEdit . . . . .	226
9.89.2.2 QEGeneralEdit . . . . .	226
9.89.3 Property Documentation . . . . .	226
9.89.3.1 arrayIndex . . . . .	226
9.89.3.2 variable . . . . .	226

9.89.3.3	variableSubstitutions	226
9.90	QEGeneralEditManager Class Reference	227
9.91	QEGenericButton Class Reference	228
9.92	QEGenericEdit Class Reference	231
9.92.1	Member Enumeration Documentation	233
9.92.1.1	DisplayAlarmStateOptions	233
9.92.1.2	UserLevels	234
9.92.2	Constructor & Destructor Documentation	234
9.92.2.1	QEGenericEdit	234
9.92.2.2	QEGenericEdit	234
9.92.3	Member Function Documentation	234
9.92.3.1	getConfirmWrite	234
9.92.3.2	getSubscribe	234
9.92.3.3	getWriteOnEnter	234
9.92.3.4	getWriteOnFinish	235
9.92.3.5	getWriteOnLoseFocus	235
9.92.3.6	setAllowFocusUpdate	235
9.92.3.7	setConfirmWrite	235
9.92.3.8	setManagedVisible	235
9.92.3.9	setSubscribe	235
9.92.3.10	setWriteOnEnter	235
9.92.3.11	setWriteOnFinish	235
9.92.3.12	setWriteOnLoseFocus	236
9.92.4	Property Documentation	236
9.92.4.1	allowDrop	236
9.92.4.2	arrayIndex	236
9.92.4.3	confirmWrite	236
9.92.4.4	defaultStyle	236
9.92.4.5	displayAlarmState	236
9.92.4.6	displayAlarmStateOption	236
9.92.4.7	int	237
9.92.4.8	styleSheet	237
9.92.4.9	subscribe	237
9.92.4.10	userLevelEnabled	237

9.92.4.11 userLevelEngineerStyle . . . . .	237
9.92.4.12 userLevelScientistStyle . . . . .	237
9.92.4.13 userLevelUserStyle . . . . .	238
9.92.4.14 userLevelVisibility . . . . .	238
9.92.4.15 variable . . . . .	238
9.92.4.16 variableAsToolTip . . . . .	238
9.92.4.17 variableSubstitutions . . . . .	238
9.92.4.18 visible . . . . .	238
9.92.4.19 writeOnEnter . . . . .	239
9.92.4.20 writeOnFinish . . . . .	239
9.92.4.21 writeOnLoseFocus . . . . .	239
9.93 QEGroupBox Class Reference . . . . .	240
9.93.1 Member Enumeration Documentation . . . . .	241
9.93.1.1 DisplayAlarmStateOptions . . . . .	241
9.93.1.2 UserLevels . . . . .	241
9.93.2 Member Function Documentation . . . . .	242
9.93.2.1 setManagedVisible . . . . .	242
9.93.3 Property Documentation . . . . .	242
9.93.3.1 allowDrop . . . . .	242
9.93.3.2 defaultStyle . . . . .	242
9.93.3.3 displayAlarmState . . . . .	242
9.93.3.4 displayAlarmStateOption . . . . .	242
9.93.3.5 int . . . . .	242
9.93.3.6 styleSheet . . . . .	243
9.93.3.7 substitutedTitle . . . . .	243
9.93.3.8 textSubstitutions . . . . .	243
9.93.3.9 userLevelEnabled . . . . .	243
9.93.3.10 userLevelEngineerStyle . . . . .	243
9.93.3.11 userLevelScientistStyle . . . . .	243
9.93.3.12 userLevelUserStyle . . . . .	244
9.93.3.13 userLevelVisibility . . . . .	244
9.93.3.14 variableAsToolTip . . . . .	244
9.93.3.15 visible . . . . .	244
9.94 QEHistogram Class Reference . . . . .	245

9.94.1	Detailed Description	246
9.94.2	Property Documentation	246
9.94.2.1	orientation	246
9.95	QEHistogramManager Class Reference	247
9.96	QEImage Class Reference	248
9.96.1	Detailed Description	278
9.96.2	Member Enumeration Documentation	278
9.96.2.1	DisplayAlarmStateOptions	278
9.96.2.2	EllipseVariableDefinitions	278
9.96.2.3	ellipseVariableDefinitions	278
9.96.2.4	FormatOptions	279
9.96.2.5	ProgramStartupOptionNames	279
9.96.2.6	ResizeOptions	279
9.96.2.7	resizeOptions	280
9.96.2.8	RotationOptions	280
9.96.2.9	selectOptions	280
9.96.2.10	TargetOptions	281
9.96.2.11	UserLevels	281
9.96.3	Constructor & Destructor Documentation	281
9.96.3.1	QEImage	281
9.96.3.2	QEImage	281
9.96.4	Member Function Documentation	281
9.96.4.1	dbValueChanged	281
9.96.4.2	setImageFile	282
9.96.4.3	setManagedVisible	282
9.96.5	Member Data Documentation	282
9.96.5.1	displayButtonBar	282
9.96.5.2	initialVertScrollPos	282
9.96.6	Property Documentation	282
9.96.6.1	allowDrop	282
9.96.6.2	areaColor	282
9.96.6.3	arguments1	282
9.96.6.4	arguments2	282
9.96.6.5	autoBrightnessContrast	283

9.96.6.6 beamColor . . . . .	283
9.96.6.7 beamXVariable . . . . .	283
9.96.6.8 beamYVariable . . . . .	283
9.96.6.9 bitDepthVariable . . . . .	283
9.96.6.10 briefInfoArea . . . . .	283
9.96.6.11 clippingHighVariable . . . . .	283
9.96.6.12 clippingLowVariable . . . . .	283
9.96.6.13 clippingOnOffVariable . . . . .	283
9.96.6.14 contrastReversal . . . . .	284
9.96.6.15 dataTypeVariable . . . . .	284
9.96.6.16 defStyle . . . . .	284
9.96.6.17 dimension1Variable . . . . .	284
9.96.6.18 dimension2Variable . . . . .	284
9.96.6.19 dimension3Variable . . . . .	284
9.96.6.20 dimensionsVariable . . . . .	284
9.96.6.21 displayAlarmState . . . . .	284
9.96.6.22 displayAlarmStateOption . . . . .	285
9.96.6.23 displayArea1Selection . . . . .	285
9.96.6.24 displayArea2Selection . . . . .	285
9.96.6.25 displayArea3Selection . . . . .	285
9.96.6.26 displayArea4Selection . . . . .	285
9.96.6.27 displayBeamSelection . . . . .	285
9.96.6.28 displayCursorPixelInfo . . . . .	285
9.96.6.29 displayEllipse . . . . .	286
9.96.6.30 displayHozSlice1Selection . . . . .	286
9.96.6.31 displayHozSlice2Selection . . . . .	286
9.96.6.32 displayHozSlice3Selection . . . . .	286
9.96.6.33 displayHozSlice4Selection . . . . .	286
9.96.6.34 displayHozSlice5Selection . . . . .	286
9.96.6.35 displayProfileSelection . . . . .	286
9.96.6.36 displayTargetSelection . . . . .	286
9.96.6.37 displayVertSlice1Selection . . . . .	286
9.96.6.38 displayVertSlice2Selection . . . . .	287
9.96.6.39 displayVertSlice3Selection . . . . .	287

9.96.6.40 displayVertSlice4Selection . . . . .	287
9.96.6.41 displayVertSlice5Selection . . . . .	287
9.96.6.42 ellipseColor . . . . .	287
9.96.6.43 ellipseHVariable . . . . .	287
9.96.6.44 ellipseWVariable . . . . .	287
9.96.6.45 ellipseXVariable . . . . .	287
9.96.6.46 ellipseYVariable . . . . .	287
9.96.6.47 enableArea1Selection . . . . .	288
9.96.6.48 enableArea2Selection . . . . .	288
9.96.6.49 enableArea3Selection . . . . .	288
9.96.6.50 enableArea4Selection . . . . .	288
9.96.6.51 enableBeamSelection . . . . .	288
9.96.6.52 enableHozSlice1Selection . . . . .	288
9.96.6.53 enableHozSlice2Selection . . . . .	288
9.96.6.54 enableHozSlice3Selection . . . . .	288
9.96.6.55 enableHozSlice4Selection . . . . .	289
9.96.6.56 enableHozSlice5Selection . . . . .	289
9.96.6.57 enableProfileSelection . . . . .	289
9.96.6.58 enableTargetSelection . . . . .	289
9.96.6.59 enableVertSlice1Selection . . . . .	289
9.96.6.60 enableVertSlice2Selection . . . . .	289
9.96.6.61 enableVertSlice3Selection . . . . .	289
9.96.6.62 enableVertSlice4Selection . . . . .	290
9.96.6.63 enableVertSlice5Selection . . . . .	290
9.96.6.64 externalControls . . . . .	290
9.96.6.65 formatOption . . . . .	290
9.96.6.66 formatVariable . . . . .	290
9.96.6.67 heightVariable . . . . .	290
9.96.6.68 horizontalFlip . . . . .	290
9.96.6.69 hozSlice1Color . . . . .	290
9.96.6.70 hozSlice2Color . . . . .	290
9.96.6.71 hozSlice3Color . . . . .	291
9.96.6.72 hozSlice4Color . . . . .	291
9.96.6.73 hozSlice5Color . . . . .	291

9.96.6.74 imageVariable . . . . .	291
9.96.6.75 initialHosScrollPos . . . . .	291
9.96.6.76 int . . . . .	291
9.96.6.77 lineProfileArrayVariable . . . . .	291
9.96.6.78 lineProfileThicknessVariable . . . . .	291
9.96.6.79 lineProfileX1Variable . . . . .	292
9.96.6.80 lineProfileX2Variable . . . . .	292
9.96.6.81 lineProfileY1Variable . . . . .	292
9.96.6.82 lineProfileY2Variable . . . . .	292
9.96.6.83 logBrightness . . . . .	292
9.96.6.84 profileColor . . . . .	292
9.96.6.85 profileHoz1ThicknessVariable . . . . .	292
9.96.6.86 profileHoz1Variable . . . . .	292
9.96.6.87 profileHoz2ThicknessVariable . . . . .	292
9.96.6.88 profileHoz2Variable . . . . .	293
9.96.6.89 profileHoz3ThicknessVariable . . . . .	293
9.96.6.90 profileHoz3Variable . . . . .	293
9.96.6.91 profileHoz4ThicknessVariable . . . . .	293
9.96.6.92 profileHoz4Variable . . . . .	293
9.96.6.93 profileHoz5ThicknessVariable . . . . .	293
9.96.6.94 profileHoz5Variable . . . . .	293
9.96.6.95 profileHozArrayVariable . . . . .	293
9.96.6.96 profileVert1ThicknessVariable . . . . .	294
9.96.6.97 profileVert1Variable . . . . .	294
9.96.6.98 profileVert2ThicknessVariable . . . . .	294
9.96.6.99 profileVert2Variable . . . . .	294
9.96.6.100profileVert3ThicknessVariable . . . . .	294
9.96.6.101profileVert3Variable . . . . .	294
9.96.6.102profileVert4ThicknessVariable . . . . .	294
9.96.6.103profileVert4Variable . . . . .	294
9.96.6.104profileVert5ThicknessVariable . . . . .	295
9.96.6.105profileVert5Variable . . . . .	295
9.96.6.106profileVertArrayVariable . . . . .	295
9.96.6.107program1 . . . . .	295

9.96.6.108program2 . . . . .	295
9.96.6.109programStartupOption1 . . . . .	295
9.96.6.110programStartupOption2 . . . . .	295
9.96.6.111regionOfInterest1HVariable . . . . .	296
9.96.6.112regionOfInterest1WVariable . . . . .	296
9.96.6.113regionOfInterest1XVariable . . . . .	296
9.96.6.114regionOfInterest1YVariable . . . . .	296
9.96.6.115regionOfInterest2HVariable . . . . .	296
9.96.6.116regionOfInterest2WVariable . . . . .	296
9.96.6.117regionOfInterest2XVariable . . . . .	296
9.96.6.118regionOfInterest2YVariable . . . . .	296
9.96.6.119regionOfInterest3HVariable . . . . .	297
9.96.6.120regionOfInterest3WVariable . . . . .	297
9.96.6.121regionOfInterest3XVariable . . . . .	297
9.96.6.122regionOfInterest3YVariable . . . . .	297
9.96.6.123regionOfInterest4HVariable . . . . .	297
9.96.6.124regionOfInterest4WVariable . . . . .	297
9.96.6.125regionOfInterest4XVariable . . . . .	297
9.96.6.126regionOfInterest4YVariable . . . . .	297
9.96.6.127resizeOption . . . . .	298
9.96.6.128rotation . . . . .	298
9.96.6.129showTime . . . . .	298
9.96.6.130styleSheet . . . . .	298
9.96.6.131targetColor . . . . .	298
9.96.6.132targetTriggerVariable . . . . .	298
9.96.6.133targetXVariable . . . . .	298
9.96.6.134targetYVariable . . . . .	298
9.96.6.135timeColor . . . . .	298
9.96.6.136URL . . . . .	299
9.96.6.137useFalseColour . . . . .	299
9.96.6.138userLevelEnabled . . . . .	299
9.96.6.139userLevelEngineerStyle . . . . .	299
9.96.6.140userLevelScientistStyle . . . . .	299
9.96.6.141userLevelUserStyle . . . . .	299

9.96.6.142userLevelVisibility . . . . .	300
9.96.6.143variableAsToolTip . . . . .	300
9.96.6.144variableSubstitutions . . . . .	300
9.96.6.145verticalFlip . . . . .	300
9.96.6.146vertSlice1Color . . . . .	300
9.96.6.147vertSlice2Color . . . . .	300
9.96.6.148vertSlice3Color . . . . .	300
9.96.6.149vertSlice4Color . . . . .	300
9.96.6.150vertSlice5Color . . . . .	301
9.96.6.151visible . . . . .	301
9.96.6.152widthVariable . . . . .	301
9.97 QEImageMarkupThickness Class Reference . . . . .	302
9.98 QEImageOptionsDialog Class Reference . . . . .	303
9.99 QEInteger Class Reference . . . . .	304
9.100QEIntegerArray Class Reference . . . . .	305
9.100.1 Detailed Description . . . . .	305
9.101 QEIntegerFormatting Class Reference . . . . .	306
9.101.1 Detailed Description . . . . .	306
9.101.2 Member Function Documentation . . . . .	306
9.101.2.1 formatInteger . . . . .	306
9.101.2.2 formatIntegerArray . . . . .	307
9.101.2.3 formatValue . . . . .	307
9.102 QELabel Class Reference . . . . .	308
9.102.1 Detailed Description . . . . .	311
9.102.2 Member Enumeration Documentation . . . . .	311
9.102.2.1 ArrayActions . . . . .	311
9.102.2.2 DisplayAlarmStateOptions . . . . .	312
9.102.2.3 Formats . . . . .	312
9.102.2.4 Notations . . . . .	312
9.102.2.5 Separators . . . . .	312
9.102.2.6 UpdateOptions . . . . .	313
9.102.2.7 updateOptions . . . . .	313
9.102.2.8 UserLevels . . . . .	313
9.102.3 Constructor & Destructor Documentation . . . . .	313

9.102.3.1 QELabel . . . . .	313
9.102.3.2 QELabel . . . . .	314
9.102.4 Member Function Documentation . . . . .	314
9.102.4.1 dbValueChanged . . . . .	314
9.102.4.2 setManagedVisible . . . . .	314
9.102.5 Property Documentation . . . . .	314
9.102.5.1 addUnits . . . . .	314
9.102.5.2 allowDrop . . . . .	314
9.102.5.3 arrayAction . . . . .	314
9.102.5.4 arrayIndex . . . . .	315
9.102.5.5 defaultStyle . . . . .	315
9.102.5.6 displayAlarmState . . . . .	315
9.102.5.7 displayAlarmStateOption . . . . .	315
9.102.5.8 format . . . . .	315
9.102.5.9 int . . . . .	315
9.102.5.10leadingZero . . . . .	315
9.102.5.11localEnumeration . . . . .	316
9.102.5.12notation . . . . .	316
9.102.5.13 pixmap0 . . . . .	316
9.102.5.14 pixmap1 . . . . .	316
9.102.5.15 pixmap2 . . . . .	317
9.102.5.16 pixmap3 . . . . .	317
9.102.5.17 pixmap4 . . . . .	317
9.102.5.18 pixmap5 . . . . .	317
9.102.5.19 pixmap6 . . . . .	317
9.102.5.20 pixmap7 . . . . .	317
9.102.5.21 precision . . . . .	317
9.102.5.22 radix . . . . .	317
9.102.5.23 separator . . . . .	317
9.102.5.24 styleSheet . . . . .	318
9.102.5.25 trailingZeros . . . . .	318
9.102.5.26 updateOption . . . . .	318
9.102.5.27 useDbPrecision . . . . .	318
9.102.5.28 userLevelEnabled . . . . .	318

9.102.5.29userLevelEngineerStyle . . . . .	318
9.102.5.30userLevelScientistStyle . . . . .	318
9.102.5.31userLevelUserStyle . . . . .	319
9.102.5.32userLevelVisibility . . . . .	319
9.102.5.33variable . . . . .	319
9.102.5.34variableAsToolTip . . . . .	319
9.102.5.35variableSubstitutions . . . . .	319
9.102.5.36visible . . . . .	319
9.103QELineEdit Class Reference . . . . .	321
9.103.1 Member Enumeration Documentation . . . . .	323
9.103.1.1 ArrayActions . . . . .	323
9.103.1.2 Formats . . . . .	323
9.103.1.3 Notations . . . . .	323
9.103.1.4 Separators . . . . .	323
9.103.2 Constructor & Destructor Documentation . . . . .	324
9.103.2.1 QELineEdit . . . . .	324
9.103.2.2 QELineEdit . . . . .	324
9.103.3 Member Function Documentation . . . . .	324
9.103.3.1 dbValueChanged . . . . .	324
9.103.4 Property Documentation . . . . .	324
9.103.4.1 addUnits . . . . .	324
9.103.4.2 arrayAction . . . . .	324
9.103.4.3 format . . . . .	325
9.103.4.4 leadingZero . . . . .	325
9.103.4.5 localEnumeration . . . . .	325
9.103.4.6 notation . . . . .	326
9.103.4.7 precision . . . . .	326
9.103.4.8 radix . . . . .	326
9.103.4.9 separator . . . . .	326
9.103.4.10trailingZeros . . . . .	326
9.103.4.11useDbPrecision . . . . .	326
9.104QELineEditManager Class Reference . . . . .	327
9.105QELink Class Reference . . . . .	328
9.106QELocalEnumeration Class Reference . . . . .	330

9.106.1 Detailed Description . . . . .	330
9.106.2 Constructor & Destructor Documentation . . . . .	330
9.106.2.1 QELocalEnumeration . . . . .	330
9.106.2.2 QELocalEnumeration . . . . .	330
9.106.3 Member Function Documentation . . . . .	331
9.106.3.1 getLocalEnumeration . . . . .	331
9.106.3.2 isDefined . . . . .	331
9.106.3.3 setLocalEnumeration . . . . .	331
9.106.3.4 text.ToDouble . . . . .	331
9.106.3.5 text.ToInt . . . . .	332
9.106.3.6 textToValue . . . . .	332
9.106.3.7 valueToText . . . . .	332
9.107 QELog Class Reference . . . . .	333
9.107.1 Member Enumeration Documentation . . . . .	335
9.107.1.1 DisplayAlarmStateOptions . . . . .	335
9.107.1.2 UserLevels . . . . .	336
9.107.2 Member Function Documentation . . . . .	336
9.107.2.1 setManagedVisible . . . . .	336
9.107.3 Property Documentation . . . . .	336
9.107.3.1 allowDrop . . . . .	336
9.107.3.2 defaultStyle . . . . .	336
9.107.3.3 displayAlarmState . . . . .	336
9.107.3.4 displayAlarmStateOption . . . . .	336
9.107.3.5 int . . . . .	337
9.107.3.6 styleSheet . . . . .	337
9.107.3.7 userLevelEnabled . . . . .	337
9.107.3.8 userLevelEngineerStyle . . . . .	337
9.107.3.9 userLevelScientistStyle . . . . .	337
9.107.3.10 userLevelUserStyle . . . . .	338
9.107.3.11 userLevelVisibility . . . . .	338
9.107.3.12 variableAsToolTip . . . . .	338
9.107.3.13 visible . . . . .	338
9.108 QELogin Class Reference . . . . .	339
9.109 QELoginDialog Class Reference . . . . .	340

9.110QE_MenuButton Class Reference . . . . .	341
9.110.1 Detailed Description . . . . .	342
9.110.2 Property Documentation . . . . .	342
9.110.2.1 allowDrop . . . . .	342
9.110.2.2 defaultSubstitutions . . . . .	342
9.110.2.3 displayAlarmStateOption . . . . .	342
9.110.2.4 variableAsToolTip . . . . .	342
9.111QE_MenuButtonData Class Reference . . . . .	343
9.112QE_MenuButtonItem Class Reference . . . . .	344
9.112.1 Detailed Description . . . . .	344
9.113QE_MenuButtonManager Class Reference . . . . .	345
9.114QE_MenuButtonModel Class Reference . . . . .	346
9.115QE_MenuButtonSetupDialog Class Reference . . . . .	348
9.116QE_MenuButtonTaskMenu Class Reference . . . . .	349
9.117QE_MenuButtonTaskMenuFactory Class Reference . . . . .	350
9.118QE_NumericEdit Class Reference . . . . .	351
9.118.1 Constructor & Destructor Documentation . . . . .	353
9.118.1.1 QE_NumericEdit . . . . .	353
9.118.1.2 QE_NumericEdit . . . . .	353
9.118.2 Member Function Documentation . . . . .	353
9.118.2.1 dbConnectionChanged . . . . .	353
9.118.2.2 dbValueChanged . . . . .	353
9.118.3 Property Documentation . . . . .	354
9.118.3.1 addUnits . . . . .	354
9.118.3.2 alignment . . . . .	354
9.118.3.3 allowFocusUpdate . . . . .	354
9.118.3.4 arrayIndex . . . . .	354
9.118.3.5 autoScale . . . . .	354
9.118.3.6 cleanText . . . . .	354
9.118.3.7 confirmWrite . . . . .	354
9.118.3.8 frame . . . . .	354
9.118.3.9 leadingZeros . . . . .	355
9.118.3.10 maximum . . . . .	355
9.118.3.11 minimum . . . . .	355

9.118.3.12notation . . . . .	355
9.118.3.13precision . . . . .	355
9.118.3.14radix . . . . .	355
9.118.3.15separator . . . . .	355
9.118.3.16variable . . . . .	355
9.118.3.17variableSubstitutions . . . . .	356
9.118.3.18writeOnChange . . . . .	356
9.118.3.19writeOnEnter . . . . .	356
9.118.3.20writeOnFinish . . . . .	356
9.118.3.21writeOnLoseFocus . . . . .	356
9.119QENumericEditManager Class Reference . . . . .	357
9.120QEPeriodic Class Reference . . . . .	358
9.120.1 Member Enumeration Documentation . . . . .	362
9.120.1.1 DisplayAlarmStateOptions . . . . .	362
9.120.1.2 UserLevels . . . . .	362
9.120.2 Member Function Documentation . . . . .	362
9.120.2.1 dbElementChanged . . . . .	362
9.120.2.2 dbValueChanged . . . . .	363
9.120.3 Member Data Documentation . . . . .	363
9.120.3.1 allowDrop . . . . .	363
9.120.4 Property Documentation . . . . .	363
9.120.4.1 displayAlarmState . . . . .	363
9.120.4.2 displayAlarmStateOption . . . . .	363
9.120.4.3 int . . . . .	363
9.120.4.4 readbackLabelVariable1 . . . . .	363
9.120.4.5 readbackLabelVariable2 . . . . .	364
9.120.4.6 subscribe . . . . .	364
9.120.4.7 userLevelEnabled . . . . .	364
9.120.4.8 userLevelEngineerStyle . . . . .	364
9.120.4.9 userLevelScientistStyle . . . . .	364
9.120.4.10 userLevelUserStyle . . . . .	364
9.120.4.11 userLevelVisibility . . . . .	365
9.120.4.12 variableAsToolTip . . . . .	365
9.120.4.13 variableSubstitutions . . . . .	365

9.120.4.14visible . . . . .	365
9.120.4.15writeButtonVariable1 . . . . .	365
9.120.4.16writeButtonVariable2 . . . . .	365
9.121QEPeriodicComponentData Class Reference . . . . .	366
9.122QEPeriodicTaskMenu Class Reference . . . . .	367
9.123QEPeriodicTaskMenuFactory Class Reference . . . . .	368
9.124QEpicsPV Class Reference . . . . .	369
9.125QEPlot Class Reference . . . . .	370
9.125.1 Member Enumeration Documentation . . . . .	374
9.125.1.1 DisplayAlarmStateOptions . . . . .	374
9.125.1.2 UserLevels . . . . .	374
9.125.2 Member Function Documentation . . . . .	374
9.125.2.1 dbValueChanged . . . . .	374
9.125.2.2 dbValueChanged . . . . .	374
9.125.2.3 setManagedVisible . . . . .	374
9.125.3 Member Data Documentation . . . . .	375
9.125.3.1 allowDrop . . . . .	375
9.125.4 Property Documentation . . . . .	375
9.125.4.1 defaultStyle . . . . .	375
9.125.4.2 displayAlarmState . . . . .	375
9.125.4.3 displayAlarmStateOption . . . . .	375
9.125.4.4 int . . . . .	375
9.125.4.5 styleSheet . . . . .	375
9.125.4.6 userLevelEnabled . . . . .	376
9.125.4.7 userLevelEngineerStyle . . . . .	376
9.125.4.8 userLevelScientistStyle . . . . .	376
9.125.4.9 userLevelUserStyle . . . . .	376
9.125.4.10userLevelVisibility . . . . .	376
9.125.4.11variable1 . . . . .	377
9.125.4.12variable2 . . . . .	377
9.125.4.13variable3 . . . . .	377
9.125.4.14variable4 . . . . .	377
9.125.4.15variableAsToolTip . . . . .	377
9.125.4.16variableSubstitutions . . . . .	377

9.125.4.1 <b>visible</b> . . . . .	377
<b>9.126QEPlotter Class Reference</b> . . . . .	378
9.126.1 Property Documentation . . . . .	382
9.126.1.1 <b>aliasNames</b> . . . . .	382
9.126.1.2 <b>variableSubstitutions</b> . . . . .	382
<b>9.127QEPlotterItemDialog Class Reference</b> . . . . .	383
<b>9.128QEPlotterManager Class Reference</b> . . . . .	384
<b>9.129QEPlotterMenu Class Reference</b> . . . . .	385
9.129.1 Detailed Description . . . . .	385
<b>9.130QEPlotterNames Class Reference</b> . . . . .	386
<b>9.131QEPlotterPushButtonSpecifications Struct Reference</b> . . . . .	388
<b>9.132QEPlotterState Class Reference</b> . . . . .	389
<b>9.133QEPlotterStateList Class Reference</b> . . . . .	390
<b>9.134QEPlotterToolBar Class Reference</b> . . . . .	391
9.134.1 Detailed Description . . . . .	391
<b>9.135QEPushButton Class Reference</b> . . . . .	392
9.135.1 Member Enumeration Documentation . . . . .	396
9.135.1.1 <b>ArrayActions</b> . . . . .	396
9.135.1.2 <b>CreationOptionNames</b> . . . . .	396
9.135.1.3 <b>DisplayAlarmStateOptions</b> . . . . .	397
9.135.1.4 <b>Formats</b> . . . . .	397
9.135.1.5 <b>Notations</b> . . . . .	398
9.135.1.6 <b>ProgramStartupOptionNames</b> . . . . .	398
9.135.1.7 <b>UpdateOptions</b> . . . . .	398
9.135.1.8 <b>UserLevels</b> . . . . .	398
9.135.2 Constructor & Destructor Documentation . . . . .	399
9.135.2.1 <b>QEPushButton</b> . . . . .	399
9.135.2.2 <b>QEPushButton</b> . . . . .	399
9.135.3 Member Function Documentation . . . . .	399
9.135.3.1 <b>clicked</b> . . . . .	399
9.135.3.2 <b>dbValueChanged</b> . . . . .	399
9.135.3.3 <b>pressed</b> . . . . .	399
9.135.3.4 <b>released</b> . . . . .	399
9.135.3.5 <b>requestAction</b> . . . . .	400

9.135.3.6 setManagedVisible . . . . .	400
9.135.4 Property Documentation . . . . .	400
9.135.4.1 addUnits . . . . .	400
9.135.4.2 alignment . . . . .	400
9.135.4.3 allowDrop . . . . .	400
9.135.4.4 altReadbackVariable . . . . .	400
9.135.4.5 arguments . . . . .	400
9.135.4.6 arrayAction . . . . .	401
9.135.4.7 arrayIndex . . . . .	401
9.135.4.8 clickCheckedText . . . . .	401
9.135.4.9 clickText . . . . .	401
9.135.4.10confirmAction . . . . .	401
9.135.4.11confirmText . . . . .	402
9.135.4.12creationOption . . . . .	402
9.135.4.13customisationName . . . . .	402
9.135.4.14defaultStyle . . . . .	402
9.135.4.15disabledRecordPolicy . . . . .	402
9.135.4.16displayAlarmState . . . . .	402
9.135.4.17displayAlarmStateOption . . . . .	403
9.135.4.18format . . . . .	403
9.135.4.19guiFile . . . . .	403
9.135.4.20int . . . . .	403
9.135.4.21labelText . . . . .	403
9.135.4.22leadingZero . . . . .	404
9.135.4.23localEnumeration . . . . .	404
9.135.4.24notation . . . . .	404
9.135.4.25password . . . . .	404
9.135.4.26 pixmap0 . . . . .	405
9.135.4.27 pixmap1 . . . . .	405
9.135.4.28 pixmap2 . . . . .	405
9.135.4.29 pixmap3 . . . . .	405
9.135.4.30 pixmap4 . . . . .	405
9.135.4.31 pixmap5 . . . . .	405
9.135.4.32 pixmap6 . . . . .	405

9.135.4.33pixmap7 . . . . .	405
9.135.4.34precision . . . . .	405
9.135.4.35pressText . . . . .	406
9.135.4.36prioritySubstitutions . . . . .	406
9.135.4.37program . . . . .	406
9.135.4.38programStartupOption . . . . .	406
9.135.4.39releaseText . . . . .	406
9.135.4.40styleSheet . . . . .	406
9.135.4.41subscribe . . . . .	406
9.135.4.42trailingZeros . . . . .	407
9.135.4.43updateOption . . . . .	407
9.135.4.44useDbPrecision . . . . .	407
9.135.4.45userLevelEnabled . . . . .	407
9.135.4.46userLevelEngineerStyle . . . . .	407
9.135.4.47userLevelScientistStyle . . . . .	407
9.135.4.48userLevelUserStyle . . . . .	408
9.135.4.49userLevelVisibility . . . . .	408
9.135.4.50variable . . . . .	408
9.135.4.51variableAsToolTip . . . . .	408
9.135.4.52variableSubstitutions . . . . .	408
9.135.4.53visible . . . . .	408
9.135.4.54writeOnClick . . . . .	408
9.135.4.55writeOnPress . . . . .	409
9.135.4.56writeOnRelease . . . . .	409
<b>9.136QEPvFrame Class Reference . . . . .</b>	<b>410</b>
<b>9.136.1 Member Function Documentation . . . . .</b>	<b>410</b>
9.136.1.1 dbConnectionChanged . . . . .	410
9.136.1.2 dbValueChanged . . . . .	411
<b>9.136.2 Property Documentation . . . . .</b>	<b>411</b>
9.136.2.1 arrayIndex . . . . .	411
9.136.2.2 variable . . . . .	411
9.136.2.3 variableSubstitutions . . . . .	411
<b>9.137QEPvFrameManager Class Reference . . . . .</b>	<b>412</b>
<b>9.138QEPvLoadSave Class Reference . . . . .</b>	<b>413</b>

9.138.1 Constructor & Destructor Documentation . . . . .	414
9.138.1.1 QEPvLoadSave . . . . .	414
9.138.2 Property Documentation . . . . .	414
9.138.2.1 configurationFileLeft . . . . .	414
9.138.2.2 confirmAction . . . . .	414
9.138.2.3 defaultDir . . . . .	414
9.138.2.4 defaultSubstitutions . . . . .	415
9.139 QEPvLoadSaveCommon Class Reference . . . . .	416
9.140 QEPvLoadSaveCompare Class Reference . . . . .	417
9.140.1 Detailed Description . . . . .	417
9.141 QEPvLoadSaveGroup Class Reference . . . . .	418
9.142 QEPvLoadSaveGroupNameDialog Class Reference . . . . .	419
9.143 QEPvLoadSaveItem Class Reference . . . . .	420
9.143.1 Detailed Description . . . . .	421
9.144 QEPvLoadSaveLeaf Class Reference . . . . .	422
9.145 QEPvLoadSaveManager Class Reference . . . . .	423
9.146 QEPvLoadSaveModel Class Reference . . . . .	424
9.146.1 Detailed Description . . . . .	425
9.147 QEPvLoadSaveTimeDialog Class Reference . . . . .	426
9.148 QEPvLoadSaveUtilities Class Reference . . . . .	427
9.149 QEPvLoadSaveValueEditDialog Class Reference . . . . .	428
9.150 QEPVNameLists Class Reference . . . . .	429
9.151 QEPvProperties Class Reference . . . . .	430
9.151.1 Property Documentation . . . . .	431
9.151.1.1 variable . . . . .	431
9.151.1.2 variableSubstitutions . . . . .	431
9.152 QEPvPropertiesManager Class Reference . . . . .	432
9.153 QERadioButton Class Reference . . . . .	433
9.153.1 Member Enumeration Documentation . . . . .	437
9.153.1.1 ArrayActions . . . . .	437
9.153.1.2 CreationOptionNames . . . . .	437
9.153.1.3 DisplayAlarmStateOptions . . . . .	438
9.153.1.4 Formats . . . . .	438
9.153.1.5 Notations . . . . .	439

9.153.1.6 ProgramStartupOptionNames . . . . .	439
9.153.1.7 Separators . . . . .	439
9.153.1.8 UpdateOptions . . . . .	439
9.153.1.9 UserLevels . . . . .	440
9.153.2 Constructor & Destructor Documentation . . . . .	440
9.153.2.1 QERadioButton . . . . .	440
9.153.2.2 QERadioButton . . . . .	440
9.153.3 Member Function Documentation . . . . .	440
9.153.3.1 clicked . . . . .	440
9.153.3.2 dbValueChanged . . . . .	440
9.153.3.3 pressed . . . . .	441
9.153.3.4 released . . . . .	441
9.153.3.5 requestAction . . . . .	441
9.153.3.6 setManagedVisible . . . . .	441
9.153.4 Property Documentation . . . . .	441
9.153.4.1 addUnits . . . . .	441
9.153.4.2 alignment . . . . .	441
9.153.4.3 allowDrop . . . . .	441
9.153.4.4 arguments . . . . .	442
9.153.4.5 arrayAction . . . . .	442
9.153.4.6 arrayIndex . . . . .	442
9.153.4.7 clickCheckedText . . . . .	442
9.153.4.8 clickText . . . . .	442
9.153.4.9 confirmAction . . . . .	443
9.153.4.10confirmText . . . . .	443
9.153.4.11creationOption . . . . .	443
9.153.4.12customisationName . . . . .	443
9.153.4.13defaultStyle . . . . .	443
9.153.4.14disabledRecordPolicy . . . . .	443
9.153.4.15displayAlarmState . . . . .	444
9.153.4.16displayAlarmStateOption . . . . .	444
9.153.4.17format . . . . .	444
9.153.4.18guiFile . . . . .	444
9.153.4.19nt . . . . .	444

9.153.4.20labelText . . . . .	444
9.153.4.21leadingZero . . . . .	445
9.153.4.22localEnumeration . . . . .	445
9.153.4.23notation . . . . .	445
9.153.4.24password . . . . .	446
9.153.4.25 pixmap0 . . . . .	446
9.153.4.26 pixmap1 . . . . .	446
9.153.4.27 pixmap2 . . . . .	446
9.153.4.28 pixmap3 . . . . .	446
9.153.4.29 pixmap4 . . . . .	446
9.153.4.30 pixmap5 . . . . .	446
9.153.4.31 pixmap6 . . . . .	446
9.153.4.32 pixmap7 . . . . .	446
9.153.4.33precision . . . . .	447
9.153.4.34pressText . . . . .	447
9.153.4.35prioritySubstitutions . . . . .	447
9.153.4.36program . . . . .	447
9.153.4.37programStartupOption . . . . .	447
9.153.4.38radix . . . . .	447
9.153.4.39releaseText . . . . .	447
9.153.4.40separator . . . . .	448
9.153.4.41stylesheet . . . . .	448
9.153.4.42subscribe . . . . .	448
9.153.4.43trailingZeros . . . . .	448
9.153.4.44updateOption . . . . .	448
9.153.4.45useDbPrecision . . . . .	448
9.153.4.46userLevelEnabled . . . . .	448
9.153.4.47userLevelEngineerStyle . . . . .	448
9.153.4.48userLevelScientistStyle . . . . .	449
9.153.4.49userLevelUserStyle . . . . .	449
9.153.4.50userLevelVisibility . . . . .	449
9.153.4.51variable . . . . .	449
9.153.4.52variableAsToolTip . . . . .	449
9.153.4.53variableSubstitutions . . . . .	449

9.153.4.54visible . . . . .	450
9.153.4.55writeOnClick . . . . .	450
9.153.4.56writeOnPress . . . . .	450
9.153.4.57writeOnRelease . . . . .	450
9.154QERadioGroup Class Reference . . . . .	451
9.154.1 Constructor & Destructor Documentation . . . . .	452
9.154.1.1 QERadioGroup . . . . .	452
9.154.1.2 QERadioGroup . . . . .	452
9.154.1.3 QERadioGroup . . . . .	453
9.154.2 Member Function Documentation . . . . .	453
9.154.2.1 dbConnectionChanged . . . . .	453
9.154.2.2 dbValueChanged . . . . .	453
9.154.3 Property Documentation . . . . .	453
9.154.3.1 arrayIndex . . . . .	453
9.154.3.2 buttonOrder . . . . .	453
9.154.3.3 buttonStyle . . . . .	453
9.154.3.4 columns . . . . .	453
9.154.3.5 localEnumeration . . . . .	454
9.154.3.6 spacing . . . . .	454
9.154.3.7 substitutedTitle . . . . .	454
9.154.3.8 useDbEnumerations . . . . .	454
9.154.3.9 variable . . . . .	454
9.154.3.10variableSubstitutions . . . . .	454
9.155QERadioGroupManager Class Reference . . . . .	455
9.156QERecipe Class Reference . . . . .	456
9.157QERecordSpec Class Reference . . . . .	458
9.158QERecordSpecList Class Reference . . . . .	459
9.159QEResizableFrame Class Reference . . . . .	460
9.159.1 Detailed Description . . . . .	461
9.159.2 Member Function Documentation . . . . .	461
9.159.2.1 setWidget . . . . .	461
9.159.3 Property Documentation . . . . .	461
9.159.3.1 allowedMaximum . . . . .	461
9.159.3.2 allowedMinimum . . . . .	461

9.159.3.3 grabbingEdge . . . . .	461
9.160QEResizableFrameManager Class Reference . . . . .	462
9.161QEScalarHistogram Class Reference . . . . .	463
9.161.1 Member Enumeration Documentation . . . . .	467
9.161.1.1 ScaleModes . . . . .	467
9.161.2 Property Documentation . . . . .	467
9.161.2.1 variableSubstitutions . . . . .	467
9.162QEScalarHistogramManager Class Reference . . . . .	468
9.163QEScratchPad Class Reference . . . . .	469
9.163.1 Detailed Description . . . . .	470
9.164QEScratchPadManager Class Reference . . . . .	471
9.165QEScratchPadMenu Class Reference . . . . .	472
9.166QEScript Class Reference . . . . .	473
9.166.1 Detailed Description . . . . .	477
9.166.2 Member Enumeration Documentation . . . . .	478
9.166.2.1 DisplayAlarmStateOptions . . . . .	478
9.166.2.2 UserLevels . . . . .	478
9.166.3 Member Function Documentation . . . . .	478
9.166.3.1 setManagedVisible . . . . .	478
9.166.4 Property Documentation . . . . .	478
9.166.4.1 allowDrop . . . . .	478
9.166.4.2 defaultStyle . . . . .	478
9.166.4.3 displayAlarmState . . . . .	479
9.166.4.4 displayAlarmStateOption . . . . .	479
9.166.4.5 int . . . . .	479
9.166.4.6 styleSheet . . . . .	479
9.166.4.7 userLevelEnabled . . . . .	479
9.166.4.8 userLevelEngineerStyle . . . . .	479
9.166.4.9 userLevelScientistStyle . . . . .	480
9.166.4.10 userLevelUserStyle . . . . .	480
9.166.4.11 userLevelVisibility . . . . .	480
9.166.4.12 variableAsToolTip . . . . .	480
9.166.4.13 visible . . . . .	480
9.167QEShape Class Reference . . . . .	482

9.167.1 Detailed Description . . . . .	487
9.167.2 Member Enumeration Documentation . . . . .	487
9.167.2.1 animationOptions . . . . .	487
9.167.2.2 DisplayAlarmStateOptions . . . . .	487
9.167.2.3 shapeOptions . . . . .	487
9.167.2.4 UserLevels . . . . .	487
9.167.3 Constructor & Destructor Documentation . . . . .	488
9.167.3.1 QEShape . . . . .	488
9.167.3.2 QEShape . . . . .	488
9.167.4 Member Function Documentation . . . . .	488
9.167.4.1 dbValueChanged1 . . . . .	488
9.167.4.2 dbValueChanged2 . . . . .	488
9.167.4.3 dbValueChanged3 . . . . .	488
9.167.4.4 dbValueChanged4 . . . . .	489
9.167.4.5 dbValueChanged5 . . . . .	489
9.167.4.6 dbValueChanged6 . . . . .	489
9.167.4.7 setManagedVisible . . . . .	489
9.167.5 Property Documentation . . . . .	489
9.167.5.1 allowDrop . . . . .	489
9.167.5.2 animation1 . . . . .	489
9.167.5.3 animation2 . . . . .	489
9.167.5.4 animation3 . . . . .	490
9.167.5.5 animation4 . . . . .	490
9.167.5.6 animation5 . . . . .	490
9.167.5.7 animation6 . . . . .	490
9.167.5.8 color1 . . . . .	490
9.167.5.9 color10 . . . . .	490
9.167.5.10color2 . . . . .	490
9.167.5.11color3 . . . . .	490
9.167.5.12color4 . . . . .	490
9.167.5.13color5 . . . . .	491
9.167.5.14color6 . . . . .	491
9.167.5.15color7 . . . . .	491
9.167.5.16color8 . . . . .	491

9.167.5.17color9 . . . . .	491
9.167.5.18defaultStyle . . . . .	491
9.167.5.19displayAlarmState . . . . .	491
9.167.5.20displayAlarmStateOption . . . . .	491
9.167.5.21int . . . . .	492
9.167.5.22offset1 . . . . .	492
9.167.5.23offset2 . . . . .	492
9.167.5.24offset3 . . . . .	492
9.167.5.25offset4 . . . . .	492
9.167.5.26offset5 . . . . .	492
9.167.5.27offset6 . . . . .	492
9.167.5.28point1 . . . . .	493
9.167.5.29point10 . . . . .	493
9.167.5.30point2 . . . . .	493
9.167.5.31point3 . . . . .	493
9.167.5.32point4 . . . . .	493
9.167.5.33point5 . . . . .	493
9.167.5.34point6 . . . . .	493
9.167.5.35point7 . . . . .	493
9.167.5.36point8 . . . . .	494
9.167.5.37point9 . . . . .	494
9.167.5.38scale2 . . . . .	494
9.167.5.39scale3 . . . . .	494
9.167.5.40scale4 . . . . .	494
9.167.5.41scale5 . . . . .	494
9.167.5.42scale6 . . . . .	494
9.167.5.43styleSheet . . . . .	494
9.167.5.44userLevelEnabled . . . . .	494
9.167.5.45userLevelEngineerStyle . . . . .	495
9.167.5.46userLevelScientistStyle . . . . .	495
9.167.5.47userLevelUserStyle . . . . .	495
9.167.5.48userLevelVisibility . . . . .	495
9.167.5.49variable1 . . . . .	496
9.167.5.50variable2 . . . . .	496

9.167.5.5 variable3 . . . . .	496
9.167.5.52 variable4 . . . . .	496
9.167.5.53 variable5 . . . . .	496
9.167.5.54 variable6 . . . . .	496
9.167.5.55 variableAsToolTip . . . . .	496
9.167.5.56 variableSubstitutions . . . . .	496
9.167.5.57 visible . . . . .	497
<b>9.168 QESimpleShape Class Reference . . . . .</b>	<b>498</b>
9.168.1 Detailed Description . . . . .	500
9.168.2 Member Enumeration Documentation . . . . .	500
9.168.2.1 DisplayAlarmStateOptions . . . . .	500
9.168.2.2 UserLevels . . . . .	501
9.168.3 Member Function Documentation . . . . .	501
9.168.3.1 dbConnectionChanged . . . . .	501
9.168.3.2 dbValueChanged . . . . .	501
9.168.3.3 setManagedVisible . . . . .	501
9.168.4 Property Documentation . . . . .	501
9.168.4.1 addUnits . . . . .	501
9.168.4.2 allowDrop . . . . .	502
9.168.4.3 arrayIndex . . . . .	502
9.168.4.4 defaultStyle . . . . .	502
9.168.4.5 displayAlarmState . . . . .	502
9.168.4.6 displayAlarmStateOption . . . . .	502
9.168.4.7 int . . . . .	502
9.168.4.8 localEnumeration . . . . .	502
9.168.4.9 styleSheet . . . . .	503
9.168.4.10 userLevelEnabled . . . . .	503
9.168.4.11 userLevelEngineerStyle . . . . .	503
9.168.4.12 userLevelScientistStyle . . . . .	504
9.168.4.13 userLevelUserStyle . . . . .	504
9.168.4.14 userLevelVisibility . . . . .	504
9.168.4.15 variable . . . . .	504
9.168.4.16 variableAsToolTip . . . . .	504
9.168.4.17 variableSubstitutions . . . . .	504

9.168.4.18visible . . . . .	505
9.169QESimpleShapeManager Class Reference . . . . .	506
9.170QESlider Class Reference . . . . .	507
9.170.1 Member Enumeration Documentation . . . . .	509
9.170.1.1 DisplayAlarmStateOptions . . . . .	509
9.170.1.2 UserLevels . . . . .	509
9.170.2 Member Function Documentation . . . . .	509
9.170.2.1 dbValueChanged . . . . .	509
9.170.2.2 setManagedVisible . . . . .	510
9.170.3 Member Data Documentation . . . . .	510
9.170.3.1 writeOnChange . . . . .	510
9.170.4 Property Documentation . . . . .	510
9.170.4.1 allowDrop . . . . .	510
9.170.4.2 allowFocusUpdate . . . . .	510
9.170.4.3 arrayIndex . . . . .	510
9.170.4.4 defaultStyle . . . . .	510
9.170.4.5 displayAlarmState . . . . .	510
9.170.4.6 displayAlarmStateOption . . . . .	511
9.170.4.7 int . . . . .	511
9.170.4.8 styleSheet . . . . .	511
9.170.4.9 subscribe . . . . .	511
9.170.4.10userLevelEnabled . . . . .	511
9.170.4.11userLevelEngineerStyle . . . . .	511
9.170.4.12userLevelScientistStyle . . . . .	512
9.170.4.13userLevelUserStyle . . . . .	512
9.170.4.14userLevelVisibility . . . . .	512
9.170.4.15variable . . . . .	512
9.170.4.16variableAsToolTip . . . . .	512
9.170.4.17variableSubstitutions . . . . .	512
9.170.4.18visible . . . . .	513
9.171QESpinBox Class Reference . . . . .	514
9.171.1 Member Enumeration Documentation . . . . .	516
9.171.1.1 DisplayAlarmStateOptions . . . . .	516
9.171.1.2 UserLevels . . . . .	516

9.171.2 Member Function Documentation . . . . .	517
9.171.2.1 dbValueChanged . . . . .	517
9.171.2.2 setManagedVisible . . . . .	517
9.171.3 Property Documentation . . . . .	517
9.171.3.1 allowDrop . . . . .	517
9.171.3.2 allowFocusUpdate . . . . .	517
9.171.3.3 arrayIndex . . . . .	517
9.171.3.4 defaultStyle . . . . .	517
9.171.3.5 displayAlarmState . . . . .	517
9.171.3.6 displayAlarmStateOption . . . . .	518
9.171.3.7 int . . . . .	518
9.171.3.8 styleSheet . . . . .	518
9.171.3.9 subscribe . . . . .	518
9.171.3.10 userLevelEnabled . . . . .	518
9.171.3.11 userLevelEngineerStyle . . . . .	518
9.171.3.12 userLevelScientistStyle . . . . .	519
9.171.3.13 userLevelUserStyle . . . . .	519
9.171.3.14 userLevelVisibility . . . . .	519
9.171.3.15 variable . . . . .	519
9.171.3.16 variableAsToolTip . . . . .	519
9.171.3.17 variableSubstitutions . . . . .	519
9.171.3.18 visible . . . . .	520
9.172 QQString Class Reference . . . . .	521
9.173 QQStringFormatting Class Reference . . . . .	522
9.173.1 Member Enumeration Documentation . . . . .	523
9.173.1.1 arrayActions . . . . .	523
9.173.1.2 formats . . . . .	523
9.173.1.3 notations . . . . .	523
9.173.1.4 separators . . . . .	523
9.174 QEStripChart Class Reference . . . . .	525
9.174.1 Property Documentation . . . . .	527
9.174.1.1 variableSubstitutions . . . . .	527
9.175 QEStripChartAdjustPVDialog Class Reference . . . . .	528
9.176 QEStripChartContextMenu Class Reference . . . . .	529

9.176.1 Constructor & Destructor Documentation . . . . .	529
9.176.1.1 QEStripChartContextMenu . . . . .	529
9.177QEStripChartDurationDialog Class Reference . . . . .	530
9.178QEStripChartItem Class Reference . . . . .	531
9.179QEStripChartNames Class Reference . . . . .	533
9.180QEStripChartPushButtonSpecifications Struct Reference . . . . .	535
9.181QEStripChartRangeDialog Class Reference . . . . .	536
9.182QEStripChartState Class Reference . . . . .	537
9.183QEStripChartStateList Class Reference . . . . .	538
9.184QEStripChartStatistics Class Reference . . . . .	539
9.185QEStripChartTimeDialog Class Reference . . . . .	540
9.186QEStripChartToolBar Class Reference . . . . .	541
9.186.1 Detailed Description . . . . .	542
9.187QESubstitutedLabel Class Reference . . . . .	543
9.187.1 Member Data Documentation . . . . .	543
9.187.1.1 labelText . . . . .	543
9.187.2 Property Documentation . . . . .	543
9.187.2.1 textSubstitutions . . . . .	543
9.188QETable Class Reference . . . . .	544
9.188.1 Detailed Description . . . . .	546
9.188.2 Constructor & Destructor Documentation . . . . .	546
9.188.2.1 QETable . . . . .	546
9.188.3 Member Function Documentation . . . . .	546
9.188.3.1 dbValueChanged . . . . .	546
9.188.4 Property Documentation . . . . .	547
9.188.4.1 colWidthMinimum . . . . .	547
9.188.4.2 displayMaximum . . . . .	547
9.188.4.3 gridStyle . . . . .	547
9.188.4.4 orientation . . . . .	547
9.188.4.5 showGrid . . . . .	547
9.188.4.6 titles . . . . .	547
9.188.4.7 variableSubstitutions . . . . .	547
9.189QETableManager Class Reference . . . . .	548
9.190QEWaveformHistogram Class Reference . . . . .	549

9.190.1 Detailed Description . . . . .	550
9.190.2 Member Enumeration Documentation . . . . .	550
9.190.2.1 ScaleModes . . . . .	550
9.191 QEWaveformHistogramManager Class Reference . . . . .	551
9.192 QNumericEdit Class Reference . . . . .	552
9.192.1 Detailed Description . . . . .	554
9.192.2 Member Enumeration Documentation . . . . .	554
9.192.2.1 Notations . . . . .	554
9.192.3 Property Documentation . . . . .	554
9.192.3.1 alignment . . . . .	554
9.192.3.2 cleanText . . . . .	554
9.192.3.3 frame . . . . .	554
9.192.3.4 leadingZeros . . . . .	554
9.192.3.5 notation . . . . .	555
9.192.3.6 precision . . . . .	555
9.192.3.7 prefix . . . . .	555
9.192.3.8 suffix . . . . .	555
9.193 QNumericEditManager Class Reference . . . . .	556
9.194 QRadioGroup Class Reference . . . . .	557
9.194.1 Member Enumeration Documentation . . . . .	558
9.194.1.1 ButtonOrders . . . . .	558
9.194.1.2 ButtonStyles . . . . .	558
9.194.2 Constructor & Destructor Documentation . . . . .	558
9.194.2.1 QRadioGroup . . . . .	558
9.194.2.2 QRadioGroup . . . . .	558
9.195 QRadioGroupManager Class Reference . . . . .	559
9.196 QSimpleshape Class Reference . . . . .	560
9.196.1 Detailed Description . . . . .	563
9.196.2 Member Enumeration Documentation . . . . .	563
9.196.2.1 TextFormats . . . . .	563
9.196.3 Constructor & Destructor Documentation . . . . .	563
9.196.3.1 QSimpleshape . . . . .	563
9.196.4 Property Documentation . . . . .	563
9.196.4.1 edgeWidth . . . . .	563

9.196.4.2 flash0 . . . . .	563
9.196.4.3 flashRate . . . . .	563
9.196.4.4 format . . . . .	563
9.196.4.5 percentSize . . . . .	564
9.196.4.6 semiCycles . . . . .	564
9.196.4.7 shape . . . . .	564
9.197QSsimpleShapeManager Class Reference . . . . .	565
9.198qcastatemachine::ReadQCaStateMachine Class Reference . . . . .	566
9.199recording Class Reference . . . . .	567
9.200imageDisplayProperties::rgbPixel Struct Reference . . . . .	568
9.201screenSelectDialog Class Reference . . . . .	569
9.202selectMenu Class Reference . . . . .	570
9.203StateMachineTemplate Class Reference . . . . .	571
9.204qcastatemachine::SubscriptionQCaStateMachine Class Reference . . . . .	572
9.205trace Class Reference . . . . .	573
9.206userInfoStruct Class Reference . . . . .	574
9.207QEPeriodic::userInfoStructArray Struct Reference . . . . .	575
9.208ValueScaling Class Reference . . . . .	576
9.209VideoWidget Class Reference . . . . .	577
9.210qcastatemachine::WriteQCaStateMachine Class Reference . . . . .	579
9.211zoomMenu Class Reference . . . . .	580

# Chapter 1

## QE framework - EPICS aware Qt Widgets and data access classes

- QE is a layered software framework for accessing EPICS data using Channel Access on a range of platforms.
- The QE framework provides object oriented C++ access to control systems using EPICS (Experimental Physics and Industrial Control System). It is based on Qt, a widely used cross-platform application development framework.
- GUI or console based applications can be written that use QE at several levels. QE includes Qt plugin libraries, EPICS aware widgets, data formatting classes, and classes for accessing raw EPICS data in a Qt friendly way.
- QE also includes an application - QEgui - for displaying forms produced by the Qt development tool ‘Designer’. Using this application a complete EPICS GUI system can be generated without writing any code. A GUI system produced in this way can interact with existing EPICS display tools such as EDM.
- QE handles much of the complexities of Channel Access including initiating and managing a channel. Applications using QE can interact with Channel Access using Qt based classes and data types. Channel Access updates are delivered using Qt’s signals and slots mechanism.

### 1.1 Documentation

Support documents can be found in the [documentation](#) section of the epicsqt sourceforge project. The framework download (available on the [epicsqt sourceforge homepage](#)) also includes this documentation as well as full Doxygen generated documentation of all the epicsqt classes and widgets.

## 1.2 License

epicsqt is distributed under the terms of the [GNU Lesser General Public License](#).

## 1.3 Platforms

epicsqt might be usable in all environments where you find [Qt](#). It is compatible with Qt >= 4.4.

## 1.4 Screenshots

- [ASgui screen shots](#)
- [other applications using epicsqt widgets](#)
- [Qt Designer](#)
- [Qt Creator](#)

Screenshots are only available in the HTML docs.

## 1.5 Downloads

Stable releases and development snapshots are available at the epicsqt [project page](#).

For getting a development snapshot from the SVN repository:

```
svn svn co https://epicsqt.svn.sourceforge.net/svnroot/epicsqt epicsqt
```

Alternatively, get a packaged file (epicsqt.tar.gz) from the [epicsqt repository site](#).

## 1.6 Installation

Read [QE\\_GettingStarted.pdf](#) in the documentation for setting up an environment for building or using the epicsqt framework.

To build the framework, open epicsqt.pro in QtCreator, ensure shadow build is turned off, and hit build.

The resultant library libQEPlugin.so will need to be installed or referenced up according to how it is to be used - see QE\_GettingStarted.pdf for details.

Any Qt specific queries? start at [the Qt Project](#)

## 1.7 Support

Visit the sourceforge epicsqt [support page](#) for assistance.

## 1.8 Related Projects

[Qwt](#), The core of a Channel Access aware plotting widget.

## 1.9 Credits:

### Authors:

Andrew Rhyder, Anthony Owen, Glenn Jackson

### Project admin:

Andrew Rhyder <[andrew.rhyder@synchrotron.org.au](mailto:andrew.rhyder@synchrotron.org.au)>



## **Chapter 2**

# **GNU Lesser General Public License**

The EPICS QT Framework is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

The EPICS QT Framework is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with the EPICS QT Framework.

If not, see <http://www.gnu.org/licenses/>

## **Chapter 3**

### **ASgui screen shots**

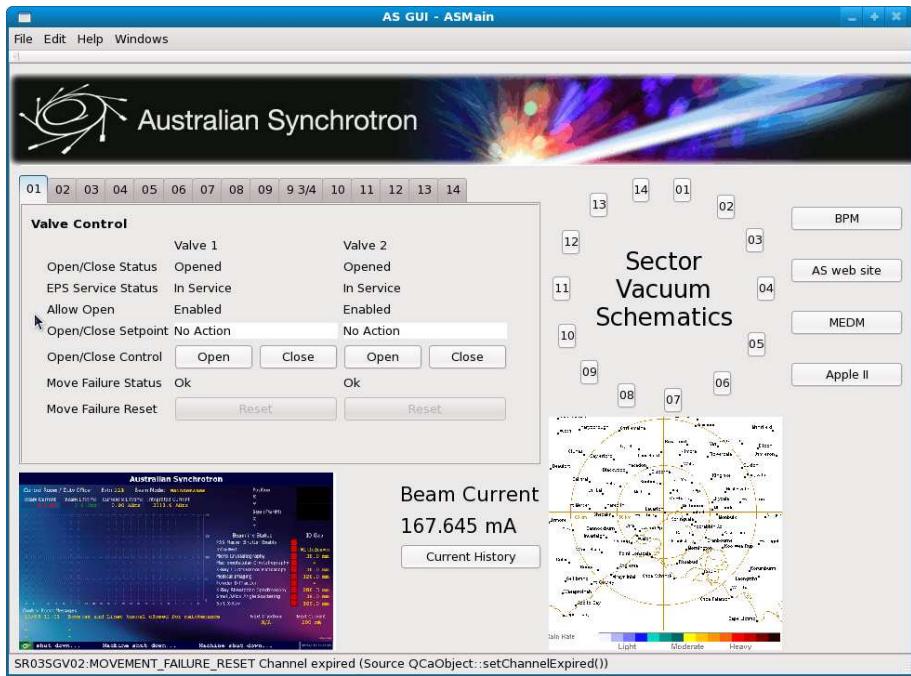


Figure 3.1: Australian Synchrotron mock up

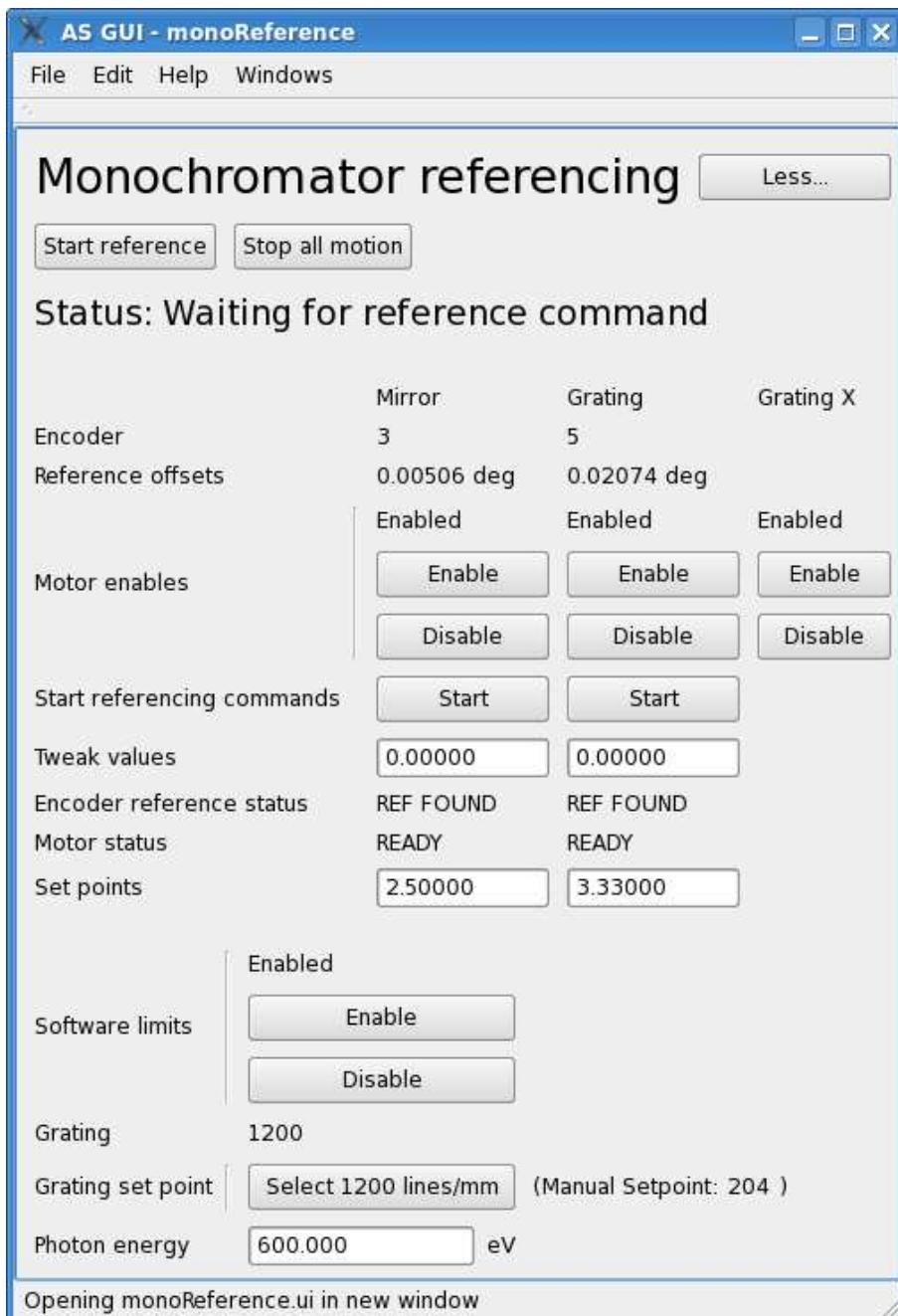


Figure 3.2: Monochromator referencing

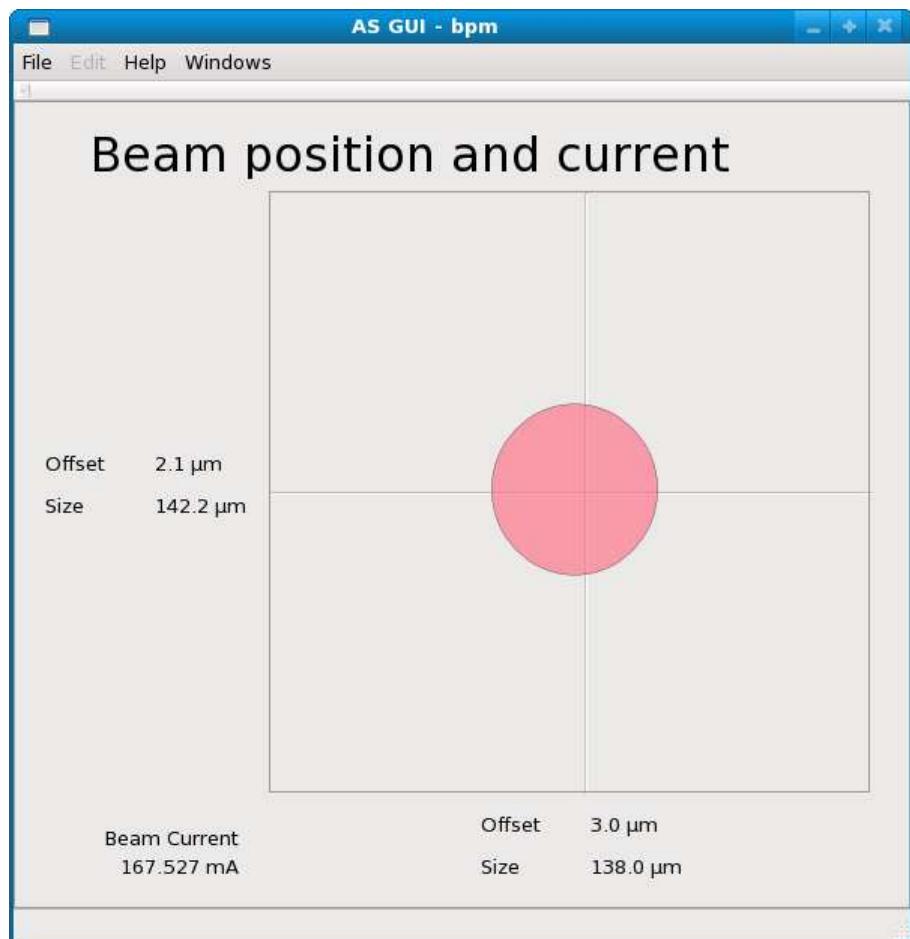


Figure 3.3: Beam position monitor

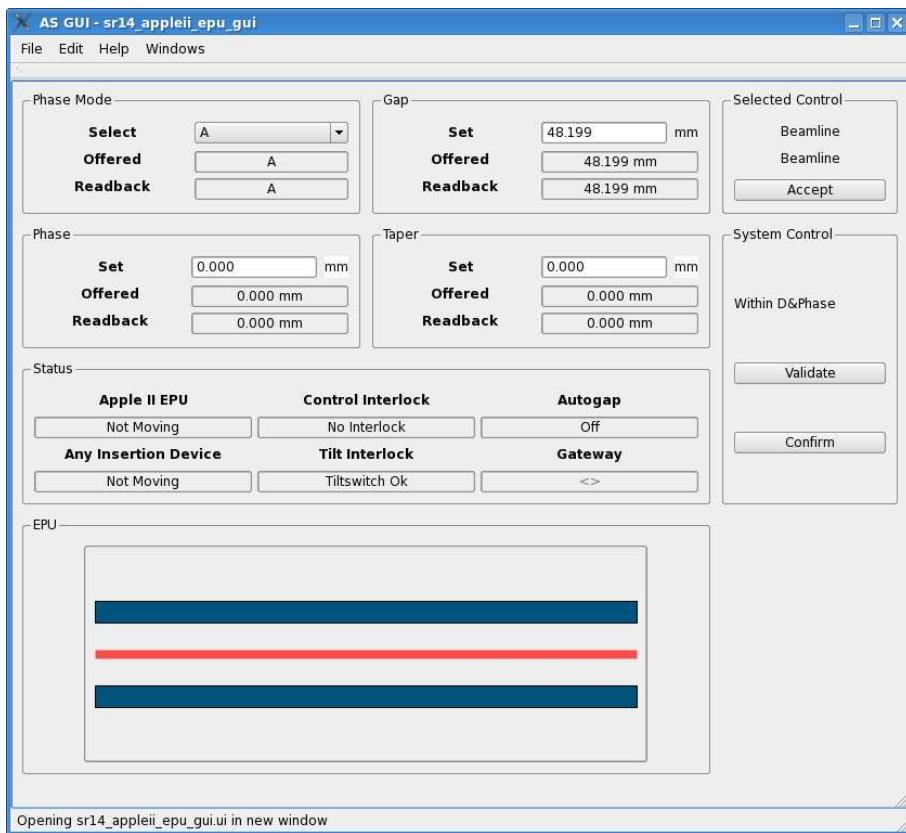


Figure 3.4: Insertion device



Figure 3.5: Injection efficiency monitor

## **Chapter 4**

# **other applications using epicsqt widgets**

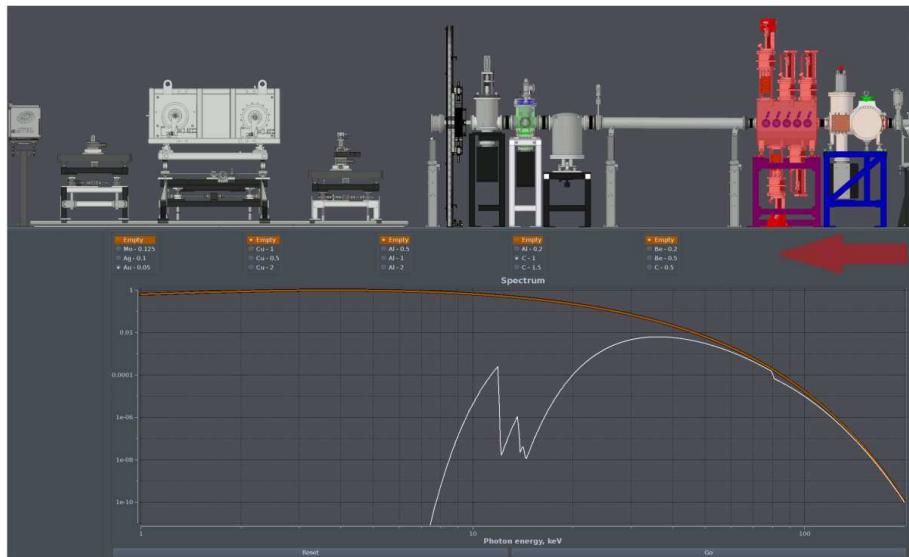


Figure 4.1: Medical Imaging beamline

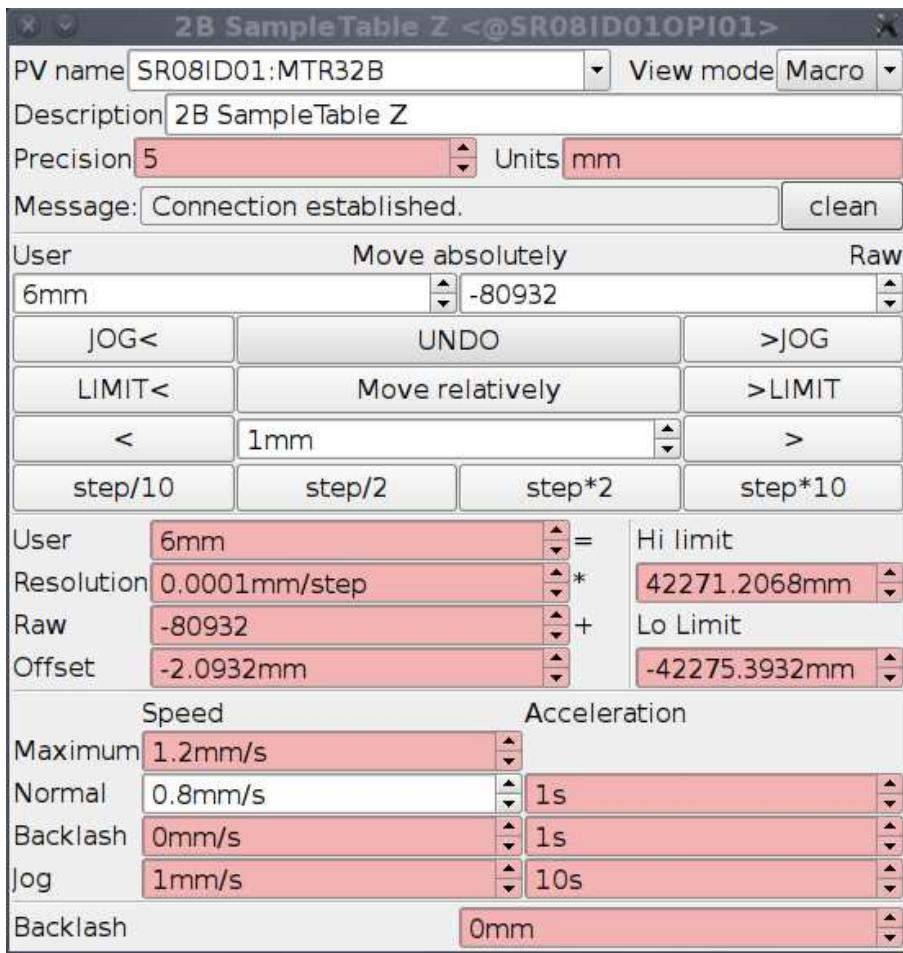


Figure 4.2: Motor controller

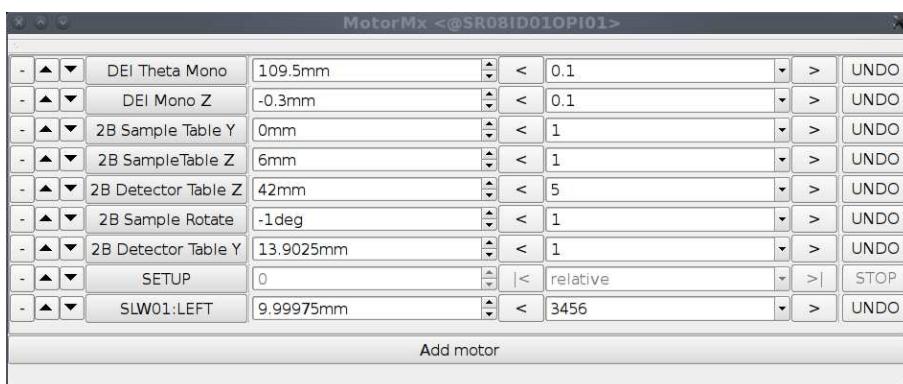


Figure 4.3: Motor controller



# **Chapter 5**

## **Qt Designer**

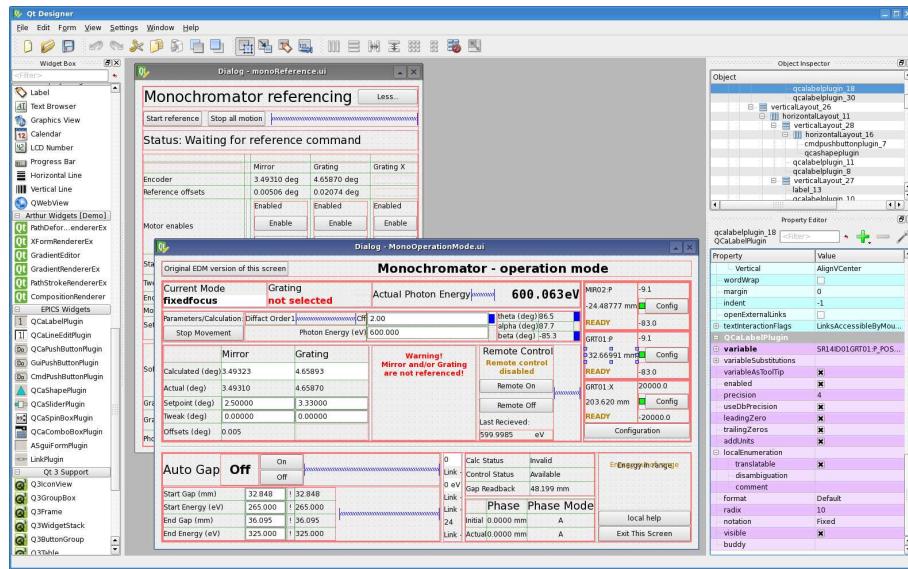


Figure 5.1: Editing multiple GUIs

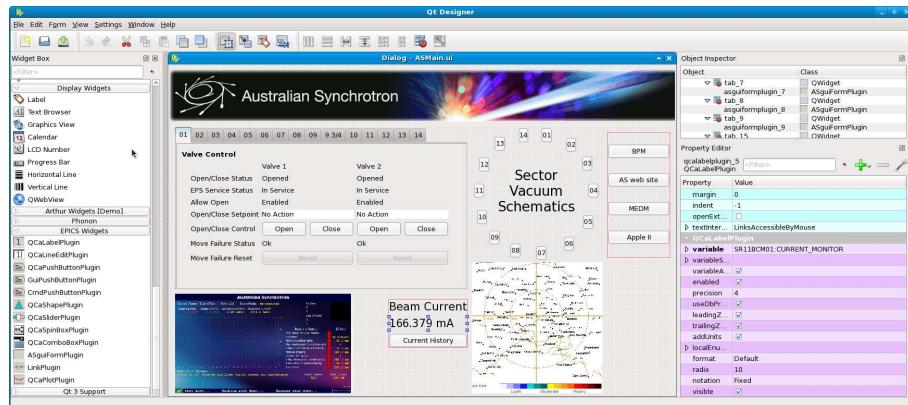
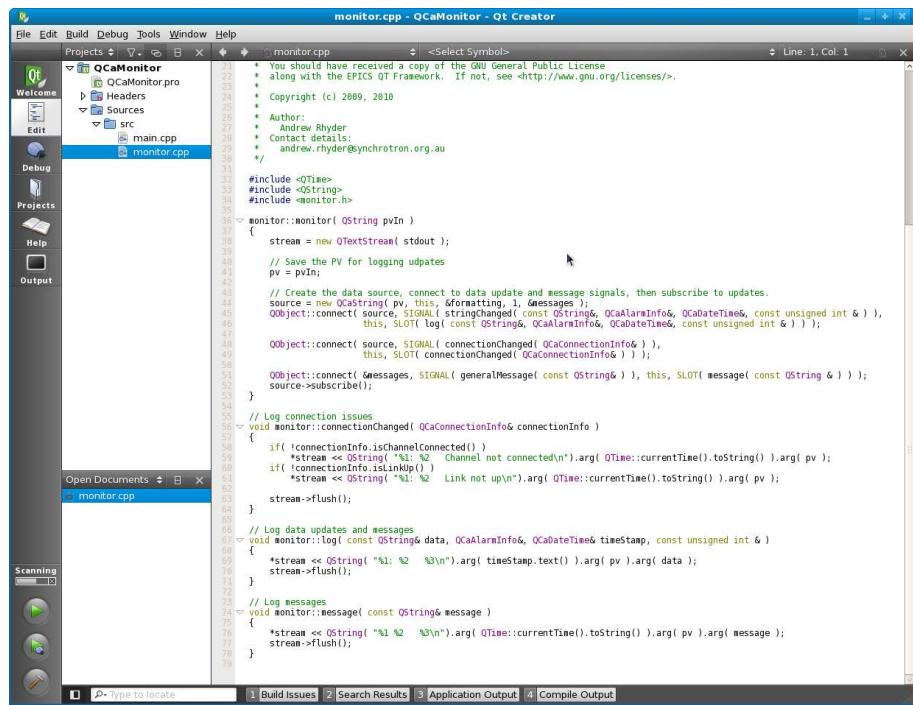


Figure 5.2: Editing a GUI

## **Chapter 6**

### **Qt Creator**



The screenshot shows the Qt Creator IDE interface. The main window displays the code for `monitor.cpp` under the project `QCaMonitor`. The code is a C++ file that includes headers for `QTime`, `QString`, and `QCaMonitor.h`. It defines a class `monitor` with methods for handling PV updates and connection changes. The code uses `QTextStream` to log messages to standard output. The Qt Creator interface includes a sidebar with icons for Edit, Debug, Projects, and Help, and a bottom navigation bar with tabs for Build Issues, Search Results, Application Output, and Compile Output.

Figure 6.1: Application using epicsqt data source classes

# Chapter 7

## Class Index

### 7.1 Class Hierarchy

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

_CopyPaste . . . . .	33
_Field . . . . .	34
_Item . . . . .	35
_QDialogItem . . . . .	36
_QPushButtonGroup . . . . .	37
_QWidgetFileBrowser . . . . .	38
_QWidgetLog . . . . .	39
_QWidgetScript . . . . .	40
areaInfo . . . . .	41
QEAnalogIndicator::Band . . . . .	42
QEAnalogIndicator::BandList . . . . .	43
QEPeriodic::elementInfoStruct . . . . .	45
FFBuffer . . . . .	46
FFThread . . . . .	47
flipRotateMenu . . . . .	48
fullScreenWindow . . . . .	49
histogram . . . . .	50
histogramScroll . . . . .	51
historicImage . . . . .	52
imageContextMenu . . . . .	53
imageDisplayProperties . . . . .	55
imageInfo . . . . .	57
QEImage . . . . .	248
imageMarkup . . . . .	58
VideoWidget . . . . .	577
imageMarkupLegendSetText . . . . .	61
imageProperties . . . . .	66
imageProcessor . . . . .	62

imagePropertiesCore . . . . .	69
imageUpdateIndicator . . . . .	70
loginWidget . . . . .	71
markupDisplayMenu . . . . .	74
markupItem . . . . .	77
markupCrosshair1 . . . . .	72
markupCrosshair2 . . . . .	73
markupEllipse . . . . .	75
markupHLine . . . . .	76
markupLine . . . . .	80
markupRegion . . . . .	81
markupText . . . . .	82
markupVLine . . . . .	83
mpegSource . . . . .	84
QEImage . . . . .	248
mpegSourceObject . . . . .	85
QEStripChartToolBar::OwnTabWidget . . . . .	86
PeriodicDialog . . . . .	87
PeriodicElementSetupForm . . . . .	88
PeriodicSetupDialog . . . . .	89
playbackTimer . . . . .	90
pointInfo . . . . .	91
profilePlot . . . . .	92
PushButtonSpecifications . . . . .	93
QAnalogSlider . . . . .	94
QEAnalogSlider . . . . .	141
QAnalogSliderManager . . . . .	98
QBitStatus . . . . .	99
QEBitStatus . . . . .	150
QCaAlarmInfo . . . . .	101
QCaConnectionInfo . . . . .	102
QCaDataPoint . . . . .	103
QCaDataPointList . . . . .	104
QCaDateTime . . . . .	105
QCaEventFilter . . . . .	106
QCaEventItem . . . . .	107
QCaEventUpdate . . . . .	108
QCaInstalledFiltersListItem . . . . .	109
qcaobject::QCaObject . . . . .	110
QEByteArray . . . . .	156
QEFloating . . . . .	204
QEInteger . . . . .	304
QString . . . . .	521
QCaVariableNamePropertyManager . . . . .	114
QEAbstractWidget . . . . .	117
QEForm . . . . .	207
QEMenuButton . . . . .	341
QENumericEdit . . . . .	351

QERadioGroup . . . . .	451
QEAnalogIndicator . . . . .	122
QEAnalogProgressBar . . . . .	129
QEAnalogSliderManager . . . . .	149
QECheckBoxManager . . . . .	175
QEComboBox . . . . .	176
QEConfiguredLayout . . . . .	183
QEConfiguredLayoutManager . . . . .	189
QEFileBrowser . . . . .	190
QEFileDialog . . . . .	197
QEFileDialogManager . . . . .	203
QEFileDialogView . . . . .	205
QEFileDialogFormatting . . . . .	206
QEFormGridManager . . . . .	217
QEFrame . . . . .	218
QEAbstractDynamicWidget . . . . .	115
QEPlotter . . . . .	378
QEScratchPad . . . . .	469
QEStripChart . . . . .	525
QETable . . . . .	544
QEFormGrid . . . . .	212
QEGeneralEdit . . . . .	225
QE_pvFrame . . . . .	410
QE_pvLoadSave . . . . .	413
QE_pvProperties . . . . .	430
QE_ResizeableFrame . . . . .	460
QE_ScalarHistogram . . . . .	463
QE_WaveformHistogram . . . . .	549
QEGeneralEditManager . . . . .	227
QEGenericButton . . . . .	228
QECheckBox . . . . .	157
QEPushButton . . . . .	392
QERadioButton . . . . .	433
QEGenericEdit . . . . .	231
QELineEdit . . . . .	321
QEGroupBox . . . . .	240
QEHistogram . . . . .	245
QEHistogramManager . . . . .	247
QEImageMarkupThickness . . . . .	302
QEImageOptionsDialog . . . . .	303
QEIntegerArray . . . . .	305
QEIntegerFormatting . . . . .	306
QELabel . . . . .	308
QELineEditManager . . . . .	327
QELink . . . . .	328
QELocalEnumeration . . . . .	330
QELog . . . . .	333
QELogin . . . . .	339

QELoginDialog . . . . .	340
QEMenuItemData . . . . .	343
QEMenuItemItem . . . . .	344
QEMenuItemManager . . . . .	345
QEMenuItemModel . . . . .	346
QEMenuItemSetupDialog . . . . .	348
QEMenuItemTaskMenu . . . . .	349
QEMenuItemTaskMenuFactory . . . . .	350
QENumericEditManager . . . . .	357
QEPeriodic . . . . .	358
QEPeriodicComponentData . . . . .	366
QEPeriodicTaskMenu . . . . .	367
QEPeriodicTaskMenuFactory . . . . .	368
QEpicsPV . . . . .	369
QEPlot . . . . .	370
QEPlotterItemDialog . . . . .	383
QEPlotterManager . . . . .	384
QEPlotterMenu . . . . .	385
QEPlotterNames . . . . .	386
QEPlotterPushButtonSpecifications . . . . .	388
QEPlotterState . . . . .	389
QEPlotterStateList . . . . .	390
QEPlotterToolBar . . . . .	391
QE PvFrameManager . . . . .	412
QE PvLoadSaveCommon . . . . .	416
QE PvLoadSaveCompare . . . . .	417
QE PvLoadSaveGroupNameDialog . . . . .	419
QE PvLoadSaveItem . . . . .	420
QE PvLoadSaveGroup . . . . .	418
QE PvLoadSaveLeaf . . . . .	422
QE PvLoadSaveManager . . . . .	423
QE PvLoadSaveModel . . . . .	424
QE PvLoadSaveTimeDialog . . . . .	426
QE PvLoadSaveUtilities . . . . .	427
QE PvLoadSaveValueEditDialog . . . . .	428
QE PvNameLists . . . . .	429
QE PvPropertiesManager . . . . .	432
QE RadioGroupManager . . . . .	455
QE Recipe . . . . .	456
QE RecordSpec . . . . .	458
QE RecordSpecList . . . . .	459
QE ResizeableFrameManager . . . . .	462
QE ScalarHistogramManager . . . . .	468
QE ScratchPadManager . . . . .	471
QE ScratchPadMenu . . . . .	472
QE Script . . . . .	473
QE Shape . . . . .	482
QE SimpleShapeManager . . . . .	506
QE Slider . . . . .	507

QESpinBox . . . . .	514
QEStringFormatting . . . . .	522
QEStripChartAdjustPVDialog . . . . .	528
QEStripChartContextMenu . . . . .	529
QEStripChartDurationDialog . . . . .	530
QEStripChartItem . . . . .	531
QEStripChartNames . . . . .	533
QEStripChartPushButtonSpecifications . . . . .	535
QEStripChartRangeDialog . . . . .	536
QEStripChartState . . . . .	537
QEStripChartStateList . . . . .	538
QEStripChartStatistics . . . . .	539
QEStripChartTimeDialog . . . . .	540
QEStripChartToolBar . . . . .	541
QESubstitutedLabel . . . . .	543
QETableManager . . . . .	548
QEWaveformHistogramManager . . . . .	551
QNumericEdit . . . . .	552
QNumericEditManager . . . . .	556
QRadioGroup . . . . .	557
QRadioGroupManager . . . . .	559
QSimpleShape . . . . .	560
QESimpleShape . . . . .	498
QSimpleShapeManager . . . . .	565
recording . . . . .	567
imageDisplayProperties::rgbPixel . . . . .	568
screenSelectDialog . . . . .	569
selectMenu . . . . .	570
StateMachineTemplate . . . . .	571
qcastatemachine::QCaStateMachine . . . . .	113
qcastatemachine::ConnectionQCaStateMachine . . . . .	44
qcastatemachine::ReadQCaStateMachine . . . . .	566
qcastatemachine::SubscriptionQCaStateMachine . . . . .	572
qcastatemachine::WriteQCaStateMachine . . . . .	579
trace . . . . .	573
userInfoStruct . . . . .	574
QEPeriodic::userInfoStructArray . . . . .	575
ValueScaling . . . . .	576
zoomMenu . . . . .	580



# Chapter 8

## Class Index

### 8.1 Class List

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

_CopyPaste . . . . .	33
_Field . . . . .	34
_Item . . . . .	35
_QDialogItem . . . . .	36
_QPushButtonGroup . . . . .	37
_QTableWidgetFileBrowser . . . . .	38
_QTableWidgetLog . . . . .	39
_QTableWidgetScript . . . . .	40
areaInfo . . . . .	41
QEAnalogIndicator::Band . . . . .	42
QEAnalogIndicator::BandList . . . . .	43
qcastatemachine::ConnectionQCaStateMachine . . . . .	44
QEPeriodic::elementInfoStruct . . . . .	45
FFBuffer . . . . .	46
FFThread . . . . .	47
flipRotateMenu . . . . .	48
fullScreenWindow . . . . .	49
histogram . . . . .	50
histogramScroll . . . . .	51
historicImage . . . . .	52
imageContextMenu . . . . .	53
imageDisplayProperties . . . . .	55
imageInfo . . . . .	57
imageMarkup . . . . .	58
imageMarkupLegendSetText . . . . .	61
imageProcessor . . . . .	62
imageProperties . . . . .	66
imagePropertiesCore . . . . .	69
imageUpdateIndicator . . . . .	70

loginWidget . . . . .	71
markupCrosshair1 . . . . .	72
markupCrosshair2 . . . . .	73
markupDisplayMenu . . . . .	74
markupEllipse . . . . .	75
markupHLine . . . . .	76
markupItem . . . . .	77
markupLine . . . . .	80
markupRegion . . . . .	81
markupText . . . . .	82
markupVLine . . . . .	83
mpegSource . . . . .	84
mpegSourceObject . . . . .	85
QEStripChartToolBar::OwnTabWidget . . . . .	86
PeriodicDialog . . . . .	87
PeriodicElementSetupForm . . . . .	88
PeriodicSetupDialog . . . . .	89
playbackTimer . . . . .	90
pointInfo . . . . .	91
profilePlot . . . . .	92
PushButtonSpecifications . . . . .	93
QAnalogSlider . . . . .	94
QAnalogSliderManager . . . . .	98
QBitStatus . . . . .	99
QCaAlarmInfo . . . . .	101
QCaConnectionInfo . . . . .	102
QCaDataPoint . . . . .	103
QCaDataPointList . . . . .	104
QCaDateTime . . . . .	105
QCaEventFilter . . . . .	106
QCaEventItem . . . . .	107
QCaEventUpdate . . . . .	108
QCaInstalledFiltersListItem . . . . .	109
qcaobject::QCaObject . . . . .	110
qcastatemachine::QCaStateMachine . . . . .	113
QCaVariableNamePropertyManager . . . . .	114
QEAbstractDynamicWidget . . . . .	115
QEAbstractWidget . . . . .	117
QEAnalogIndicator . . . . .	122
QEAnalogProgressBar . . . . .	129
QEAnalogSlider . . . . .	141
QEAnalogSliderManager . . . . .	149
QEBitStatus . . . . .	150
QEByteArray . . . . .	156
QECheckBox . . . . .	157
QECheckBoxManager . . . . .	175
QEComboBox . . . . .	176
QEConfiguredLayout . . . . .	183
QEConfiguredLayoutManager . . . . .	189
QEFileBrowser . . . . .	190

QEFileDialog . . . . .	197
QEFileDialogManager . . . . .	203
QEFloating . . . . .	204
QEFloatingArray . . . . .	205
QEFloatingFormatting . . . . .	206
QEForm . . . . .	207
QEFormGrid . . . . .	212
QEFormGridManager . . . . .	217
QEFrame . . . . .	218
QEGeneralEdit (The <a href="#">QEGeneralEdit</a> class This class provides a general PV edit widget, presenting one off a <a href="#">QELineEdit</a> , a <a href="#">QENumericEdit</a> or a <a href="#">QERadioGroup</a> for string, numerical and enumeration data kinds respectively) . . . . .	225
QEGeneralEditManager . . . . .	227
QEGenericButton . . . . .	228
QEGenericEdit . . . . .	231
QEGroupBox . . . . .	240
QEHistogram . . . . .	245
QEHistogramManager . . . . .	247
QEImage . . . . .	248
QEImageMarkupThickness . . . . .	302
QEImageOptionsDialog . . . . .	303
QEInteger . . . . .	304
QEIntegerArray . . . . .	305
QEIntegerFormatting . . . . .	306
QELabel . . . . .	308
QELineEdit . . . . .	321
QELineEditManager . . . . .	327
QELink . . . . .	328
QELocalEnumeration . . . . .	330
QELog . . . . .	333
QELogin . . . . .	339
QELoginDialog . . . . .	340
QEMenuItem . . . . .	341
QEMenuItemData . . . . .	343
QEMenuItemItem . . . . .	344
QEMenuItemManager . . . . .	345
QEMenuItemModel . . . . .	346
QEMenuItemSetupDialog . . . . .	348
QEMenuItemTaskMenu . . . . .	349
QEMenuItemTaskMenuFactory . . . . .	350
QENumericEdit . . . . .	351
QENumericEditManager . . . . .	357
QEPeriodic . . . . .	358
QEPeriodicComponentData . . . . .	366
QEPeriodicTaskMenu . . . . .	367
QEPeriodicTaskMenuFactory . . . . .	368
QEpicsPV . . . . .	369
QEPlot . . . . .	370
QEPlotter . . . . .	378

QEPlotterItemDialog . . . . .	383
QEPlotterManager . . . . .	384
QEPlotterMenu . . . . .	385
QEPlotterNames . . . . .	386
QEPlotterPushButtonSpecifications . . . . .	388
QEPlotterState . . . . .	389
QEPlotterStateList . . . . .	390
QEPlotterToolBar . . . . .	391
QEPushButton . . . . .	392
QEPvFrame . . . . .	410
QEPvFrameManager . . . . .	412
QEPvLoadSave . . . . .	413
QEPvLoadSaveCommon . . . . .	416
QEPvLoadSaveCompare . . . . .	417
QEPvLoadSaveGroup . . . . .	418
QEPvLoadSaveGroupNameDialog . . . . .	419
QEPvLoadSaveItem . . . . .	420
QEPvLoadSaveLeaf . . . . .	422
QEPvLoadSaveManager . . . . .	423
QEPvLoadSaveModel . . . . .	424
QEPvLoadSaveTimeDialog . . . . .	426
QEPvLoadSaveUtilities . . . . .	427
QEPvLoadSaveValueEditDialog . . . . .	428
QEPVNameLists . . . . .	429
QEPvProperties . . . . .	430
QEPvPropertiesManager . . . . .	432
QERadioButton . . . . .	433
QERadioGroup . . . . .	451
QERadioGroupManager . . . . .	455
QERecipe . . . . .	456
QERecordSpec . . . . .	458
QERecordSpecList . . . . .	459
QEResizableFrame . . . . .	460
QEResizableFrameManager . . . . .	462
QEScalarHistogram . . . . .	463
QEScalarHistogramManager . . . . .	468
QEScratchPad . . . . .	469
QEScratchPadManager . . . . .	471
QEScratchPadMenu . . . . .	472
QEScript . . . . .	473
QEShape . . . . .	482
QESimpleShape . . . . .	498
QESimpleShapeManager . . . . .	506
QESlider . . . . .	507
QESpinBox . . . . .	514
QESTring . . . . .	521
QESTringFormatting . . . . .	522
QEStripChart . . . . .	525
QEStripChartAdjustPVDIALOG . . . . .	528
QEStripChartContextMenu . . . . .	529

QEStripChartDurationDialog . . . . .	530
QEStripChartItem . . . . .	531
QEStripChartNames . . . . .	533
QEStripChartPushButtonSpecifications . . . . .	535
QEStripChartRangeDialog . . . . .	536
QEStripChartState . . . . .	537
QEStripChartStateList . . . . .	538
QEStripChartStatistics . . . . .	539
QEStripChartTimeDialog . . . . .	540
QEStripChartToolBar (This class holds all the StripChart tool bar widgets ) . . . . .	541
QESubstitutedLabel . . . . .	543
QETable . . . . .	544
QETableManager . . . . .	548
QEWaveformHistogram . . . . .	549
QEWaveformHistogramManager . . . . .	551
QNumericEdit . . . . .	552
QNumericEditManager . . . . .	556
QRadioButton . . . . .	557
QRadioButtonManager . . . . .	559
QSimpleShape . . . . .	560
QSimpleShapeManager . . . . .	565
qcastatemachine::ReadQCaStateMachine . . . . .	566
recording . . . . .	567
imageDisplayProperties::rgbPixel . . . . .	568
screenSelectDialog . . . . .	569
selectMenu . . . . .	570
StateMachineTemplate . . . . .	571
qcastatemachine::SubscriptionQCaStateMachine . . . . .	572
trace . . . . .	573
userInfoStruct . . . . .	574
QEPeriodic::userInfoStructArray . . . . .	575
ValueScaling . . . . .	576
VideoWidget . . . . .	577
qcastatemachine::WriteQCaStateMachine . . . . .	579
zoomMenu . . . . .	580



# Chapter 9

## Class Documentation

### 9.1 \_CopyPaste Class Reference

#### Public Member Functions

- **\_CopyPaste** (bool pEnable, QString pProgram, QString pParameters, QString pWorkingDirectory, int pTimeOut, bool pStop, bool pLog)
- void **setEnable** (bool pEnable)
- bool **getEnable** ()
- void **setProgram** (QString pProgram)
- QString **getProgram** ()
- void **setParameters** (QString pParameters)
- QString **getParameters** ()
- void **setWorkingDirectory** (QString pWorkingDirectory)
- QString **getWorkingDirectory** ()
- void **setTimeOut** (int pTimeOut)
- int **getTimeOut** ()
- void **setStop** (bool pStop)
- bool **getStop** ()
- void **setLog** (bool pLog)
- bool **getLog** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScript/QEScript.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScript/QEScript.cpp

## 9.2 Field Class Reference

### Public Member Functions

- QEWidget \* **getWidget** ()
- void **setWidget** (QString \*pValue)
- QString **getName** ()
- void **setName** (QString pValue)
- QString **getProcessVariable** ()
- void **setProcessVariable** (QString pValue)
- void **setJoin** (bool pValue)
- bool **getJoin** ()
- int **getType** ()
- void **setType** (int pValue)
- QString **getGroup** ()
- void **setGroup** (QString pValue)
- QString **getVisible** ()
- void **setVisible** (QString pValue)
- QString **getEditable** ()
- void **setEditable** (QString pValue)
- bool **getVisibility** ()
- void **setVisibility** (bool pValue)

### Public Attributes

- QEWidget \* **qeWidget**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConf
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConf

## 9.3 \_Item Class Reference

### Public Member Functions

- void **setName** (QString pValue)
- QString **getName** ()
- void **setSubstitution** (QString pValue)
- QString **getSubstitution** ()
- void **setVisible** (QString pValue)
- QString **getVisible** ()

### Public Attributes

- QList< [\\_Field](#) \* > **fieldList**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

## 9.4 \_QDialogItem Class Reference

### Public Member Functions

- **\_QDialogItem** (QWidget \*pParent=0, QString pItemName="", QString pGroupName="", QList< [\\_Field](#) \* > \*pCurrentFieldList=0, Qt::WindowFlags pF=0)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConf
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConf

## 9.5 \_QPushButtonGroup Class Reference

### Public Slots

- void **buttonGroupClicked** ()

### Public Member Functions

- **\_QPushButtonGroup** (QWidget \*pParent=0, QString pItemName="", QString pGroupName="", QList< [\\_Field](#) \* > \*pCurrentFieldList=0)
- void **mouseReleaseEvent** (QMouseEvent \*qMouseEvent)
- void **keyPressEvent** (QKeyEvent \*pKeyEvent)
- void **showDialogGroup** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

## 9.6 \_QTableWidgetFileBrowser Class Reference

### Public Member Functions

- **\_QTableWidgetFileBrowser** (QWidget \*pParent=0)
- void **refreshSize** ()
- void **resizeEvent** (QResizeEvent \*)
- void **resize** (int w, int h)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileBrowser/QEFileBrowse
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileBrowser/QEFileBrowse

## 9.7 \_QTableWidgetLog Class Reference

### Public Member Functions

- **\_QTableWidgetLog** (QWidget \*parent=0)
- void **refreshSize** ()
- void **resizeEvent** (QResizeEvent \*)
- void **resize** (int w, int h)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELog/QELog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELog/QELog.cpp

## 9.8 `_QTableWidgetScript` Class Reference

### Public Member Functions

- `_QTableWidgetScript (QWidget *pParent=0)`
- `void refreshSize ()`
- `void resizeEvent (QResizeEvent *)`
- `void resize (int w, int h)`

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScript/QEScript.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScript/QEScript.cpp`

## 9.9 areaInfo Class Reference

### Public Member Functions

- void **setX1** (long x)
- void **setY1** (long y)
- void **setX2** (long x)
- void **setY2** (long y)
- void **setX** (long x)
- void **setY** (long y)
- void **setW** (long w)
- void **setH** (long h)
- void **setPoint1** (QPoint p1In)
- void **setPoint2** (QPoint p2In)
- void **clearX1** ()
- void **clearY1** ()
- void **clearX2** ()
- void **clearY2** ()
- void **clearX** ()
- void **clearY** ()
- void **clearW** ()
- void **clearH** ()
- bool **getStatus** ()
- QRect **getArea** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImage.h

## 9.10 QEAnalogIndicator::Band Struct Reference

### Public Attributes

- double **lower**
- double **upper**
- QColor **colour**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalog

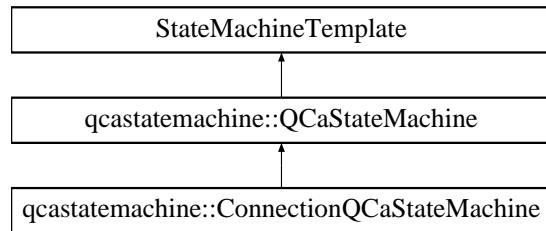
## 9.11 QEAnalogIndicator::BandList Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalogIndicator.h

## 9.12 qcastatemachine::ConnectionQCaStateMachine Class Reference

Inheritance diagram for qcastatemachine::ConnectionQCaStateMachine::



### Public Member Functions

- **ConnectionQCaStateMachine** (void \*parent)
- bool **process** (int requestedState)

### Static Public Attributes

- static int **disconnectedCount** = 0
- static int **connectedCount** = 0

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.cpp

## 9.13 QEPeriodic::elementInfoStruct Struct Reference

### Public Attributes

- unsigned int **number**
- double **atomicWeight**
- QString **name**
- QString **symbol**
- double **meltingPoint**
- double **boilingPoint**
- double **density**
- unsigned int **group**
- double **ionizationEnergy**
- unsigned int **tableRow**
- unsigned int **tableCol**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h

## 9.14 FFBuffer Class Reference

### Public Member Functions

- void **reserve** ()
- void **release** ()
- bool **grabFree** ()

### Public Attributes

- QMutex \* **mutex**
- unsigned char \* **mem**
- AVFrame \* **pFrame**
- PixelFormat **pix\_fmt**
- int **width**
- int **height**
- int **refs**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.cpp

## 9.15 FFThread Class Reference

### Public Slots

- void **stopGracefully** ()

### Signals

- void **updateSignal** ([FFBuffer](#) \*buf)

### Public Member Functions

- **FFThread** (const QString &url, QObject \*parent)
- void **run** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.cpp

## 9.16 flipRotateMenu Class Reference

### Public Member Functions

- **flipRotateMenu** (QWidget \*parent=0)
- imageContextMenu::imageContextMenuOptions **getFlipRotate** (const QPoint &pos)
- void **setChecked** (const int rotation, const bool flipH, const bool flipV)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/flipRotateMenu.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/flipRotateMenu.cpp

## 9.17 fullScreenWindow Class Reference

### Signals

- void **fullScreenResize ()**

### Public Member Functions

- **fullScreenWindow (QWidget \*parent=0)**

### Protected Member Functions

- void **resizeEvent (QResizeEvent \*event)**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/fullScreenWindow.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/fullScreenWindow.cpp

## 9.18 histogram Class Reference

### Public Member Functions

- **histogram** (QWidget \*parent, [imageDisplayProperties](#) \*idp)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/brightnessContrast.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/brightnessContrast.c

## 9.19 histogramScroll Class Reference

### Public Member Functions

- **histogramScroll** (QWidget \*parent, [imageDisplayProperties](#) \*idp)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/brightnessContrast.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/brightnessContrast.cpp

## 9.20 historicImage Class Reference

### Public Member Functions

- **historicImage** (QByteArray **image**, unsigned long **dataSize**, QCaAlarmInfo &**alarmInfo**, QCaDateTime &**time**)

### Public Attributes

- QByteArray **image**
- unsigned long **dataSize**
- QCaAlarmInfo **alarmInfo**
- QCaDateTime **time**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/recording.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImage.cpp

## 9.21 imageContextMenu Class Reference

### Public Types

- enum **imageContextMenuOptions** {  
    **ICM\_NONE** = contextMenu::CM\_SPECIFIC\_WIDGETS\_START\_HERE,  
    **ICM\_SAVE**, **ICM\_PAUSE**, **ICM\_ENABLE\_TIME**,  
    **ICM\_ENABLE\_CURSOR\_PIXEL**, **ICM\_ABOUT\_IMAGE**, **ICM\_ENABLE\_VERT1**, **ICM\_ENABLE\_VERT2**,  
    **ICM\_ENABLE\_VERT3**, **ICM\_ENABLE\_VERT4**, **ICM\_ENABLE\_VERT5**, **ICM\_ENABLE\_HOZ1**,  
    **ICM\_ENABLE\_HOZ2**, **ICM\_ENABLE\_HOZ3**, **ICM\_ENABLE\_HOZ4**,  
    **ICM\_ENABLE\_HOZ5**,  
    **ICM\_ENABLE\_AREA1**, **ICM\_ENABLE\_AREA2**, **ICM\_ENABLE\_AREA3**, **ICM\_ENABLE\_AREA4**,  
    **ICM\_ENABLE\_LINE**, **ICM\_ENABLE\_TARGET**, **ICM\_ENABLE\_BEAM**, **ICM\_DISPLAY\_BUTTON\_BAR**,  
    **ICM\_DISPLAY\_IMAGE\_DISPLAY\_PROPERTIES**, **ICM\_DISPLAY\_RECORDER**, **ICM\_ZOOM\_SELECTED**, **ICM\_ZOOM\_FIT**,  
    **ICM\_ZOOM\_PLUS**, **ICM\_ZOOM\_MINUS**, **ICM\_ZOOM\_10**, **ICM\_ZOOM\_25**,  
    **ICM\_ZOOM\_50**, **ICM\_ZOOM\_75**, **ICM\_ZOOM\_100**, **ICM\_ZOOM\_150**,  
    **ICM\_ZOOM\_200**, **ICM\_ZOOM\_300**, **ICM\_ZOOM\_400**, **ICM\_ROTATE\_NONE**,  
    **ICM\_ROTATE\_RIGHT**, **ICM\_ROTATE\_LEFT**, **ICM\_ROTATE\_180**,  
    **ICM\_FLIP\_HORIZONTAL**,  
    **ICM\_FLIP\_VERTICAL**, **ICM\_SELECT\_PAN**, **ICM\_SELECT\_HSLICE1**,  
    **ICM\_SELECT\_HSLICE2**,  
    **ICM\_SELECT\_HSLICE3**, **ICM\_SELECT\_HSLICE4**, **ICM\_SELECT\_HSLICE5**, **ICM\_SELECT\_VSLICE1**,  
    **ICM\_SELECT\_VSLICE2**, **ICM\_SELECT\_VSLICE3**, **ICM\_SELECT\_VSLICE4**, **ICM\_SELECT\_VSLICE5**,  
    **ICM\_SELECT\_AREA1**, **ICM\_SELECT\_AREA2**, **ICM\_SELECT\_AREA3**, **ICM\_SELECT\_AREA4**,  
    **ICM\_SELECT\_PROFILE**, **ICM\_SELECT\_TARGET**, **ICM\_SELECT\_BEAM**, **ICM\_CLEAR\_MARKUP**,  
    **ICM\_SET\_LEGEND**, **ICM\_THICKNESS\_ONE\_MARKUP**, **ICM\_THICKNESS\_SELECT\_MARKUP**, **ICM\_COPY\_PLOT\_DATA**,  
    **ICM\_FULL\_SCREEN**, **ICM\_DISPLAY\_HSLICE1**, **ICM\_DISPLAY\_HSLICE2**, **ICM\_DISPLAY\_HSLICE3**,  
    **ICM\_DISPLAY\_HSLICE4**, **ICM\_DISPLAY\_HSLICE5**, **ICM\_DISPLAY\_VSLICE1**, **ICM\_DISPLAY\_VSLICE2**,

```
ICM_DISPLAY_VSLICE3, ICM_DISPLAY_VSLICE4, ICM_DISPLAY_-
VSLICE5, ICM_DISPLAY_AREA1,
ICM_DISPLAY_AREA2, ICM_DISPLAY_AREA3, ICM_DISPLAY_-
AREA4, ICM_DISPLAY_PROFILE,
ICM_DISPLAY_TARGET, ICM_DISPLAY_BEAM, ICM_DISPLAY_-
TIMESTAMP, ICM_DISPLAY_ELLIPSE,
ICM_OPTIONS }
```

## Public Member Functions

- **imageContextMenu** (QWidget \*parent=0)
- void **getContextMenuItemOption** (const QPoint &, imageContextMenuOptions \*option, bool \*checked)
- void **addMenuItem** (const QString &title, const bool checkable, const bool checked, const imageContextMenuOptions option)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageContextMenu.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageContextMenu.cpp

## 9.22 imageDisplayProperties Class Reference

### Classes

- struct [rgbPixel](#)

### Signals

- void **brightnessContrastAutoImage** ()
- void **imageDisplayPropertiesChange** ()

### Public Member Functions

- void **setBrightnessContrast** (const unsigned int max, const unsigned int min)
- void **setAutoBrightnessContrast** (bool autoBrightnessContrast)
- void **setContrastReversal** (bool contrastReversal)
- void **setLog** (bool log)
- void **setFalseColour** (bool falseColour)
- bool **getAutoBrightnessContrast** ()
- bool **getContrastReversal** ()
- bool **getLog** ()
- bool **getFalseColour** ()
- int **getLowPixel** ()
- int **getHighPixel** ()
- void **setStatistics** (unsigned int minPIn, unsigned int maxPIn, unsigned int bitDepth, unsigned int binsIn[HISTOGRAM\_BINS], [rgbPixel](#) pixelLookup[256])
- void **showStatistics** ()
- void **setHistZoom** (int value)
- int **getHistZoom** ()
- bool **statisticsValid** ()

### Public Attributes

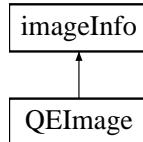
- int **zeroValue**
- int **fullValue**
- bool **defaultFullValue**
- unsigned int **range**
- unsigned int **maxP**
- unsigned int **minP**
- unsigned int **depth**
- unsigned int **bins** [HISTOGRAM\_BINS]
- bool **statisticsSet**
- [rgbPixel](#) \* **pixelLookup**
- QLabel \* **histXLabel**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/brightnessContrast.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/brightnessContrast.c

## 9.23 imageInfo Class Reference

Inheritance diagram for imageInfo::



### Public Member Functions

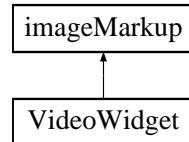
- void **showInfo** (bool show)
- QLayout \* **getInfoWidget** ()
- void **infoShow** (const bool show)
- void **infoUpdateTarget** ()
- void **infoUpdateTarget** (const int x, const int y)
- void **infoUpdateBeam** ()
- void **infoUpdateBeam** (const int x, const int y)
- void **infoUpdateVertProfile** ()
- void **infoUpdateVertProfile** (const int x, const unsigned int thickness)
- void **infoUpdateHozProfile** ()
- void **infoUpdateHozProfile** (const int y, const unsigned int thickness)
- void **infoUpdateProfile** ()
- void **infoUpdateProfile** (const QPoint start, const QPoint end, const unsigned int thickness)
- void **infoUpdateRegion** (const unsigned int region)
- void **infoUpdateRegion** (const unsigned int region, const int x1, const int y1, const int x2, const int y2)
- void **infoUpdatePixel** ()
- void **infoUpdatePixel** (const QPoint pos, int value)
- void **infoUpdateZoom** ()
- void **infoUpdateZoom** (int value, const double XStretch, const double YStretch)
  
- void **infoUpdatePaused** ()
- void **infoUpdatePaused** (bool paused)
- void **setBriefInfoArea** (const bool briefIn)
- bool **getBriefInfoArea** ()
- void **freshImage** (QDateTime &time)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageInfo.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageInfo.cpp

## 9.24 imageMarkup Class Reference

Inheritance diagram for imageMarkup::



### Public Types

- enum **markupIds** {
   
MARKUP\_ID\_REGION1, MARKUP\_ID\_REGION2, MARKUP\_ID\_-  
REGION3, MARKUP\_ID\_REGION4,  
MARKUP\_ID\_H1\_SLICE, MARKUP\_ID\_H2\_SLICE, MARKUP\_ID\_-  
H3\_SLICE, MARKUP\_ID\_H4\_SLICE,  
MARKUP\_ID\_H5\_SLICE, MARKUP\_ID\_V1\_SLICE, MARKUP\_ID\_-  
V2\_SLICE, MARKUP\_ID\_V3\_SLICE,  
MARKUP\_ID\_V4\_SLICE, MARKUP\_ID\_V5\_SLICE, MARKUP\_ID\_-  
LINE, MARKUP\_ID\_TARGET,  
MARKUP\_ID\_BEAM, MARKUP\_ID\_TIMESTAMP, MARKUP\_ID\_-  
ELLIPSE, MARKUP\_ID\_COUNT,  
MARKUP\_ID\_NONE }
- enum **beamAndTargetOptions** { CROSSHAIR1, CROSSHAIR2 }

### Public Member Functions

- void **setShowTime** (bool visibleIn)
- bool **getShowTime** ()
- markupIds **getMode** ()
- void **setMode** (markupIds modeIn)
- void **setMarkupColor** (markupIds mode, QColor markupColorIn)
- QColor **getMarkupColor** (markupIds mode)
- bool **showMarkupMenu** (const QPoint &pos, const QPoint &globalPos)
- void **markupRegionValueChange** (int areaIndex, QRect area, bool display-  
Markups)
- void **markupH1ProfileChange** (int y, bool displayMarkups)
- void **markupH2ProfileChange** (int y, bool displayMarkups)
- void **markupH3ProfileChange** (int y, bool displayMarkups)
- void **markupH4ProfileChange** (int y, bool displayMarkups)
- void **markupH5ProfileChange** (int y, bool displayMarkups)
- void **markupV1ProfileChange** (int x, bool displayMarkups)
- void **markupV2ProfileChange** (int x, bool displayMarkups)

- void **markupV3ProfileChange** (int x, bool displayMarkups)
- void **markupV4ProfileChange** (int x, bool displayMarkups)
- void **markupV5ProfileChange** (int x, bool displayMarkups)
- void **markupLineProfileChange** (QPoint start, QPoint end, bool displayMarkups)
- void **markupTargetValueChange** (QPoint point, bool displayMarkups)
- void **markupBeamValueChange** (QPoint point, bool displayMarkups)
- void **markupEllipseValueChange** (QPoint point1, QPoint point2, bool displayMarkups)
- void **markupValueChange** (int markup, bool displayMarkups, QPoint p1, QPoint p2=QPoint())
- QCursor **getCircleCursor** ()
- QCursor **getTargetCursor** ()
- QCursor **getVLineCursor** ()
- QCursor **getHLineCursor** ()
- QCursor **getLineCursor** ()
- QCursor **getRegionCursor** ()
- virtual void **markupSetCursor** (QCursor cursor)=0
- void **setMarkupLegend** (markupIds mode, QString legend)
- QString **getMarkupLegend** (markupIds mode)
- void **clearMarkup** (markupIds markupId)
- void **showMarkup** (markupIds markupId)
- void **displayMarkup** (markupIds markupId, bool state)
- bool **isMarkupVisible** (markupIds mode)
- double **getZoomScale** ()
- QSize **getImageSize** ()
- void **setImageSize** (const QSize &imageSizeIn)
- beamAndTargetOptions **getTargetOption** ()
- void **setTargetOption** (beamAndTargetOptions option)
- beamAndTargetOptions **getBeamOption** ()
- void **setBeamOption** (beamAndTargetOptions option)
- void **setBeamOrTargetOption** (markupIds item, beamAndTargetOptions option)

## Public Attributes

- QVector< [markupItem](#) \* > **items**
- QPoint **grabOffset**
- bool **markupAreasStale**
- QFont **legendFont**
- QFontMetrics \* **legendFontMetrics**

## Protected Member Functions

- void **drawMarkups** (QPainter &p, const QRect &rect)
- bool **anyVisibleMarkups** ()
- QCursor **getDefaultMarkupCursor** ()
- void **setMarkupTime** (QCaDateTime &time)
- bool **markupMousePressEvent** (QMouseEvent \*event, bool panning)
- bool **markupMouseReleaseEvent** (QMouseEvent \*event, bool panning)
- bool **markupMouseMoveEvent** (QMouseEvent \*event, bool panning)
- void **markupResize** (const double scale)
- virtual void **markupChange** (QVector< QRect > &changedAreas)=0
- virtual void **markupAction** (markupIds mode, bool complete, bool clearing, QPoint point1, QPoint point2, unsigned int thickness)=0

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

## 9.25 imageMarkupLegendSetText Class Reference

### Public Member Functions

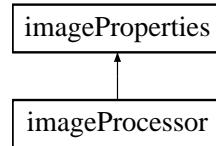
- **imageMarkupLegendSetText** (QString existingLegend, QWidget \*parent=0)
- QString **getLegend** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageMarkupLegendSetText.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageMarkupLegendSetText.cpp

## 9.26 imageProcessor Class Reference

#include <imageProcessor.h> Inheritance diagram for imageProcessor::



### Signals

- void **imageBuilt** (QImage image, QString error)  
*An image has been generated from image data and is now ready for presentation.*

### Public Member Functions

- **imageProcessor ()**  
*Constructor.*
- **~imageProcessor ()**  
*Destructor.*
- void **setImage** (const QByteArray &imageIn, unsigned long dataSize)  
*Save the image data for analysis processing and display.*
- void **buildImage** ()  
*Generate a new image.*
- bool **setWidth** (unsigned long uValue)  
*Set the image width.*
- bool **setHeight** (unsigned long uValue)  
*Set the image height.*
- bool **setNumDimensions** (unsigned long uValue)  
*Set the number of dimensions.*
- bool **setDimension0** (unsigned long uValue)  
*Set the first dimension (width if two dimensions, bytes per element if three dimensions).*
- bool **setDimension1** (unsigned long uValue)  
*Set the second dimension (height if two dimensions, width if three dimensions).*

- bool `setDimension2` (unsigned long uValue)  
*Set the third dimension (unused if two dimensions, height if three dimensions).*
- void `setClippingOn` (bool clippingOnIn)  
*Set clipping flag. If true, `setClippingLow()` and `setClippingHigh()` are used to set clipping values.*
- void `setClippingLow` (unsigned int value)  
*Set pixel value below which low clip colour is displayed.*
- void `setClippingHigh` (unsigned int value)  
*Set pixel value above which high clip colour is displayed.*
- int `getScanOption` ()  
*Determine the way the input pixel data must be scanned to accommodate the required rotate and flip options.*
- void `getPixelTranslation` ()  
*Generate a lookup table to convert raw pixel values to display pixel values.*
- unsigned int `maxPixelValue` ()  
*Determine the maximum pixel value for the current format.*
- unsigned int `rotatedImageBuffWidth` ()  
*Return the image width following any rotation.*
- unsigned int `rotatedImageBuffHeight` ()  
*Return the image height following any rotation.*
- `imageDisplayProperties::rgbPixel getFalseColor` (const unsigned char value)  
*Get a false color representation for an entry from the color lookup table.*
- int `getElementCount` ()  
*Determine the element count expected based on the available dimensions.*
- bool `validateDimensions` ()  
*Determine if the image dimensional information is valid.*
- void `getPixelRange` (const QRect &area, unsigned int \*min, unsigned int \*max)  
*Determine the range of pixel values an area of the image.*
- bool `hasImage` ()  
*Return true if the current image is empty.*
- const unsigned char \* `getImageDataPtr` (QPoint &pos)

*Return a pointer to pixel data in the original image data.*

- int [getPixelValueFromData](#) (const unsigned char \*ptr)  
*Return a number representing a pixel intensity given a pointer into an image data buffer.*
- double [getFloatingPixelValueFromData](#) (const unsigned char \*ptr)  
*Return a floating point number representing a pixel intensity given a pointer into an image data buffer.*
- QImage [copyImage](#) ()  
*Return a QImage based on the current image.*
- void [generateVSliceData](#) (QVector< QPointF > &vSliceData, int x, unsigned int thickness)  
*Generate a series of pixel values from a vertical slice through the current image.*
- void [generateHSliceData](#) (QVector< QPointF > &hSliceData, int y, unsigned int thickness)  
*Generate a series of pixel values from a horizontal slice through the current image.*
- void [generateProfileData](#) (QVector< QPointF > &profileData, QPoint point1, QPoint point2, unsigned int thickness)  
*Generate a series of pseudo pixel values from an arbitrary line between two pixels.*
- QRect [rotateFlipToDataRectangle](#) (const QRect &rect)  
*Transform a rectangle from the image to the original data according to current rotation and flip options.*
- QRect [rotateFlipToDataRectangle](#) (const QPoint &pos1, const QPoint &pos2)  
*Transform a rectangle from the image to the original data according to current rotation and flip options.*
- QPoint [rotateFlipToDataPoint](#) (const QPoint &pos)  
*Transform a point from the image to the original data according to current rotation and flip options.*
- QRect [rotateFlipToImageRectangle](#) (const QRect &rect)  
*Transform a rectangle from the original data to the image according to current rotation and flip options.*
- QRect [rotateFlipToImageRectangle](#) (const QPoint &pos1, const QPoint &pos2)  
*Transform a rectangle from the original data to the image according to current rotation and flip options.*
- QPoint [rotateFlipToImagePoint](#) (const QPoint &pos)  
*Transform a point from the original data to the image according to current rotation and flip options.*

- void **run ()**

## Public Attributes

- QWaitCondition **imageSync**
- QMutex **imageWait**
- QMutex **imageLock**
- bool **finishNow**
- [imagePropertiesCore \\* next](#)

### 9.26.1 Detailed Description

This class generates images for presentation from raw image data and formatting information such as brightness, contrast, flip, rotate, canvas size, etc. The work is performed in a dedicated thread .

### 9.26.2 Member Function Documentation

#### 9.26.2.1 int imageProcessor::getPixelValueFromData (const unsigned char \* *ptr*)

Return a number representing a pixel intensity given a pointer into an image data buffer.

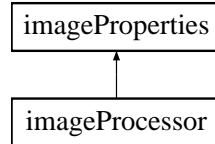
!! not done - copy of RGB1  
!! not done - copy of RGB1

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageProcessor.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageProcessor.cpp

## 9.27 imageProperties Class Reference

#include <imageProperties.h> Inheritance diagram for imageProperties::



### Public Types

- enum **rotationOptions** { ROTATION\_0, ROTATION\_90\_RIGHT, ROTATION\_-90\_LEFT, ROTATION\_180 }

### Public Member Functions

- **imageProperties ()**  
*Constructor.*
- void **setRotation** (**rotationOptions** rotationIn)
- **rotationOptions** **getRotation** ()
- void **setFlipVert** (bool flipVertIn)
- bool **getFlipVert** ()
- void **setFlipHoz** (bool flipHozIn)
- bool **getFlipHoz** ()
- void **setImageBuffWidth** (unsigned long imageBuffWidthIn)
- void **setImageBuffHeight** (unsigned long imageBuffHeightIn)
- unsigned long **getImageBuffWidth** ()
- unsigned long **getImageBuffHeight** ()
- **imageDataFormats::formatOptions** **getFormat** ()
- void **setFormat** (**imageDataFormats::formatOptions** formatIn)
- bool **setFormat** (const QString &text)
- void **setBitDepth** (unsigned int bitDepthIn)
- unsigned int **getBitDepth** ()
- void **setElementsPerPixel** (unsigned long elementsPerPixelIn)
- void **setImageDisplayProperties** (**imageDisplayProperties** \*imageDisplayPropsIn)
- void **setWidthHeightFromDimensions** ()  
*// Update the image dimensions (width and height) from the area detector dimension variables.*
- void **invalidatePixelLookup** ()  
*recalculate (when next required) pixel summary information*

- `QString getInfoText ()`

*Generate textual information regarding the current image.*

## Protected Attributes

- `imageDisplayProperties * imageDisplayProps`
- `imageDataFormats::formatOptions formatOption`
- `unsigned int bitDepth`
- `unsigned long imageDataSize`
- `unsigned long elementsPerPixel`
- `unsigned long bytesPerPixel`
- `QByteArray imageData`
- `unsigned long receivedImageSize`
- `QString previousMessageText`
- `QImage image`
- `unsigned long imageBuffWidth`
- `unsigned long imageBuffHeight`
- `unsigned long numDimensions`
- `unsigned long imageDimension0`
- `unsigned long imageDimension1`
- `unsigned long imageDimension2`
- `bool pixelLookupValid`
- `imageDisplayProperties::rgbPixel pixelLookup [256]`
- `int pixelLow`
- `int pixelHigh`
- `bool clippingOn`
- `unsigned int clippingLow`
- `unsigned int clippingHigh`
- `rotationOptions rotation`
- `bool flipVert`
- `bool flipHoz`

### 9.27.1 Detailed Description

This class manages the image attributes required for generating a QImage from a QByteArray holding CA image data. It is used as the base class for the [imageProcessor](#) class. Note, while this class holds and manages all the information needed to process an image, a snapshot of all the information required for processing an image in a seperate thread is made by the [imagePropertiesCore](#) class.

## 9.27.2 Member Enumeration Documentation

### 9.27.2.1 enum imageProperties::rotationOptions

Image rotation options

**Enumerator:**

*ROTATION\_0* No image rotation.

*ROTATION\_90\_RIGHT* Rotate image 90 degrees clockwise.

*ROTATION\_90\_LEFT* Rotate image 90 degrees anticlockwise.

*ROTATION\_180* Rotate image 180 degrees.

## 9.27.3 Constructor & Destructor Documentation

### 9.27.3.1 imageProperties::imageProperties ()

Constructor. Construction. Set all image attributes to sensible defaults.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageProperties.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageProperties.cpp

## 9.28 imagePropertiesCore Class Reference

### Public Member Functions

- **imagePropertiesCore** (QByteArray imageDataIn, unsigned long imageBufWidthIn, unsigned long imageBuffHeightIn, int scanOptionIn, unsigned long bytesPerPixelIn, int pixelLowIn, int pixelHighIn, unsigned int bitDepthIn, [imageDisplayProperties::rgbPixel](#) \*pixelLookupIn, imageDataFormats::formatOptions formatOptionIn, unsigned long imageDataSizeIn, [imageDisplayProperties](#) \*imageDisplayPropsIn, unsigned int rotatedImageBufWidthIn, unsigned int rotatedImageBuffHeightIn)
- QImage [buildImageCore](#) ()

#### 9.28.1 Member Function Documentation

##### 9.28.1.1 QImage imagePropertiesCore::buildImageCore ()

```
!! not done yet - just do the same as RGB1 for the time being and hope
!! not done yet - just do the same as RGB1 for the time being and hope
!! not done yet. do the same as for YUV422
!! not done yet. do the same as for YUV422
```

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageProperties.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageProcessor.cpp

## 9.29 imageUpdateIndicator Class Reference

### Public Member Functions

- void **freshImage ()**
- void **paintEvent (QPaintEvent \*)**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageInfo.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/imageInfo.cpp

## 9.30 loginWidget Class Reference

### Public Member Functions

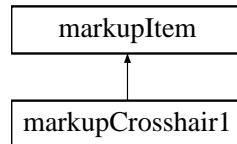
- **loginWidget** ([QELogin](#) \*ownerIn)
- userLevelTypes::userLevels **getUserType** ()
- [QString](#) **getPassword** ()
- void **clearPassword** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELogin/QELogin.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELogin/QELogin.cpp

## 9.31 markupCrosshair1 Class Reference

Inheritance diagram for markupCrosshair1::



### Public Member Functions

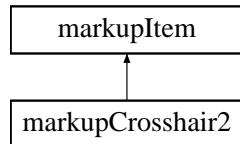
- **markupCrosshair1** ([imageMarkup](#) \*ownerIn, const bool interactiveIn, const bool reportOnMoveIn, const QString legendIn)
- void **startDrawing** (const QPoint pos)
- void **setArea** ()
- void **drawMarkup** ( QPainter &p )
- void **moveTo** (const QPoint pos)
- bool **isOver** (const QPoint point, QCursor \*cursor)
- QPoint **origin** ()
- QCursor **cursorForHandle** (const markupItem::markupHandles handle)
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()
- void **nonInteractiveUpdate** (QPoint p1, QPoint p2)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupTarget.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupTarget.cpp

## 9.32 markupCrosshair2 Class Reference

Inheritance diagram for markupCrosshair2::



### Public Member Functions

- **markupCrosshair2** (`imageMarkup *ownerIn`, `const bool interactiveIn`, `const bool reportOnMoveIn`, `const QString legendIn`)
- **void startDrawing** (`const QPoint pos`)
- **void setArea** ()
- **void drawMarkup** (`QPainter &p`)
- **void moveTo** (`const QPoint pos`)
- **bool isOver** (`const QPoint point`, `QCursor *cursor`)
- **QPoint origin** ()
- **QCursor cursorForHandle** (`const markupItem::markupHandles handle`)
- **QPoint getPoint1** ()
- **QPoint getPoint2** ()
- **QCursor defaultCursor** ()
- **void nonInteractiveUpdate** (`QPoint p1`, `QPoint p2`)

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupBeam.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupBeam.cpp`

## 9.33 markupDisplayMenu Class Reference

### Public Member Functions

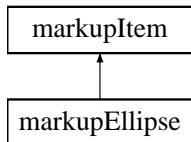
- **markupDisplayMenu** (QWidget \*parent=0)
- void **setDisplayed** (imageContextMenu::imageContextMenuOptions option, bool state)
- void **setItemText** (imageContextMenu::imageContextMenuOptions option, QString title)
- bool **isDisplayed** (imageContextMenu::imageContextMenuOptions option)
- void **enable** (imageContextMenu::imageContextMenuOptions option, bool state)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupDisplayMen
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupDisplayMen

## 9.34 markupEllipse Class Reference

Inheritance diagram for markupEllipse::



### Public Member Functions

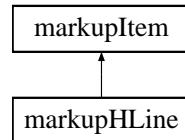
- **markupEllipse** ([imageMarkup](#) \*ownerIn, const bool interactiveIn, const bool reportOnMoveIn, const [QString](#) legendIn)
- void **startDrawing** (const [QPoint](#) pos)
- void **setArea** ()
- void **drawMarkup** ( [QPainter](#) &p)
- void **moveTo** (const [QPoint](#) pos)
- bool **isOver** (const [QPoint](#) point, [QCursor](#) \*cursor)
- [QPoint](#) **origin** ()
- [QCursor](#) **cursorForHandle** (const [markupItem::markupHandles](#) handle)
- [QPoint](#) **getPoint1** ()
- [QPoint](#) **getPoint2** ()
- [QCursor](#) **defaultCursor** ()
- void **nonInteractiveUpdate** ([QPoint](#) p1, [QPoint](#) p2)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupEllipse.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupEllipse.cpp

## 9.35 markupHLine Class Reference

Inheritance diagram for markupHLine::



### Public Member Functions

- **markupHLine** ([imageMarkup](#) \*ownerIn, const bool interactiveIn, const bool reportOnMoveIn, const QString legendIn)
- void **startDrawing** (const QPoint pos)
- void **setArea** ()
- void **drawMarkup** ( QPainter &p )
- void **moveTo** (const QPoint pos)
- bool **isOver** (const QPoint point, QCursor \*cursor)
- QPoint **origin** ()
- QCursors **cursorForHandle** (const markupItem::markupHandles handle)
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursors **defaultCursor** ()
- void **nonInteractiveUpdate** (QPoint p1, QPoint p2)

#### 9.35.1 Member Function Documentation

##### 9.35.1.1 void markupHLine::drawMarkup ( QPainter & p ) [virtual]

!! draw the handle in the middle of the existing view, not the entire image

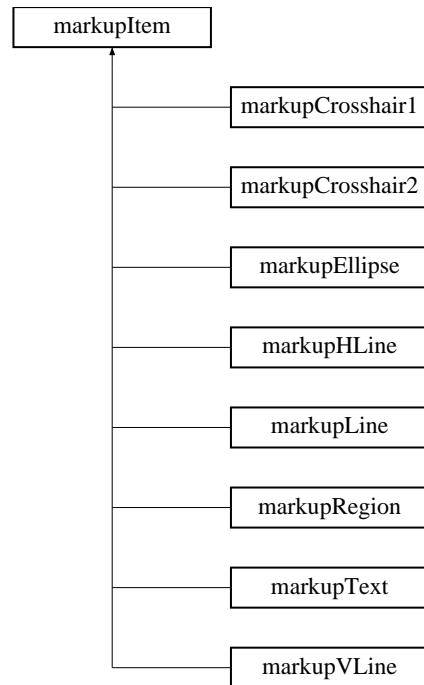
Implements [markupItem](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupHLine.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupHLine.cpp

## 9.36 markupItem Class Reference

Inheritance diagram for markupItem::



### Public Types

- enum **markupHandles** {  
    **MARKUP\_HANDLE\_NONE**, **MARKUP\_HANDLE\_START**, **MARKUP\_HANDLE\_END**, **MARKUP\_HANDLE\_CENTER**,  
    **MARKUP\_HANDLE\_TL**,   **MARKUP\_HANDLE\_TR**,   **MARKUP\_HANDLE\_BL**, **MARKUP\_HANDLE\_BR**,  
    **MARKUP\_HANDLE\_T**, **MARKUP\_HANDLE\_B**, **MARKUP\_HANDLE\_L**, **MARKUP\_HANDLE\_R** }

### Public Member Functions

- void **drawMarkupItem** (QPainter &p)
- QSize **getImageSize** ()
- virtual QPoint **origin** ()=0
- virtual void **moveTo** (const QPoint pos)=0
- virtual void **startDrawing** (const QPoint pos)=0
- virtual bool **isOver** (const QPoint point, QCursors \*cursor)=0

- virtual QCursor **cursorForHandle** (const markupItem::markupHandles handle)=0
- virtual QPoint **getPoint1** ()=0
- virtual QPoint **getPoint2** ()=0
- virtual QCursor **defaultCursor** ()=0
- virtual void **nonInteractiveUpdate** (QPoint, QPoint)
- void **setThickness** (const unsigned int thicknessIn)
- unsigned int **getThickness** ()
- void **setLegend** (const QString legendIn)
- const QString **getLegend** ()
- void **setColor** (QColor colorIn)
- QColor **getColor** ()

## Public Attributes

- QRect **area**
- QRect **scalableArea**
- bool **visible**
- bool **interactive**
- bool **reportOnMove**
- QColor **color**

## Protected Types

- enum **isOverOptions** { **OVER\_LINE**, **OVER\_BORDER**, **OVER\_AREA** }
- enum **legendJustification** { **ABOVE\_RIGHT**, **BELOW\_LEFT**, **BELOW\_RIGHT** }

## Protected Member Functions

- **markupItem** ([imageMarkup](#) \*ownerIn, const isOverOptions over, const bool interactiveIn, const bool reportOnMoveIn, const QString legendIn)
- virtual void **setArea** ()=0
- virtual void **drawMarkup** (QPainter &p)=0
- bool **pointIsNear** (QPoint p1, QPoint p)
- const QSize **getLegendSize** ()
- void **addLegendArea** ()
- const QPoint **getLegendTextOrigin** (QPoint posScaled)
- void **setLegendOffset** (QPoint offset, legendJustification just)
- const QPoint **getLegendOffset** ()
- void **drawLegend** (QPainter &p, QPoint posScaled)
- QPoint **limitPointToImage** (const QPoint pos)
- double **getZoomScale** ()

## Protected Attributes

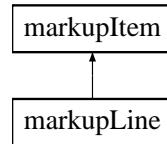
- markupHandles **activeHandle**
- **imageMarkup** \* **owner**
- unsigned int **thickness**
- unsigned int **maxThickness**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupItem.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupItem.cpp

## 9.37 markupLine Class Reference

Inheritance diagram for markupLine::



### Public Member Functions

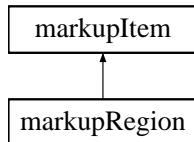
- **markupLine** (*imageMarkup* \*ownerIn, const bool interactiveIn, const bool reportOnMoveIn, const QString legendIn)
- void **startDrawing** (const QPoint pos)
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (const QPoint pos)
- bool **isOver** (const QPoint point, QCursor \*cursor)
- QPoint **origin** ()
- QCursor **cursorForHandle** (const markupItem::markupHandles handle)
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()
- void **nonInteractiveUpdate** (QPoint p1, QPoint p2)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupLine.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupLine.cpp

## 9.38 markupRegion Class Reference

Inheritance diagram for markupRegion::



### Public Member Functions

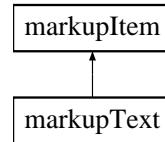
- **markupRegion** ([imageMarkup](#) \*ownerIn, const bool interactiveIn, const bool reportOnMoveIn, const QString legendIn)
- void **startDrawing** (const QPoint pos)
- void **setArea** ()
- void **drawMarkup** ( QPainter &p )
- void **moveTo** (const QPoint pos)
- bool **isOver** (const QPoint point, QCursor \*cursor)
- QPoint **origin** ()
- QCursor **cursorForHandle** (const markupItem::markupHandles handle)
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()
- void **nonInteractiveUpdate** (QPoint p1, QPoint p2)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupRegion.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupRegion.cpp

## 9.39 markupText Class Reference

Inheritance diagram for markupText::



### Public Member Functions

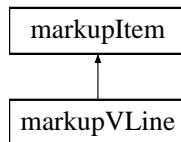
- **markupText** (`imageMarkup *ownerIn`, `const bool interactiveIn`, `const bool reportOnMoveIn`, `const QString legendIn`)
- `void setText (QString textIn)`
- `void startDrawing (const QPoint pos)`
- `void setArea ()`
- `void drawMarkup ( QPainter &p )`
- `void moveTo (const QPoint pos)`
- `bool isOver (const QPoint point, QCursor *cursor)`
- `QPoint origin ()`
- `QCursor cursorForHandle (const markupItem::markupHandles handle)`
- `QPoint getPoint1 ()`
- `QPoint getPoint2 ()`
- `QCursor defaultCursor ()`

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupText.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupText.cpp`

## 9.40 markupVLine Class Reference

Inheritance diagram for markupVLine::



### Public Member Functions

- **markupVLine** (`imageMarkup *ownerIn, const bool interactiveIn, const bool reportOnMoveIn, const QString legendIn)`
- **void startDrawing** (`const QPoint pos`)
- **void setArea** ()
- **void drawMarkup** (`QPainter &p`)
- **void moveTo** (`const QPoint pos`)
- **bool isOver** (`const QPoint point, QCursors *cursor`)
- **QPoint origin** ()
- **QCursors cursorForHandle** (`const markupItem::markupHandles handle`)
- **QPoint getPoint1** ()
- **QPoint getPoint2** ()
- **QCursors defaultCursor** ()
- **void nonInteractiveUpdate** (`QPoint p1, QPoint p2`)

#### 9.40.1 Member Function Documentation

##### 9.40.1.1 void markupVLine::drawMarkup (QPainter & p) [virtual]

!! draw the handle in the middle of the existing view, not the entire image

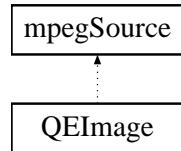
Implements [markupItem](#).

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupVLine.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/markupVLine.cpp`

## 9.41 mpegSource Class Reference

Inheritance diagram for mpegSource::



### Public Member Functions

- void [updateImage \(FFBuffer \\*buf\)](#)
- void [setURL \(QString\)](#)
- void [startStream \(\)](#)
- void [stopStream \(\)](#)

### Protected Member Functions

- QString [getURL \(\)](#)
- void [setURL \(QString urlIn\)](#)
- void [stopStream \(\)](#)
- void [startStream \(\)](#)

#### 9.41.1 Member Function Documentation

##### 9.41.1.1 void mpegSource::updateImage (FFBuffer \* *buf*)

!!?? \* 3 for color only

!! Since the [QEImage](#) widget handles (or should handle) CA image data in all the formats that are expected in this mpeg stream !! perhaps this formatting here should be simply packaging the data in a QByteArray and delivering it, rather than perform any conversion.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImage.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.cpp

## 9.42 mpegSourceObject Class Reference

### Public Slots

- void **updateImage** ([FFBuffer](#) \*buf)

### Signals

- void **aboutToQuit** ()

### Public Member Functions

- **mpegSourceObject** ([mpegSource](#) \*msIn)
- void **sentAboutToQuit** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/mpeg.cpp

## 9.43 QEStripChartToolBar::OwnTabWidget Class Reference

### Public Member Functions

- **OwnTabWidget** ([QEStripChartToolBar](#) \*parent)

### Public Attributes

- QPushButton \* **pushButtons** [NUMBER\_OF\_BUTTONS]
- QLabel \* **yScaleStatus**
- QLabel \* **timeStatus**
- QLabel \* **durationStatus**
- QLabel \* **numberOfOutstandingRequests**
- QLabel \* **timeModeStatus**
- QComboBox \* **predefinedComboBox**
- QPushButton \* **loadButton**
- QPushButton \* **saveAsButton**
- QLabel \* **timeRefLabel**
- QLabel \* **time1**
- QLabel \* **time2**
- QLabel \* **timeDeltaLabel**
- QLabel \* **timeDelta**
- QLabel \* **valueRefLabel**
- QLabel \* **value1**
- QLabel \* **value2**
- QLabel \* **valueDelta1**
- QLabel \* **value3**
- QLabel \* **value4**
- QLabel \* **valueDelta2**
- QLabel \* **placeHolder2**
- QLabel \* **placeHolder3**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartTc

## 9.44 PeriodicDialog Class Reference

### Public Member Functions

- **PeriodicDialog** (QWidget \*parent=0)
- QString **getElement** ()
- void **setElement** (QString elementIn, QList< bool > &enabledList, QList< QString > &elementList)

### Protected Member Functions

- void **changeEvent** (QEvent \*e)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/PeriodicDialog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/PeriodicDialog.cpp

## 9.45 PeriodicElementSetupForm Class Reference

### Public Member Functions

- **PeriodicElementSetupForm** (QWidget \*parent=0)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/PeriodicElementSetupForm.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/PeriodicElementSetupForm.cpp

## 9.46 PeriodicSetupDialog Class Reference

### Public Member Functions

- **PeriodicSetupDialog** (QWidget \*parent=0)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/PeriodicSetupDialog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/PeriodicSetupDialog.cpp

## 9.47 playbackTimer Class Reference

### Public Member Functions

- **playbackTimer** ([recording](#) \*recorderIn)
- void **timerEvent** (QTimerEvent \*event)

### Public Attributes

- [recording](#) \* recorder

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/recording.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/recording.cpp

## 9.48 pointInfo Class Reference

### Public Member Functions

- void **setX** (long x)
- void **setY** (long y)
- void **setPoint** (QPoint pIn)
- void **clearX** ()
- void **clearY** ()
- bool **getStatus** ()
- QPoint **getPoint** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImage.h

## 9.49 profilePlot Class Reference

### Public Types

- enum **plotDirections** { PROFILEPLOT\_LR, PROFILEPLOT\_RL, PROFILEPLOT\_TB, PROFILEPLOT\_BT }

### Public Member Functions

- **profilePlot** (plotDirections plotDirectionIn)
- void **setProfile** (QVector< QPointF > \*profile, double minX, double maxX, double minY, double maxY, QString title, QPoint start, QPoint end, unsigned int thicknessIn)
- void **clearProfile** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/profilePlot.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/profilePlot.cpp

## 9.50 PushButtonSpecifications Struct Reference

### Public Attributes

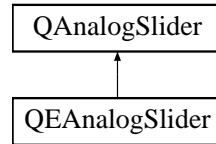
- int **side**
- int **gap**
- int **width**
- bool **isIcon**
- const QString **captionOrIcon**
- const QString **toolTip**
- const char \* **member**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSave.cpp

## 9.51 QAnalogSlider Class Reference

#include <QAnalogSlider.h> Inheritance diagram for QAnalogSlider::



### Public Slots

- void **setValue** (const double value)

### Signals

- void **valueChanged** (const double value)
- void **appliedValue** (const double value)

### Public Member Functions

- **QAnalogSlider** (QWidget \*parent=0)
- virtual ~**QAnalogSlider** ()

*Destruction.*

- QSize **sizeHint** () const
- void **setColourBandList** (const QEColourBandList &bandList)
- QEColourBandList **getColourBandList** () const
- double **getValue** () const
- void **setDesignPrecision** (const int precision)

*Access function for precision - refer to precision property for details.*

- int **getDesignPrecision** () const

*Access function for precision - refer to precision property for details.*

- void **setDesignMinimum** (const double minimum)

*Access function for minimum - refer to minimum property for details.*

- double **getDesignMinimum** () const

*Access function for minimum - refer to minimum property for details.*

- void **setDesignMaximum** (const double maximum)

*Access function for maximum - refer to maximum property for details.*

- double `getDesignMaximum () const`  
*Access function for `maximum` - refer to `maximum` property for details.*
- void `setMinorInterval (const double minorInterval)`  
*Access function for `minorInterval` - refer to `minorInterval` property for details.*
- double `getMinorInterval () const`  
*Access function for `minorInterval` - refer to `minorInterval` property for details.*
- void `setMajorInterval (const double majorInterval)`  
*Access function for `majorInterval` - refer to `majorInterval` property for details.*
- double `getMajorInterval () const`  
*Access function for `majorInterval` - refer to `majorInterval` property for details.*
- void `setLeftText (const QString &leftText)`  
*Access function for `leftText` - refer to `leftText` property for details.*
- QString `getLeftText () const`  
*Access function for `leftText` - refer to `leftText` property for details.*
- void `setCentreText (const QString &centreText)`  
*Access function for `centreText` - refer to `centreText` property for details.*
- QString `getCentreText () const`  
*Access function for `centreText` - refer to `centreText` property for details.*
- void `setRightText (const QString &rightText)`  
*Access function for `rightText` - refer to `rightText` property for details.*
- QString `getRightText () const`  
*Access function for `rightText` - refer to `rightText` property for details.*
- void `setShowSaveRevert (const bool show)`
- bool `getShowSaveRevert () const`
- void `setShowApply (const bool show)`
- bool `getShowApply () const`

## Protected Slots

- virtual void `applyButtonClicked (bool)`

## Protected Member Functions

- void **setIsActive** (const bool value)
- bool **getIsActive** ()
- QEAxisPainter \* **getAxisPainter** ()
- virtual int **getPrecision** () const
- virtual double **getMinimum** () const
- virtual double **getMaximum** () const
- void **updateAxisAndSlider** ()

## Properties

- int **precision**
- double **minimum**

*Specifies the minimum allowed value.*

- double **maximum**

*Specifies the maximum allowed value.*

- double **minorInterval**
- double **majorInterval**
- bool **tracking**
- QString **leftText**

*Controls the left, centre and right texts.*

- QString **centreText**
- QString **rightText**
- bool **showSaveRevert**

*Enables/disables the save-revert capability.*

- bool **showApply**

*Enables/disables the apply value capability.*

- double **value**

*Specifies the value.*

### 9.51.1 Detailed Description

[QAnalogSlider](#) is a non EPICS aware slider that provides an analog equivalent of the QSlider. It is deemed analog as it can be set by/emits floating point (double) values as opposed to integer values. It is also decorated with a scale and text showing the current value; it also provides a save-restore capability.

### 9.51.2 Constructor & Destructor Documentation

**9.51.2.1 `QAnalogSlider::QAnalogSlider (QWidget *parent = 0) [explicit]`**

Create with default title.

### 9.51.3 Property Documentation

**9.51.3.1 `double QAnalogSlider::majorInterval [read, write]`**

Minor scale interval. Only applies for linear scale (not log scale)

**9.51.3.2 `double QAnalogSlider::minorInterval [read, write]`**

Minor scale interval. Only applies for linear scale (not log scale)

**9.51.3.3 `int QAnalogSlider::precision [read, write]`**

Precision used for the display and editing of numbers. The default is 4. Strictly speaking, this should be an unsigned int, but designer int properties editor much 'nicer'.

**9.51.3.4 `bool QAnalogSlider::tracking [read, write]`**

Controls when valueChanged signal is emitted. If tracking is enabled (the default), the slider emits the valueChanged () signal while the slider is being dragged. If tracking is disabled, the slider emits the valueChanged () signal only when the user releases the slider.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogSlider/QAnalogSlider.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogSlider/QAnalogSlider.cpp

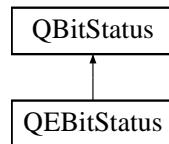
## 9.52 QAnalogSliderManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogSlider/QAnalogSliderManager.h

## 9.53 QBitStatus Class Reference

Inheritance diagram for QBitStatus::



### Public Types

- enum **Orientations** { **LSB\_On\_Right**, **LSB\_On\_Bottom**, **LSB\_On\_Left**, **LSB\_On\_Top** }
- enum **Shapes** { **Rectangle**, **Circle** }

### Public Slots

- void **setValue** (const int value)

### Public Member Functions

- **QBitStatus** (QWidget \*parent=0)
- virtual QSize **sizeHint** () const
- void **setBorderColour** (const QColor value)
- QColor **getBorderColour** ()
- void **setOnColour** (const QColor value)
- QColor **getOnColour** ()
- void **setOffColour** (const QColor value)
- QColor **getOffColour** ()
- void **setInvalidColour** (const QColor value)
- QColor **getInvalidColour** ()
- void **setClearColour** (const QColor value)
- QColor **getClearColour** ()
- void **setDrawBorder** (const bool value)
- bool **getDrawBorder** ()
- void **setNumberOfBits** (const int value)
- int **getNumberOfBits** ()
- void **setGap** (const int value)
- int **getGap** ()
- void **setShift** (const int value)
- int **getShift** ()
- void **setOnClearMask** (const QString value)
- QString **getOnClearMask** ()
- void **setOffClearMask** (const QString value)

- `QString getOffClearMask ()`
- `void setReversePolarityMask (const QString value)`
- `QString getReversePolarityMask ()`
- `void setIsValid (const bool value)`
- `bool getIsValid ()`
- `void setOrientation (const enum Orientations value)`
- `enum Orientations getOrientation ()`
- `void setShape (const enum Shapes value)`
- `enum Shapes getShape ()`
- `int getValue ()`

## Protected Member Functions

- `void setIsActive (const bool value)`
- `bool getIsActive ()`

## Properties

- `int value`
- `int number_of_bits`
- `int shift`
- `Orientations orientation`
- `Shapes shape`
- `int gap`
- `QString reversePolarityMask`
- `QString onClearMask`
- `QString offClearMask`
- `QColor boarderColour`
- `QColor invalidColour`
- `QColor onColour`
- `QColor offColour`
- `QColor clearColour`
- `bool drawBorder`
- `bool isValid`
- `bool isActive`

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEBitStatus/QBitStatus.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEBitStatus/QBitStatus.cpp`

## 9.54 QCaAlarmInfo Class Reference

### Public Member Functions

- **QCaAlarmInfo** (unsigned short statusIn, unsigned short severityIn)
- bool **operator==** (const [QCaAlarmInfo](#) &other) const
- bool **operator!=** (const [QCaAlarmInfo](#) &other) const
- QString **statusName** () const
- QString **severityName** () const
- bool **isInAlarm** () const
- bool **isMinor** () const
- bool **isMajor** () const
- bool **isValid** () const
- QString **style** () const
- QString **getStyleColorName** () const
- QString **getColorName** () const
- QCAALARMINFO\_SEVERITY **getSeverity** () const

### Static Public Member Functions

- static QCAALARMINFO\_SEVERITY **getInvalidSeverity** ()
- static void **setStyleColorNames** (const QStringList &styleColorNames)
- static QStringList **getStyleColorNames** ()
- static void **setColorNames** (const QStringList &colorNames)
- static QStringList **getColorNames** ()
- static QStringList **getDefaultStyleColorNames** ()
- static QStringList **getDefaultColorNames** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaAlarmInfo.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaAlarmInfo.cpp

## 9.55 QCaConnectionInfo Class Reference

### Public Member Functions

- **QCaConnectionInfo** (unsigned short channelStateIn, unsigned short linkStateIn, QString recordName)
- bool **isChannelConnected** ()
- bool **isLinkUp** ()
- QString **variable** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaConnectionInfo.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaConnectionInfo.cpp

## 9.56 QCaDataPoint Class Reference

### Public Member Functions

- bool **isDisplayable () const**
- QString **toString () const**
- QString **toString (const QCaDateTime &originDateTime) const**

### Public Attributes

- double **value**
- QCaDateTime **datetime**
- QCaAlarmInfo **alarm**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaDataPoint.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaDataPoint.cpp

## 9.57 QCaDataPointList Class Reference

### Public Member Functions

- void **reserve** (const int size)
- void **clear** ()
- void **removeLast** ()
- void **removeFirst** ()
- void **append** (const [QCaDataPointList](#) &other)
- void **append** (const [QCaDataPoint](#) &other)
- void **replace** (const int i, const [QCaDataPoint](#) &t)
- int **count** () const
- [QCaDataPoint](#) **value** (const int j) const
- [QCaDataPoint](#) **last** () const
- void **truncate** (const int position)
- int **indexBeforeTime** (const [QCaDateTime](#) &searchTime, const int defaultIndex) const
- const [QCaDataPoint](#) \* **findNearestPoint** (const [QCaDateTime](#) &searchTime) const
- void **resample** (const [QCaDataPointList](#) &source, const double interval, const [QCaDateTime](#) &endTime)
- void **compact** (const [QCaDataPointList](#) &source)
- void **toStream** (QTextStream &target, bool withIndex, bool withRelativeTime) const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaDataPoint.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaDataPoint.cpp

## 9.58 QCaDateTime Class Reference

### Public Member Functions

- **QCaDateTime** (QDateTime dt)
- **QCaDateTime & operator=** (const **QCaDateTime** &other)
- **QCaDateTime** (unsigned long seconds, unsigned long nanoseconds)
- **QString text** ()
- **QCaDateTime addSeconds** (const double seconds) const
- double **secondsTo** (const QDateTime &target) const
- double **floating** (const QDateTime &base) const
- unsigned long **getSeconds** () const

*Recover original EPICS time constructor parameters.*

- unsigned long **getNanoSeconds** () const

### 9.58.1 Member Function Documentation

#### 9.58.1.1 QCaDateTime QCaDateTime::addSeconds (const double *seconds*) const

Equivalent of addSecs and secsTo in base class, save that we specify the seconds as a floating point number.

#### 9.58.1.2 double QCaDateTime::floating (const QDateTime & *base*) const

Duration in seconds from base time to this time. Note: this is the opposite sense to the parent QDateTime daysTo, secsTo and msecstos functions. Phase out

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaDateTime.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaDateTime.cpp

## 9.59 QCaEventFilter Class Reference

### Public Member Functions

- void **addFilter** (QObject \*objectIn)
- void **deleteFilter** (QObject \*objectIn)
- bool **eventFilter** (QObject \*watched, QEvent \*e)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaEventFilter.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaEventFilter.cpp

## 9.60 QCaEventItem Class Reference

### Public Member Functions

- **QCaEventItem** ([QCaEventUpdate](#) \*newEvent)

### Public Attributes

- [QCaEventUpdate](#) \* **event**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaEventUpdate.h

## 9.61 QCaEventUpdate Class Reference

### Public Member Functions

- **QCaEventUpdate** ([qcaobject::QCaObject](#) \*emitterObjectIn, long newReason, void \*newDataPtr)

### Public Attributes

- bool **acceptThisEvent**
- [qcaobject::QCaObject](#) \* **emitterObject**
- long **reason**
- void \* **dataPtr**

### Static Public Attributes

- static QEvent::Type **EVENT\_UPDATE\_TYPE** = QEvent::User

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaEventUpdate.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaEventUpdate.cpp

## 9.62 QCaInstalledFiltersListItem Class Reference

### Public Member Functions

- **QCaInstalledFiltersListItem** (QObject \*eventObjectIn)

### Public Attributes

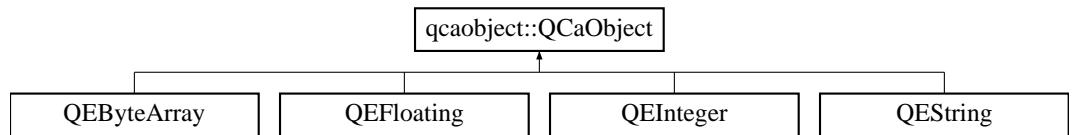
- QObject \* **eventObject**
- long **referenceCount**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaEventFilter.h

## 9.63 qcaobject::QCaObject Class Reference

Inheritance diagram for qcaobject::QCaObject::



### Public Types

- enum **priorities** { **QE\_PRIORITY\_LOW**, **QE\_PRIORITY\_NORMAL**, **QE\_PRIORITY\_HIGH** }

### Public Slots

- bool **writeData** (const QVariant &value)
- bool **writeDataElement** (const QVariant &elementValue)
- void **resendLastData** ()

### Signals

- void **dataChanged** (const QVariant &value, QCaAlarmInfo &alarmInfo, QCaDateTime &timeStamp, const unsigned int &variableIndex)
- void **dataChanged** (const QByteArray &value, unsigned long dataSize, QCaAlarmInfo &alarmInfo, QCaDateTime &timeStamp, const unsigned int &variableIndex)
- void **connectionChanged** (QCaConnectionInfo &connectionInfo, const unsigned int &variableIndex)
- void **connectionChanged** (QCaConnectionInfo &connectionInfo)

### Public Member Functions

- **QCaObject** (const QString &recordName, QObject \*eventObject, const unsigned int variableIndex, unsigned char signalsToSendIn=SIG\_VARIANT, priorities priorityIn=QE\_PRIORITY\_NORMAL)
- **QCaObject** (const QString &recordName, QObject \*eventObject, const unsigned int variableIndex, UserMessage \*userMessageIn, unsigned char signalsToSendIn=SIG\_VARIANT, priorities priorityIn=QE\_PRIORITY\_NORMAL)
- bool **subscribe** ()
- bool **singleShotRead** ()
- bool **dataTypeKnown** ()
- unsigned int **getVariableIndex** () const

- bool **createChannel** ()
- void **deleteChannel** ()
- bool **createSubscription** ()
- bool **getChannel** ()
- bool **putChannel** ()
- bool **isChannelConnected** ()
- void **startConnectionTimer** ()
- void **stopConnectionTimer** ()
- void **setUserMessage** (UserMessage \*userMessageIn)
- void **enableWriteCallbacks** (bool enable)
- bool **isWriteCallbacksEnabled** ()
- void **setRequestedElementCount** (unsigned int elementCount)
- QString **getRecordName** ()
- QString **getEgu** ()
- QStringList **getEnumerations** ()
- unsigned int **getPrecision** ()
- QCaAlarmInfo **getAlarmInfo** ()
- QCaDateTime **getDateTime** ()
- double **getDisplayLimitUpper** ()
- double **getDisplayLimitLower** ()
- double **getAlarmLimitUpper** ()
- double **getAlarmLimitLower** ()
- double **getWarningLimitUpper** ()
- double **getWarningLimitLower** ()
- double **getControlLimitUpper** ()
- double **getControlLimitLower** ()
- generic::generic\_types **getDataType** ()
- QString **getHostName** ()
- QString **getFieldType** ()
- unsigned long **getElementCount** ()
- bool **getReadAccess** ()
- bool **getWriteAccess** ()
- void **setArrayIndex** (const int index)
- int **getArrayIndex** () const
- void **getLastData** (bool &isDefined, QVariant &value, QCaAlarmInfo &alarmInfo, QCaDateTime &timeStamp)
- bool **getChannelIsConnected** () const
- bool **getIsLinkUp** () const
- bool **getDataIsAvailable** () const
- QString **getStringValue** () const
- long **getIntegerValue** () const
- double **getFloatingValue** () const
- QVector< long > **getIntegerArray** () const
- QVector< double > **getFloatingArray** () const

## Static Public Member Functions

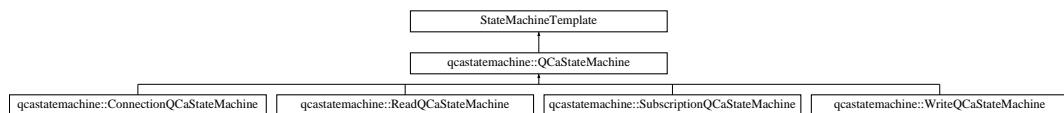
- static void **deletingEventStatic** ([QCaEventUpdate](#) \*dataUpdateEvent)
- static void **processEventStatic** ([QCaEventUpdate](#) \*dataUpdateEvent)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaObject.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaObject.cpp

## 9.64 qcastatemachine::QCaStateMachine Class Reference

Inheritance diagram for qcastatemachine::QCaStateMachine:::



### Public Member Functions

- **QCaStateMachine** (void \*parent)
- virtual bool **process** (int requestedState)=0

### Public Attributes

- QMutex **lock**
- bool **pending**
- bool **active**
- bool **expired**
- void \* **myWorker**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.cpp

## 9.65 QCaVariableNamePropertyManager Class Reference

### Signals

- void **newVariableNameProperty** (QString variable, QString Substitutions, unsigned int variableIndex)

### Public Member Functions

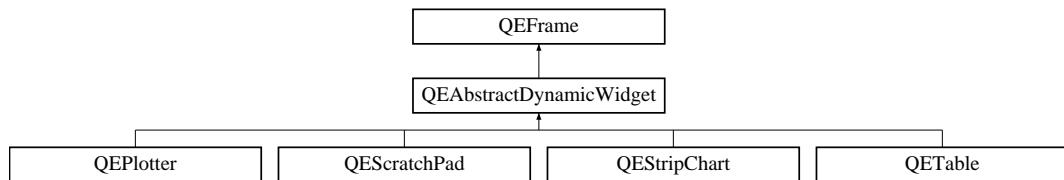
- QString **getVariableNameProperty** () const
- void **setVariableNameProperty** (QString variableNamePropertyIn)
- QString **getSubstitutionsProperty** () const
- void **setSubstitutionsProperty** (QString substitutionsPropertyIn)
- void **setVariableIndex** (unsigned int variableIndexIn)
- unsigned int **getVariableIndex** () const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaVariableNamePropertyManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaVariableNamePropertyManager.cpp

## 9.66 QEAbstractDynamicWidget Class Reference

#include <QEAbstractDynamicWidget.h>  
Inheritance diagram for QEAbstractDynamicWidget::



### Public Types

- enum **OwnContextMenuOptions** { **ADWCM\_NONE** = CM\_SPECIFIC\_WIDGETS\_START\_HERE, **ADWCM\_LOAD\_WIDGET\_CONFIG**, **ADWCM\_SAVE\_WIDGET\_CONFIG**, **ADWCM\_SUB\_CLASS\_WIDGETS\_START\_HERE** }

### Public Member Functions

- **QEAbstractDynamicWidget** (QWidget \*parent=0)
- void **setDefaultDir** (const QString &defaultDir)
- QString **getDefaultDir** () const
- void **paste** (QVariant s)
- void **addPvNameList** (const QStringList &pvNameList)
- void **addPvNameSet** (const QString &pvNameSet)
- virtual int **addPvName** (const QString &pvName)=0

### Protected Slots

- void **loadNamedWidetConfiguration** (const QString &filename)
- void **saveNamedWidetConfiguration** (const QString &filename)
- void **loadWidgetConfiguration** ()
- void **saveWidgetConfiguration** ()

### Protected Member Functions

- QMenu \* **buildContextMenu** ()
- void **contextMenuTriggered** (int selectedItemNum)
- QString **getPersistantRootName** () const
- QString **getPersistantName** () const

## Properties

- `QString defaultDir`

### 9.66.1 Detailed Description

Provides a common abstract base class for dynamic widgets, i.e. dynamic in the sense that the user can add/removed and modify PVs used by the widget at run time. It has been specifically designed to be a common base class for the `QEStripChart`, `QEScratchPad`, `QEPlotter` and `QETable` widgets. This not only minimises maintainance, but helps ensure we maintain a common look and feel user experiance.

### 9.66.2 Property Documentation

#### 9.66.2.1 `QString QEAbstractDynamicWidget::defaultDir [read, write]`

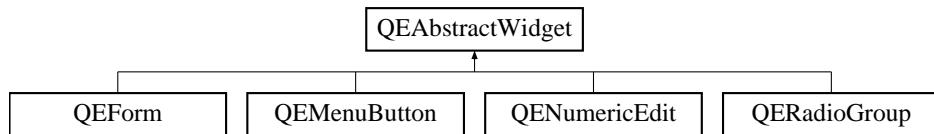
Default directory used for loading/saving files. Default to null string which is interpreted as the current directory.

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAbstractWidget/QEAbstractDynamicWidget.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAbstractWidget/QEAbstractDynamicWidget.cpp`

## 9.67 QEAbstractWidget Class Reference

Inheritance diagram for QEAbstractWidget::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setManagedVisible` (bool v)

### Public Member Functions

- `UserLevels getUserLevelVisibilityProperty ()`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void `setUserLevelVisibilityProperty (UserLevels level)`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- `UserLevels getUserLevelEnabledProperty ()`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void `setUserLevelEnabledProperty (UserLevels level)`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- void [setDisplayAlarmStateOptionProperty](#) (DisplayAlarmStateOptions option)  
*Access function for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property for details.*
- [QEAbstractWidget](#) (QWidget \*parent=0)

## Properties

- bool [variableAsToolTip](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [styleSheet](#)
- QString [defaultStyle](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- UserLevels [userLevelVisibility](#)
- UserLevels [userLevelEnabled](#)
- bool [displayAlarmState](#)
- DisplayAlarmStateOptions [displayAlarmStateOption](#)

### 9.67.1 Member Enumeration Documentation

#### 9.67.1.1 enum QEAbstractWidget::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

- Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.
- Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.
- WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.67.1.2 enum QEAbstractWidget::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and [userLevel](#) enumeration for details.

##### Enumerator:

- User* Refer to USERLEVEL\_USER for details.
- Scientist* Refer to USERLEVEL\_SCIENTIST for details.
- Engineer* Refer to USERLEVEL\_ENGINEER for details.

## 9.67.2 Member Function Documentation

### 9.67.2.1 void QEAbstractWidget::setManagedVisible (bool v) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.67.3 Property Documentation

### 9.67.3.1 bool QEAbstractWidget::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

Reimplemented in [QEForm](#), and [QEMenuButton](#).

### 9.67.3.2 QString QEAbstractWidget::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

### 9.67.3.3 bool QEAbstractWidget::displayAlarmState [read, write]

DEPRECATED. USE `displayAlarmStateOption` INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

### 9.67.3.4 DisplayAlarmStateOptions QEAbstractWidget::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

Reimplemented in [QEForm](#), and [QEMenuButton](#).

**9.67.3.5 unsigned QEAbstractWidget::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

Reimplemented in [QEForm](#).

**9.67.3.6 QString QEAbstractWidget::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.67.3.7 UserLevels QEAbstractWidget::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.67.3.8 QString QEAbstractWidget::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.67.3.9 QString QEAbstractWidget::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.67.3.10 QString QEAbstractWidget::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.67.3.11 UserLevels QEAbstractWidget::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.67.3.12 bool QEAbstractWidget::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

Reimplemented in [QEForm](#), and [QEActionButton](#).

**9.67.3.13 bool QEAbstractWidget::visible [read, write]**

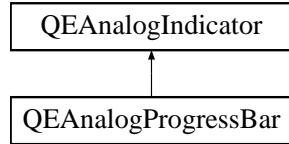
Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAbstractWidget/QEAbstractWidget.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAbstractWidget/QEAbstractWidget.cpp

## 9.68 QEAnalogIndicator Class Reference

#include <QEAnalogIndicator.h>  
Inheritance diagram for QEAnalogIndicator::



### Classes

- struct [Band](#)
- class [BandList](#)

### Public Types

- enum [Orientations](#) { [Left\\_To\\_Right](#), [Top\\_To\\_Bottom](#), [Right\\_To\\_Left](#), [Bottom\\_To\\_Top](#) }
- enum [Modes](#) { [Bar](#), [Scale](#), [Meter](#) }

### Public Slots

- void [setRange](#) (const double MinimumIn, const double MaximumIn)
- void [setValue](#) (const double ValueIn)

### Public Member Functions

- [QEAnalogIndicator](#) (QWidget \*parent=0)  
*Constructor.*
- virtual [~QEAnalogIndicator](#) ()  
*Destructor.*
- virtual QSize [sizeHint](#) () const  
*Size hint.*
- double [getValue](#) () const  
*Access function for `value` property - refer to `value` property for details.*
- void [setMinimum](#) (const double value)  
*Access function for `minimum` - refer to `minimum` property for details.*

- double [getMinimum \(\) const](#)  
*Access function for minimum - refer to [minimum](#) property for details.*
- void [setMaximum \(const double value\)](#)  
*Access function for maximum - refer to [maximum](#) property for details.*
- double [getMaximum \(\) const](#)  
*Access function for maximum - refer to [maximum](#) property for details.*
- void [setOrientation \(const enum Orientations value\)](#)  
*Access function for orientation - refer to [orientation](#) property for details.*
- enum [Orientations getOrientation \(\) const](#)  
*Access function for orientation - refer to [orientation](#) property for details.*
- void [setMode \(const enum Modes value\)](#)  
*Access function for mode - refer to [mode](#) property for details.*
- enum [Modes getMode \(\) const](#)  
*Access function for mode - refer to [mode](#) property for details.*
- void [setCentreAngle \(const int value\)](#)  
*Access function for centreAngle - refer to [centreAngle](#) property for details.*
- int [getCentreAngle \(\) const](#)  
*Access function for centreAngle - refer to [centreAngle](#) property for details.*
- void [setSpanAngle \(const int value\)](#)  
*Access function for spanAngle - refer to [spanAngle](#) property for details.*
- int [getSpanAngle \(\) const](#)  
*Access function for spanAngle - refer to [spanAngle](#) property for details.*
- void [setMinorInterval \(const double value\)](#)  
*Access function for minorInterval - refer to [minorInterval](#) property for details.*
- double [getMinorInterval \(\) const](#)  
*Access function for minorInterval - refer to [minorInterval](#) property for details.*
- void [setMajorInterval \(const double value\)](#)  
*Access function for majorInterval - refer to [majorInterval](#) property for details.*
- double [getMajorInterval \(\) const](#)  
*Access function for majorInterval - refer to [majorInterval](#) property for details.*
- void [setLogScaleInterval \(const int value\)](#)

*Access function for `logScaleInterval` - refer to `logScaleInterval` property for details.*

- int `getLogScaleInterval () const`  
*Access function for `logScaleInterval` - refer to `logScaleInterval` property for details.*
- void `setBorderColour (const QColor value)`  
*Access function for `borderColour` - refer to `borderColour` property for details.*
- QColor `getBorderColour () const`  
*Access function for `borderColour` - refer to `borderColour` property for details.*
- void `setForegroundColour (const QColor value)`  
*Access function for `foregroundColour` - refer to `foregroundColour` property for details.*
- QColor `getForegroundColour () const`  
*Access function for `foregroundColour` - refer to `foregroundColour` property for details.*
- void `setBackgroundColour (const QColor value)`  
*Access function for `backgroundColour` - refer to `backgroundColour` property for details.*
- QColor `getBackgroundColour () const`  
*Access function for `backgroundColour` - refer to `backgroundColour` property for details.*
- void `setFontColour (const QColor value)`  
*Access function for `fontColour` - refer to `fontColour` property for details.*
- QColor `getFontColour () const`  
*Access function for `fontColour` - refer to `fontColour` property for details.*
- void  `setShowText (const bool value)`  
*Access function for `showText` - refer to `showText` property for details.*
- bool `getShowText () const`  
*Access function for `showText` - refer to `showText` property for details.*
- void  `setShowScale (const bool value)`  
*Access function for `showScale` - refer to `showScale` property for details.*
- bool `getShowScale () const`  
*Access function for `showScale` - refer to `showScale` property for details.*
- void  `setLogScale (const bool value)`  
*Access function for `logScale` - refer to `logScale` property for details.*

- bool `getLogScale () const`

*Access function for `logScale` - refer to `logScale` property for details.*

## Protected Member Functions

- virtual QString `getTextImage ()`
- virtual `BandList getBandList ()`
- void `setIsActive (const bool value)`
- bool `getIsActive () const`

## Properties

- double `value`
- double `minimum`
- double `maximum`
- double `minorInterval`
- double `majorInterval`
- int `logScaleInterval`
- bool `showText`
- bool `showScale`
- bool `logScale`
- `Modes mode`
- `Orientations orientation`
- int `centreAngle`
- int `spanAngle`
- QColor `borderColour`
- QColor `backgroundColour`
- QColor `foregroundColour`
- QColor `fontColour`
- bool `isActive`

*Alternative to `isEnabled`. Default is true.*

### 9.68.1 Detailed Description

This class provides a non CA aware graphical analog indicator base class. It supports a number of display modes including Bar, Scale and Meter.

When in Bar mode, it mimics QProgressBar and provides an analog progress bar widget.

## 9.68.2 Member Enumeration Documentation

### 9.68.2.1 enum QEAnalogIndicator::Modes

The type of analog indicator used to represent the value

**Enumerator:**

*Bar* Bar (solid bar from minimum up to current value).

*Scale* Scale (diamond marker tracks current value).

*Meter* Meter (Needle moving across an arc scale).

### 9.68.2.2 enum QEAnalogIndicator::Orientations

The orientation of Bar and Scale indicators

**Enumerator:**

*Left\_To\_Right* Left to right.

*Top\_To\_Bottom* Top to bottom.

*Right\_To\_Left* Right to left.

*Bottom\_To\_Top* Bottom to top.

## 9.68.3 Property Documentation

### 9.68.3.1 QColor QEAnalogIndicator::backgroundColour [read, write]

Background colour

### 9.68.3.2 QColor QEAnalogIndicator::borderColour [read, write]

Border colour

### 9.68.3.3 int QEAnalogIndicator::centreAngle [read, write]

The angle in degreed of the line that Meter indicators are centered around. Zero represents a vertical centerline and angles increment clockwise.

### 9.68.3.4 QColor QEAnalogIndicator::fontColour [read, write]

Font colour

### 9.68.3.5 QColor QEAnalogIndicator::foregroundColour [read, write]

Foreground colour

**9.68.3.6 bool QEAnalogIndicator::logScale [read, write]**

If set, use a logarithmic scale. If clear, use a linear scale

**9.68.3.7 int QEAnalogIndicator::logScaleInterval [read, write]**

Log scale interval.

**9.68.3.8 double QEAnalogIndicator::majorInterval [read, write]**

Minor scale interval. Only applies for linear scale (not log scale)

**9.68.3.9 double QEAnalogIndicator::maximum [read, write]**

Maximum indicated value.

**9.68.3.10 double QEAnalogIndicator::minimum [read, write]**

Minimum indicated value.

**9.68.3.11 double QEAnalogIndicator::minorInterval [read, write]**

Minor scale interval. Only applies for linear scale (not log scale)

**9.68.3.12 Modes QEAnalogIndicator::mode [read, write]**

Selects what type of indicator is used (refer to Modes)

**9.68.3.13 Orientations QEAnalogIndicator::orientation [read, write]**

The orientation of Bar and Scale indicators (refer to Orientations)

**9.68.3.14 bool QEAnalogIndicator::showScale [read, write]**

If set, show the scale

**9.68.3.15 bool QEAnalogIndicator::showText [read, write]**

If set, show textual representation of value on the indicator

**9.68.3.16 int QEAnalogIndicator::spanAngle [read, write]**

The span of the Meter scale arc in degrees Typical meters are 180 deg and 270 deg

**9.68.3.17 double QEAnalogIndicator::value [read, write]**

Current indicated value.

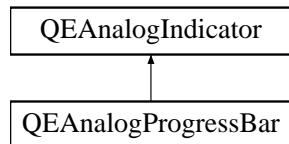
Reimplemented in [QEAnalogProgressBar](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalog
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalog

## 9.69 QEAnalogProgressBar Class Reference

Inheritance diagram for QEAnalogProgressBar::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }
- enum `AlarmSeverityDisplayModes` { `foreground`, `background` }
- enum `Formats` {
 `Default` = QQStringFormatting::FORMAT\_DEFAULT, `Floating` = QQStringFormatting::FORMAT\_FLOATING, `Integer` = QQStringFormatting::FORMAT\_INTEGER, `UnsignedInteger` = QQStringFormatting::FORMAT\_UNSIGNEDINTEGER,
 `Time` = QQStringFormatting::FORMAT\_TIME, `LocalEnumeration` = QQStringFormatting::FORMAT\_LOCAL\_ENUMERATE
 }
- enum `Separators` { `NoSeparator` = QQStringFormatting::SEPARATOR\_NONE, `Comma` = QQStringFormatting::SEPARATOR\_COMMA, `Under_score` = QQStringFormatting::SEPARATOR\_UNDERSCORE, `Space` = QQStringFormatting::SEPARATOR\_SPACE }
- enum `Notations` { `Fixed` = QQStringFormatting::NOTATION\_FIXED, `Scientific` = QQStringFormatting::NOTATION\_SCIENTIFIC, `Automatic` = QQStringFormatting::NOTATION\_AUTOMATIC }
- enum `ArrayActions` { `Append` = QQStringFormatting::APPEND, `Ascii` = QQStringFormatting::ASCII, `Index` = QQStringFormatting::INDEX }

### Public Slots

- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const QString &out)
- void `dbValueChanged` (const int &out)

- void **dbValueChanged** (const long &out)
- void **dbValueChanged** (const qlonglong &out)
- void **dbValueChanged** (const double &out)
- void **dbValueChanged** (const bool &out)
- void **dbConnectionChanged** (const bool &isConnected)
- void **requestResend** ()

*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*

## Public Member Functions

- **UserLevels getUserLevelVisibilityProperty ()**  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void **setUserLevelVisibilityProperty (UserLevels level)**  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- **UserLevels getUserLevelEnabledProperty ()**  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void **setUserLevelEnabledProperty (UserLevels level)**  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- **DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void **setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void **setFormatProperty (Formats format)**  
*Access function for `format` property - refer to `format` property for details.*
- **Formats getFormatProperty ()**  
*Access function for `format` property - refer to `format` property for details.*
- void **setSeparatorProperty (const Separators notation)**  
*Access function for `separator` property - refer to `separator` property for details.*
- **Separators getSeparatorProperty () const**  
*Access function for `separator` property - refer to `separator` property for details.*

- void [setNotationProperty \(Notations notation\)](#)  
*Access function for [notation](#) property - refer to [notation](#) property for details.*
- [Notations getNotationProperty \(\)](#)  
*Access function for [notation](#) property - refer to [notation](#) property for details.*
- void [setArrayActionProperty \(ArrayActions arrayAction\)](#)  
*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*
- [ArrayActions getArrayActionProperty \(\)](#)  
*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*
- [QEAnalogProgressBar \(QWidget \\*parent=0\)](#)
- [QEAnalogProgressBar \(const QString &variableName, QWidget \\*parent=0\)](#)
- virtual ~[QEAnalogProgressBar \(\)](#)  
*Destruction.*
- void [setUseDbDisplayLimits \(bool useDbDisplayLimitsIn\)](#)  
*Access function for [useDbDisplayLimits](#) property - refer to [useDbDisplayLimits](#) property for details.*
- bool [getUseDbDisplayLimits \(\)](#)  
*Access function for [useDbDisplayLimits](#) property - refer to [useDbDisplayLimits](#) property for details.*
- void [setAlarmSeverityDisplayStyle \(AlarmSeverityDisplayModes value\)](#)  
*Access function for [AlarmSeverityDisplayStyle](#) property - refer to [AlarmSeverityDisplayStyle](#) property for details.*
- AlarmSeverityDisplayModes [getAlarmSeverityDisplayStyle \(\)](#)  
*Access function for [AlarmSeverityDisplayStyle](#) property - refer to [AlarmSeverityDisplayStyle](#) property for details.*

## Protected Member Functions

- void [establishConnection \(unsigned int variableIndex\)](#)
- void [stringFormattingChange \(\)](#)
- void [dragEnterEvent \(QDragEnterEvent \\*event\)](#)
- void [dropEvent \(QDropEvent \\*event\)](#)
- void [mousePressEvent \(QMouseEvent \\*event\)](#)
- void [setDrop \(QVariant drop\)](#)
- QVariant [getDrop \(\)](#)
- QString [copyVariable \(\)](#)
- QVariant [copyData \(\)](#)
- QString [getTextImage \(\)](#)
- BandList [getBandList \(\)](#)

## Properties

- `QString variable`
- `QString variableSubstitutions`
- `int arrayIndex`
- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- `AlarmSeverityDisplayModes alarmSeverityDisplayStyle`
- `bool useDbDisplayLimits`
- `int value`
- `bool isActive`

*Alternative to isEnabled. Default is true.*

- `int precision`
- `bool useDbPrecision`
- `bool leadingZero`
- `bool trailingZeros`
- `bool addUnits`
- `QString localEnumeration`
- `Formats format`
- `int radix`
- `Separators separator`
- `Notations notation`
- `ArrayActions arrayAction`

### 9.69.1 Member Enumeration Documentation

#### 9.69.1.1 enum QEAnalogProgressBar::ArrayActions

User friendly enumerations for arrayAction property - refer to `QEStringFormatting::arrayActions` for details.

**Enumerator:**

**Append** Refer to `QEStringFormatting::APPEND` for details.

**Ascii** Refer to `QEStringFormatting::ASCII` for details.

**Index** Refer to `QEStringFormatting::INDEX` for details.

### 9.69.1.2 enum QEAnalogProgressBar::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and displayAlarmStateOptions enumeration for details.

#### Enumerator:

*Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

### 9.69.1.3 enum QEAnalogProgressBar::Formats

User friendly enumerations for format property - refer to [QEStringFormatting::formats](#) for details.

#### Enumerator:

*Default* Format as best appropriate for the data type.

*Floating* Format as a floating point number.

*Integer* Format as an integer.

*UnsignedInteger* Format as an unsigned integer.

*Time* Format as a time.

*LocalEnumeration* Format as a selection from the [localEnumeration](#) property.

### 9.69.1.4 enum QEAnalogProgressBar::Notations

User friendly enumerations for notation property - refer to [QEStringFormatting::notations](#) for details.

#### Enumerator:

*Fixed* Refer to [QEStringFormatting::NOTATION\\_FIXED](#) for details.

*Scientific* Refer to [QEStringFormatting::NOTATION\\_SCIENTIFIC](#) for details.

*Automatic* Refer to [QEStringFormatting::NOTATION\\_AUTOMATIC](#) for details.

### 9.69.1.5 enum QEAnalogProgressBar::Separators

User friendly enumerations for separator property - refer to [QEStringFormatting::formats](#) for details.

#### Enumerator:

*NoSeparator* Use no separator.

*Comma* Use ',' as separator.

*Underscore* Use '\_' as separator.

*Space* Use ' ' as separator.

#### 9.69.1.6 enum QEAnalogProgressBar::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and userLevel enumeration for details.

##### Enumerator:

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

#### 9.69.2 Constructor & Destructor Documentation

##### 9.69.2.1 QEAnalogProgressBar::QEAnalogProgressBar (QWidget \* *parent* = 0)

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

##### 9.69.2.2 QEAnalogProgressBar::QEAnalogProgressBar (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

#### 9.69.3 Member Function Documentation

##### 9.69.3.1 void QEAnalogProgressBar::dbConnectionChanged (const bool & *isConnected*) [signal]

Sent when the widget state updated following a channel connection change Applied to provary varible.

##### 9.69.3.2 void QEAnalogProgressBar::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.69.3.3 void QEAnalogProgressBar::setManagedVisible (bool v) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.69.4 Property Documentation

**9.69.4.1 bool QEAnalogProgressBar::addUnits [read, write]**

If true (default), add engineering units supplied with the data.

**9.69.4.2 AlarmSeverityDisplayModes QEAnalogProgressBar::alarmSeverityDisplayMode [read, write]**

Visualise the EPICS alarm severity

**9.69.4.3 bool QEAnalogProgressBar::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.69.4.4 ArrayActions QEAnalogProgressBar::arrayAction [read, write]**

Text formatting option for array data. Default is ASCII. Options are:

- ASCII - treat array as a single text string. For example an array of three characters 'a' 'b' 'c' will be formatted as 'abc'.
- APPEND - treat array as an array of numbers and format a string containing them all with a space between each. For example, an array of three numbers 10, 11 and 12 will be formatted as '10 11 12'.
- INDEX - Extract a single item from the array. The item is then formatted as any other non array data would be. The item selected is determined by the arrayIndex property. For example, if arrayIndex property is 1, an array of three numbers 10, 11 and 12 will be formatted as '11'.

**9.69.4.5 int QEAnalogProgressBar::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.69.4.6 QString QEAnalogProgressBar::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.69.4.7 bool QEAnalogProgressBar::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.69.4.8 DisplayAlarmStateOptions QEAnalogProgress-  
Bar::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.69.4.9 Formats QEAnalogProgressBar::format [read, write]**

Format to apply to data. Default is 'Default' in which case the data type supplied with the data determines how the data is formatted. For all other options, an attempt is made to format the data as requested (whatever its native form).

**9.69.4.10 unsigned QEAnalogProgressBar::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.69.4.11 bool QEAnalogProgressBar::leadingZero [read, write]**

If true (default), always add a leading zero when formatting numbers.

**9.69.4.12 QString QEAnalogProgressBar::localEnumeration [read, write]**

An enumeration list used to data values. Used only when the formatting option is 'local enumeration'. Value is converted to an integer and used to select a string from this list.

Format is:

`[[<|<=|=|=|>|=|>]value1|*] : string1 , [[<|<=|=|=|>|=|>]value2|*] : string2 , [[<|<=|=|=|>|=|>]value3|*] : string3 , ...`

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
 >= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

`0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm" <2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than 2" 3:"Beamline Available", *:"" "Pump Off":"OH NO!, the pump is OFF!","Pump On":"It's OK, the pump is on"`

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10:""'

A range of numbers can be covered by a pair of values as in the following example:  
`>=4:"Between 4 and 8",<=8:"Between 4 and 8"`

**9.69.4.13 Notations QEAnalogProgressBar::notation [read, write]**

Notation used for numerical formatting. Default is fixed.

**9.69.4.14 int QEAnalogProgressBar::precision [read, write]**

Precision used when formatting floating point numbers. The default is 4. This is only used if useDbPrecision is false.

**9.69.4.15 int QEAnalogProgressBar::radix [read, write]**

Base used for when formatting integers. Default is 10 (duh!)

**9.69.4.16 Separators QEAnalogProgressBar::separator [read, write]**

Separators used for integer and fixed point formatting. Default is None.

**9.69.4.17 QString QEAnalogProgressBar::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.69.4.18 bool QEAnalogProgressBar::trailingZeros [read, write]**

If true (default), always remove any trailing zeros when formatting numbers.

**9.69.4.19 bool QEAnalogProgressBar::useDbDisplayLimits [read, write]**

Use the EPICS database display limits

**9.69.4.20 bool QEAnalogProgressBar::useDbPrecision [read, write]**

If true (default), format floating point numbers using the precision supplied with the data. If false, the precision property is used.

**9.69.4.21 UserLevels QEAnalogProgressBar::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer' .

**9.69.4.22 QString QEAnalogProgressBar::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.69.4.23 QString QEAnalogProgressBar::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example,

'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.69.4.24 **QString QEAnalogProgressBar::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.69.4.25 **UserLevels QEAnalogProgressBar::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel() Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### 9.69.4.26 **int QEAnalogProgressBar::value [read, write]**

Current indicated value.

Reimplemented from [QEAnalogIndicator](#).

#### 9.69.4.27 **QString QEAnalogProgressBar::variable [read, write]**

EPICS variable name (CA PV)

#### 9.69.4.28 **bool QEAnalogProgressBar::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### 9.69.4.29 **QString QEAnalogProgressBar::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For exam-

ple, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### 9.69.4.30 bool QEAnalogProgressBar::visible [read, write]

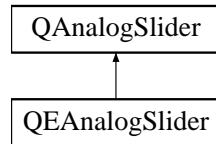
Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogProgressBar/QEAnalogProgressBar.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogProgressBar/QEAnalogProgressBar.cpp

## 9.70 QEAnalogSlider Class Reference

Inheritance diagram for QEAnalogSlider::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const long &out)

### Public Member Functions

- `UserLevels getUserLevelVisibilityProperty ()`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void `setUserLevelVisibilityProperty (UserLevels level)`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- `UserLevels getUserLevelEnabledProperty ()`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void `setUserLevelEnabledProperty (UserLevels level)`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- **DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- **void setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- **QEAnalogSlider (QWidget \*parent=0)**
- **QEAnalogSlider (const QString &variableName, const QString &readbackName="", QWidget \*parent=0)**
- **virtual ~QEAnalogSlider ()**  
*Destruction.*
- **void activated ()**
- **void writeNow ()**
- **void setVariableNameSubstitutionsProperty (const QString &substitutions)**
- **void setReadbackNameProperty (const QString &variableName)**
- **QString getReadbackNameProperty () const**
- **void setReadbackArrayIndex (const int arrayIndex)**
- **int getReadbackArrayIndex () const**
- **void setSubstitutionsProperty (const QString &substitutions)**
- **QString getSubstitutionsProperty () const**
- **void setContinuousWrite (const bool value)**
- **bool getContinuousWrite () const**
- **void setAutoScale (const bool value)**
- **bool getAutoScale () const**
- **void setAlarmColours (const bool value)**
- **bool getAlarmColours () const**

## Protected Member Functions

- **void establishConnection (unsigned int variableIndex)**
- **qcaobject::QCaObject \* createQcaItem (unsigned int variableIndex)**
- **void dragEnterEvent (QDragEnterEvent \*event)**
- **void dropEvent (QDropEvent \*event)**
- **void mousePressEvent (QMouseEvent \*event)**
- **QString copyVariable ()**
- **QVariant copyData ()**
- **void paste (QVariant s)**
- **int getPrecision () const**
- **double getMinimum () const**
- **double getMaximum () const**

## Properties

- `QString variable`
- `QString variableSubstitutions`
- `int arrayIndex`
- `QString readbackVariable`
- `int readbackArrayIndex`
- `bool continuousWrite`
- `bool autoScale`
- `bool axisAlarmColours`
- `double value`

*Specifies the value.*

- `QString leftText`

*Controls the left, centre and right texts.*

- `QString centreText`
- `QString rightText`
- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`

### 9.70.1 Member Enumeration Documentation

#### 9.70.1.1 enum QEAnalogSlider::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

##### Enumerator:

**Never** Refer to `DISPLAY_ALARM_STATE_NEVER` for details.

**Always** Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.

**WhenInAlarm** Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

### 9.70.1.2 enum QEAnalogSlider::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

#### Enumerator:

*User* Refer to `USERLEVEL_USER` for details.

*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.

*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

## 9.70.2 Constructor & Destructor Documentation

### 9.70.2.1 QEAnalogSlider::QEAnalogSlider (QWidget \* *parent* = 0) [explicit]

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

### 9.70.2.2 QEAnalogSlider::QEAnalogSlider (const QString & *variableName*, const QString & *readbackName* = "", QWidget \* *parent* = 0) [explicit]

Create with a variables. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

## 9.70.3 Member Function Documentation

### 9.70.3.1 void QEAnalogSlider::dbValueChanged (const long & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget. Note: this widget emits the numeric enumeration value as opposed to the associated text.

### 9.70.3.2 void QEAnalogSlider::setManagedVisible (bool *v*) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.70.4 Property Documentation

### 9.70.4.1 bool QEAnalogSlider::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

### 9.70.4.2 int QEAnalogSlider::arrayIndex [read, write]

Index used to select a single item of data for processing. The default is 0.

### 9.70.4.3 bool QEAnalogSlider::autoScale [read, write]

If true (default), display and editing of numbers using the precision, and control limits supplied with the data. If false, the precision, leadingZeros, minimum and maximum properties are used.

### 9.70.4.4 bool QEAnalogSlider::axisAlarmColours [read, write]

If true the background axis colour reflect the alarm and warning values of the (setpoint) variable. If false (default) then axis uses widget colour.

### 9.70.4.5 bool QEAnalogSlider::continuousWrite [read, write]

If true the widget writes to the PV as the slider is moved. If false (default) a write only occurs when apply button click.

### 9.70.4.6 QString QEAnalogSlider::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

### 9.70.4.7 bool QEAnalogSlider::displayAlarmState [read, write]

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.70.4.8 DisplayAlarmStateOptions**

**QEAnalogSlider::displayAlarmStateOption**  
[**read, write**]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.70.4.9 unsigned QEAnalogSlider::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.70.4.10 QString QEAnalogSlider::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.70.4.11 UserLevels QEAnalogSlider::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.70.4.12 QString QEAnalogSlider::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.70.4.13 QString QEAnalogSlider::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.70.4.14 QString QEAnalogSlider::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.70.4.15 UserLevels QEAnalogSlider::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.70.4.16 QString QEAnalogSlider::variable [read, write]**

EPICS variable name (CA PV)

**9.70.4.17 bool QEAnalogSlider::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.70.4.18 QString QEAnalogSlider::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

**9.70.4.19 bool QEAnalogSlider::visible [read, write]**

Display the widget. Default is true. Setting this property false is usefull if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogSlider/QEAnalogSl...
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogSlider/QEAnalogSl...

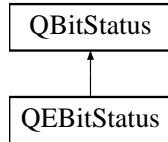
## 9.71 QEAnalogSliderManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEAnalogSlider/QEAnalogSliderManager.h

## 9.72 QEBitStatus Class Reference

Inheritance diagram for QEBitStatus:::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const QString &out)
- void `dbValueChanged` (const int &out)
- void `dbValueChanged` (const long &out)
- void `dbValueChanged` (const qlonglong &out)
- void `dbValueChanged` (const double &out)
- void `dbValueChanged` (const bool &out)
- void `dbConnectionChanged` (const bool &isConnected)

### Public Member Functions

- `UserLevels getUserLevelVisibilityProperty ()`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void `setUserLevelVisibilityProperty (UserLevels level)`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- `UserLevels getUserLevelEnabledProperty ()`

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- void `setUserLevelEnabledProperty` (`UserLevels` level)  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void `setDisplayAlarmStateOptionProperty` (`DisplayAlarmStateOptions` option)  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- `QEBitStatus` (`QWidget` \*parent=0)
- `QEBitStatus` (`const QString &variableName`, `QWidget` \*parent=0)

## Protected Member Functions

- `qcaobject::QCaObject *createQcaItem` (`unsigned int variableIndex`)
- void `establishConnection` (`unsigned int variableIndex`)
- void `dragEnterEvent` (`QDragEnterEvent *event`)
- void `dropEvent` (`QDropEvent *event`)
- void `mousePressEvent` (`QMouseEvent *event`)
- `QString copyVariable ()`
- `QVariant copyData ()`

## Properties

- `QString variable`
- `QString variableSubstitutions`
- `int arrayIndex`
- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`

- double **value**
- bool **isActive**
- bool **isValid**

### 9.72.1 Member Enumeration Documentation

#### 9.72.1.1 enum QEBitStatus::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

*Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.72.1.2 enum QEBitStatus::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and [userLevel](#) enumeration for details.

##### Enumerator:

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

### 9.72.2 Member Function Documentation

#### 9.72.2.1 void QEBitStatus::dbConnectionChanged (const bool & *isConnected*) [signal]

Sent when the widget state updated following a channel connection change Applied to provary varible.

#### 9.72.2.2 void QEBitStatus::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.72.2.3 void QEBitStatus::setManagedVisible (bool v) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

### 9.72.3 Property Documentation

**9.72.3.1 bool QEBitStatus::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.72.3.2 int QEBitStatus::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.72.3.3 QString QEBitStatus::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.72.3.4 bool QEBitStatus::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.72.3.5 DisplayAlarmStateOptions QEBitStatus::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.72.3.6 `unsigned QEBitStatus::int [read, write]`**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.72.3.7 `QString QEBitStatus::styleSheet [read, write]`**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.72.3.8 `UserLevels QEBitStatus::userLevelEnabled [read, write]`**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.72.3.9 `QString QEBitStatus::userLevelEngineerStyle [read, write]`**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.72.3.10 `QString QEBitStatus::userLevelScientistStyle [read, write]`**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.72.3.11 `QString QEBitStatus::userLevelUserStyle [read, write]`**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string

will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.72.3.12 UserLevels QEBitStatus::userLevelVisibility [read, write]

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### 9.72.3.13 QString QEBitStatus::variable [read, write]

EPICS variable name (CA PV)

#### 9.72.3.14 bool QEBitStatus::variableAsToolTip [read, write]

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### 9.72.3.15 QString QEBitStatus::variableSubstitutions [read, write]

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### 9.72.3.16 bool QEBitStatus::visible [read, write]

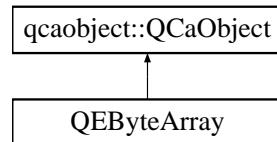
Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEBitStatus/QEBitStatus.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEBitStatus/QEBitStatus.cpp

## 9.73 QEByteArray Class Reference

Inheritance diagram for QEByteArray::



### Public Slots

- void **writeByteArray** (const QByteArray &data)

### Signals

- void **byteArrayConnectionChanged** ([QCaConnectionInfo](#) &connectionInfo, const unsigned int &variableIndex)
- void **byteArrayChanged** (const QByteArray &value, unsigned long dataSize, [QCaAlarmInfo](#) &alarmInfo, [QCaDateTime](#) &timeStamp, const unsigned int &variableIndex)

### Public Member Functions

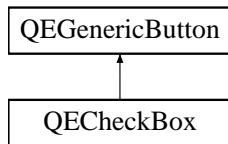
- **QEByteArray** (QString recordName, QObject \*eventObject, unsigned int variableIndexIn)
- **QEByteArray** (QString recordName, QObject \*eventObject, unsigned int variableIndexIn, UserMessage \*userMessageIn)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEByteArray.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEByteArray.cpp

## 9.74 QECheckBox Class Reference

Inheritance diagram for QECheckBox::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }
- enum `Formats` {
 `Default` = QEStringFormatting::FORMAT\_DEFAULT, `Floating` = QEStringFormatting::FORMAT\_FLOATING, `Integer` = QEStringFormatting::FORMAT\_INTEGER, `UnsignedInteger` = QEStringFormatting::FORMAT\_UNSIGNEDINTEGER,
 `Time` = QEStringFormatting::FORMAT\_TIME, `LocalEnumeration` = QEStringFormatting::FORMAT\_LOCAL\_ENUMERATE }
- enum `Separators` { `NoSeparator` = QEStringFormatting::SEPARATOR\_NONE, `Comma` = QEStringFormatting::SEPARATOR\_COMMA, `Under_score` = QEStringFormatting::SEPARATOR\_UNDERSCORE, `Space` = QEStringFormatting::SEPARATOR\_SPACE }
- enum `Notations` { `Fixed` = QEStringFormatting::NOTATION\_FIXED, `Scientific` = QEStringFormatting::NOTATION\_SCIENTIFIC, `Automatic` = QEStringFormatting::NOTATION\_AUTOMATIC }
- enum `ArrayActions` { `Append` = QEStringFormatting::APPEND, `Ascii` = QEStringFormatting::ASCII, `Index` = QEStringFormatting::INDEX }
- enum `UpdateOptions` { `Text` = QEGenericButton::UPDATE\_TEXT, `Icon` = QEGenericButton::UPDATE\_ICON, `TextAndIcon` = QEGenericButton::UPDATE\_TEXT\_AND\_ICON, `State` = QEGenericButton::UPDATE\_STATE }

*User friendly enumerations for updateOption property - refer to QEGenericButton::updateOptions for details.*

- enum `ProgramStartupOptionNames` { `None` = applicationLauncher::PSO\_NONE, `Terminal` = applicationLauncher::PSO\_TERMINAL, `LogOutput` = applicationLauncher::PSO\_LOGOUTPUT, `StdOutput` = applicationLauncher::PSO\_STDOUPUT }

- enum `CreationOptionNames` {
   
`Open` = QEActionRequests::OptionOpen,     `NewTab` = QEActionRequests::OptionNewTab,     `NewWindow` = QEActionRequests::OptionNewWindow,     `DockTop` = QEActionRequests::OptionTopDockWindow,
   
`DockBottom` = QEActionRequests::OptionBottomDockWindow,     `DockLeft` = QEActionRequests::OptionLeftDockWindow,     `DockRight` = QEActionRequests::OptionRightDockWindow,     `DockTopTabbed` = QEActionRequests::OptionTopDockWindowTabbed,
   
`DockBottomTabbed` = QEActionRequests::OptionBottomDockWindowTabbed,     `DockLeftTabbed` = QEActionRequests::OptionLeftDockWindowTabbed,     `DockRightTabbed` = QEActionRequests::OptionRightDockWindowTabbed,     `DockFloating` = QEActionRequests::OptionFloatingDockWindow }

*Creation options. Used to indicate how to present a GUI when requesting a new GUI be created. Open a new window, open a new tab, or replace the current window.*

## Public Slots

- void `requestAction` (const QEActionRequests &request)
- void `setDefaultStyle` (const QString &style)
   
*Update the default style applied to this widget.*
- void `setManagedVisible` (bool v)

## Signals

- void `dbValueChanged` (const QString &out)
- void `dbValueChanged` (const int &out)
- void `dbValueChanged` (const long &out)
- void `dbValueChanged` (const qlonglong &out)
- void `dbValueChanged` (const double &out)
- void `dbValueChanged` (const bool &out)
- void `dbConnectionChanged` (const bool &isConnected)
   
*Sent when the widget state updated following a channel connection change.*
- void `requestResend` ()
   
*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*
- void `newGui` (const QEActionRequests &request)
   
*Internal use only. Request a new GUI is created. Typically, this is caught by the QEGui application.*
- void `pressed` (int value)
- void `released` (int value)

- void [clicked](#) (int value)
- void [programCompleted](#) ()

*Program started by button has completed.*

## Public Member Functions

- [QECheckBox](#) (QWidget \*parent=0)
- [QECheckBox](#) (const QString &variableName, QWidget \*parent=0)
- void [writeNow](#) ()
- [UserLevels getUserLevelVisibilityProperty](#) ()  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- void [setUserLevelVisibilityProperty](#) (UserLevels level)  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- [UserLevels getUserLevelEnabledProperty](#) ()  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- void [setUserLevelEnabledProperty](#) (UserLevels level)  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- [DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty](#) ()  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void [setDisplayAlarmStateOptionProperty](#) (DisplayAlarmStateOptions option)  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void [setFormatProperty](#) (Formats format)  
*Access function for format property - refer to format property for details.*
- [Formats getFormatProperty](#) ()  
*Access function for format property - refer to format property for details.*
- void [setSeparatorProperty](#) (const Separators notation)  
*Access function for separator property - refer to separator property for details.*
- [Separators getSeparatorProperty](#) () const  
*Access function for separator property - refer to separator property for details.*
- void [setNotationProperty](#) (Notations notation)

*Access function for [notation](#) property - refer to [notation](#) property for details.*

- [Notations getNotationProperty \(\)](#)

*Access function for [notation](#) property - refer to [notation](#) property for details.*

- void [setArrayActionProperty \(ArrayActions arrayAction\)](#)

*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*

- [ArrayActions getArrayActionProperty \(\)](#)

*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*

## Properties

- [QString variable](#)
- [QString variableSubstitutions](#)
- int [arrayIndex](#)
- bool [subscribe](#)
- bool [variableAsToolTip](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- [QString styleSheet](#)
- [QString defaultStyle](#)
- [QString userLevelUserStyle](#)
- [QString userLevelScientistStyle](#)
- [QString userLevelEngineerStyle](#)
- [UserLevels userLevelVisibility](#)
- [UserLevels userLevelEnabled](#)
- bool [displayAlarmState](#)
- [DisplayAlarmStateOptions displayAlarmStateOption](#)
- int [precision](#)
- bool [useDbPrecision](#)
- bool [leadingZero](#)
- bool [trailingZeros](#)
- bool [addUnits](#)
- [QString localEnumeration](#)
- [Formats format](#)
- int [radix](#)
- [Separators separator](#)
- [Notations notation](#)
- [ArrayActions arrayAction](#)
- [QEWidgetProperties::DisabledRecordPolicy disabledRecordPolicy](#)
- [Qt::Alignment alignment](#)
- [UpdateOptions updateOption](#)
- [QPixmap pixmap0](#)

- QPixmap  [pixmap1](#)
- QPixmap  [pixmap2](#)
- QPixmap  [pixmap3](#)
- QPixmap  [pixmap4](#)
- QPixmap  [pixmap5](#)
- QPixmap  [pixmap6](#)
- QPixmap  [pixmap7](#)
- QString  [password](#)
- bool  [confirmAction](#)
- QString  [confirmText](#)
- bool  [writeOnPress](#)
- bool  [writeOnRelease](#)
- bool  [writeOnClick](#)
- QString  [pressText](#)
- QString  [releaseText](#)
- QString  [clickText](#)
- QString  [clickCheckedText](#)
- QString  [labelText](#)
- QString  [program](#)
- QStringList  [arguments](#)
- ProgramStartupOptionNames  [programStartupOption](#)
- QString  [guiFile](#)
- CreationOptionNames  [creationOption](#)
- QString  [prioritySubstitutions](#)
- QString  [customisationName](#)

## 9.74.1 Member Enumeration Documentation

### 9.74.1.1 enum QECheckBox::ArrayActions

User friendly enumerations for arrayAction property - refer to [QEStringFormatting::arrayActions](#) for details.

#### Enumerator:

*Append* Refer to [QEStringFormatting::APPEND](#) for details.

*Ascii* Refer to [QEStringFormatting::ASCII](#) for details.

*Index* Refer to [QEStringFormatting::INDEX](#) for details.

### 9.74.1.2 enum QECheckBox::CreationOptionNames

Creation options. Used to indicate how to present a GUI when requesting a new GUI be created. Open a new window, open a new tab, or replace the current window.

#### Enumerator:

*Open* Replace the current GUI with the new GUI.

**NewTab** Open new GUI in a new tab.

**NewWindow** Open new GUI in a new window.

**DockTop** Open new GUI in a top dock window.

**DockBottom** Open new GUI in a bottom dock window.

**DockLeft** Open new GUI in a left dock window.

**DockRight** Open new GUI in a right dock window.

**DockTopTabbed** Open new GUI in a top dock window (tabbed with any existing dock in that area).

**DockBottomTabbed** Open new GUI in a bottom dock window (tabbed with any existing dock in that area).

**DockLeftTabbed** Open new GUI in a left dock window (tabbed with any existing dock in that area).

**DockRightTabbed** Open new GUI in a right dock window (tabbed with any existing dock in that area).

**DockFloating** Open new GUI in a floating dock window.

#### 9.74.1.3 enum QECheckBox::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

**Never** Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

**Always** Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

**WhenInAlarm** Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.74.1.4 enum QECheckBox::Formats

User friendly enumerations for format property - refer to [QEStringFormatting::formats](#) for details.

##### Enumerator:

**Default** Format as best appropriate for the data type.

**Floating** Format as a floating point number.

**Integer** Format as an integer.

**UnsignedInteger** Format as an unsigned integer.

**Time** Format as a time.

**LocalEnumeration** Format as a selection from the [localEnumeration](#) property.

#### 9.74.1.5 enum QECheckBox::Notations

User friendly enumerations for notation property - refer to [QEStringFormatting::notations](#) for details.

##### Enumerator:

*Fixed* Refer to [QEStringFormatting::NOTATION\\_FIXED](#) for details.

*Scientific* Refer to [QEStringFormatting::NOTATION\\_SCIENTIFIC](#) for details.

*Automatic* Refer to [QEStringFormatting::NOTATION\\_AUTOMATIC](#) for details.

#### 9.74.1.6 enum QECheckBox::ProgramStartupOptionNames

Startup options. Just run the command, run the command within a terminal, or display the output in QE message system.

##### Enumerator:

*None* Just run the program.

*Terminal* Run the program in a terminal (in Windows a command interpreter will also be started, so the program may be a built-in command like 'dir').

*LogOutput* Run the program, and log the output in the QE message system.

*StdOutput* Run the program, and send output to standard output and standard error.

#### 9.74.1.7 enum QECheckBox::Separators

User friendly enumerations for separator property - refer to [QEStringFormatting::formats](#) for details.

##### Enumerator:

*NoSeparator* Use no separator.

*Comma* Use ',' as separator.

*Underscore* Use '\_' as separator.

*Space* Use ' ' as separator.

#### 9.74.1.8 enum QECheckBox::UpdateOptions

User friendly enumerations for updateOption property - refer to [QEGenericButton::updateOptions](#) for details.

##### Enumerator:

*Text* Data updates will update the button text.

**Icon** Data updates will update the button icon.

**TextAndIcon** Data updates will update the button text and icon.

**State** Data updates will update the button state (checked or unchecked).

#### 9.74.1.9 enum QECheckBox::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

##### Enumerator:

**User** Refer to `USERLEVEL_USER` for details.

**Scientist** Refer to `USERLEVEL_SCIENTIST` for details.

**Engineer** Refer to `USERLEVEL_ENGINEER` for details.

### 9.74.2 Constructor & Destructor Documentation

#### 9.74.2.1 QECheckBox::QECheckBox (QWidget \* *parent* = 0)

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

#### 9.74.2.2 QECheckBox::QECheckBox (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

### 9.74.3 Member Function Documentation

#### 9.74.3.1 void QECheckBox::clicked (int *value*) [signal]

Button has been Clicked. The value emitted is the integer interpretation of the `clickText` property (or the `clickCheckedText` property if the button was checked)

#### 9.74.3.2 void QECheckBox::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.74.3.3 void QECheckBox::pressed (int *value*) [signal]**

Button has been Pressed. The value emitted is the integer interpretation of the pressText property

**9.74.3.4 void QECheckBox::released (int *value*) [signal]**

Button has been Released The value emitted is the integer interpretation of the releaseText property

**9.74.3.5 void QECheckBox::requestAction (const QEActionRequests & *request*) [inline, slot]**

Default slot used to create a new GUI if there is no slot indicated in the ContainerProfile class. This slot is typically used when the button is pressed within the Designer preview window to allow the operation of the button to be tested. If an application does not specify a slot to use for creating new windows (through the ContainerProfile class) a window will still be created through this slot, but it will not respect the window creation options or any other window related application constraints. For example, the QEGui application does provide a slot for creating new GUIs in the ContainerProfile class which respects the creation options, knows how to add tabs in the application, and extend the application's window menu in the menu bar.

**9.74.3.6 void QECheckBox::setManagedVisible (bool *v*) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.74.4 Property Documentation

**9.74.4.1 bool QECheckBox::addUnits [read, write]**

If true (default), add engineering units supplied with the data.

**9.74.4.2 Qt::Alignment QECheckBox::alignment [read, write]**

Set the buttons text alignment. Left justification is particularly useful when displaying quickly changing numeric data updates.

**9.74.4.3 bool QECheckBox::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.74.4.4 QStringList QECheckBox::arguments [read, write]**

Arguments for program specified in the 'program' property.

**9.74.4.5 ArrayActions QECheckBox::arrayAction [read, write]**

Text formatting option for array data. Default is ASCII. Options are:

- ASCII - treat array as a single text string. For example an array of three characters 'a' 'b' 'c' will be formatted as 'abc'.
- APPEND - treat array as an array of numbers and format a string containing them all with a space between each. For example, an array of three numbers 10, 11 and 12 will be formatted as '10 11 12'.
- INDEX - Extract a single item from the array. The item is then formatted as any other non array data would be. The item selected is determined by the arrayIndex property. For example, if arrayIndex property is 1, an array of three numbers 10, 11 and 12 will be formatted as '11'.

**9.74.4.6 int QECheckBox::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.74.4.7 QString QECheckBox::clickCheckedText [read, write]**

Text used to compare with text written or read to determine if push button should be marked as checked. Note, must be an exact match following formatting of data updates. When writing values, the 'pressText', 'ReleaseText', or 'clickedtext' must match this property to cause the button to be checked when the write occurs.

Good example: formatting set to display a data value of '1' as 'On', clickCheckedText is 'On', clickText is 'On'. In this example, the push button will be checked when a data update occurs with a value of 1 or when the button is clicked.

Bad example: formatting set to display a data value of '1' as 'On', clickCheckedText is 'On', clickText is '1'. In this example, the push button will be checked when a data update occurs with a value of 1 but, although a valid value will be written when clicked, the button will not be checked when clicked as '1' is not the same as 'On'.

Reimplemented from [QEGenericButton](#).

**9.74.4.8 QString QECheckBox::clickText [read, write]**

Value written when user clicks button if 'writeOnClick' property is true

Reimplemented from [QEGenericButton](#).

**9.74.4.9 bool QECheckBox::confirmAction [read, write]**

If true, a dialog will be presented asking the user to confirm if the button action should be carried out

**9.74.4.10 QString QECheckBox::confirmText [read, write]**

Text used to confirm action if confirmation dialog is presented

Reimplemented from [QEGenericButton](#).

**9.74.4.11 CreationOptionNames QECheckBox::creationOption [read, write]**

Creation options when opening a new GUI. Open a new window, open a new tab, or replace the current window. The creation option is supplied when the button generates a newGui signal. Application code connected to this signal should honour this request if possible. When used within the QEGui application, the QEGui application creates a new window, new tab, or replaces the current window as appropriate.

Reimplemented from [QEGenericButton](#).

**9.74.4.12 QString QECheckBox::customisationName [read, write]**

Window customisation name. This name will be used to select a set of window customisations including menu items and tool bar buttons. Applications such as QEGui can load .xml files containing named sets of window customisations. This property is used to select a set loaded from these files. The selected set of customisations will be applied to the main window containing the new GUI. Customisations are not applied if the GUI is opened as a dock.

Reimplemented from [QEGenericButton](#).

**9.74.4.13 QString QECheckBox::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.74.4.14 QEWidgetProperties::DisabledRecordPolicy  
QECheckBox::disabledRecordPolicy [read, write]**

Set the widget's disabled record policy, i.e. the action to be taken when the underlying record is disabled, i.e. when the associated record's DISA and DISV field values are equal. Note: this is only applicable IOC process variables. When the policy is ignore, then no special action is taken. This is the default policy. When the policy is grayout, the widget's style is set as if disconnected when the record is disabled.

Reimplemented from [QEGenericButton](#).

**9.74.4.15 bool QECheckBox::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.74.4.16 DisplayAlarmStateOptions QECheckBox::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.74.4.17 Formats QECheckBox::format [read, write]**

Format to apply to data. Default is 'Default' in which case the data type supplied with the data determines how the data is formatted. For all other options, an attempt is made to format the data as requested (whatever its native form).

**9.74.4.18 QString QECheckBox::guiFile [read, write]**

File name of GUI to be presented on button click. File name can be absolute, relative to the path of the QEform in which the [QEPushButton](#) is located, relative to the any path in the path list published in the ContainerProfile class, or relative to the current path. See [QEWidget::openQEFfile\(\)](#) in [QEWidget.cpp](#) for details.

**9.74.4.19 unsigned QECheckBox::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.74.4.20 QString QECheckBox::labelText [read, write]**

Button label text (prior to substitution). Macro substitutions will be applied to this text and the result will be set as the button text. Used when data updates are not being represented in the button text. IF NOT LEFT EMPTY, THIS TEXT WILL TAKE PRIORITY OVER THE PUSH BUTTON 'text' PROPERTY! For example, a button

in a sub form may have a 'labelText' property of 'Turn Pump On'. When the sub form is used twice in a main form with substitutions PUMPNUM=1 and PUMPNUM=2 respectively, the two identical buttons in the sub forms will have the labels 'Turn Pump 1 On' and 'Turn Pump 2 On' respectively.

Reimplemented from [QEGenericButton](#).

#### **9.74.4.21 bool QECheckBox::leadingZero [read, write]**

If true (default), always add a leading zero when formatting numbers.

#### **9.74.4.22 QString QECheckBox::localEnumeration [read, write]**

An enumeration list used to data values. Used only when the formatting option is 'local enumeration'. Value is converted to an integer and used to select a string from this list.

Format is:

[[<|<=|=|=|>]value1]\* : string1 , [[<|<=|=|=|>]value2]\* : string2 ,  
[[<|<=|=|=|>]value3]\* : string3 , ...

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
>= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm"  
<2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than  
2" 3:"Beamline Available", \*:"" "Pump Off":"OH NO!, the pump is OFF!","Pump  
On":"It's OK, the pump is on"

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10":""

A range of numbers can be covered by a pair of values as in the following example:  
>=4:"Between 4 and 8",<=8:"Between 4 and 8"

#### **9.74.4.23 Notations QECheckBox::notation [read, write]**

Notation used for numerical formatting. Default is fixed.

**9.74.4.24 QString QECheckBox::password [read, write]**

Password user will need to enter before any action is taken

Reimplemented from [QEGenericButton](#).

**9.74.4.25 QPixmap QECheckBox:: pixmap0 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 0

**9.74.4.26 QPixmap QECheckBox:: pixmap1 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 1

**9.74.4.27 QPixmap QECheckBox:: pixmap2 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 2

**9.74.4.28 QPixmap QECheckBox:: pixmap3 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 3

**9.74.4.29 QPixmap QECheckBox:: pixmap4 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 4

**9.74.4.30 QPixmap QECheckBox:: pixmap5 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 5

**9.74.4.31 QPixmap QECheckBox:: pixmap6 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 6

**9.74.4.32 QPixmap QECheckBox:: pixmap7 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 7

**9.74.4.33 int QECheckBox::precision [read, write]**

Precision used when formatting floating point numbers. The default is 4. This is only used if useDbPrecision is false.

**9.74.4.34 QString QECheckBox::pressText [read, write]**

Value written when user presses button if 'writeOnPress' property is true

Reimplemented from [QEGenericButton](#).

**9.74.4.35 QString QECheckBox::prioritySubstitutions [read, write]**

Overriding macro substitutions. These macro substitions take precedence over any existing macro substitutions defined by the variableSubstitutions property, any parent forms, or the application containing the button. These macro substitutions are particularly usefull when the button's function is to reload the same form but with different macro substitutions. The variableSubstitutions property cannot be used for this since, although they are added to the list of macro substitions applied to the new form, they are appended to the list and the existing macro substitutions take precedence.

Reimplemented from [QEGenericButton](#).

**9.74.4.36 QString QECheckBox::program [read, write]**

Program to run when the button is clicked. No attempt to run a program is made if this property is empty. Example: firefox

**9.74.4.37 ProgramStartupOptionNames QECheckBox::programStartupOption [read, write]**

Startup options. Just run the command, run the command within a terminal, or display the output in QE message system.

**9.74.4.38 int QECheckBox::radix [read, write]**

Base used for when formatting integers. Default is 10 (duh!)

**9.74.4.39 QString QECheckBox::releaseText [read, write]**

Value written when user releases button if 'writeOnRelease' property is true

Reimplemented from [QEGenericButton](#).

**9.74.4.40 Separators QECheckBox::separator [read, write]**

Separators used for integer and fixed point formatting. Default is None.

**9.74.4.41 QString QECheckBox::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.74.4.42 bool QECheckBox::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.74.4.43 bool QECheckBox::trailingZeros [read, write]**

If true (default), always remove any trailing zeros when formatting numbers.

**9.74.4.44 UpdateOptions QECheckBox::updateOption [read, write]**

Update options (text, pixmap, both, or state (checked or unchecked)

Reimplemented from [QEGenericButton](#).

**9.74.4.45 bool QECheckBox::useDbPrecision [read, write]**

If true (default), format floating point numbers using the precision supplied with the data. If false, the precision property is used.

**9.74.4.46 UserLevels QECheckBox::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.74.4.47 QString QECheckBox::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string

will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.74.4.48 QString QECheckBox::userLevelScientistStyle [read, write]

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.74.4.49 QString QECheckBox::userLevelUserStyle [read, write]

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.74.4.50 UserLevels QECheckBox::userLevelVisibility [read, write]

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### 9.74.4.51 QString QECheckBox::variable [read, write]

EPICS variable name (CA PV)

#### 9.74.4.52 bool QECheckBox::variableAsToolTip [read, write]

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### 9.74.4.53 QString QECheckBox::variableSubstitutions [read, write]

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

**9.74.4.54 bool QECheckBox::visible [read, write]**

Display the widget. Default is true. Setting this property false is usefull if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

**9.74.4.55 bool QECheckBox::writeOnClick [read, write]**

If true, the 'clickText' property is written when the button is clicked. Default is true

Reimplemented from [QEGenericButton](#).

**9.74.4.56 bool QECheckBox::writeOnPress [read, write]**

If true, the 'pressText' property is written when the button is pressed. Default is false

Reimplemented from [QEGenericButton](#).

**9.74.4.57 bool QECheckBox::writeOnRelease [read, write]**

If true, the 'releaseText' property is written when the button is released. Default is false

Reimplemented from [QEGenericButton](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QECheckBox.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QECheckBox.cpp

## 9.75 QECheckBoxManager Class Reference

### Public Member Functions

- **QECheckBoxManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QECheckBoxManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QECheckBoxManager.cpp

## 9.76 QEComboBox Class Reference

### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setDefaultStyle` (const QString &style)  
*Update the default style applied to this widget.*
- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const QString &out)
- void `dbValueChanged` (const int &out)
- void `dbValueChanged` (const long &out)
- void `dbValueChanged` (const qlonglong &out)
- void `dbValueChanged` (const double &out)
- void `dbValueChanged` (const bool &out)
- void `dbConnectionChanged` (const bool &isConnected)  
*Sent when the widget state updated following a channel connection change.*
- void `userChange` (const QString &oldValue, const QString &newValue, const QString &lastValue)  
*Internal use only. Used by `QEConfiguredLayout` to be notified when one of its widgets has written something.*

### Public Member Functions

- `QEComboBox` (QWidget \*parent=0)
- `QEComboBox` (const QString &variableName, QWidget \*parent=0)
- void `setWriteOnChange` (bool writeOnChangeIn)
- bool `getWriteOnChange` () const
- void `setSubscribe` (bool subscribe)
- bool `getSubscribe` () const
- void `setUseDbEnumerations` (bool `useDbEnumerations`)

- `bool getUseDbEnumerations () const`
- `void setLocalEnumerations (const QString &localEnumerations)`
- `QString getLocalEnumerations () const`
- `void setAllowFocusUpdate (bool allowFocusUpdate)`
- `bool getAllowFocusUpdate () const`
- `void writeNow ()`
- `UserLevels getUserLevelVisibilityProperty ()`  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- `void setUserLevelVisibilityProperty (UserLevels level)`  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- `UserLevels getUserLevelEnabledProperty ()`  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- `void setUserLevelEnabledProperty (UserLevels level)`  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- `void setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)`  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*

## Protected Member Functions

- `void establishConnection (unsigned int variableIndex)`
- `void dragEnterEvent (QDragEnterEvent *event)`
- `void dropEvent (QDropEvent *event)`
- `void setDrop (QVariant drop)`
- `QVariant getDrop ()`
- `QString copyVariable ()`
- `QVariant copyData ()`
- `void paste (QVariant s)`

## Protected Attributes

- `QEIntegerFormatting integerFormatting`
- `QELocalEnumeration localEnumerations`
- `bool useDbEnumerations`
- `bool writeOnChange`

## Properties

- QString variable
- QString variableSubstitutions
- int arrayIndex
- bool subscribe
- bool allowFocusUpdate
- bool variableAsToolTip
- bool allowDrop
- bool visible
- unsigned int
- QString styleSheet
- QString defaultStyle
- QString userLevelUserStyle
- QString userLevelScientistStyle
- QString userLevelEngineerStyle
- UserLevels userLevelVisibility
- UserLevels userLevelEnabled
- bool displayAlarmState
- DisplayAlarmStateOptions displayAlarmStateOption
- QString localEnumeration

### 9.76.1 Member Enumeration Documentation

#### 9.76.1.1 enum QEComboBox::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

**Enumerator:**

*Never* Refer to `DISPLAY_ALARM_STATE_NEVER` for details.

*Always* Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.

*WhenInAlarm* Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

#### 9.76.1.2 enum QEComboBox::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

**Enumerator:**

*User* Refer to `USERLEVEL_USER` for details.

*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.

*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

### 9.76.2 Member Function Documentation

#### 9.76.2.1 void QEComboBox::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change. Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

#### 9.76.2.2 void QEComboBox::setManagedVisible (bool *v*) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

### 9.76.3 Member Data Documentation

#### 9.76.3.1 bool QEComboBox::useDbEnumerations [read, write, protected]

Use database enumerations - defaults to true

#### 9.76.3.2 bool QEComboBox::writeOnChange [read, write, protected]

Sets if this widget writes any changes as the user selects values (the QComboBox 'activated' signal is emitted). Default is 'true' (writes any changes when the QComboBox 'activated' signal is emitted).

### 9.76.4 Property Documentation

#### 9.76.4.1 bool QEComboBox::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

#### 9.76.4.2 bool QEComboBox::allowFocusUpdate [read, write]

Allow updated while widget has focus - defaults to false

#### 9.76.4.3 int QEComboBox::arrayIndex [read, write]

Index used to select a single item of data for processing. The default is 0.

**9.76.4.4 QString QEComboBox::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.76.4.5 bool QEComboBox::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.76.4.6 DisplayAlarmStateOptions QEComboBox::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.76.4.7 unsigned QEComboBox::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.76.4.8 QString QEComboBox::localEnumeration [read, write]**

Enumrations values used when useDbEnumerations is false.

**9.76.4.9 QString QEComboBox::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.76.4.10 bool QEComboBox::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.76.4.11 UserLevels QEComboBox::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.76.4.12 QString QEComboBox::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.76.4.13 QString QEComboBox::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.76.4.14 QString QEComboBox::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.76.4.15 UserLevels QEComboBox::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.76.4.16 QString QEComboBox::variable [read, write]**

EPICS variable name (CA PV)

**9.76.4.17 bool QEComboBox::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.76.4.18 QString QEComboBox::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

**9.76.4.19 bool QEComboBox::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEComboBox/QEComboBox.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEComboBox/QEComboBox.cpp

## 9.77 QEConfiguredLayout Class Reference

### Public Types

- enum **configurationTypesProperty** { **File** = FROM\_FILE, **Text** = FROM\_TEXT }
- enum **optionsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }
- enum **UserLevels** { **User** = userLevelTypes::USERLEVEL\_USER, **Scientist** = userLevelTypes::USERLEVEL\_SCIENTIST, **Engineer** = userLevelTypes::USERLEVEL\_ENGINEER }
- enum **DisplayAlarmStateOptions** { **Never** = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, **Always** = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, **WhenInAlarm** = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void [setManagedVisible](#) (bool v)

### Public Member Functions

- **QEConfiguredLayout** (QWidget \*pParent=0, bool pSubscription=true)
- void **setItemDescription** (QString pValue)
- QString **getItemDescription** ()
- void **setShowItemList** (bool pValue)
- bool **getShowItemList** ()
- void **setConfigurationType** (int pValue)
- int **getConfigurationType** ()
- void **setConfigurationFile** (QString pValue)
- QString **getConfigurationFile** ()
- void **setConfigurationText** (QString pValue)
- QString **getConfigurationText** ()
- void **setOptionsLayout** (int pValue)
- int **getOptionsLayout** ()
- void **setCurrentUserType** (int pValue)
- int **getCurrentUserType** ()
- void **refreshFields** ()
- void **userLevelChanged** (userLevelTypes::userLevels pValue)
- void **setConfigurationTypeProperty** (configurationTypesProperty pConfigurationType)
- configurationTypesProperty **getConfigurationTypeProperty** ()
- void **setOptionsLayoutProperty** (optionsLayoutProperty pOptionsLayout)
- optionsLayoutProperty **getOptionsLayoutProperty** ()
- UserLevels **getUserLevelVisibilityProperty** ()

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- void `setUserLevelVisibilityProperty (UserLevels level)`

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- `UserLevels getUserLevelEnabledProperty ()`

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- void `setUserLevelEnabledProperty (UserLevels level)`

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- void `setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)`

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

## Public Attributes

- `QList< _Item * > itemList`
- `QList< _Field * > currentFieldList`

## Protected Attributes

- `QLabel * qLabelItemDescription`
- `QComboBox * qComboBoxItemList`
- `QVBoxLayout * qVBoxLayoutFields`
- `QScrollArea * qScrollArea`
- `QString configurationFile`
- `QString configurationText`
- `int configurationType`
- `int optionsLayout`
- `int currentUserType`
- `bool subscription`

## Properties

- `QString itemDescription`
- `bool showItemList`
- `configurationTypesProperty configurationType`
- `optionsLayoutProperty optionsLayout`

*Change the order of the widgets. Valid orders are: TOP, BOTTOM, LEFT and RIGHT.*

- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`

### 9.77.1 Member Enumeration Documentation

#### 9.77.1.1 enum QEConfiguredLayout::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

##### Enumerator:

**Never** Refer to `DISPLAY_ALARM_STATE_NEVER` for details.

**Always** Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.

**WhenInAlarm** Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

#### 9.77.1.2 enum QEConfiguredLayout::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

##### Enumerator:

**User** Refer to `USERLEVEL_USER` for details.

**Scientist** Refer to `USERLEVEL_SCIENTIST` for details.

**Engineer** Refer to `USERLEVEL_ENGINEER` for details.

## 9.77.2 Member Function Documentation

### 9.77.2.1 void QEConfiguredLayout::setManagedVisible (bool *v*) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.77.3 Property Documentation

### 9.77.3.1 bool QEConfiguredLayout::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

### 9.77.3.2 QString QEConfiguredLayout::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

### 9.77.3.3 bool QEConfiguredLayout::displayAlarmState [read, write]

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

### 9.77.3.4 DisplayAlarmStateOptions QEConfiguredLayout::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

### 9.77.3.5 unsigned QEConfiguredLayout::int [read, write]

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For

example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

#### 9.77.3.6 `QString QEConfiguredLayout::styleSheet [read, write]`

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

#### 9.77.3.7 `UserLevels QEConfiguredLayout::userLevelEnabled [read, write]`

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmaticaly through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

#### 9.77.3.8 `QString QEConfiguredLayout::userLevelEngineerStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.77.3.9 `QString QEConfiguredLayout::userLevelScientistStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.77.3.10 `QString QEConfiguredLayout::userLevelUserStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string

will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.77.3.11 UserLevels QEConfiguredLayout::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.77.3.12 bool QEConfiguredLayout::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### **9.77.3.13 bool QEConfiguredLayout::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

## 9.78 QEConfiguredLayoutManager Class Reference

### Public Member Functions

- **QEConfiguredLayoutManager** (QObject \*pParent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*pParent)
- void **initialize** (QDesignerFormEditorInterface \*pCore)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

## 9.79 QEFileBrowser Class Reference

```
#include <QEFileBrowser.h>
```

### Public Types

- enum **optionsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }
- enum **UserLevels** { **User** = userLevelTypes::USERLEVEL\_USER, **Scientist** = userLevelTypes::USERLEVEL\_SCIENTIST, **Engineer** = userLevelTypes::USERLEVEL\_ENGINEER }
- enum **DisplayAlarmStateOptions** { **Never** = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, **Always** = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, **WhenInAlarm** = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void **setManagedVisible** (bool v)

### Signals

- void **selected** (QString pFilename)

### Public Member Functions

- **QEFileBrowser** (QWidget \*pParent=0)
- void **setVariableName** (QString pValue)
- QString **getVariableName** ()
- void **setVariableNameSubstitutions** (QString pValue)
- QString **getVariableNameSubstitutions** ()
- void **setDirectoryPath** (QString pValue)
- QString **getDirectoryPath** ()
- void **setShowDirectoryPath** (bool pValue)
- bool **getShowDirectoryPath** ()
- void **setShowDirectoryBrowser** (bool pValue)
- bool **getShowDirectoryBrowser** ()
- void **setShowRefresh** (bool pValue)
- bool **getShowRefresh** ()
- void **setShowTable** (bool pValue)
- bool **getShowTable** ()
- void **setShowColumnTime** (bool pValue)
- bool **getShowColumnTime** ()
- void **setShowColumnSize** (bool pValue)
- bool **getShowColumnSize** ()

- void **setShowColumnFilename** (bool pValue)
- bool **getShowColumnFilename** ()
- void **setShowFileExtension** (bool pValue)
- bool **getShowFileExtension** ()
- void **setFileFilter** (QString pValue)
- QString **getFileFilter** ()
- void **setFileDialogDirectoriesOnly** (bool pValue)
- bool **getFileDialogDirectoriesOnly** ()
- void **setOptionsLayout** (int pValue)
- int **getOptionsLayout** ()
- void **updateTable** ()
- void **setOptionsLayoutProperty** (optionsLayoutProperty pOptionsLayout)
- optionsLayoutProperty **getOptionsLayoutProperty** ()
- UserLevels **getUserLevelVisibilityProperty** ()  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- void **setUserLevelVisibilityProperty** (UserLevels level)  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- UserLevels **getUserLevelEnabledProperty** ()  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- void **setUserLevelEnabledProperty** (UserLevels level)  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- DisplayAlarmStateOptions **getDisplayAlarmStateOptionProperty** ()  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void **setDisplayAlarmStateOptionProperty** (DisplayAlarmStateOptions option)  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*

## Protected Attributes

- QLEnEdit \* **qeLineEditDirectoryPath**
- QPushButton \* **qPushButtonDirectoryBrowser**
- QPushButton \* **qPushButtonRefresh**
- \_QTableWidgetFileBrowser \* **qTableWidgetFileBrowser**
- QString **fileFilter**
- bool **showFileExtension**

*Show/hide the extension of files.*

- bool **fileDialogDirectoriesOnly**  
*Enable/disable the browsing of directories-only when opening the dialog window.*
- int **optionsLayout**

## Properties

- QString **variable**
- QString **variableSubstitutions**
- QString **directoryPath**  
*Default directory where to browse files when **QEFileBrowser** is launched for the first time.*
- bool **showDirectoryPath**  
*Show/hide directory path line edit where the user can specify the directory to browse files.*
- bool **showDirectoryBrowser**  
*Show/hide button to open the dialog window to browse for directories and files.*
- bool **showRefresh**  
*Show/hide button to refresh the table containing the list of files being browsed.*
- bool **showTable**  
*Show/hide table containing the list of files being browsed.*
- bool **showColumnTime**  
*Show/hide column containing the time of creation of files.*
- bool **showColumnSize**  
*Show/hide column containing the size (in bytes) of files.*
- bool **showColumnFilename**  
*Show/hide column containing the name of files.*
- optionsLayoutProperty **optionsLayout**  
*Change the order of the widgets. Valid orders are: TOP, BOTTOM, LEFT and RIG.*
  - bool **variableAsToolTip**
  - bool **allowDrop**
  - bool **visible**
  - unsigned **int**
  - QString **styleSheet**
  - QString **defaultStyle**
  - QString **userLevelUserStyle**

- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`

### 9.79.1 Detailed Description

This class is a EPICS aware widget. The `QEFileBrowser` widget allows the user to browse existing files from a certain directory.

### 9.79.2 Member Enumeration Documentation

#### 9.79.2.1 enum QEFileBrowser::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

##### Enumerator:

- Never* Refer to `DISPLAY_ALARM_STATE_NEVER` for details.  
*Always* Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.  
*WhenInAlarm* Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

#### 9.79.2.2 enum QEFileBrowser::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

##### Enumerator:

- User* Refer to `USERLEVEL_USER` for details.  
*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.  
*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

### 9.79.3 Member Function Documentation

#### 9.79.3.1 void QEFileBrowser::selected (QString *pFilename*) [signal]

Signal that is generated every time the user double-clicks a certain file. This signal emits a string that contains the full path and the name of the selected file. This signal may be captured by other widgets that perform further operations (for instance, the `QEImage` displays the content of this file if it is a graphical one).

**9.79.3.2 void QEFileBrowser::setManagedVisible (bool v) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

**9.79.4 Member Data Documentation****9.79.4.1 QString QEFileBrowser::fileFilter [read, write, protected]**

Specify which files to browse. To specify more than one filter, please separate them with a ";". Example: \*.py;\*.ui (this will only display files with an extension .py or .ui).

**9.79.5 Property Documentation****9.79.5.1 bool QEFileBrowser::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.79.5.2 QString QEFileBrowser::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.79.5.3 bool QEFileBrowser::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.79.5.4 DisplayAlarmStateOptions QEFileBrowser::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.79.5.5 unsigned QEFileBrowser::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.79.5.6 QString QEFileBrowser::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.79.5.7 UserLevels QEFileBrowser::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.79.5.8 QString QEFileBrowser::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.79.5.9 QString QEFileBrowser::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.79.5.10 QString QEFileBrowser::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string

will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.79.5.11 UserLevels QEFileBrowser::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.79.5.12 QString QEFileBrowser::variable [read, write]**

EPICS variable name (CA PV). This variable is used for both writing and reading the directory to be used by the widget.

#### **9.79.5.13 bool QEFileBrowser::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### **9.79.5.14 QString QEFileBrowser::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### **9.79.5.15 bool QEFileBrowser::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileBrowser/QEFileBrowser.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileBrowser/QEFileBrowser.cpp

## 9.80 QEFileDialog Class Reference

```
#include <QEFileDialog.h>
```

### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setImageFileName` (const QString &text)
- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const QString &out)
- void `requestResend` ()

*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*

### Public Member Functions

- `QEFileDialog` (QWidget \*parent=0)
- `QEFileDialog` (const QString &variableName, QWidget \*parent=0)
- void `setVariableNameProperty` (QString variableName)

*Property access function for `variable` property. This has special behaviour to work well within designer.*

- QString `getVariableNameProperty` ()

*Property access function for `variable` property. This has special behaviour to work well within designer.*

- void `setVariableNameSubstitutionsProperty` (QString variableNameSubstitutions)

*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*

- QString `getVariableNameSubstitutionsProperty` ()

*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*

- **UserLevels getUserLevelVisibilityProperty ()**  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void **setUserLevelVisibilityProperty (UserLevels level)**  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- **UserLevels getUserLevelEnabledProperty ()**  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void **setUserLevelEnabledProperty (UserLevels level)**  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- **DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void **setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

## Properties

- `QString variable`
- `QString variableSubstitutions`
- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`

### 9.80.1 Detailed Description

This class is a EPICS aware image widget based on the Qt label widget. When a variable is defined, the label pixmap will be updated with the file specified by the variable. The label will be disabled if the variable is invalid. It is tightly integrated with the base class QEWidget which provides generic support such as macro substitutions, drag/drop, and standard properties.

### 9.80.2 Member Enumeration Documentation

#### 9.80.2.1 enum QEFileImage::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

*Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.80.2.2 enum QEFileImage::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and [userLevel](#) enumeration for details.

##### Enumerator:

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

### 9.80.3 Constructor & Destructor Documentation

#### 9.80.3.1 QEFileImage::QEFileImage (QWidget \* *parent* = 0)

Create without a variable. Use [setVariableNameProperty\(\)](#) and [setSubstitutionsProperty\(\)](#) to define a variable and, optionally, macro substitutions later.

#### 9.80.3.2 QEFileImage::QEFileImage (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

### 9.80.4 Member Function Documentation

#### 9.80.4.1 void QEFileImage::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

#### 9.80.4.2 void QEFileImage::setManagedVisible (bool *v*) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

### 9.80.5 Property Documentation

#### 9.80.5.1 bool QEFileImage::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

#### 9.80.5.2 QString QEFileImage::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

#### 9.80.5.3 bool QEFileImage::displayAlarmState [read, write]

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.80.5.4 DisplayAlarmStateOptions QEFileImage::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm' If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.80.5.5 unsigned QEFileImage::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.80.5.6 QString QEFileImage::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.80.5.7 UserLevels QEFileImage::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.80.5.8 QString QEFileImage::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.80.5.9 QString QEFileImage::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.80.5.10 QString QEFileImage::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string

will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.80.5.11 UserLevels QEFileDialog::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.80.5.12 QString QEFileDialog::variable [read, write]**

EPICS variable name (CA PV)

#### **9.80.5.13 bool QEFileDialog::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### **9.80.5.14 QString QEFileDialog::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### **9.80.5.15 bool QEFileDialog::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileDialog/QEFileDialog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileDialog/QEFileDialog.cpp

## 9.81 QEFileImageManager Class Reference

### Public Member Functions

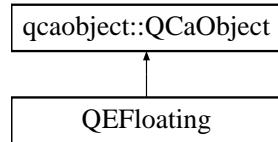
- **QEFileImageManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileDialog/QEFileDialog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFileDialog/QEFileDialog.cpp

## 9.82 QEFloating Class Reference

Inheritance diagram for QEFloating::



### Public Slots

- void **writeFloating** (const double &data)
- void **writeFloatingElement** (const double &data)
- void **writeFloating** (const QVector< double > &data)

### Signals

- void **floatingConnectionChanged** (QCaConnectionInfo &connectionInfo, const unsigned int &variableIndex)
- void **floatingChanged** (const double &value, QCaAlarmInfo &alarmInfo, QCaDateTime &timeStamp, const unsigned int &variableIndex)
- void **floatingArrayChanged** (const QVector< double > &values, QCaAlarmInfo &alarmInfo, QCaDateTime &timeStamp, const unsigned int &variableIndex)

### Public Member Functions

- **QEFloating** (QString recordName, QObject \*eventObject, QEFloatingFormatting \*floatingFormattingIn, unsigned int variableIndexIn)
- **QEFloating** (QString recordName, QObject \*eventObject, QEFloatingFormatting \*floatingFormattingIn, unsigned int variableIndexIn, UserMessage \*userMessageIn)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEFloating.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEFloating.cpp

## 9.83 QEFloatinArray Class Reference

```
#include <QEFloatinArray.h>
```

### Public Member Functions

- **QEFloatinArray** (int size)
- **QEFloatinArray** (int size, const double &t)
- **QEFloatinArray** (const QVector< double > &other)
- double **minimumValue** (const double &defaultValue=0.0)
- double **maximumValue** (const double &defaultValue=0.0)
- **QEFloatinArray calcDyByDx** (const QVector< double > &x)
- **QEFloatinArray medianFilter** (const int window)

#### 9.83.1 Detailed Description

This class provides short hand for QVector<double> together with some basic double vector operations.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEFloatinArray.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEFloatinArray.cpp

## 9.84 QEFloatingFormatting Class Reference

### Public Types

- enum **formats** {  
    **FORMAT\_e** = 'e', **FORMAT\_E** = 'E', **FORMAT\_f** = 'f', **FORMAT\_g** = 'g',  
    **FORMAT\_G** = 'G' }

### Public Member Functions

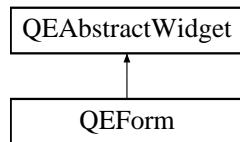
- double **formatFloating** (const QVariant &value)
- QVector< double > **formatFloatingArray** (const QVariant &value)
- QVariant **formatValue** (const double &floatingValue, generic::generic\_types valueType)
- QVariant **formatValue** (const QVector< double > &floatingValue, generic::generic\_types valueType)
- void **setPrecision** (unsigned int precision)
- void **setFormat** (formats format)
- unsigned int **getPrecision** ()
- int **getFormat** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEFloatingFormatting.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEFloatingFormatting.cpp

## 9.85 QEForm Class Reference

Inheritance diagram for QEForm::



### Public Types

- enum **MessageFilterOptions** { **Match** = UserMessage::MESSAGE\_FILTER\_\_MATCH, **None** = UserMessage::MESSAGE\_FILTER\_NONE }

### Public Slots

- bool [readUiFile \(\)](#)  
*Find a widget within the ui loaded by the [QEForm](#). Returns NULL if no UI is loaded yet or if the named widget can't be found.*
- void [requestAction \(const QEActionRequests &request\)](#)  
*Read a .ui file and present it within this [QEForm](#).*

### Signals

- void [formLoaded \(bool fileLoaded\)](#)

### Public Member Functions

- [QEForm \(QWidget \\*parent=0\)](#)
- [QEForm \(const QString &uifileNameIn, QWidget \\*parent=0\)](#)
- void [commonInit \(const bool alertIfUINoFoundIn, const bool loadManuallyIn\)](#)
- void [setQEGuiTitle \(const QString titleIn\)](#)
- QString [getQEGuiTitle \(\)](#)  
*Set the title to be used as the window or form title. (note, also set when reading a .ui file).*
- QString [getFullName \(\)](#)  
*Get the title to be used as the window or form title.*
- QString [getUiFileName \(\)](#)  
*Get the standard, absolute UI file name.*

- void **setFileMonitoringEnabled** (bool fileMonitoringEnabled)  
*Get the fully substituted file name (Not the uiFile property).*
- bool **getFileMonitoringEnabled** ()  
*Set flag indicating if form should take account of file monitoring.*
- void **setHandleGuiLaunchRequests** (bool handleGuiLaunchRequests)  
*Get flag indicating if form should take account of file monitoring.*
- bool **getHandleGuiLaunchRequests** ()  
*Set flag indicating form should handle gui form launch requests.*
- void **setResizeContents** (bool resizeContentsIn)  
*Get flag indicating form should handle gui form launch requests.*
- bool **getResizeContents** ()  
*Set flag indicating form should resize contents to match form size (otherwise resize form to match contents).*
- QString **getContainedFrameworkVersion** ()  
*Get flag indicating form should resize contents to match form size (otherwise resize form to match contents).*
- QString **getUniqueIdentifier** ()  
*Get the version of the first QE widget (if any) of QE widgets by QUILoader.*
- void **setUniqueIdentifier** (QString name)  
*Get a unique identifier string for this form. This identifier should be persistant across application runs as it is based on the QEForm's position in the widget hierarchy. The same widget will generate the same identifier when opened within the same GUI.*
- int **getDisconnectedCount** ()  
*Set a unique identifier string for this form. This identifier should be persistant across application runs as it is based on the QEForm's position in the widget hierarchy. The same widget will generate the same identifier when opened within the same GUI.*
- int **getConnectedCount** ()  
*Return the count of disconnected variables.*
- QWidget \* **getChild** (QString name)  
*Return the count of connected variables.*
- void **setUiFileNameProperty** (QString uiFileName)
- QString **getUiFileNameProperty** ()
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)

- `QString getVariableNameSubstitutionsProperty ()`
- `MessageFilterOptions getMessageFormFilter ()`
- `void setMessageFormFilter (MessageFilterOptions messageFormFilter)`
- `MessageFilterOptions getMessageSourceFilter ()`
- `void setMessageSourceFilter (MessageFilterOptions messageSourceFilter)`

## Protected Member Functions

- `void establishConnection (unsigned int variableIndex)`
- `void clearUiFileNames ()`

## Protected Attributes

- `QString uiFileName`
- `QString fullUiFileName`
- `bool handleGuiLaunchRequests`
- `bool resizeContents`

## Properties

- `QString uiFile`
- `QString variableSubstitutions`
- `unsigned int`
- `MessageFilterOptions messageFormFilter`
- `MessageFilterOptions messageSourceFilter`
- `bool variableAsToolTip`
- `bool allowDrop`
- `DisplayAlarmStateOptions displayAlarmStateOption`

### 9.85.1 Member Data Documentation

#### 9.85.1.1 bool QEForm::handleGuiLaunchRequests [read, write, protected]

If true, the `QEForm` widget publishes its own slot for launching new GUIs so all QE widgets within it will use the `QEForm`'s mechanism for launching new GUIs, rather than any mechanism the application may provide (through the ContainerProfile mechanism)

#### 9.85.1.2 bool QEForm::resizeContents [read, write, protected]

If set, the `QEForm` will resize the top level widget of the .ui file it opens (and set other size and border related properties) to match itself. This is useful if the `QEForm` is used as a sub form within a main form (possibly another `QEForm`) and you want to control the size of the `QEForm` being used as a sub form. If clear, the `QEForm` will resize itself

(and set other size and border related properties) to match the top level widget of the .ui file it opens. This is useful if the [QEForm](#) is used as a sub form within a main form (possibly another [QEForm](#)) and you want the main form to resize to match the size of the [QEForm](#) being used as a sub form, or you want the sub form border decorations (such as frame shape and shadow) to be displayed.

## 9.85.2 Property Documentation

### 9.85.2.1 bool QEForm::allowDrop [read, write]

allowDrop is added as a non-designable property here only to hide the implementation present in [QEAbstractWidget](#)

Reimplemented from [QEAbstractWidget](#).

### 9.85.2.2 DisplayAlarmStateOptions QEForm::displayAlarmStateOption [read, write]

displayAlarmStateOption is added as a non-designable property here only to hide the implementation present in [QEAbstractWidget](#)

Reimplemented from [QEAbstractWidget](#).

### 9.85.2.3 unsigned QEForm::int [read, write]

Widgets or applications that use messages from the framework have the option of filtering on this ID. Messages that the [QEForm](#) widget catches with its message filters will be regenerated using this ID

Reimplemented from [QEAbstractWidget](#).

### 9.85.2.4 MessageFilterOptions QEForm::messageFormFilter [read, write]

Message filter that attempts to match messages sent through the QE message logging system based on the automatically generated message form ID. This filter will match form ID of the message to the form ID of this QEform as follows:

Any - A message will always be accepted. Match - A message will be accepted if it comes from a QE widget within this form. None - The message will not be matched based on the form the message comes from. (It may still be accepted based on the message source ID.) Matched messages will be resent with the messageSourceId of this [QEForm](#)

**9.85.2.5 MessageFilterOptions QEForm::messageSourceFilter [read, write]**

!?!? is this a valid property. Resending messages based on the source ID is unnecessary as they will be sent on with the same source ID? Message filter that attempts to match messages sent through the QE message logging system based on the messageSourceId of the widget that generatedd the messge. This filter will match message message source ID of the message to the message source ID of this QEform as follows:

Any - A message will always be accepted. Match - A message will be accepted if the message source ID matches this [QEForm](#). None - The message will not be matched based of message source ID (It may still be accepted based on the message form ID.) Matched messages will be resend with the messageSourceId of this [QEForm](#).

**9.85.2.6 QString QEForm::uiFile [read, write]**

File name of the .ui file being presented within the [QEForm](#) widget.

**9.85.2.7 bool QEForm::variableAsToolTip [read, write]**

variableAsToolTip is added as a non-designable property here only to hide the implementation present in [QEAbstractWidget](#)

Reimplemented from [QEAbstractWidget](#).

**9.85.2.8 QString QEForm::variableSubstitutions [read, write]**

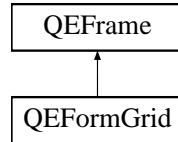
Macro substitutions to be applied to this widget, and all QE widgets that are opened when the .ui file is presented. Note, despite the name, the macro substitutions are general macro substitutions, and do not just apply to a variable name (in fact a [QEForm](#) widget does not even have a variable name property).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEForm/QEForm.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEForm/QEForm.cpp

## 9.86 QEFormGrid Class Reference

#include <QEFormGrid.h> Inheritance diagram for QEFormGrid::



### Classes

- class **MacroData**

### Public Types

- enum **GridOrders** { **RowMajor** = 0, **ColMajor** = 1 }

### Public Member Functions

- **QEFormGrid** (QWidget \*parent=0)
- **QEFormGrid** (const QString &uiFile, const int number, const int cols, QWidget \*parent=0)
- virtual ~**QEFormGrid** ()  
*Destruiction.*
- void **setUiFile** (QString uiFileName)
- QString **getUiFile** ()
- void **setGridVariableSubstitutions** (QString variableSubstitutions)
- QString **getGridVariableSubstitutions** ()
- void **setNumber** (int n)
- int **getNumber** ()
- void **setColumns** (int n)
- int **getColumns** ()
- void **setGridOrder** (**GridOrders** go)
- **GridOrders** **getGridOrder** ()
- void **setMargin** (int n)
- int **getMargin** ()
- void **setSpacing** (int n)
- int **getSpacing** ()
- int **getRows** ()

### Protected Member Functions

- QSize **sizeHint** () const

## Properties

- `QString uiFile`
- `QString variableSubstitutions`
- `int number`
- `int columns`
- `GridOrders gridOrder`
- `int margin`
- `int spacing`
- `QString slotMacroPrefix`
- `int slotNumberOffset`
- `int slotNumberWidth`
- `QStringList slotStrings`
- `QString rowMacroPrefix`
- `int rowNumberOffset`
- `int rowNumberWidth`
- `QStringList rowStrings`
- `QString colMacroPrefix`
- `int colNumberOffset`
- `int colNumberWidth`
- `QStringList colStrings`
- `bool variableAsToolTip`
- `bool allowDrop`
- `DisplayAlarmStateOptions displayAlarmStateOption`

## Friends

- class `MacroData`

### 9.86.1 Detailed Description

This class load a grid QEForms.

### 9.86.2 Member Enumeration Documentation

#### 9.86.2.1 enum QEFormGrid::GridOrders

`GridOrders` specifies how grid elements are layed out.

Row major (example 10, items, 3 cols): 0 1 2 3 4 5 6 7 8 9 - -

Col major (example 10, items, 3 cols): 0 4 8 1 5 9 2 6 - 3 7 -

### 9.86.3 Constructor & Destructor Documentation

#### 9.86.3.1 `QEFormGrid::QEFormGrid (QWidget *parent = 0) [explicit]`

Create a grid widget with default parameters.

### 9.86.4 Property Documentation

#### 9.86.4.1 `bool QEFormGrid::allowDrop [read, write]`

allowDrop is added as a non-designable property here only to hide the implementation present in [QEFrame](#)

Reimplemented from [QEFrame](#).

#### 9.86.4.2 `QString QEFormGrid::colMacroPrefix [read, write]`

Specified formal slot macro name prefix Default value: COL

#### 9.86.4.3 `int QEFormGrid::colNumberOffset [read, write]`

Specifies the COL macro number offset. Default value: 1.

#### 9.86.4.4 `int QEFormGrid::colNumberWidth [read, write]`

Specifies the COL macro width format. Default value: 2.

#### 9.86.4.5 `QStringList QEFormGrid::colStrings [read, write]`

Specifies the COLNAME macro values applied to each QEFrom grid element in a specific column. Indexing of the colStrings is NOT impacted by the colNumberOffset. Default value: ""

#### 9.86.4.6 `int QEFormGrid::columns [read, write]`

Specifies the number of columns. This is restricted to the range 1 to 200. Default value: 1.

#### 9.86.4.7 `DisplayAlarmStateOptions QEFormGrid::displayAlarmStateOption [read, write]`

displayAlarmStateOption is added as a non-designable property here only to hide the implementation present in [QEFrame](#)

Reimplemented from [QEFrame](#).

**9.86.4.8 GridOrders QEFormGrid::gridOrder [read, write]**

Specifies the gridOrder. Default value: RowMajor

**9.86.4.9 int QEFormGrid::margin [read, write]**

Margin of the internal QGridLayout object. Default value: 2.

**9.86.4.10 int QEFormGrid::number [read, write]**

The total number of QEForms. This is restricted to the range 1 to 2000. Default value: 4.

**9.86.4.11 QString QEFormGrid::rowMacroPrefix [read, write]**

Specified formal slot macro name prefix Default value: ROW

**9.86.4.12 int QEFormGrid::rowNumberOffset [read, write]**

Specifies the ROW macro number offset. Default value: 1.

**9.86.4.13 int QEFormGrid::rowNumberWidth [read, write]**

Specifies the COL macro width format. Default value: 2.

**9.86.4.14 QStringList QEFormGrid::rowStrings [read, write]**

Specifies the ROWNAME macro values applied to each QEFrom grid element in a spfci row. Indexing of the rowStrings is NOT impacted by the rowNumberOffset. Default value: ""

**9.86.4.15 QString QEFormGrid::slotMacroPrefix [read, write]**

Specified formal slot macro name prefix Default value: SLOT

**9.86.4.16 int QEFormGrid::slotNumberOffset [read, write]**

Specifies the SLOT macro number offset. Default value: 1.

**9.86.4.17 int QEFormGrid::slotNumberWidth [read, write]**

Specifies the SLOT macro width format. Default value: 2.

**9.86.4.18 QStringList QEFormGrid::slotStrings [read, write]**

Specifies the SLOTNAME macro values applied to each QEFrom grid element. Slot number allocation is impacted by the gridOrder property. Indexing of the slotStrings is NOT impacted by the slotNumberOffset. Default value: ""

**9.86.4.19 int QEFormGrid::spacing [read, write]**

Spacing of the internal QGridLayout object. Default value: 2.

**9.86.4.20 QString QEFormGrid::uiFile [read, write]**

The uiFile loaded into each [QEForm](#) element. Default value: ""

**9.86.4.21 bool QEFormGrid::variableAsToolTip [read, write]**

variableAsToolTip is added as a non-designable property here only to hide the implementation present in [QEFrame](#)

Reimplemented from [QEFrame](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFormGrid/QEFormGrid.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFormGrid/QEFormGrid.cpp

## 9.87 QEFormGridManager Class Reference

### Public Member Functions

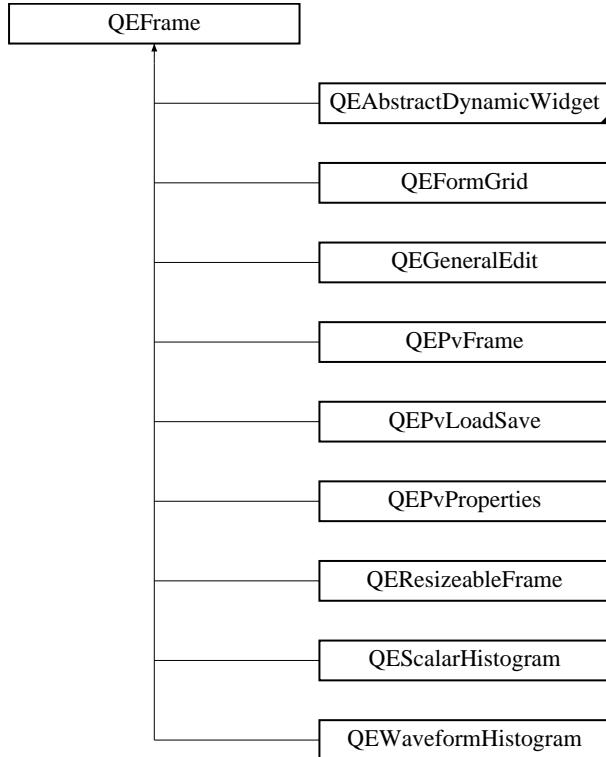
- **QEFormGridManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFormGrid/QEFormGridManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFormGrid/QEFormGridManager.cpp

## 9.88 QEFrame Class Reference

Inheritance diagram for QEFrame::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setManagedVisible` (bool v)
- void `setSelectPixmap` (const int index)

### Public Member Functions

- `UserLevels getUserLevelVisibilityProperty ()`

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- void `setUserLevelVisibilityProperty (UserLevels level)`

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- UserLevels `getUserLevelEnabledProperty ()`

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- void `setUserLevelEnabledProperty (UserLevels level)`

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- DisplayAlarmStateOptions `getDisplayAlarmStateOptionProperty ()`

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- void `setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)`

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- QEFrame (QWidget \*parent=0)

- QSize `sizeHint () const`

- void `setScaledContents (bool scaledContentsIn)`

- bool `getScaledContents () const`

- int `getSelectedPixmap () const`

## Protected Member Functions

- void `paintEvent (QPaintEvent *event)`
- void `pixmapUpdated (const int index)`

## Properties

- bool `variableAsToolTip`
- bool `allowDrop`
- bool `visible`
- unsigned `int`
- QString `styleSheet`
- QString `defaultStyle`
- QString `userLevelUserStyle`
- QString `userLevelScientistStyle`
- QString `userLevelEngineerStyle`
- UserLevels `userLevelVisibility`

- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- `QPixmap pixmap`
- `bool scaledContents`
- `QPixmap pixmap0`
- `QPixmap pixmap1`
- `QPixmap pixmap2`
- `QPixmap pixmap3`
- `QPixmap pixmap4`
- `QPixmap pixmap5`
- `QPixmap pixmap6`
- `QPixmap pixmap7`

### 9.88.1 Member Enumeration Documentation

#### 9.88.1.1 enum QEFrame::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

**Enumerator:**

*Never* Refer to `DISPLAY_ALARM_STATE_NEVER` for details.

*Always* Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.

*WhenInAlarm* Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

#### 9.88.1.2 enum QEFrame::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

**Enumerator:**

*User* Refer to `USERLEVEL_USER` for details.

*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.

*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

### 9.88.2 Member Function Documentation

#### 9.88.2.1 void QEFrame::setManagedVisible (bool v) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

### 9.88.3 Property Documentation

#### 9.88.3.1 bool QEFrame::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

Reimplemented in [QEFormGrid](#).

#### 9.88.3.2 QString QEFrame::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

#### 9.88.3.3 bool QEFrame::displayAlarmState [read, write]

DEPRECATED. USE `displayAlarmStateOption` INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.88.3.4 DisplayAlarmStateOptions QEFrame::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

Reimplemented in [QEFormGrid](#).

#### 9.88.3.5 unsigned QEFrame::int [read, write]

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

#### 9.88.3.6 QPixmap QEFrame::pixmap [read, write]

Pixmap for frame background. Similar operation to pixmap property for a QLabel. Deprecated, and is an alias for pixmap0

**9.88.3.7 QPixmap QEFrame:: pixmap0 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 0.

**9.88.3.8 QPixmap QEFrame:: pixmap1 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 1.

**9.88.3.9 QPixmap QEFrame:: pixmap2 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 2.

**9.88.3.10 QPixmap QEFrame:: pixmap3 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 3.

**9.88.3.11 QPixmap QEFrame:: pixmap4 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 4.

**9.88.3.12 QPixmap QEFrame:: pixmap5 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 5.

**9.88.3.13 QPixmap QEFrame:: pixmap6 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 6.

**9.88.3.14 QPixmap QEFrame:: pixmap7 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 7.

**9.88.3.15 bool QEFrame:: scaledContents [read, write]**

Flag the pixmap for the background is to be scaled to fit the frame. Similar operation to scaledContents property for a QLabel.

**9.88.3.16 QString QEFrame::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.88.3.17 UserLevels QEFrame::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.88.3.18 QString QEFrame::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.88.3.19 QString QEFrame::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.88.3.20 QString QEFrame::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.88.3.21 UserLevels QEFrame::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user

mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.88.3.22 bool QEFrame::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

Reimplemented in [QEFormGrid](#).

#### **9.88.3.23 bool QEFrame::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

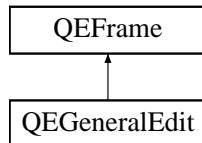
The documentation for this class was generated from the following files:

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFrame/QEFrame.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFrame/QEFrame.cpp

## 9.89 QEGeneralEdit Class Reference

The [QEGeneralEdit](#) class This class provides a general PV edit widget, presenting one off a [QELineEdit](#), a [QENumericEdit](#) or a [QERadioGroup](#) for string, numerical and enumeration data kinds respectively.

```
#include <QEGeneralEdit.h>
```



### Public Member Functions

- [QEGeneralEdit](#) (QWidget \*parent=0)
  - [QEGeneralEdit](#) (const QString &variableName, QWidget \*parent=0)
  - virtual ~[QEGeneralEdit](#) ()
- Destruction.*
- void [setArrayIndex](#) (const int arrayIndex)

### Protected Member Functions

- QSize [sizeHint](#) () const
- void [establishConnection](#) (unsigned int variableIndex)
- [qcaobject::QCaObject](#) \* [createQcaItem](#) (unsigned int variableIndex)
- void [dragEnterEvent](#) (QDragEnterEvent \*event)
- void [dropEvent](#) (QDropEvent \*event)
- void [mousePressEvent](#) (QMouseEvent \*event)
- void [setDrop](#) (QVariant drop)
- QVariant [getDrop](#) ()
- QString [copyVariable](#) ()
- QVariant [copyData](#) ()
- void [paste](#) (QVariant s)

### Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- int [arrayIndex](#)

### 9.89.1 Detailed Description

The [QEGeneralEdit](#) class This class provides a general PV edit widget, presenting one off a [QELineEdit](#), a [QENumericEdit](#) or a [QERadioGroup](#) for string, numerical and enumeration data kinds respectively. This widget is intended for use within a qegui predefined form, displayed in response to context menu request to edit an arbitrary PV.

### 9.89.2 Constructor & Destructor Documentation

#### 9.89.2.1 **QEGeneralEdit::QEGeneralEdit (QWidget \* *parent* = 0) [explicit]**

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

#### 9.89.2.2 **QEGeneralEdit::QEGeneralEdit (const QString & *variableName*, QWidget \* *parent* = 0) [explicit]**

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

### 9.89.3 Property Documentation

#### 9.89.3.1 **int QEGeneralEdit::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

#### 9.89.3.2 **QString QEGeneralEdit::variable [read, write]**

EPICS variable name (CA PV)

#### 9.89.3.3 **QString QEGeneralEdit::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEGeneralEdit/QEGeneralEdit.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEGeneralEdit/QEGeneralEdit.cpp

## 9.90 QEGeneralEditManager Class Reference

### Public Member Functions

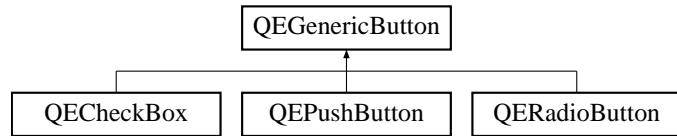
- **QEGeneralEditManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEGeneralEdit/QEGeneralEditManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEGeneralEdit/QEGeneralEditManager.cpp

## 9.91 QEGenericButton Class Reference

Inheritance diagram for QEGenericButton::



### Public Types

- enum **VariableAllocation** {
   
    **VAR\_PRIMARY** = 0, **VAR\_READBACK**, **VAR\_DISA**, **VAR\_DISV**,
   
    **NUMBER\_OF\_VARIABLES** }
- enum **updateOptions** { **UPDATE\_TEXT**, **UPDATE\_ICON**, **UPDATE\_TEXT\_AND\_ICON**, **UPDATE\_STATE** }

### Public Member Functions

- **QEGenericButton** (QAbstractButton \*owner)
- void **setSubscribe** (bool subscribe)
- bool **getSubscribe** ()
- void **setUpdateOption** (updateOptions updateOptionIn)
- updateOptions **getUpdateOption** ()
- void **setTextAlignment** (Qt::Alignment alignment)
- Qt::Alignment **getTextAlignment** ()
- void **setPassword** (QString password)
- QString **getPassword** ()
- void **setConfirmAction** (bool confirmRequiredIn)
- bool **getConfirmAction** ()
- void **setConfirmText** (QString confirmTextIn)
- QString **getConfirmText** ()
- void **setWriteOnPress** (bool writeOnPress)
- bool **getWriteOnPress** ()
- void **setWriteOnRelease** (bool writeOnRelease)
- bool **getWriteOnRelease** ()
- void **setWriteOnClick** (bool writeOnClick)
- bool **getWriteOnClick** ()
- void **setPressText** (QString pressText)
- QString **getPressText** ()
- void **setReleaseText** (QString releaseTextIn)
- QString **getReleaseText** ()
- void **setClickText** (QString clickTextIn)
- QString **getClickText** ()

- void **setClickCheckedText** (QString clickCheckedTextIn)
- QString **getClickCheckedText** ()
- void **setProgram** (QString program)
- QString **getProgram** ()
- void **setArguments** (QStringList arguments)
- QStringList  **getArguments** ()
- void **setProgramStartupOption** (applicationLauncher::programStartupOptions programStartupOptionIn)
- applicationLauncher::programStartupOptions **getProgramStartupOption** ()
- void **setGuiName** (QString guiName)
- QString **getGuiName** ()
- void **setPrioritySubstitutions** (QString prioritySubstitutionsIn)
- QString **getPrioritySubstitutions** ()
- void **setCustomisationName** (QString customisationNameIn)
- QString **getCustomisationName** ()
- void **setCreationOption** (QEActionRequests::Options creationOption)
- QEActionRequests::Options **getCreationOption** ()
- void **setLabelTextProperty** (QString labelTextIn)
- QString **getLabelTextProperty** ()
- void **setDisabledRecordPolicy** (const QEWidgetProperties::DisabledRecordPolicy disabledRecordPolicy)
- QEWidgetProperties::DisabledRecordPolicy **getDisabledRecordPolicy** () const
- void **writeClickedNow** (const bool checked=false)

## Protected Member Functions

- void **useGenericNewVariableName** (const QString &variableName, const QString &variableNameSubstitutions, const unsigned int variableIndex)
- void **connectionChanged** (QCaConnectionInfo &connectionInfo, const unsigned int &variableIndex)
- void **setGenericButtonText** (const QString &text, QCaAlarmInfo &alarmInfo, QCaDateTime &, const unsigned int &variableIndex)
- void **setGenericDISAvalue** (const long &value, QCaAlarmInfo &alarmInfo, QCaDateTime &, const unsigned int &variableIndex)
- void **setGenericDISVvalue** (const long &value, QCaAlarmInfo &alarmInfo, QCaDateTime &, const unsigned int &variableIndex)
- void **userPressed** ()
- void **userReleased** ()
- void **userClicked** (bool checked)
- void **processWriteNow** (const bool checked)
- virtual updateOptions **getDefaultUpdateOption** ()=0
- void **setup** ()
- void **establishConnection** (unsigned int variableIndex)
- void **calcStyleOption** ()

## Protected Attributes

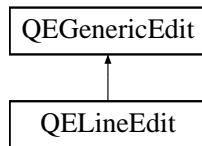
- applicationLauncher **programLauncher**
- QESingleVariableMethods \* **altReadback**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QEGenericButton.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QEGenericButton.c

## 9.92 QEGenericEdit Class Reference

Inheritance diagram for QEGenericEdit::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setManagedVisible` (bool v)
- void `setDefaultStyle` (const QString &style)

*Update the default style applied to this widget.*

### Signals

- void `userChange` (const QVariant &oldValue, const QVariant &newValue, const QVariant &lastValue)
 

*Internal use only. Used by `QEConfiguredLayout` to be notified when one of its widgets has written something.*
- void `requestResend` ()
 

*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*

### Public Member Functions

- `UserLevels getUserLevelVisibilityProperty` ()
 

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- void `setUserLevelVisibilityProperty (UserLevels level)`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- `UserLevels getUserLevelEnabledProperty ()`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void `setUserLevelEnabledProperty (UserLevels level)`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void `setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)`  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- `QEGenericEdit (QWidget *parent=0)`
- `QEGenericEdit (const QString &variableName, QWidget *parent=0)`
- void `setWriteOnLoseFocus (bool writeOnLoseFocus)`
- bool `getWriteOnLoseFocus ()`
- void `setWriteOnEnter (bool writeOnEnter)`
- bool `getWriteOnEnter ()`
- void `setWriteOnFinish (bool writeOnFinish)`
- bool `getWriteOnFinish ()`
- void `setConfirmWrite (bool confirmWrite)`
- bool `getConfirmWrite ()`
- void `setAllowFocusUpdate (bool allowFocusUpdate)`
- bool `getAllowFocusUpdate () const`  
*Returns 'true' if this widget configured to allow updates while it has focus.*
- void `setSubscribe (bool subscribe)`
- bool `getSubscribe ()`
- void `writeValue (qcaobject::QCaObject *qca, QVariant newValue)`
- void `writeNow ()`

## Protected Member Functions

- void `setDataIfNoFocus (const QVariant &value, QCaAlarmInfo &alarmInfo, QCaDateTime &dateTime)`
- bool `getIsConnected () const`
- bool `getIsFirstUpdate () const`
- virtual void `setValue (const QVariant &value)=0`
- virtual QVariant `getValue ()=0`
- virtual bool `writeData (const QVariant &value, QString &message)=0`

## Protected Attributes

- QVariant **lastValue**
- QVariant **lastUserValue**
- bool **messageDialogPresent**
- bool **writeFailMessageDialogPresent**
- bool **isConnected**

## Properties

- QString **text**
- QString **variable**
- QString **variableSubstitutions**
- int **arrayIndex**
- bool **subscribe**
- bool **writeOnLoseFocus**
- bool **writeOnEnter**
- bool **writeOnFinish**
- bool **confirmWrite**
- bool **allowFocusUpdate**

*Allow updated while widget has focus - defaults to false.*

- bool **variableAsToolTip**
- bool **allowDrop**
- bool **visible**
- unsigned **int**
- QString **styleSheet**
- QString **defaultStyle**
- QString **userLevelUserStyle**
- QString **userLevelScientistStyle**
- QString **userLevelEngineerStyle**
- **UserLevels userLevelVisibility**
- **UserLevels userLevelEnabled**
- bool **displayAlarmState**
- DisplayAlarmStateOptions **displayAlarmStateOption**

### 9.92.1 Member Enumeration Documentation

#### 9.92.1.1 enum QEGenericEdit::DisplayAlarmStateOptions

User friendly enumerations for **displayAlarmStateOption** property - refer to **displayAlarmStateOption** property and **displayAlarmStateOptions** enumeration for details.

##### Enumerator:

**Never** Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

### 9.92.1.2 enum QEGenericEdit::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and userLevel enumeration for details.

#### Enumerator:

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

## 9.92.2 Constructor & Destructor Documentation

### 9.92.2.1 QEGenericEdit::QEGenericEdit (QWidget \* *parent* = 0)

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

### 9.92.2.2 QEGenericEdit::QEGenericEdit (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

## 9.92.3 Member Function Documentation

### 9.92.3.1 bool QEGenericEdit::getConfirmWrite ()

Returns 'true' if this widget will ask for confirmation (using a dialog box) prior to writing data.

### 9.92.3.2 bool QEGenericEdit::getSubscribe ()

Returns 'true' if this widget subscribes for data updates and displays current data.

### 9.92.3.3 bool QEGenericEdit::getWriteOnEnter ()

Returns 'true' if this widget writes any changes when the user presses 'enter'.

**9.92.3.4 bool QEGenericEdit::getWriteOnFinish ()**

Returns 'true' if this widget writes any changes when the user finished editing (the QLineEdit 'editingFinished' signal is emitted).

**9.92.3.5 bool QEGenericEdit::getWriteOnLoseFocus ()**

Returns 'true' if this widget automatically writes any changes when it loses focus.

**9.92.3.6 void QEGenericEdit::setAllowFocusUpdate (bool *allowFocusUpdate*)**

Sets if this widget configured to allow updates while it has focus. Default is 'false'.

**9.92.3.7 void QEGenericEdit::setConfirmWrite (bool *confirmWrite*)**

Sets if this widget will ask for confirmation (using a dialog box) prior to writing data. Default is 'false' (will not ask for confirmation (using a dialog box) prior to writing data).

**9.92.3.8 void QEGenericEdit::setManagedVisible (bool *v*) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

**9.92.3.9 void QEGenericEdit::setSubscribe (bool *subscribe*)**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.92.3.10 void QEGenericEdit::setWriteOnEnter (bool *writeOnEnter*)**

Sets if this widget writes any changes when the user presses 'enter'. Note, the current value will be written even if the user has not changed it. Default is 'true' (writes any changes when the user presses 'enter').

**9.92.3.11 void QEGenericEdit::setWriteOnFinish (bool *writeOnFinish*)**

Sets if this widget writes any changes when the user finished editing (the QLineEdit 'editingFinished' signal is emitted). No writing occurs if no changes were made. Default is 'true' (writes any changes when the QLineEdit 'editingFinished' signal is emitted).

**9.92.3.12 void QEGenericEdit::setWriteOnLoseFocus (bool *writeOnLoseFocus*)**

Sets if this widget automatically writes any changes when it loses focus. Default is 'false' (does not write any changes when it loses focus).

**9.92.4 Property Documentation****9.92.4.1 bool QEGenericEdit::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.92.4.2 int QEGenericEdit::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.92.4.3 bool QEGenericEdit::confirmWrite [read, write]**

Sets if this widget will ask for confirmation (using a dialog box) prior to writing data. Default is 'false' (will not ask for confirmation (using a dialog box) prior to writing data).

**9.92.4.4 QString QEGenericEdit::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.92.4.5 bool QEGenericEdit::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.92.4.6 DisplayAlarmStateOptions QEGenericEdit::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of

standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.92.4.7 **unsigned QEGenericEdit::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

#### 9.92.4.8 **QString QEGenericEdit::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

#### 9.92.4.9 **bool QEGenericEdit::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

#### 9.92.4.10 **UserLevels QEGenericEdit::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

#### 9.92.4.11 **QString QEGenericEdit::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.92.4.12 **QString QEGenericEdit::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager

class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.92.4.13 QString QEGenericEdit::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.92.4.14 UserLevels QEGenericEdit::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.92.4.15 QString QEGenericEdit::variable [read, write]**

EPICS variable name (CA PV)

#### **9.92.4.16 bool QEGenericEdit::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### **9.92.4.17 QString QEGenericEdit::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### **9.92.4.18 bool QEGenericEdit::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

**9.92.4.19 bool QEGenericEdit::writeOnEnter [read, write]**

Sets if this widget writes any changes when the user presses 'enter'. Note, the current value will be written even if the user has not changed it. Default is 'true' (writes any changes when the user presses 'enter').

**9.92.4.20 bool QEGenericEdit::writeOnFinish [read, write]**

Sets if this widget writes any changes when the user finished editing (the QLineEdit 'editingFinished' signal is emitted). No writing occurs if no changes were made. Default is 'true' (writes any changes when the QLineEdit 'editingFinished' signal is emitted).

**9.92.4.21 bool QEGenericEdit::writeOnLoseFocus [read, write]**

Sets if this widget automatically writes any changes when it loses focus. Default is 'false' (does not write any changes when it loses focus).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELineEdit/QEGenericEdit.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELineEdit/QEGenericEdit.cpp

## 9.93 QEGroupBox Class Reference

### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setManagedVisible` (bool v)

### Public Member Functions

- `UserLevels getUserLevelVisibilityProperty ()`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void `setUserLevelVisibilityProperty (UserLevels level)`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- `UserLevels getUserLevelEnabledProperty ()`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void `setUserLevelEnabledProperty (UserLevels level)`  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void `setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)`  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- `QEGroupBox (QWidget *parent=0)`
- `QEGroupBox (const QString &title, QWidget *parent=0)`
- `QSize sizeHint () const`

## Protected Member Functions

- virtual void **setSubstitutionsProperty** (QString macroSubstitutionsIn)
- QString **getSubstitutionsProperty** ()

## Properties

- bool **variableAsToolTip**
- bool **allowDrop**
- bool **visible**
- unsigned **int**
- QString **styleSheet**
- QString **defaultStyle**
- QString **userLevelUserStyle**
- QString **userLevelScientistStyle**
- QString **userLevelEngineerStyle**
- **UserLevels userLevelVisibility**
- **UserLevels userLevelEnabled**
- bool **displayAlarmState**
- **DisplayAlarmStateOptions displayAlarmStateOption**
- QString **substitutedTitle**
- QString **textSubstitutions**

### 9.93.1 Member Enumeration Documentation

#### 9.93.1.1 enum QEGroupBox::DisplayAlarmStateOptions

User friendly enumerations for **displayAlarmStateOption** property - refer to **displayAlarmStateOption** property and **displayAlarmStateOptions** enumeration for details.

##### Enumerator:

- Never** Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.  
**Always** Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.  
**WhenInAlarm** Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.93.1.2 enum QEGroupBox::UserLevels

User friendly enumerations for **userLevelVisibility** and **userLevelEnabled** properties - refer to **userLevelVisibility** and **userLevelEnabled** properties and **userLevel** enumeration for details.

##### Enumerator:

- User** Refer to USERLEVEL\_USER for details.  
**Scientist** Refer to USERLEVEL\_SCIENTIST for details.  
**Engineer** Refer to USERLEVEL\_ENGINEER for details.

## 9.93.2 Member Function Documentation

### 9.93.2.1 void QEGroupBox::setManagedVisible (bool *v*) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.93.3 Property Documentation

### 9.93.3.1 bool QEGroupBox::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

### 9.93.3.2 QString QEGroupBox::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

### 9.93.3.3 bool QEGroupBox::displayAlarmState [read, write]

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

### 9.93.3.4 DisplayAlarmStateOptions QEGroupBox::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

### 9.93.3.5 unsigned QEGroupBox::int [read, write]

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.93.3.6 QString QEGroupBox::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.93.3.7 QString QEGroupBox::substitutedTitle [read, write]**

Group box title text to be substituted. This text will be copied to the group box title text after applying any macro substitutions from the textSubstitutions property

**9.93.3.8 QString QEGroupBox::textSubstitutions [read, write]**

Text substitutions. These substitutions are applied to the 'substitutedTitle' property prior to copying it to the label text.

**9.93.3.9 UserLevels QEGroupBox::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.93.3.10 QString QEGroupBox::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.93.3.11 QString QEGroupBox::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.93.3.12 QString QEGroupBox::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.93.3.13 UserLevels QEGroupBox::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.93.3.14 bool QEGroupBox::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.93.3.15 bool QEGroupBox::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEGroupBox/QEGroupBox.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEGroupBox/QEGroupBox.c

## 9.94 QEHistogram Class Reference

```
#include <QEHistogram.h>
```

### Public Types

- `typedef QVector< double > DataArray`

### Public Slots

- `void setColour (const int index, const QColor &value)`
- `void setValue (const int index, const double value)`
- `void setValues (const DataArray &values)`

### Signals

- `void mouseIndexChanged (const int index)`
- `void mouseIndexPressed (const int index, const Qt::MouseButton button)`

### Public Member Functions

- `QEHistogram (QWidget *parent=0)`
- `virtual QSize sizeHint () const`
- `int count () const`
- `int indexOfPosition (const int x, const int y) const`
- `int indexOfPosition (const QPoint &p) const`
- `QRect positionOfIndex (const int index) const`
- `void clearValue (const int index)`
- `void clearColour (const int index)`
- `void clear ()`
- `double value (const int index) const`
- `DataArray values () const`

### Protected Member Functions

- `bool eventFilter (QObject *obj, QEvent *event)`
- `void fontChange (const QFont &f)`

### Properties

- `bool autoBarGapWidths`
- `int barWidth`
- `int gap`
- `int margin`

- bool **autoScale**
- double **minimum**
- double **maximum**
- double **baseLine**
- bool **drawAixes**
- bool **showScale**
- int **precision**
- bool **showGrid**
- bool **logScale**
- QColor **backgroundColour**
- QColor **barColour**
- bool **drawBorder**
- Qt::Orientation **orientation**
- int **testSize**

### 9.94.1 Detailed Description

The [QEHistogram](#) class is a non-EPICS aware [histogram](#) widget. The value of, i.e. the length of each bar, and colour may be set independently.

### 9.94.2 Property Documentation

#### 9.94.2.1 Qt::Orientation QEHistogram::orientation [[read](#), [write](#)]

orientation horizontal (default) or vertical. Horizontal means each element displayed horizontally from left to right with the bar representing the value increasing vertically from bottom to top.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEHistogram.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEHistogram.c

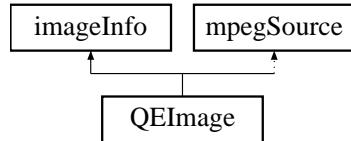
## 9.95 QEHistogramManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEHistogramManager.h

## 9.96 QEImage Class Reference

#include <QEImage.h> Inheritance diagram for QEImage::



### Public Types

- enum `selectOptions` {
   
    `SO_NONE, SO_PANNING, SO_VSLICE1, SO_VSLICE2,`
  
    `SO_VSLICE3, SO_VSLICE4, SO_VSLICE5, SO_HSLICE1,`
  
    `SO_HSLICE2, SO_HSLICE3, SO_HSLICE4, SO_HSLICE5,`
  
    `SO_AREA1, SO_AREA2, SO_AREA3, SO_AREA4,`
  
    `SO_PROFILE, SO_TARGET, SO_BEAM }`
- enum `imageUses` { `IMAGE_USE_DISPLAY, IMAGE_USE_SAVE,`
`IMAGE_USE_DISPLAY_AND_SAVE` }
- enum `resizeOptions` { `RESIZE_OPTION_ZOOM, RESIZE_OPTION_FIT` }
- enum `ellipseVariableDefinitions` { `BOUNding_RECTANGLE, CENTRE_-`
`AND_SIZE` }
- enum `UserLevels` { `User = userLevelTypes::USERLEVEL_USER,`
`Scientist = userLevelTypes::USERLEVEL_SCIENTIST, Engineer =`
`userLevelTypes::USERLEVEL_ENGINEER` }
- enum `DisplayAlarmStateOptions` { `Never = standardProperties::DISPLAY_-`
`ALARM_STATE_NEVER, Always = standardProperties::DISPLAY_-`
`ALARM_STATE_ALWAYS, WhenInAlarm = standardProperties::DISPLAY_-`
`ALARM_STATE_WHEN_IN_ALARM` }
- enum `FormatOptions` {
   
    `Mono = imageDataFormats::MONO, Bayer = imageDataFormats::BAYERRG,`
  
    `BayerGB = imageDataFormats::BAYERGB, BayerBG = imageDataFor-`
  
    `mats::BAYERBG,`
  
    `BayerGR = imageDataFormats::BAYERGR, BayerRG = imageDataFor-`
  
    `mats::BAYERRG, rgb1 = imageDataFormats::RGB1, rgb2 = imageDataFor-`
  
    `mats::RGB2,`
  
    `rgb3 = imageDataFormats::RGB3, yuv444 = imageDataFormats::YUV444,`
  
    `yuv422 = imageDataFormats::YUV422, yuv421 = imageDataFor-`
  
    `mats::YUV421 }`
- enum `EllipseVariableDefinitions` { `BoundingRectangle = BOUNDING_-`
`RECTANGLE, CenterAndSize = CENTRE_AND_SIZE` }
- enum `TargetOptions` { `DottedFullCrosshair = VideoWidget::CROSSHAIR1,`
`SolidSmallCrosshair = VideoWidget::CROSSHAIR2` }

- enum `ResizeOptions` { `Zoom` = QEImage::RESIZE\_OPTION\_ZOOM, `Fit` = QEImage::RESIZE\_OPTION\_FIT }
- enum `RotationOptions` { `NoRotation` = imageProperties::ROTATION\_0, `Rotate90Right` = imageProperties::ROTATION\_90\_RIGHT, `Rotate90Left` = imageProperties::ROTATION\_90\_LEFT, `Rotate180` = imageProperties::ROTATION\_180 }
- enum `ProgramStartupOptionNames` { `None` = applicationLauncher::PSO\_NONE, `Terminal` = applicationLauncher::PSO\_TERMINAL, `LogOutput` = applicationLauncher::PSO\_LOGOUTPUT, `StdOutput` = applicationLauncher::PSO\_STDOUPUT }

## Public Slots

- void `setImageFile` (QString name)
- void `setSelectPanMode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectVSliceMode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectHSliceMode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectArea1Mode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectArea2Mode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectArea3Mode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectArea4Mode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectProfileMode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectTargetMode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectBeamMode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectVSlice1Mode` ()

*Framework use only. Slot to allow external setting of selection menu options.*

- void `setSelectVSlice2Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectVSlice3Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectVSlice4Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectVSlice5Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectHSlice1Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectHSlice2Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectHSlice3Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectHSlice4Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `setSelectHSlice5Mode ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `pauseClicked ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `saveClicked ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `targetClicked ()`  
*Framework use only. Slot to allow external setting of selection menu options.*
- void `imageDisplayPropsDestroyed (QObject *)`  
*Framework use only. Slot to catch deletion of components (such as profile plots) that have been passed to the application for presentation.*
- void `vSliceDisplayDestroyed (QObject *)`  
*Framework use only. Slot to catch deletion of components (such as profile plots) that have been passed to the application for presentation.*
- void `hSliceDisplayDestroyed (QObject *)`

*Framework use only. Slot to catch deletion of components (such as profile plots) that have been passed to the application for presentation.*

- void **profileDisplayDestroyed** (QObject \*)

*Framework use only. Slot to catch deletion of components (such as profile plots) that have been passed to the application for presentation.*

- void **recorderDestroyed** (QObject \*)

*Framework use only. Slot to catch deletion of components (such as profile plots) that have been passed to the application for presentation.*

- void **showProfile** ()

*Show the arbitrary line (profile) markup - refer to [enableProfileSelection](#) property and [displayMarkups](#) property for details.*

- void **showProfile** (bool show)

*Show or hide the arbitrary line (profile) markup. Note that when hiding if its PV changes it will reshown unless [DisplayMarkups](#) has been set to off - refer to [enableProfileSelection](#) property and [displayMarkups](#) property for details.*

- void **hideProfile** ()

*Hide the arbitrary line (profile) markup but note that if its PV changes it will reshown unless [DisplayMarkups](#) has been set to off - refer to [enableProfileSelection](#) property and [displayMarkups](#) property for details.*

- void **showArea1** ()

*Show the area1 markup - refer to [enableArea1Selection](#) property and [displayMarkups](#) property for details.*

- void **showArea1** (bool show)

*Show or hide the area1 markup. Note that when hiding if its PV changes it will reshown unless [DisplayMarkups](#) has been set to off - refer to [enableArea1Selection](#) property and [displayMarkups](#) property for details*

- void **hideArea1** ()

*Hide the area1 markup but note that if its PV changes it will reshown unless [DisplayMarkups](#) has been set to off - refer to [enableArea1Selection](#) property and [displayMarkups](#) property for details.*

- void **setDisplayMarkupsOn** ()

*Set markup display to on to show all markups that change either due to user or PV activity, even if their [setDisplay????Selection](#) is off - refer to [displayMarkups](#) property for details.*

- void **setDisplayMarkupsOn** (bool on)

*Set markup display to on or off show all markups that change either due to user or PV activity, even if their [setDisplay????Selection](#) is off - refer to [displayMarkups](#) property for details.*

- void [setDisplayMarkupsOff \(\)](#)  
*Set markup display to off to stop PV controlled pvs from showing even if they change, unless their setDisplay????Selection is on - refer to displayMarkups property for details.*
- void [setManagedVisible \(bool v\)](#)

## Signals

- void [dbValueChanged \(const QString &out\)](#)
- void [requestResend \(\)](#)  
*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*
- void [componentHostRequest \(const QEActionRequests &request\)](#)

## Public Member Functions

- [QEImage \(QWidget \\*parent=0\)](#)
- [QEImage \(const QString &variableName, QWidget \\*parent=0\)](#)
- [~QEImage \(\)](#)  
*Destructor.*
- [selectOptions getSelectionOption \(\)](#)
- void [setBitDepth \(unsigned int bitDepthIn\)](#)  
*Access function for bitDepth property - refer to bitDepth property for details.*
- unsigned int [getBitDepth \(\)](#)  
*Access function for bitDepth property - refer to bitDepth property for details.*
- void [setFormatOption \(imageDataFormats::formatOptions formatOption\)](#)  
*Access function for formatOption property - refer to formatOption property for details.*
- imageDataFormats::formatOptions [getFormatOption \(\)](#)  
*Access function for formatOption property - refer to formatOption property for details.*
- void [setResizeOption \(resizeOptions resizeOptionIn\)](#)  
*Access function for resizeOption property - refer to resizeOption property for details.*
- [resizeOptions getResizeOption \(\)](#)  
*Access function for resizeOption property - refer to resizeOption property for details.*
- void [setZoom \(int zoomIn\)](#)  
*Access function for zoom property - refer to zoom property for details.*

- int `getZoom ()`  
*Access function for `zoom` property - refer to `zoom` property for details.*
- void `setXStretch` (double XStretchIn)  
*Access function for `XStretch` property - refer to `XStretch` property for details.*
- double `getXStretch ()`  
*Access function for `XStretch` property - refer to `XStretch` property for details.*
- void `setYStretch` (double YStretchIn)  
*Access function for `YStretch` property - refer to `YStretch` property for details.*
- double `getYStretch ()`  
*Access function for `YStretch` property - refer to `YStretch` property for details.*
- void `setRotation` (`imageProperties::rotationOptions` rotationIn)  
*Access function for `rotation` property - refer to `rotation` property for details.*
- `imageProperties::rotationOptions getRotation ()`  
*Access function for `rotation` property - refer to `rotation` property for details.*
- void `setHorizontalFlip` (bool flipHozIn)  
*Access function for `horizontalFlip` property - refer to `horizontalFlip` property for details.*
- bool `getHorizontalFlip ()`  
*Access function for `horizontalFlip` property - refer to `horizontalFlip` property for details.*
- void `setVerticalFlip` (bool flipVertIn)  
*Access function for `verticalFlip` property - refer to `verticalFlip` property for details.*
- bool `getVerticalFlip ()`  
*Access function for `verticalFlip` property - refer to `verticalFlip` property for details.*
- void `setInitialHozScrollPos` (int initialHosScrollPosIn)  
*Access function for `initialHosScrollPos` property - refer to `initialHosScrollPos` property for details.*
- int `getInitialHozScrollPos ()`  
*Access function for `initialHosScrollPos` property - refer to `initialHosScrollPos` property for details.*
- void `setInitialVertScrollPos` (int initialVertScrollPosIn)  
*Access function for `initialVertScrollPos` property - refer to `initialVertScrollPos` property for details.*

- int `getInitialVertScrollPos ()`  
*Access function for `initialVertScrollPos` property - refer to `initialVertScrollPos` property for details.*
- void `setDisplayButtonBar (bool displayButtonBarIn)`  
*Access function for `displayButtonBar` property - refer to `displayButtonBar` property for details.*
- bool `getDisplayButtonBar ()`  
*Access function for `displayButtonBar` property - refer to `displayButtonBar` property for details.*
- void `setShowTime (bool pValue)`  
*Access function for `showTime` property - refer to `showTime` property for details.*
- bool `getShowTime ()`  
*Access function for `showTime` property - refer to `showTime` property for details.*
- void `setUseFalseColour (bool pValue)`  
*Access function for `useFalseColour` property - refer to `useFalseColour` property for details.*
- bool `getUseFalseColour ()`  
*Access function for `useFalseColour` property - refer to `useFalseColour` property for details.*
- void `setVertSlice1MarkupColor (QColor pValue)`  
*Access function for `vertSliceColor` property - refer to `vertSliceColor` property for details.*
- QColor `getVertSlice1MarkupColor ()`  
*Access function for `vertSliceColor` property - refer to `vertSliceColor` property for details.*
- void `setVertSlice2MarkupColor (QColor pValue)`  
*Access function for `vertSlice2Color` property - refer to `vertSlice2Color` property for details.*
- QColor `getVertSlice2MarkupColor ()`  
*Access function for `vertSlice2Color` property - refer to `vertSlice2Color` property for details.*
- void `setVertSlice3MarkupColor (QColor pValue)`  
*Access function for `vertSlice3Color` property - refer to `vertSlice3Color` property for details.*
- QColor `getVertSlice3MarkupColor ()`

*Access function for `vertSlice3Color` property - refer to `vertSlice3Color` property for details.*

- void `setVertSlice4MarkupColor (QColor pValue)`

*Access function for `vertSlice4Color` property - refer to `vertSlice4Color` property for details.*

- QColor `getVertSlice4MarkupColor ()`

*Access function for `vertSlice4Color` property - refer to `vertSlice4Color` property for details.*

- void `setVertSlice5MarkupColor (QColor pValue)`

*Access function for `vertSlice5Color` property - refer to `vertSlice5Color` property for details.*

- QColor `getVertSlice5MarkupColor ()`

*Access function for `vertSlice5Color` property - refer to `vertSlice5Color` property for details.*

- void `setHozSlice1MarkupColor (QColor pValue)`

*Access function for `hozSliceColor` property - refer to `hozSliceColor` property for details.*

- QColor `getHozSlice1MarkupColor ()`

*Access function for `hozSliceColor` property - refer to `hozSliceColor` property for details.*

- void `setHozSlice2MarkupColor (QColor pValue)`

*Access function for `hozSlice2Color` property - refer to `hozSlice2Color` property for details.*

- QColor `getHozSlice2MarkupColor ()`

*Access function for `hozSlice2Color` property - refer to `hozSlice2Color` property for details.*

- void `setHozSlice3MarkupColor (QColor pValue)`

*Access function for `hozSlice3Color` property - refer to `hozSlice3Color` property for details.*

- QColor `getHozSlice3MarkupColor ()`

*Access function for `hozSlice3Color` property - refer to `hozSlice3Color` property for details.*

- void `setHozSlice4MarkupColor (QColor pValue)`

*Access function for `hozSlice4Color` property - refer to `hozSlice4Color` property for details.*

- QColor `getHozSlice4MarkupColor ()`

*Access function for `hozSlice4Color` property - refer to `hozSlice4Color` property for details.*

- void `setHozSlice5MarkupColor` (QColor pValue)

*Access function for `hozSlice5Color` property - refer to `hozSlice5Color` property for details.*

- QColor `getHozSlice5MarkupColor` ()

*Access function for `hozSlice5Color` property - refer to `hozSlice5Color` property for details.*

- void `setProfileMarkupColor` (QColor pValue)

*Access function for `profileColor` property - refer to `profileColor` property for details.*

- QColor `getProfileMarkupColor` ()

*Access function for `profileColor` property - refer to `profileColor` property for details.*

- void `setAreaMarkupColor` (QColor pValue)

*Access function for `areaColor` property - refer to `areaColor` property for details.*

- QColor `getAreaMarkupColor` ()

*Access function for `areaColor` property - refer to `areaColor` property for details.*

- void `setTargetMarkupColor` (QColor pValue)

*Access function for `targetColor` property - refer to `targetColor` property for details.*

- QColor `getTargetMarkupColor` ()

*Access function for `targetColor` property - refer to `targetColor` property for details.*

- void `setBeamMarkupColor` (QColor pValue)

*Access function for `beamColor` property - refer to `beamColor` property for details.*

- QColor `getBeamMarkupColor` ()

*Access function for `beamColor` property - refer to `beamColor` property for details.*

- void `setTimeMarkupColor` (QColor pValue)

*Access function for `timeColor` property - refer to `timeColor` property for details.*

- QColor `getTimeMarkupColor` ()

*Access function for `timeColor` property - refer to `timeColor` property for details.*

- void `setEllipseMarkupColor` (QColor markupColor)

*Access function for `ellipseColor` property - refer to `ellipseColor` property for details.*

- QColor `getEllipseMarkupColor` ()

*Access function for `ellipseColor` property - refer to `ellipseColor` property for details.*

- void **setDisplayCursorPixelInfo** (bool displayCursorPixelInfo)  
*Access function for `displayCursorPixelInfo` property - refer to `displayCursorPixelInfo` property for details.*
- bool **getDisplayCursorPixelInfo** ()  
*Access function for `displayCursorPixelInfo` property - refer to `displayCursorPixelInfo` property for details.*
- void **setContrastReversal** (bool contrastReversalIn)  
*Access function for `contrastReversal` property - refer to `contrastReversal` property for details.*
- bool **getContrastReversal** ()  
*Access function for `contrastReversal` property - refer to `contrastReversal` property for details.*
- void **setLog** (bool log)  
*Access function for `logBrightness` property - refer to `logBrightness` property for details.*
- bool **getLog** ()  
*Access function for `logBrightness` property - refer to `logBrightness` property for details.*
- void **setEnableVertSlice1Selection** (bool enableVSliceSelection)  
*Access function for `enableVertSlice1Selection` property - refer to `enableVertSlice1Selection` property for details.*
- bool **getEnableVertSlice1Selection** ()  
*Access function for `enableVertSlice1Selection` property - refer to `enableVertSlice1Selection` property for details.*
- void **setEnableVertSlice2Selection** (bool enableVSliceSelection)  
*Access function for `enableVertSlice2Selection` property - refer to `enableVertSlice2Selection` property for details.*
- bool **getEnableVertSlice2Selection** ()  
*Access function for `enableVertSlice2Selection` property - refer to `enableVertSlice2Selection` property for details.*
- void **setEnableVertSlice3Selection** (bool enableVSliceSelection)  
*Access function for `enableVertSlice3Selection` property - refer to `enableVertSlice3Selection` property for details.*
- bool **getEnableVertSlice3Selection** ()  
*Access function for `enableVertSlice3Selection` property - refer to `enableVertSlice3Selection` property for details.*

- void `setEnableVertSlice4Selection` (bool enableVSliceSelection)  
*Access function for `enableVertSlice4Selection` property - refer to `enableVertSlice4Selection` property for details.*
- bool `getEnableVertSlice4Selection` ()  
*Access function for `enableVertSlice4Selection` property - refer to `enableVertSlice4Selection` property for details.*
- void `setEnableVertSlice5Selection` (bool enableVSliceSelection)  
*Access function for `enableVertSlice5Selection` property - refer to `enableVertSlice5Selection` property for details.*
- bool `getEnableVertSlice5Selection` ()  
*Access function for `enableVertSlice5Selection` property - refer to `enableVertSlice5Selection` property for details.*
- void `setEnableHozSlice1Selection` (bool enableHSliceSelection)  
*Access function for `enableHozSlice1Selection` property - refer to `enableHozSlice1Selection` property for details.*
- bool `getEnableHozSlice1Selection` ()  
*Access function for `enableHozSlice1Selection` property - refer to `enableHozSlice1Selection` property for details.*
- void `setEnableHozSlice2Selection` (bool enableHSliceSelection)  
*Access function for `enableHozSlice2Selection` property - refer to `enableHozSlice2Selection` property for details.*
- bool `getEnableHozSlice2Selection` ()  
*Access function for `enableHozSlice2Selection` property - refer to `enableHozSlice2Selection` property for details.*
- void `setEnableHozSlice3Selection` (bool enableHSliceSelection)  
*Access function for `enableHozSlice3Selection` property - refer to `enableHozSlice3Selection` property for details.*
- bool `getEnableHozSlice3Selection` ()  
*Access function for `enableHozSlice3Selection` property - refer to `enableHozSlice3Selection` property for details.*
- void `setEnableHozSlice4Selection` (bool enableHSliceSelection)  
*Access function for `enableHozSlice4Selection` property - refer to `enableHozSlice4Selection` property for details.*
- bool `getEnableHozSlice4Selection` ()  
*Access function for `enableHozSlice4Selection` property - refer to `enableHozSlice4Selection` property for details.*

- void **setEnableHozSlice5Selection** (bool enableHSliceSelection)  
*Access function for `enableHozSlice5Selection` property - refer to `enableHozSlice5Selection` property for details.*
- bool **getEnableHozSlice5Selection** ()  
*Access function for `enableHozSlice5Selection` property - refer to `enableHozSlice5Selection` property for details.*
- void **setEnableArea1Selection** (bool enableAreaSelectionIn)  
*Access function for `enableArea1Selection` property - refer to `enableArea1Selection` property for details.*
- bool **getEnableArea1Selection** ()  
*Access function for `enableArea1Selection` property - refer to `enableArea1Selection` property for details.*
- void **setEnableArea2Selection** (bool enableAreaSelectionIn)  
*Access function for `enableArea2Selection` property - refer to `enableArea2Selection` property for details.*
- bool **getEnableArea2Selection** ()  
*Access function for `enableArea2Selection` property - refer to `enableArea2Selection` property for details.*
- void **setEnableArea3Selection** (bool enableAreaSelectionIn)  
*Access function for `enableArea3Selection` property - refer to `enableArea3Selection` property for details.*
- bool **getEnableArea3Selection** ()  
*Access function for `enableArea3Selection` property - refer to `enableArea3Selection` property for details.*
- void **setEnableArea4Selection** (bool enableAreaSelectionIn)  
*Access function for `enableArea4Selection` property - refer to `enableArea4Selection` property for details.*
- bool **getEnableArea4Selection** ()  
*Access function for `enableArea4Selection` property - refer to `enableArea4Selection` property for details.*
- void **setEnableProfileSelection** (bool enableProfileSelectionIn)  
*Access function for `enableProfileSelection` property - refer to `enableProfileSelection` property for details.*
- bool **getEnableProfileSelection** ()  
*Access function for `enableProfileSelection` property - refer to `enableProfileSelection` property for details.*

- void `setEnableTargetSelection` (bool enableTargetSelectionIn)  
*Access function for `enableTargetSelection` property - refer to `enableTargetSelection` property for details.*
- bool `getEnableTargetSelection` ()  
*Access function for `enableTargetSelection` property - refer to `enableTargetSelection` property for details.*
- void `setEnableBeamSelection` (bool enableBeamSelectionIn)  
*Access function for `enableBeamSelection` property - refer to `enableBeamSelection` property for details.*
- bool `getEnableBeamSelection` ()  
*Access function for `enableBeamSelection` property - refer to `enableBeamSelection` property for details.*
- void `setEnableImageDisplayProperties` (bool enableImageDisplayPropertiesIn)  
*Access function for `enableImageDisplayProperties` property - refer to `enableImageDisplayProperties` property for details.*
- bool `getEnableImageDisplayProperties` ()  
*Access function for `enableImageDisplayProperties` property - refer to `enableImageDisplayProperties` property for details.*
- void `setEnableRecording` (bool enableRecordingIn)  
*Access function for `enableRecording` property - refer to `enableRecording` property for details.*
- bool `getEnableRecording` ()  
*Access function for `enableRecording` property - refer to `enableRecording` property for details.*
- void `setAutoBrightnessContrast` (bool autoBrightnessContrastIn)  
*Access function for `autoBrightnessContrast` property - refer to `autoBrightnessContrast` property for details.*
- bool `getAutoBrightnessContrast` ()  
*Access function for `autoBrightnessContrast` property - refer to `autoBrightnessContrast` property for details.*
- void `setExternalControls` (bool externalControlsIn)  
*Access function for `externalControls` property - refer to `externalControls` property for details.*
- bool `getExternalControls` ()  
*Access function for `externalControls` property - refer to `externalControls` property for details.*

- void [setFullContextMenu](#) (bool fullContextMenuIn)  
*Access function for fullContextMenu property - refer to fullContextMenu property for details.*
- bool [getFullContextMenu](#) ()  
*Access function for fullContextMenu property - refer to fullContextMenu property for details.*
- void [setEnableProfilePresentation](#) (bool enableProfilePresentationIn)  
*Access function for enableProfilePresentation property - refer to enableProfilePresentation property for details.*
- bool [getEnableProfilePresentation](#) ()  
*Access function for enableProfilePresentation property - refer to enableProfilePresentation property for details.*
- void [setEnableHozSlicePresentation](#) (bool enableHozSlicePresentationIn)  
*Access function for enableHozSlicePresentation property - refer to enableHozSlicePresentation property for details.*
- bool [getEnableHozSlicePresentation](#) ()  
*Access function for enableHozSlicePresentation property - refer to enableHozSlicePresentation property for details.*
- void [setEnableVertSlicePresentation](#) (bool enableVertSlicePresentationIn)  
*Access function for enableVertSlicePresentation property - refer to enableVertSlicePresentation property for details.*
- bool [getEnableVertSlicePresentation](#) ()  
*Access function for enableVertSlicePresentation property - refer to enableVertSlicePresentation property for details.*
- void [setDisplayVertSlice1Selection](#) (bool displayVSliceSelection)  
*Access function for displayVertSlice1Selection property - refer to displayVertSlice1Selection property for details.*
- bool [getDisplayVertSlice1Selection](#) ()  
*Access function for displayVertSlice1Selection property - refer to displayVertSlice1Selection property for details.*
- void [setDisplayVertSlice2Selection](#) (bool displayVSliceSelection)  
*Access function for displayVertSlice2Selection property - refer to displayVertSlice2Selection property for details.*
- bool [getDisplayVertSlice2Selection](#) ()  
*Access function for displayVertSlice2Selection property - refer to displayVertSlice2Selection property for details.*

- void `setDisplayVertSlice3Selection` (bool displayVSliceSelection)  
*Access function for `displayVertSlice3Selection` property - refer to `displayVertSlice3Selection` property for details.*
- bool `getDisplayVertSlice3Selection` ()  
*Access function for `displayVertSlice3Selection` property - refer to `displayVertSlice3Selection` property for details.*
- void `setDisplayVertSlice4Selection` (bool displayVSliceSelection)  
*Access function for `displayVertSlice4Selection` property - refer to `displayVertSlice4Selection` property for details.*
- bool `getDisplayVertSlice4Selection` ()  
*Access function for `displayVertSlice4Selection` property - refer to `displayVertSlice4Selection` property for details.*
- void `setDisplayVertSlice5Selection` (bool displayVSliceSelection)  
*Access function for `displayVertSlice5Selection` property - refer to `displayVertSlice5Selection` property for details.*
- bool `getDisplayVertSlice5Selection` ()  
*Access function for `displayVertSlice5Selection` property - refer to `displayVertSlice5Selection` property for details.*
- void `setDisplayHozSlice1Selection` (bool displayHSliceSelection)  
*Access function for `displayHozSlice1Selection` property - refer to `displayHozSlice1Selection` property for details.*
- bool `getDisplayHozSlice1Selection` ()  
*Access function for `displayHozSlice1Selection` property - refer to `displayHozSlice1Selection` property for details.*
- void `setDisplayHozSlice2Selection` (bool displayHSliceSelection)  
*Access function for `displayHozSlice2Selection` property - refer to `displayHozSlice2Selection` property for details.*
- bool `getDisplayHozSlice2Selection` ()  
*Access function for `displayHozSlice2Selection` property - refer to `displayHozSlice2Selection` property for details.*
- void `setDisplayHozSlice3Selection` (bool displayHSliceSelection)  
*Access function for `displayHozSlice3Selection` property - refer to `displayHozSlice3Selection` property for details.*
- bool `getDisplayHozSlice3Selection` ()  
*Access function for `displayHozSlice3Selection` property - refer to `displayHozSlice3Selection` property for details.*

- void [`setDisplayHozSlice4Selection`](#) (bool displayHSliceSelection)  
*Access function for `displayHozSlice4Selection` property - refer to `displayHozSlice4Selection` property for details.*
- bool [`getDisplayHozSlice4Selection`](#) ()  
*Access function for `displayHozSlice4Selection` property - refer to `displayHozSlice4Selection` property for details.*
- void [`setDisplayHozSlice5Selection`](#) (bool displayHSliceSelection)  
*Access function for `displayHozSlice5Selection` property - refer to `displayHozSlice5Selection` property for details.*
- bool [`getDisplayHozSlice5Selection`](#) ()  
*Access function for `displayHozSlice5Selection` property - refer to `displayHozSlice5Selection` property for details.*
- void [`setDisplayArea1Selection`](#) (bool displayAreaSelection)  
*Access function for `displayArea1Selection` property - refer to `displayArea1Selection` property for details.*
- bool [`getDisplayArea1Selection`](#) ()  
*Access function for `displayArea1Selection` property - refer to `displayArea1Selection` property for details.*
- void [`setDisplayArea2Selection`](#) (bool displayAreaSelection)  
*Access function for `displayArea2Selection` property - refer to `displayArea2Selection` property for details.*
- bool [`getDisplayArea2Selection`](#) ()  
*Access function for `displayArea2Selection` property - refer to `displayArea2Selection` property for details.*
- void [`setDisplayArea3Selection`](#) (bool displayAreaSelection)  
*Access function for `displayArea3Selection` property - refer to `displayArea3Selection` property for details.*
- bool [`getDisplayArea3Selection`](#) ()  
*Access function for `displayArea3Selection` property - refer to `displayArea3Selection` property for details.*
- void [`setDisplayArea4Selection`](#) (bool displayAreaSelection)  
*Access function for `displayArea4Selection` property - refer to `displayArea4Selection` property for details.*
- bool [`getDisplayArea4Selection`](#) ()  
*Access function for `displayArea4Selection` property - refer to `displayArea4Selection` property for details.*

- void **setDisplayProfileSelection** (bool displayProfileSelection)  
*Access function for `displayProfileSelection` property - refer to `displayProfileSelection` property for details.*
- bool **getDisplayProfileSelection** ()  
*Access function for `displayProfileSelection` property - refer to `displayProfileSelection` property for details.*
- void **setDisplayTargetSelection** (bool displayTargetSelection)  
*Access function for `displayTargetSelection` property - refer to `displayTargetSelection` property for details.*
- bool **getDisplayTargetSelection** ()  
*Access function for `displayTargetSelection` property - refer to `displayTargetSelection` property for details.*
- void **setDisplayBeamSelection** (bool displayBeamSelection)  
*Access function for `displayBeamSelection` property - refer to `displayBeamSelection` property for details.*
- bool **getDisplayBeamSelection** ()  
*Access function for `displayBeamSelection` property - refer to `displayBeamSelection` property for details.*
- void **setDisplayEllipse** (bool displayEllipse)  
*Access function for `displayEllipse` property - refer to `displayEllipse` property for details.*
- bool **getDisplayEllipse** ()  
*Access function for `displayEllipse` property - refer to `displayEllipse` property for details.*
- **ellipseVariableDefinitions getEllipseVariableDefinition** ()  
*Access function for `ellipseVariableDefinition` property - refer to `ellipseVariableDefinition` property for details.*
- void **setEllipseVariableDefinition** (ellipseVariableDefinitions def)  
*Access function for `ellipseVariableDefinition` property - refer to `ellipseVariableDefinition` property for details.*
- void **setDisplayMarkups** (bool displayMarkupsIn)  
*Access function for `displayMarkups` property - refer to `displayMarkups` property for details.*
- bool **getDisplayMarkups** ()  
*Access function for `displayMarkups` property - refer to `displayMarkups` property for details.*

- void `setName` (QString nameIn)  
*Access function for name property - refer to name property for details.*
- QString `getName` ()  
*Access function for name property - refer to name property for details.*
- void `setProgram1` (QString program)  
*Access function for program1 property - refer to program1 property for details.*
- QString `getProgram1` ()  
*Access function for program1 property - refer to program1 property for details.*
- void `setProgram2` (QString program)  
*Access function for program2 property - refer to program2 property for details.*
- QString `getProgram2` ()  
*Access function for program2 property - refer to program2 property for details.*
- void `setArguments1` (QStringList arguments)  
*Access function for arguments1 property - refer to arguments1 property for details.*
- QStringList `getArguments1` ()  
*Access function for arguments1 property - refer to arguments1 property for details.*
- void `setArguments2` (QStringList arguments)  
*Access function for arguments2 property - refer to arguments2 property for details.*
- QStringList `getArguments2` ()  
*Access function for arguments2 property - refer to arguments2 property for details.*
- void `setProgramStartupOption1` (applicationLauncher::programStartupOptions programStartupOption)  
*Access function for programStartupOption1 property - refer to programStartupOption1 property for details.*
- applicationLauncher::programStartupOptions `getProgramStartupOption1` ()  
*Access function for programStartupOption1 property - refer to programStartupOption1 property for details.*
- void `setProgramStartupOption2` (applicationLauncher::programStartupOptions programStartupOption)  
*Access function for programStartupOption2 property - refer to programStartupOption2 property for details.*
- applicationLauncher::programStartupOptions `getProgramStartupOption2` ()  
*Access function for programStartupOption2 property - refer to programStartupOption2 property for details.*

- **QString getHozSlice1Legend ()**  
*Access function for `hozSlice1Legend` property - refer to `hozSlice1Legend` property for details.*
- **void setHozSlice1Legend (QString legend)**  
*Access function for `hozSlice1Legend` property - refer to `hozSlice1Legend` property for details.*
- **QString getHozSlice2Legend ()**  
*Access function for `hozSlice2Legend` property - refer to `hozSlice2Legend` property for details.*
- **void setHozSlice2Legend (QString legend)**  
*Access function for `hozSlice2Legend` property - refer to `hozSlice2Legend` property for details.*
- **QString getHozSlice3Legend ()**  
*Access function for `hozSlice3Legend` property - refer to `hozSlice3Legend` property for details.*
- **void setHozSlice3Legend (QString legend)**  
*Access function for `hozSlice3Legend` property - refer to `hozSlice3Legend` property for details.*
- **QString getHozSlice4Legend ()**  
*Access function for `hozSlice4Legend` property - refer to `hozSlice4Legend` property for details.*
- **void setHozSlice4Legend (QString legend)**  
*Access function for `hozSlice4Legend` property - refer to `hozSlice4Legend` property for details.*
- **QString getHozSlice5Legend ()**  
*Access function for `hozSlice5Legend` property - refer to `hozSlice5Legend` property for details.*
- **void setHozSlice5Legend (QString legend)**  
*Access function for `hozSlice5Legend` property - refer to `hozSlice5Legend` property for details.*
- **QString getVertSlice1Legend ()**  
*Access function for `vertSlice1Legend` property - refer to `vertSlice1Legend` property for details.*
- **void setVertSlice1Legend (QString legend)**  
*Access function for `vertSlice1Legend` property - refer to `vertSlice1Legend` property for details.*

- `QString getVertSlice2Legend ()`  
*Access function for `vertSlice2Legend` property - refer to `vertSlice2Legend` property for details.*
- `void setVertSlice2Legend (QString legend)`  
*Access function for `vertSlice2Legend` property - refer to `vertSlice2Legend` property for details.*
- `QString getVertSlice3Legend ()`  
*Access function for `vertSlice3Legend` property - refer to `vertSlice3Legend` property for details.*
- `void setVertSlice3Legend (QString legend)`  
*Access function for `vertSlice3Legend` property - refer to `vertSlice3Legend` property for details.*
- `QString getVertSlice4Legend ()`  
*Access function for `vertSlice4Legend` property - refer to `vertSlice4Legend` property for details.*
- `void setVertSlice4Legend (QString legend)`  
*Access function for `vertSlice4Legend` property - refer to `vertSlice4Legend` property for details.*
- `QString getVertSlice5Legend ()`  
*Access function for `vertSlice5Legend` property - refer to `vertSlice5Legend` property for details.*
- `void setVertSlice5Legend (QString legend)`  
*Access function for `vertSlice5Legend` property - refer to `vertSlice5Legend` property for details.*
- `QString getprofileLegend ()`  
*Access function for `profileLegend` property - refer to `profileLegend` property for details.*
- `void setProfileLegend (QString legend)`  
*Access function for `profileLegend` property - refer to `profileLegend` property for details.*
- `QString getAreaSelection1Legend ()`  
*Access function for `areaSelection1Legend` property - refer to `areaSelection1Legend` property for details.*
- `void setAreaSelection1Legend (QString legend)`  
*Access function for `areaSelection1Legend` property - refer to `areaSelection1Legend` property for details.*

- **QString getAreaSelection2Legend ()**  
*Access function for `areaSelection2Legend` property - refer to `areaSelection2Legend` property for details.*
- **void setAreaSelection2Legend (QString legend)**  
*Access function for `areaSelection2Legend` property - refer to `areaSelection2Legend` property for details.*
- **QString getAreaSelection3Legend ()**  
*Access function for `areaSelection3Legend` property - refer to `areaSelection3Legend` property for details.*
- **void setAreaSelection3Legend (QString legend)**  
*Access function for `areaSelection3Legend` property - refer to `areaSelection3Legend` property for details.*
- **QString getAreaSelection4Legend ()**  
*Access function for `areaSelection4Legend` property - refer to `areaSelection4Legend` property for details.*
- **void setAreaSelection4Legend (QString legend)**  
*Access function for `areaSelection4Legend` property - refer to `areaSelection4Legend` property for details.*
- **QString getTargetLegend ()**  
*Access function for `targetLegend` property - refer to `targetLegend` property for details.*
- **void setTargetLegend (QString legend)**  
*Access function for `targetLegend` property - refer to `targetLegend` property for details.*
- **QString getBeamLegend ()**  
*Access function for `beamLegend` property - refer to `beamLegend` property for details.*
- **void setBeamLegend (QString legend)**  
*Access function for `beamLegend` property - refer to `beamLegend` property for details.*
- **QString getEllipseLegend ()**  
*Access function for `ellipseLegend` property - refer to `ellipseLegend` property for details.*
- **void setEllipseLegend (QString legend)**  
*Access function for `ellipseLegend` property - refer to `ellipseLegend` property for details.*
- **bool getFullScreen ()**  
*Access function for `fullScreen` property - refer to `fullScreen` property for details.*

- void [setFullScreen](#) (bool fullScreenIn)  
*Access function for fullScreen property - refer to fullScreen property for details.*
- void [setSubstitutedUrl](#) (QString urlIn)  
*Access function for URL property - refer to URL property for details.*
- QString [getSubstitutedUrl](#) ()  
*Access function for URL property - refer to URL property for details.*
- void [setVariableNameSubstitutionsProperty](#) (QString variableNameSubstitutions)  
*Property access function for variableSubstitutions property. This has special behaviour to work well within designer.*
- QString [getVariableNameSubstitutionsProperty](#) ()  
*Property access function for variableSubstitutions property. This has special behaviour to work well within designer.*
- UserLevels [getUserLevelVisibilityProperty](#) ()  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- void [setUserLevelVisibilityProperty](#) (UserLevels level)  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- UserLevels [getUserLevelEnabledProperty](#) ()  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- void [setUserLevelEnabledProperty](#) (UserLevels level)  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- DisplayAlarmStateOptions [getDisplayAlarmStateOptionProperty](#) ()  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void [setDisplayAlarmStateOptionProperty](#) (DisplayAlarmStateOptions option)  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void [setFormatOptionProperty](#) (FormatOptions formatOption)  
*Access function for formatOption property - refer to formatOption property for details.*
- FormatOptions [getFormatOptionProperty](#) ()

*Access function for `formatOption` property - refer to `formatOption` property for details.*

- void `setBitDepthProperty` (unsigned int bitDepth)

*Access function for `bitDepth` property - refer to `bitDepth` property for details.*

- unsigned int `getBitDepthProperty` ()

*Access function for `bitDepth` property - refer to `bitDepth` property for details.*

- `EllipseVariableDefinitions getEllipseVariableDefinitionProperty` ()

*Access function for `EllipseVariableDefinition` property - refer to `EllipseVariableDefinition` property for details.*

- void `setEllipseVariableDefinitionProperty` (`EllipseVariableDefinitions` variableUsage)

*Access function for `EllipseVariableDefinitions` property - refer to `EllipseVariableDefinitions` property for details.*

- `TargetOptions getTargetOptionProperty` ()

*Access function for `targetOption` property - refer to `targetOption` property for details.*

- void `setTargetOptionProperty` (`TargetOptions` option)

*Access function for `targetOption` property - refer to `targetOption` property for details.*

- `TargetOptions getBeamOptionProperty` ()

*Access function for `beamOption` property - refer to `beamOption` property for details.*

- void `setBeamOptionProperty` (`TargetOptions` option)

*Access function for `beamOption` property - refer to `beamOption` property for details.*

- void `setResizeOptionProperty` (`ResizeOptions` resizeOption)

*Access function for `resizeOption` property - refer to `resizeOption` property for details.*

- `ResizeOptions getResizeOptionProperty` ()

*Access function for `resizeOption` property - refer to `resizeOption` property for details.*

- void `setRotationProperty` (`RotationOptions` rotation)

*Access function for `rotation` property - refer to `rotation` property for details.*

- `RotationOptions getRotationProperty` ()

*Access function for `rotation` property - refer to `rotation` property for details.*

- void `setProgramStartupOptionProperty1` (`ProgramStartupOptionNames` programStartupOption)

*Access function for `ProgramStartupOptionNames1` property - refer to `ProgramStartupOptionNames1` property for details.*

- `ProgramStartupOptionNames getProgramStartupOptionProperty1 ()`  
*Access function for ProgramStartupOptionNames1 property - refer to ProgramStartupOptionNames1 property for details.*
- `void setProgramStartupOptionProperty2 (ProgramStartupOptionNames programStartupOption)`  
*Access function for ProgramStartupOptionNames2 property - refer to ProgramStartupOptionNames2 property for details.*
- `ProgramStartupOptionNames getProgramStartupOptionProperty2 ()`  
*Access function for ProgramStartupOptionNames2 property - refer to ProgramStartupOptionNames2 property for details.*

## Protected Types

- enum `variableIndexes {`  
`IMAGE_VARIABLE, FORMAT_VARIABLE, BIT_DEPTH_VARIABLE,`  
`DATA_TYPE_VARIABLE,`  
`WIDTH_VARIABLE, HEIGHT_VARIABLE, NUM_DIMENSIONS_-`  
`VARIABLE, DIMENSION_0_VARIABLE,`  
`DIMENSION_1_VARIABLE, DIMENSION_2_VARIABLE, ROI1_X_-`  
`VARIABLE, ROI1_Y_VARIABLE,`  
`ROI1_W_VARIABLE, ROI1_H_VARIABLE, ROI2_X_VARIABLE,`  
`ROI2_Y_VARIABLE,`  
`ROI2_W_VARIABLE, ROI2_H_VARIABLE, ROI3_X_VARIABLE,`  
`ROI3_Y_VARIABLE,`  
`ROI3_W_VARIABLE, ROI3_H_VARIABLE, ROI4_X_VARIABLE,`  
`ROI4_Y_VARIABLE,`  
`ROI4_W_VARIABLE, ROI4_H_VARIABLE, TARGET_X_VARIABLE,`  
`TARGET_Y_VARIABLE,`  
`BEAM_X_VARIABLE, BEAM_Y_VARIABLE, TARGET_TRIGGER_-`  
`VARIABLE, CLIPPING_ONOFF_VARIABLE,`  
`CLIPPING_LOW_VARIABLE, CLIPPING_HIGH_VARIABLE,`  
`PROFILE_H1_VARIABLE, PROFILE_H1_THICKNESS_VARIABLE,`  
`PROFILE_H2_VARIABLE, PROFILE_H2_THICKNESS_VARIABLE,`  
`PROFILE_H3_VARIABLE, PROFILE_H3_THICKNESS_VARIABLE,`  
`PROFILE_H4_VARIABLE, PROFILE_H4_THICKNESS_VARIABLE,`  
`PROFILE_H5_VARIABLE, PROFILE_H5_THICKNESS_VARIABLE,`  
`PROFILE_V1_VARIABLE, PROFILE_V1_THICKNESS_VARIABLE,`  
`PROFILE_V2_VARIABLE, PROFILE_V2_THICKNESS_VARIABLE,`  
`PROFILE_V3_VARIABLE, PROFILE_V3_THICKNESS_VARIABLE,`  
`PROFILE_V4_VARIABLE, PROFILE_V4_THICKNESS_VARIABLE,`

```

PROFILE_V5_VARIABLE, PROFILE_V5_THICKNESS_VARIABLE,
LINE_PROFILE_X1_VARIABLE, LINE_PROFILE_Y1_VARIABLE,
LINE_PROFILE_X2_VARIABLE, LINE_PROFILE_Y2_VARIABLE,
LINE_PROFILE_THICKNESS_VARIABLE, PROFILE_H_ARRAY,
PROFILE_V_ARRAY, PROFILE_LINE_ARRAY, ELLIPSE_X-
VARIABLE, ELLIPSE_Y_VARIABLE,
ELLIPSE_W_VARIABLE, ELLIPSE_H_VARIABLE, QEIMAGE_-
NUM_VARIABLES }

```

## Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **redisplayAllMarkups** ()
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant v)
- void **resizeEvent** (QResizeEvent \*)

## Protected Attributes

- QEStringFormatting **stringFormatting**
- QEIntegerFormatting **integerFormatting**
- QEFloatingFormatting **floatingFormatting**
- resizeOptions **resizeOption**
- int **zoom**  
*Zoom percentage. Used when resizeOption is Zoom.*
- double **XStretch**  
*Stretch X factor. Used when generating canvas in which fully processed image is presented.*
- double **YStretch**  
*Stretch Y factor. Used when generating canvas in which fully processed image is presented.*
- int **initialHozScrollPos**
- int **initialVertScrollPos**
- bool **displayButtonBar**

## Properties

- `QString imageVariable`
- `QString formatVariable`
- `QString bitDepthVariable`
- `QString dataTypeVariable`
- `QString widthVariable`
- `QString heightVariable`
- `QString dimensionsVariable`
- `QString dimension1Variable`
- `QString dimension2Variable`
- `QString dimension3Variable`
- `QString regionOfInterest1XVariable`
- `QString regionOfInterest1YVariable`
- `QString regionOfInterest1WVariable`
- `QString regionOfInterest1HVariable`
- `QString regionOfInterest2XVariable`
- `QString regionOfInterest2YVariable`
- `QString regionOfInterest2WVariable`
- `QString regionOfInterest2HVariable`
- `QString regionOfInterest3XVariable`
- `QString regionOfInterest3YVariable`
- `QString regionOfInterest3WVariable`
- `QString regionOfInterest3HVariable`
- `QString regionOfInterest4XVariable`
- `QString regionOfInterest4YVariable`
- `QString regionOfInterest4WVariable`
- `QString regionOfInterest4HVariable`
- `QString targetXVariable`
- `QString targetYVariable`
- `QString beamXVariable`
- `QString beamYVariable`
- `QString targetTriggerVariable`
- `QString clippingOnOffVariable`
- `QString clippingLowVariable`
- `QString clippingHighVariable`
- `QString profileHozVariable`
- `QString profileHoz1Variable`
- `QString profileHozThicknessVariable`
- `QString profileHoz1ThicknessVariable`
- `QString profileHoz2Variable`
- `QString profileHoz2ThicknessVariable`
- `QString profileHoz3Variable`
- `QString profileHoz3ThicknessVariable`
- `QString profileHoz4Variable`
- `QString profileHoz4ThicknessVariable`
- `QString profileHoz5Variable`

- `QString profileHoz5ThicknessVariable`
- `QString profileVertVariable`
- `QString profileVert1Variable`
- `QString profileVertThicknessVariable`
- `QString profileVert1ThicknessVariable`
- `QString profileVert2Variable`
- `QString profileVert2ThicknessVariable`
- `QString profileVert3Variable`
- `QString profileVert3ThicknessVariable`
- `QString profileVert4Variable`
- `QString profileVert4ThicknessVariable`
- `QString profileVert5Variable`
- `QString profileVert5ThicknessVariable`
- `QString lineProfileX1Variable`
- `QString lineProfileY1Variable`
- `QString lineProfileX2Variable`
- `QString lineProfileY2Variable`
- `QString lineProfileThicknessVariable`
- `QString profileHozArrayVariable`
- `QString profileVertArrayVariable`
- `QString lineProfileArrayVariable`
- `QString ellipseXVariable`
- `QString ellipseYVariable`
- `QString ellipseWVariable`
- `QString ellipseHVariable`
- `QString variableSubstitutions`
- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- `FormatOptions formatOption`
- `bool enableVertSliceSelection`
- `bool enableVertSlice1Selection`
- `bool enableVertSlice2Selection`
- `bool enableVertSlice3Selection`
- `bool enableVertSlice4Selection`
- `bool enableVertSlice5Selection`

- bool **enableHozSliceSelection**
- bool **enableHozSlice1Selection**
- bool **enableHozSlice2Selection**
- bool **enableHozSlice3Selection**
- bool **enableHozSlice4Selection**
- bool **enableHozSlice5Selection**
- bool **enableProfileSelection**
- bool **enableArea1Selection**
- bool **enableArea2Selection**
- bool **enableArea3Selection**
- bool **enableArea4Selection**
- bool **enableTargetSelection**
- bool **enableBeamSelection**
- QString **hozSliceLegend**
- QString **hozSlice1Legend**

*Name of horizontal slice 1 markup.*

- QString **hozSlice2Legend**

*Name of horizontal slice 2 markup.*

- QString **hozSlice3Legend**

*Name of horizontal slice 3 markup.*

- QString **hozSlice4Legend**

*Name of horizontal slice 4 markup.*

- QString **hozSlice5Legend**

*Name of horizontal slice 5 markup.*

- QString **vertSliceLegend**

- QString **vertSlice1Legend**

*Name of vertical slice 1 markup.*

- QString **vertSlice2Legend**

*Name of vertical slice 2 markup.*

- QString **vertSlice3Legend**

*Name of vertical slice 3 markup.*

- QString **vertSlice4Legend**

*Name of vertical slice 4 markup.*

- QString **vertSlice5Legend**

*Name of vertical slice 5 markup.*

- QString **profileLegend**

*Name of arbitrary profile markup.*

- **QString areaSelection1Legend**  
*Name of area selection 1 markup.*
- **QString areaSelection2Legend**  
*Name of area selection 2 markup.*
- **QString areaSelection3Legend**  
*Name of area selection 3 markup.*
- **QString areaSelection4Legend**  
*Name of area selection 4 markup.*
- **QString targetLegend**  
*Name of target markup.*
- **QString beamLegend**  
*Name of beam markup.*
- **QString ellipseLegend**  
*Name of ellipse markup.*
- **bool displayVertSliceSelection**
- **bool displayVertSlice1Selection**
- **bool displayVertSlice2Selection**
- **bool displayVertSlice3Selection**
- **bool displayVertSlice4Selection**
- **bool displayVertSlice5Selection**
- **bool displayHozSliceSelection**
- **bool displayHozSlice1Selection**
- **bool displayHozSlice2Selection**
- **bool displayHozSlice3Selection**
- **bool displayHozSlice4Selection**
- **bool displayHozSlice5Selection**
- **bool displayProfileSelection**
- **bool displayArea1Selection**
- **bool displayArea2Selection**
- **bool displayArea3Selection**
- **bool displayArea4Selection**
- **bool displayTargetSelection**
- **bool displayBeamSelection**
- **bool displayEllipse**
- **EllipseVariableDefinitions ellipseVariableDefinition**

*Definition of how ellipse variables are to be used.*

- `TargetOptions targetOption`

*Definition of target markup options.*

- `TargetOptions beamOption`

*Definition of beam markup options.*

- `bool displayCursorPixelInfo`
- `bool contrastReversal`
- `bool logBrightness`
- `bool showTime`
- `bool useFalseColour`
- `QColor vertSliceColor`
- `QColor vertSlice1Color`
- `QColor vertSlice2Color`
- `QColor vertSlice3Color`
- `QColor vertSlice4Color`
- `QColor vertSlice5Color`
- `QColor hozSliceColor`
- `QColor hozSlice1Color`
- `QColor hozSlice2Color`
- `QColor hozSlice3Color`
- `QColor hozSlice4Color`
- `QColor hozSlice5Color`
- `QColor profileColor`
- `QColor areaColor`
- `QColor beamColor`
- `QColor targetColor`
- `QColor timeColor`
- `QColor ellipseColor`
- `ResizeOptions resizeOption`
- `RotationOptions rotation`
- `bool verticalFlip`
- `bool horizontalFlip`
- `int initialHosScrollPos`
- `bool enableImageDisplayProperties`

*If true, the local Image Display Properties controls are displayed.*

- `bool enableRecording`

*If true, the recording controls are displayed.*

- `bool autoBrightnessContrast`
- `bool externalControls`
- `bool briefInfoArea`
- `QString program1`
- `QStringList arguments1`
- `ProgramStartupOptionNames programStartupOption1`

- [QString program2](#)
- [QStringList arguments2](#)
- [ProgramStartupOptionNames programStartupOption2](#)
- [QString URL](#)

### 9.96.1 Detailed Description

This class is a EPICS aware image widget. When image related variables are defined the image will be displayed. Many PVs may be defined to allow user interaction, such as selecting regions of interest. It is tightly integrated with the base class QEWidget which provides generic support such as macro substitutions, drag/drop, and standard properties.

### 9.96.2 Member Enumeration Documentation

#### 9.96.2.1 enum QEImage::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and displayAlarmStateOptions enumeration for details.

**Enumerator:**

*Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.96.2.2 enum QEImage::EllipseVariableDefinitions

User friendly enumerations for [ellipseVariableDefinition](#) property - refer to [ellipseVariableDefinition](#) property for details.

**Enumerator:**

*BoundingRectangle* Refer to BOUNDING\_RECTANGLE for details.

*CenterAndSize* Refer to CENTRE\_AND\_SIZE for details.

#### 9.96.2.3 enum QEImage::ellipseVariableDefinitions

Options for the use of ellipse markup variables.

**Enumerator:**

*BOUNDING\_RECTANGLE* Variables define bounding rectangle of ellipse.

### 9.96.2.4 enum QEImage::FormatOptions

User friendly enumerations for `formatOption` property - refer to `formatOption` property and `formatOptions` enumeration for details.

#### Enumerator:

- Mono*** Grey scale.
- Bayer*** Colour (Bayer Red Green).
- BayerGB*** Colour (Bayer Green Blue).
- BayerBG*** Colour (Bayer Blue Green).
- BayerGR*** Colour (Bayer Green Red).
- BayerRG*** Colour (Bayer Red Green).
- rgb1*** Colour (24 bit RGB).
- rgb2*** Colour (??? bit RGB).
- rgb3*** Colour (??? bit RGB).
- yuv444*** Colour (????).
- yuv422*** Colour (????).

### 9.96.2.5 enum QEImage::ProgramStartupOptionNames

Startup options. Just run the command, run the command within a terminal, or display the output in QE message system.

#### Enumerator:

- None*** Just run the program.
- Terminal*** Run the program in a terminal (in Windows a command interpreter will also be started, so the program may be a built-in command like 'dir').
- LogOutput*** Run the program, and log the output in the QE message system.
- StdOutput*** Run the program, and send output to standard output and standard error.

### 9.96.2.6 enum QEImage::ResizeOptions

User friendly enumerations for `resizeOption` property

#### Enumerator:

- Zoom*** Zoom to selected percentage.
- Fit*** Zoom to fit the current window size.

### 9.96.2.7 enum QEImage::resizeOptions

Image resize options

**Enumerator:**

*RESIZE\_OPTION\_ZOOM* Zoom to selected percentage.

*RESIZE\_OPTION\_FIT* Zoom to fit the current window size.

### 9.96.2.8 enum QEImage::RotationOptions

User friendly enumerations for [rotation](#) property

**Enumerator:**

*NoRotation* No image rotation.

*Rotate90Right* Rotate image 90 degrees clockwise.

*Rotate90Left* Rotate image 90 degrees anticlockwise.

*Rotate180* Rotate image 180 degrees.

### 9.96.2.9 enum QEImage::selectOptions

Internal use only. Selection options. What will happen when the user interacts with the image area

**Enumerator:**

*SO\_NONE* Do nothing.

*SO\_PANNING* User is panning.

*SO\_VSLICE1* Select the vertical slice 1 point.

*SO\_VSLICE2* Select the vertical slice 2 point.

*SO\_VSLICE3* Select the vertical slice 3 point.

*SO\_VSLICE4* Select the vertical slice 4 point.

*SO\_VSLICE5* Select the vertical slice 5 point.

*SO\_HSLICE1* Select the horizontal slice 1 point.

*SO\_HSLICE2* Select the horizontal slice 2 point.

*SO\_HSLICE3* Select the horizontal slice 3 point.

*SO\_HSLICE4* Select the horizontal slice 4 point.

*SO\_HSLICE5* Select the horizontal slice 5 point.

*SO\_AREA4* User is selecting an area (for region of interest).

*SO\_PROFILE* Select an arbitrary line across the image (to determine a profile).

*SO\_TARGET* Mark the target point.

*SO\_BEAM* Mark the current beam location.

### 9.96.2.10 enum QEImage::TargetOptions

User friendly enumerations for targetOptions property - refer to targetOptions property for details.

**Enumerator:**

*DottedFullCrosshair* Refer to CROSSHAIR1 for details.

*SolidSmallCrosshair* Refer to CROSSHAIR2 for details.

### 9.96.2.11 enum QEImage::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and userLevel enumeration for details.

**Enumerator:**

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

## 9.96.3 Constructor & Destructor Documentation

### 9.96.3.1 QEImage::QEImage (QWidget \* *parent* = 0)

Create without a variable. Use `setVariableName'n'Property()` - where 'n' is a number from 0 to 40 - and `setSubstitutionsProperty()` to define variables and, optionally, macro substitutions later. Note, each variable property is named by function (such as `imageVariable` and `widthVariable`) but given a numeric get and set property access function such as `setVariableName22Property()`. Refer to the property definitions to determine what 'set' and 'get' function is used for each variable, or use Qt library functions to set or get the variable names by name.

### 9.96.3.2 QEImage::QEImage (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a variable. A connection is automatically established. The variable is set up as the first variable. This is consistent with other widgets, but will not result in an updating image as the width and height variables are required as a minimum.

## 9.96.4 Member Function Documentation

### 9.96.4.1 void QEImage::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change. Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.96.4.2 void QEImage::setImageFile (QString *name*) [slot]**

!! memcpy will be more efficient.

**9.96.4.3 void QEImage::setManagedVisible (bool *v*) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.96.5 Member Data Documentation

**9.96.5.1 bool QEImage::displayButtonBar [read, write, protected]**

If true, a button bar will be displayed above the image. If not displayed, all buttons in the button bar are still available in the right click menu.

**9.96.5.2 int QEImage::initialVertScrollPos [read, write, protected]**

Sets the initial position of the vertical scroll bar, if present. Used to set up an initial view when zoomed in.

## 9.96.6 Property Documentation

**9.96.6.1 bool QEImage::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.96.6.2 QColor QEImage::areaColor [read, write]**

Used to select the color of the area selection markups.

**9.96.6.3 QStringList QEImage::arguments1 [read, write]**

Arguments for program specified in the 'program1' property.

**9.96.6.4 QStringList QEImage::arguments2 [read, write]**

Arguments for program specified in the 'program2' property.

**9.96.6.5 bool QEImage::autoBrightnessContrast [read, write]**

If true, auto set local brightness and contrast when any area is selected. The brightness and contrast is set to use the full range of pixels in the selected area.

**9.96.6.6 QColor QEImage::beamColor [read, write]**

Used to select the color of the beam marker.

**9.96.6.7 QString QEImage::beamXVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the selected beam X position.

**9.96.6.8 QString QEImage::beamYVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the selected beam Y position.

**9.96.6.9 QString QEImage::bitDepthVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read the bit depth of the image.

**9.96.6.10 bool QEImage::briefInfoArea [read, write]**

If true, the information area will be brief (one row)

**9.96.6.11 QString QEImage::clippingHighVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector clipping high level.

**9.96.6.12 QString QEImage::clippingLowVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector clipping low level.

**9.96.6.13 QString QEImage::clippingOnOffVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector clipping on/off command.

**9.96.6.14 bool QEImage::contrastReversal [read, write]**

If true, the image will undergo contrast reversal.

**9.96.6.15 QString QEImage::dataTypeVariable [read, write]**

EPICS variable name (CA PV). This variable is used to infer the bit depth of the image.

**9.96.6.16 QString QEImage::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.96.6.17 QString QEImage::dimension1Variable [read, write]**

EPICS variable name (CA PV). This variable is used to read the first area detector dimension of the image. If there are 2 dimensions, this will be the image width. If there are 3 dimensions, this will be the number of elements per pixel.

**9.96.6.18 QString QEImage::dimension2Variable [read, write]**

EPICS variable name (CA PV). This variable is used to read the second area detector dimension of the image. If there are 2 dimensions, this will be the image height. If there are 3 dimensions, this will be the image width.

**9.96.6.19 QString QEImage::dimension3Variable [read, write]**

EPICS variable name (CA PV). This variable is used to read the third area detector dimension of the image. If there are 3 dimensions, this will be the image height.

**9.96.6.20 QString QEImage::dimensionsVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read the number of area detector dimensions of the image. If used, this will be 2 (one element per pixel arranged by width and height) or 3 (multiple elements per pixel arranged by pixel, width and height)

**9.96.6.21 bool QEImage::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of

standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### **9.96.6.22 bool QEImage::displayAlarmStateOptions [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### **9.96.6.23 bool QEImage::displayArea1Selection [read, write]**

If true, selected area 1 will be displayed on the image. Note, this property is ignored unless the [enableArea1Selection](#) property is true.

#### **9.96.6.24 bool QEImage::displayArea2Selection [read, write]**

If true, selected area 2 will be displayed on the image. Note, this property is ignored unless the [enableArea2Selection](#) property is true.

#### **9.96.6.25 bool QEImage::displayArea3Selection [read, write]**

If true, selected area 3 will be displayed on the image. Note, this property is ignored unless the [enableArea3Selection](#) property is true.

#### **9.96.6.26 bool QEImage::displayArea4Selection [read, write]**

If true, selected area 4 will be displayed on the image. Note, this property is ignored unless the [enableArea4Selection](#) property is true.

#### **9.96.6.27 bool QEImage::displayBeamSelection [read, write]**

If true, beam selection will be displayed on the image. Note, this property is ignored unless the [enableBeamSelection](#) property is true.

#### **9.96.6.28 bool QEImage::displayCursorPixelInfo [read, write]**

If true, an area will be presented under the image with textual information about the pixel under the cursor, and for other selections such as selected areas.

**9.96.6.29 bool QEImage::displayEllipse [read, write]**

If true, the ellipse markup will be displayed on the image.

**9.96.6.30 bool QEImage::displayHozSlice1Selection [read, write]**

If true, the selected horizontal slice will be displayed on the image. Note, this property is ignored unless the [enableHozSlice1Selection](#) property is true.

**9.96.6.31 bool QEImage::displayHozSlice2Selection [read, write]**

If true, the selected horizontal slice will be displayed on the image. Note, this property is ignored unless the [enableHozSlice2Selection](#) property is true.

**9.96.6.32 bool QEImage::displayHozSlice3Selection [read, write]**

If true, the selected horizontal slice will be displayed on the image. Note, this property is ignored unless the [enableHozSlice3Selection](#) property is true.

**9.96.6.33 bool QEImage::displayHozSlice4Selection [read, write]**

If true, the selected horizontal slice will be displayed on the image. Note, this property is ignored unless the [enableHozSlice4Selection](#) property is true.

**9.96.6.34 bool QEImage::displayHozSlice5Selection [read, write]**

If true, the selected horizontal slice will be displayed on the image. Note, this property is ignored unless the [enableHozSlice5Selection](#) property is true.

**9.96.6.35 bool QEImage::displayProfileSelection [read, write]**

If true, the selected arbitrary line will be displayed on the image. Note, this property is ignored unless the [enableProfileSelection](#) property is true.

**9.96.6.36 bool QEImage::displayTargetSelection [read, write]**

If true, target selection will be displayed on the image. Note, this property is ignored unless the [enableTargetSelection](#) property is true.

**9.96.6.37 bool QEImage::displayVertSlice1Selection [read, write]**

If true, the selected vertical slice 1 will be displayed on the image. Note, this property is ignored unless the [enableVertSlice1Selection](#) property is true.

**9.96.6.38 bool QEImage::displayVertSlice2Selection [read, write]**

If true, the selected vertical slice 2 will be displayed on the image. Note, this property is ignored unless the [enableVertSlice2Selection](#) property is true.

**9.96.6.39 bool QEImage::displayVertSlice3Selection [read, write]**

If true, the selected vertical slice 3 will be displayed on the image. Note, this property is ignored unless the [enableVertSlice3Selection](#) property is true.

**9.96.6.40 bool QEImage::displayVertSlice4Selection [read, write]**

If true, the selected vertical slice 4 will be displayed on the image. Note, this property is ignored unless the [enableVertSlice4Selection](#) property is true.

**9.96.6.41 bool QEImage::displayVertSlice5Selection [read, write]**

If true, the selected vertical slice 5 will be displayed on the image. Note, this property is ignored unless the [enableVertSlice5Selection](#) property is true.

**9.96.6.42 QColor QEImage::ellipseColor [read, write]**

Used to select the color of the ellipse marker.

**9.96.6.43 QString QEImage::ellipseHVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read an ellipse height

**9.96.6.44 QString QEImage::ellipseWVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read an ellipse width.

**9.96.6.45 QString QEImage::ellipseXVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read an ellipse X (center or top left corner of bounding rectangle depending on property [ellipseDefinition](#)).

**9.96.6.46 QString QEImage::ellipseYVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read an ellipse Y (center or top left corner of bounding rectangle depending on property [ellipseDefinition](#)).

**9.96.6.47 bool QEImage::enableArea1Selection [read, write]**

If true, the user will be able to select area 1. These are used for selection of Region of Interests, and for zooming to area 1

**9.96.6.48 bool QEImage::enableArea2Selection [read, write]**

If true, the user will be able to select area 2. These are used for selection of Region of Interests, and for zooming to area 2

**9.96.6.49 bool QEImage::enableArea3Selection [read, write]**

If true, the user will be able to select area 3. These are used for selection of Region of Interests, and for zooming to area 3

**9.96.6.50 bool QEImage::enableArea4Selection [read, write]**

If true, the user will be able to select area 4. These are used for selection of Region of Interests, and for zooming to area 4

**9.96.6.51 bool QEImage::enableBeamSelection [read, write]**

If true, the user will be able to select points on the image to mark a beam position. This can be used for automatic beam positioning.

**9.96.6.52 bool QEImage::enableHozSlice1Selection [read, write]**

If true, the option to select a horizontal slice through the image will be available to the user. This will be used to generate a horizontal pixel profile, and write the position of the slice to the optional variable specified by the `profileHoz1Variable` property. The profile will only be presented to the user if `enableHozSlicePresentation` property is true.

**9.96.6.53 bool QEImage::enableHozSlice2Selection [read, write]**

If true, the option to select a second horizontal slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the `profileHoz2Variable` property.

**9.96.6.54 bool QEImage::enableHozSlice3Selection [read, write]**

If true, the option to select a third horizontal slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the `profileHoz3Variable` property.

**9.96.6.55 bool QEImage::enableHozSlice4Selection [read, write]**

If true, the option to select a fourth horizontal slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the [profileHoz4Variable](#) property.

**9.96.6.56 bool QEImage::enableHozSlice5Selection [read, write]**

If true, the option to select a fifth horizontal slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the [profileHoz5Variable](#) property.

**9.96.6.57 bool QEImage::enableProfileSelection [read, write]**

If true, the option to select an arbitrary line through any part of the image will be available to the user. This will be used to generate a pixel profile.

**9.96.6.58 bool QEImage::enableTargetSelection [read, write]**

If true, the user will be able to select points on the image to mark a target position. This can be used for automatic beam positioning.

**9.96.6.59 bool QEImage::enableVertSlice1Selection [read, write]**

If true, the option to select a vertical slice through the image will be available to the user. This will be used to generate a horizontal pixel profile, and write the position of the slice to the optional variable specified by the [profileVert1Variable](#) property. The profile will only be presented to the user if enableVertSlicePresentation property is true.

**9.96.6.60 bool QEImage::enableVertSlice2Selection [read, write]**

If true, the option to select a second vertical slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the [profileVert2Variable](#) property.

**9.96.6.61 bool QEImage::enableVertSlice3Selection [read, write]**

If true, the option to select a third vertical slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the [profileVert3Variable](#) property.

**9.96.6.62 bool QEImage::enableVertSlice4Selection [read, write]**

If true, the option to select a fourth vertical slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the [profileVert4Variable](#) property.

**9.96.6.63 bool QEImage::enableVertSlice5Selection [read, write]**

If true, the option to select a fifth vertical slice through the image will be available to the user. This will be used to write the position of the slice to the optional variable specified by the [profileVert5Variable](#) property.

**9.96.6.64 bool QEImage::externalControls [read, write]**

If true, image controls and views such as brightness controls and profile plots are hosted by the application as dock windows, toolbars, etc. Refer to the ContainerProfile class and the windowCustomisation class to see how this class asks an application to act as a host.

**9.96.6.65 FormatOptions QEImage::formatOption [read, write]**

Video format. EPICS data type size will typically be adequate for the number of bits required (one byte for 8 bits, 2 bytes for 12 and 16 bits), but can be larger (4 bytes for 24 bits.)

**9.96.6.66 QString QEImage::formatVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read the format of the image.

**9.96.6.67 QString QEImage::heightVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read the height of the image.

**9.96.6.68 bool QEImage::horizontalFlip [read, write]**

If true, flip image horizontally.

**9.96.6.69 QColor QEImage::hozSlice1Color [read, write]**

Used to select the color of the horizontal slice 1 markup.

**9.96.6.70 QColor QEImage::hozSlice2Color [read, write]**

Used to select the color of the horizontal slice 2 markup.

**9.96.6.71 QColor QEImage::hozSlice3Color [read, write]**

Used to select the color of the horizontal slice 3 markup.

**9.96.6.72 QColor QEImage::hozSlice4Color [read, write]**

Used to select the color of the horizontal slice 4 markup.

**9.96.6.73 QColor QEImage::hozSlice5Color [read, write]**

Used to select the color of the horizontal slice 5 markup.

**9.96.6.74 QString QEImage::imageVariable [read, write]**

EPICS variable name (CA PV). This variable is used as the source the image waveform.

**9.96.6.75 int QEImage::initialHosScrollPos [read, write]**

Sets the initial position of the horizontal scroll bar, if present. Used to set up an initial view when zoomed in.

**9.96.6.76 unsigned QEImage::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

Bit depth. Note, EPICS data type size will typically be adequate for the number of bits required (one byte for up to 8 bits, 2 bytes for up to 16 bits, etc), but can be larger (for example, 4 bytes for 24 bits) and may be larger than necessary (4 bytes for 8 bits).

**9.96.6.77 QString QEImage::lineProfileArrayVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector arbitrary line profile array.

**9.96.6.78 QString QEImage::lineProfileThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector arbitrary line profile end Y.

**9.96.6.79 QString QEImage::lineProfileX1Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector arbitrary line profile start X.

**9.96.6.80 QString QEImage::lineProfileX2Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector arbitrary line profile end X.

**9.96.6.81 QString QEImage::lineProfileY1Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector arbitrary line profile start Y.

**9.96.6.82 QString QEImage::lineProfileY2Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector arbitrary line profile end Y.

**9.96.6.83 bool QEImage::logBrightness [read, write]**

If true, the image will be displayed using a logarithmic brightness scale.

**9.96.6.84 QColor QEImage::profileColor [read, write]**

Used to select the color of the arbitrary profile line markup.

**9.96.6.85 QString QEImage::profileHoz1ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector first horizontal profile thickness.

**9.96.6.86 QString QEImage::profileHoz1Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector first horizontal profile.

**9.96.6.87 QString QEImage::profileHoz2ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector second horizontal profile thickness.

**9.96.6.88 QString QEImage::profileHoz2Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector second horizontal profile.

**9.96.6.89 QString QEImage::profileHoz3ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector third horizontal profile thickness.

**9.96.6.90 QString QEImage::profileHoz3Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector third horizontal profile.

**9.96.6.91 QString QEImage::profileHoz4ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fourth horizontal profile thickness.

**9.96.6.92 QString QEImage::profileHoz4Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fourth horizontal profile.

**9.96.6.93 QString QEImage::profileHoz5ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fifth horizontal profile thickness.

**9.96.6.94 QString QEImage::profileHoz5Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fifth horizontal profile.

**9.96.6.95 QString QEImage::profileHozArrayVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector horizontal profile array.

**9.96.6.96 QString QEImage::profileVert1ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector first vertical profile.

**9.96.6.97 QString QEImage::profileVert1Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector first vertical profile.

**9.96.6.98 QString QEImage::profileVert2ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector second vertical profile.

**9.96.6.99 QString QEImage::profileVert2Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector second vertical profile.

**9.96.6.100 QString QEImage::profileVert3ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector third vertical profile.

**9.96.6.101 QString QEImage::profileVert3Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector third vertical profile.

**9.96.6.102 QString QEImage::profileVert4ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fourth vertical profile.

**9.96.6.103 QString QEImage::profileVert4Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fourth vertical profile.

**9.96.6.104 QString QEImage::profileVert5ThicknessVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fifth vertical profile.

**9.96.6.105 QString QEImage::profileVert5Variable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector fifth vertical profile.

**9.96.6.106 QString QEImage::profileVertArrayVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the areadetector vertical profile array.

**9.96.6.107 QString QEImage::program1 [read, write]**

Program to run when a request is made to pass on the current image to the first external application. No attempt to run a program is made if this property is empty. Example: paint.exe

**9.96.6.108 QString QEImage::program2 [read, write]**

Program to run when a request is made to pass on the current image to the second external application. No attempt to run a program is made if this property is empty. Example: paint.exe

**9.96.6.109 ProgramStartupOptionNames QEImage::programStartupOption1 [read, write]**

Startup options for the program specified in the 'program1' property. Just run the command, run the command within a terminal, or display the output in QE message system.

**9.96.6.110 ProgramStartupOptionNames QEImage::programStartupOption2 [read, write]**

Startup options for the program specified in the 'program2' property. Just run the command, run the command within a terminal, or display the output in QE message system.

**9.96.6.111 QString QEImage::regionOfInterest1HVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the first region of interest height.

**9.96.6.112 QString QEImage::regionOfInterest1WVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the first region of interest width.

**9.96.6.113 QString QEImage::regionOfInterest1XVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the first region of interest X position.

**9.96.6.114 QString QEImage::regionOfInterest1YVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the first region of interest Y position.

**9.96.6.115 QString QEImage::regionOfInterest2HVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the second region of interest height.

**9.96.6.116 QString QEImage::regionOfInterest2WVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the second region of interest width.

**9.96.6.117 QString QEImage::regionOfInterest2XVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the second region of interest X position.

**9.96.6.118 QString QEImage::regionOfInterest2YVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the second region of interest Y position.

**9.96.6.119 QString QEImage::regionOfInterest3HVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the third region of interest height.

**9.96.6.120 QString QEImage::regionOfInterest3WVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the third region of interest width.

**9.96.6.121 QString QEImage::regionOfInterest3XVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the third region of interest X position.

**9.96.6.122 QString QEImage::regionOfInterest3YVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the third region of interest Y position.

**9.96.6.123 QString QEImage::regionOfInterest4HVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the fourth region of interest height.

**9.96.6.124 QString QEImage::regionOfInterest4WVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the fourth region of interest width.

**9.96.6.125 QString QEImage::regionOfInterest4XVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the fourth region of interest X position.

**9.96.6.126 QString QEImage::regionOfInterest4YVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the fourth region of interest Y position.

**9.96.6.127 ResizeOptions QEImage::resizeOption [read, write]**

Resize option. Zoom to zoom to the percentage given by the [zoom](#) property, or fit to the window size.

**9.96.6.128 RotationOptions QEImage::rotation [read, write]**

Image rotation option.

**9.96.6.129 bool QEImage::showTime [read, write]**

If true, the image timestamp will be written in the top left of the image.

**9.96.6.130 QString QEImage::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.96.6.131 QColor QEImage::targetColor [read, write]**

Used to select the color of the target marker.

**9.96.6.132 QString QEImage::targetTriggerVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write a 'trigger' to initiate movement of the target into the beam as defined by the target and beam X and Y positions.

**9.96.6.133 QString QEImage::targetXVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the selected target X position.

**9.96.6.134 QString QEImage::targetYVariable [read, write]**

EPICS variable name (CA PV). This variable is used to write the selected target Y position.

**9.96.6.135 QColor QEImage::timeColor [read, write]**

Used to select the color of the timestamp.

**9.96.6.136 QString QEImage::URL [read, write]**

MPEG stream URL. If this is specified, this will be used as the source of the image in preference to variables (variables defining the image data, width, and height will be ignored)

**9.96.6.137 bool QEImage::useFalseColour [read, write]**

If true, the apply false colour to the image.

**9.96.6.138 UserLevels QEImage::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmaticaly through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessable to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessable to engineers maintaining the facility should be visible at 'Engineer'.

**9.96.6.139 QString QEImage::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.96.6.140 QString QEImage::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.96.6.141 QString QEImage::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.96.6.142 QEImage::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.96.6.143 bool QEImage::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.96.6.144 QString QEImage::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'CAM=1, NAME = "Image 1"' These substitutions are applied to all the variable names.

**9.96.6.145 bool QEImage::verticalFlip [read, write]**

If true, flip image vertically.

**9.96.6.146 QColor QEImage::vertSlice1Color [read, write]**

Used to select the color of the vertical slice 1 markup.

**9.96.6.147 QColor QEImage::vertSlice2Color [read, write]**

Used to select the color of the vertical slice 2 markup.

**9.96.6.148 QColor QEImage::vertSlice3Color [read, write]**

Used to select the color of the vertical slice 3 markup.

**9.96.6.149 QColor QEImage::vertSlice4Color [read, write]**

Used to select the color of the vertical slice 4 markup.

**9.96.6.150 QColor QEImage::vertSlice5Color [read, write]**

Used to select the color of the vertical slice 5 markup.

**9.96.6.151 bool QEImage::visible [read, write]**

Display the widget. Default is true. Setting this property false is usefull if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

**9.96.6.152 QString QEImage::widthVariable [read, write]**

EPICS variable name (CA PV). This variable is used to read the width of the image.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImage.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImage.cpp

## 9.97 QEImageMarkupThickness Class Reference

### Public Member Functions

- **QEImageMarkupThickness** (QWidget \*parent=0)
- void **setThickness** (unsigned int thicknessIn)
- unsigned int **getThickness** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImageMarkupTh
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImageMarkupTh

## 9.98 QEImageOptionsDialog Class Reference

### Signals

- void **optionChange** (imageContextMenu::imageContextMenuOptions option, bool checked)

### Public Member Functions

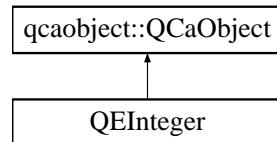
- **QEImageOptionsDialog** (QWidget \*parent=0)
- void **initialise** ()
- void **optionSet** (imageContextMenu::imageContextMenuOptions option, bool checked)
- bool **optionGet** (imageContextMenu::imageContextMenuOptions option)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImageOptionsDialog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/QEImageOptionsDialog.cpp

## 9.99 QEInteger Class Reference

Inheritance diagram for QEInteger::



### Public Slots

- void **writeInteger** (const long &data)
- void **writeIntegerElement** (const long &data)
- void **writeInteger** (const QVector< long > &data)

### Signals

- void **integerConnectionChanged** ([QCaConnectionInfo](#) &connectionInfo, const unsigned int &variableIndex)
- void **integerChanged** (const long &value, [QCaAlarmInfo](#) &alarmInfo, [QCaDateTime](#) &timeStamp, const unsigned int &variableIndex)
- void **integerArrayChanged** (const QVector< long > &values, [QCaAlarmInfo](#) &alarmInfo, [QCaDateTime](#) &timeStamp, const unsigned int &variableIndex)

### Public Member Functions

- **QEInteger** (QString recordName, QObject \*eventObject, [QEIntegerFormatting](#) \*integerFormattingIn, unsigned int variableIndexIn)
- **QEInteger** (QString recordName, QObject \*eventObject, [QEIntegerFormatting](#) \*integerFormattingIn, unsigned int variableIndexIn, UserMessage \*userMessageIn)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEInteger.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEInteger.cpp

## 9.100 QEIntegerArray Class Reference

```
#include <QEIntegerArray.h>
```

### Public Member Functions

- **QEIntegerArray** (int size)
- **QEIntegerArray** (int size, const long &t)
- **QEIntegerArray** (const QVector< long > &other)
- long **minimumValue** (const long &defaultValue=0)
- long **maximumValue** (const long &defaultValue=0)

### 9.100.1 Detailed Description

This class provides short hand for QVector<long> together with some basic long vector operations.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEIntegerArray.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEIntegerArray.cpp

## 9.101 QEIntegerFormatting Class Reference

```
#include <QEIntegerFormatting.h>
```

### Public Member Functions

- [QEIntegerFormatting \(\)](#)

*Constructor.*

- long [formatInteger \(const QVariant &value\)](#)
- QVector< long > [formatIntegerArray \(const QVariant &value\)](#)
- QVariant [formatValue \(const long &integerValue, generic::generic\\_types valueType\)](#)
- QVariant [formatValue \(const QVector< long > &integerValue, generic::generic\\_types valueType\)](#)
- void [setRadix \(unsigned int radix\)](#)

*Set the radix used for all conversions. Default is 10.*

- unsigned int [getPrecision \(\)](#)

*Get the precision used for all conversions.*

- unsigned int [getRadix \(\)](#)

*Get the radix used for all conversions.*

### 9.101.1 Detailed Description

This class holds formatting instructions and uses them to convert between an integer and a QVariant of any type. It is generally set up with its formatting instructions and then passed to a [QEInteger](#) class that will sink and source integer data to widgets or other code. It is used to convert data to and from a QCaObject (which sources and sinks data in the form of a QVariant where the QVariant reflects the underlying variable data type) and the [QEInteger](#) class. An example of a requirement for integer data is a combo box which must determine an integer index to select a menu option.

### 9.101.2 Member Function Documentation

#### 9.101.2.1 long QEIntegerFormatting::formatInteger (const QVariant & value)

Given a data value of any type, format it as an integer according to the formatting instructions held by the class. This is used to convert the QVariant value received from a QCaObject, which is still based on the data variable type, to an integer.

**9.101.2.2 `QVector< long > QEIntegerFormatting::formatIntegerArray (const QVariant & value)`**

Given a data value of any type, format it as an array of integers according to the formatting instructions held by the class. This is used to convert the QVariant value received from a QCaObject, which is still based on the data variable type, to an integer array. Typically used where the input QVariant value is an array of data values, but will work for any QVariant type.

**9.101.2.3 `QVariant QEIntegerFormatting::formatValue (const long & integerValue, generic::generic_types valueType)`**

Given an integer value, format it as a data value of the specified type, according to the formatting instructions held by the class. This is used when writing integer data to a QCaObject.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEIntegerFormatting.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEIntegerFormatting.cpp

## 9.102 QELabel Class Reference

```
#include <QELabel.h>
```

### Public Types

- enum `updateOptions` { `UPDATE_TEXT`, `UPDATE_PIXMAP` }
- enum `UserLevels` { `User` = `userLevelTypes::USERLEVEL_USER`, `Scientist` = `userLevelTypes::USERLEVEL_SCIENTIST`, `Engineer` = `userLevelTypes::USERLEVEL_ENGINEER` }
- enum `DisplayAlarmStateOptions` { `Never` = `standardProperties::DISPLAY_ALARM_STATE_NEVER`, `Always` = `standardProperties::DISPLAY_ALARM_STATE_ALWAYS`, `WhenInAlarm` = `standardProperties::DISPLAY_ALARM_STATE_WHEN_IN_ALARM` }
- enum `Formats` {
   
`Default` = `QStringFormatting::FORMAT_DEFAULT`, `Floating` = `QStringFormatting::FORMAT_FLOATING`, `Integer` = `QStringFormatting::FORMAT_INTEGER`, `UnsignedInteger` = `QStringFormatting::FORMAT_UNSIGNEDINTEGER`,
   
`Time` = `QStringFormatting::FORMAT_TIME`, `LocalEnumeration` = `QStringFormatting::FORMAT_LOCAL_ENUMERATE` }
- enum `Separators` { `NoSeparator` = `QStringFormatting::SEPARATOR_NONE`, `Comma` = `QStringFormatting::SEPARATOR_COMMA`, `Underline` = `QStringFormatting::SEPARATOR_UNDERSCORE`, `Space` = `QStringFormatting::SEPARATOR_SPACE` }
- enum `Notations` { `Fixed` = `QStringFormatting::NOTATION_FIXED`, `Scientific` = `QStringFormatting::NOTATION_SCIENTIFIC`, `Automatic` = `QStringFormatting::NOTATION_AUTOMATIC` }
- enum `ArrayActions` { `Append` = `QStringFormatting::APPEND`, `Ascii` = `QStringFormatting::ASCII`, `Index` = `QStringFormatting::INDEX` }
- enum `UpdateOptions` { `Text` = `QELabel::UPDATE_TEXT`, `Picture` = `QELabel::UPDATE_PIXMAP` }

*User friendly enumerations for updateOption property - refer to `QELabel::updateOptions` for details.*

### Public Slots

- void `setDefaultStyle` (const QString &style)
   
*Update the default style applied to this widget.*
- void `setManagedVisible` (bool v)

## Signals

- void **dbValueChanged** (const QString &out)
- void **dbValueChanged** (const int &out)
- void **dbValueChanged** (const long &out)
- void **dbValueChanged** (const qlonglong &out)
- void **dbValueChanged** (const double &out)
- void **dbValueChanged** (const bool &out)
- void **dbConnectionChanged** (const bool &isConnected)

*Sent when the widget state updated following a channel connection change.*

- void **requestResend** ()

*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*

## Public Member Functions

- **QELabel** (QWidget \*parent=0)
- **QELabel** (const QString &variableName, QWidget \*parent=0)
- **UserLevels getUserLevelVisibilityProperty** ()

*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*

- void **setUserLevelVisibilityProperty** (UserLevels level)

*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*

- **UserLevels getUserLevelEnabledProperty** ()

*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*

- void **setUserLevelEnabledProperty** (UserLevels level)

*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*

- **DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty** ()

*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*

- void **setDisplayAlarmStateOptionProperty** (DisplayAlarmStateOptions option)

*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*

- void **setFormatProperty** (Formats format)

*Access function for format property - refer to format property for details.*

- **Formats getFormatProperty ()**  
*Access function for `format` property - refer to `format` property for details.*
- **void setSeparatorProperty (const Separators notation)**  
*Access function for `separator` property - refer to `separator` property for details.*
- **Separators getSeparatorProperty () const**  
*Access function for `separator` property - refer to `separator` property for details.*
- **void setNotationProperty (Notations notation)**  
*Access function for `notation` property - refer to `notation` property for details.*
- **Notations getNotationProperty ()**  
*Access function for `notation` property - refer to `notation` property for details.*
- **void setArrayActionProperty (ArrayActions arrayAction)**  
*Access function for `arrayAction` property - refer to `arrayAction` property for details.*
- **ArrayActions getArrayActionProperty ()**  
*Access function for `arrayAction` property - refer to `arrayAction` property for details.*
- **void setUpdateOptionProperty (UpdateOptions updateOption)**  
*Access function for `updateOption` property - refer to `updateOption` property for details.*
- **UpdateOptions getUpdateOptionProperty ()**  
*Access function for `updateOption` property - refer to `updateOption` property for details.*

## Properties

- `QString variable`
- `QString variableSubstitutions`
- `int arrayIndex`
- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`

- bool `displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- int `precision`
- bool `useDbPrecision`
- bool `leadingZero`
- bool `trailingZeros`
- bool `addUnits`
- `QString localEnumeration`
- `Formats format`
- int `radix`
- `Separators separator`
- `Notations notation`
- `ArrayActions arrayAction`
- `QString text`
- `UpdateOptions updateOption`
- `QPixmap pixmap0`
- `QPixmap pixmap1`
- `QPixmap pixmap2`
- `QPixmap pixmap3`
- `QPixmap pixmap4`
- `QPixmap pixmap5`
- `QPixmap pixmap6`
- `QPixmap pixmap7`

### 9.102.1 Detailed Description

This class is a EPICS aware label widget based on the Qt label widget. When a variable is defined, the label text (or optionally the background pixmap) will be updated. The label will be disabled if the variable is invalid. It is tightly integrated with the base class QEWidget which provides generic support such as macro substitutions, drag/drop, and standard properties.

### 9.102.2 Member Enumeration Documentation

#### 9.102.2.1 enum QELabel::ArrayActions

User friendly enumerations for arrayAction property - refer to `QEStringFormatting::arrayActions` for details.

**Enumerator:**

- Append* Refer to `QEStringFormatting::APPEND` for details.
- Ascii* Refer to `QEStringFormatting::ASCII` for details.
- Index* Refer to `QEStringFormatting::INDEX` for details.

### 9.102.2.2 enum QELabel::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

#### Enumerator:

*Never* Refer to [DISPLAY\\_ALARM\\_STATE\\_NEVER](#) for details.

*Always* Refer to [DISPLAY\\_ALARM\\_STATE\\_ALWAYS](#) for details.

*WhenInAlarm* Refer to [DISPLAY\\_ALARM\\_STATE\\_WHEN\\_IN\\_ALARM](#) for details.

### 9.102.2.3 enum QELabel::Formats

User friendly enumerations for format property - refer to [QEStringFormatting::formats](#) for details.

#### Enumerator:

*Default* Format as best appropriate for the data type.

*Floating* Format as a floating point number.

*Integer* Format as an integer.

*UnsignedInteger* Format as an unsigned integer.

*Time* Format as a time.

*LocalEnumeration* Format as a selection from the [localEnumeration](#) property.

### 9.102.2.4 enum QELabel::Notations

User friendly enumerations for notation property - refer to [QEStringFormatting::notations](#) for details.

#### Enumerator:

*Fixed* Refer to [QEStringFormatting::NOTATION\\_FIXED](#) for details.

*Scientific* Refer to [QEStringFormatting::NOTATION\\_SCIENTIFIC](#) for details.

*Automatic* Refer to [QEStringFormatting::NOTATION\\_AUTOMATIC](#) for details.

### 9.102.2.5 enum QELabel::Separators

User friendly enumerations for separator property - refer to [QEStringFormatting::formats](#) for details.

#### Enumerator:

*NoSeparator* Use no separator.

**Comma** Use ',' as separator.

**Underscore** Use '\_' as separator.

**Space** Use ' ' as separator.

### 9.102.2.6 enum QELabel::UpdateOptions

User friendly enumerations for updateOption property - refer to [QELabel::updateOptions](#) for details.

#### Enumerator:

**Text** Data updates will update the label text.

**Picture** Data updates will update the label icon.

### 9.102.2.7 enum QELabel::updateOptions

Options for updating the label. The formatted text is used to update the label text, or select a background pixmap.

#### Enumerator:

**UPDATE\_TEXT** Update the label text.

**UPDATE\_PIXMAP** Update the label background pixmap.

### 9.102.2.8 enum QELabel::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and userLevel enumeration for details.

#### Enumerator:

**User** Refer to USERLEVEL\_USER for details.

**Scientist** Refer to USERLEVEL\_SCIENTIST for details.

**Engineer** Refer to USERLEVEL\_ENGINEER for details.

## 9.102.3 Constructor & Destructor Documentation

### 9.102.3.1 QELabel::QELabel (QWidget \* *parent* = 0)

Create without a variable. Use [setVariableNameProperty\(\)](#) and [setSubstitutionsProperty\(\)](#) to define a variable and, optionally, macro substitutions later.

**9.102.3.2 QELabel::QELabel (const QString & *variableName*, QWidget \* *parent* = 0)**

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

**9.102.4 Member Function Documentation****9.102.4.1 void QELabel::dbValueChanged (const QString & *out*) [signal]**

Sent when the widget is updated following a data change. Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.102.4.2 void QELabel::setManagedVisible (bool *v*) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

**9.102.5 Property Documentation****9.102.5.1 bool QELabel::addUnits [read, write]**

If true (default), add engineering units supplied with the data.

**9.102.5.2 bool QELabel::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.102.5.3 ArrayActions QELabel::arrayAction [read, write]**

Text formatting option for array data. Default is ASCII. Options are:

- ASCII - treat array as a single text string. For example an array of three characters 'a' 'b' 'c' will be formatted as 'abc'.
- APPEND - treat array as an array of numbers and format a string containing them all with a space between each. For example, an array of three numbers 10, 11 and 12 will be formatted as '10 11 12'.
- INDEX - Extract a single item from the array. The item is then formatted as any other non array data would be. The item selected is determined by the arrayIndex property. For example, if arrayIndex property is 1, an array of three numbers 10, 11 and 12 will be formatted as '11'.

**9.102.5.4 int QELabel::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.102.5.5 QString QELabel::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.102.5.6 bool QELabel::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.102.5.7 DisplayAlarmStateOptions QELabel::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.102.5.8 Formats QELabel::format [read, write]**

Format to apply to data. Default is 'Default' in which case the data type supplied with the data determines how the data is formatted. For all other options, an attempt is made to format the data as requested (whatever its native form).

**9.102.5.9 unsigned QELabel::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.102.5.10 bool QELabel::leadingZero [read, write]**

If true (default), always add a leading zero when formatting numbers.

### 9.102.5.11 QString QELabel::localEnumeration [read, write]

An enumeration list used to data values. Used only when the formatting option is 'local enumeration'. Value is converted to an integer and used to select a string from this list.

Format is:

[[<|<=|=|=|>|=|>]value1|\*] : string1 , [[<|<=|=|=|>|=|>]value2|\*] : string2 ,  
[[<|<=|=|=|>|=|>]value3|\*] : string3 , ...

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
>= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm"  
<2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than 2"  
3:"Beamline Available", \*:"Pump Off":"OH NO!, the pump is OFF!","Pump On":"It's OK, the pump is on"

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10":""'

A range of numbers can be covered by a pair of values as in the following example:  
>=4:"Between 4 and 8",<=8:"Between 4 and 8"

### 9.102.5.12 Notations QELabel::notation [read, write]

Notation used for numerical formatting. Default is fixed.

### 9.102.5.13 QPixmap QELabel:: pixmap0 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 0.

### 9.102.5.14 QPixmap QELabel:: pixmap1 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 1.

**9.102.5.15 QPixmap QELabel:: pixmap2 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 2.

**9.102.5.16 QPixmap QELabel:: pixmap3 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 3.

**9.102.5.17 QPixmap QELabel:: pixmap4 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 4.

**9.102.5.18 QPixmap QELabel:: pixmap5 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 5.

**9.102.5.19 QPixmap QELabel:: pixmap6 [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 6.

**9.102.5.20 int QELabel:: precision [read, write]**

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 7.

**9.102.5.21 int QELabel:: precision [read, write]**

Precision used when formatting floating point numbers. The default is 4. This is only used if useDbPrecision is false.

**9.102.5.22 int QELabel:: radix [read, write]**

Base used for when formatting integers. Default is 10 (duh!)

**9.102.5.23 Separators QELabel:: separator [read, write]**

Separators used for integer and fixed point formatting. Default is None.

**9.102.5.24 QString QELabel::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.102.5.25 bool QELabel::trailingZeros [read, write]**

If true (default), always remove any trailing zeros when formatting numbers.

**9.102.5.26 UpdateOptions QELabel::updateOption [read, write]**

Determines if data updates the label text, or the label pixmap. For both options all normal string formatting is applied. If Text, the formatted text is simply presented as the label text. If Picture, the FORMATTED text is then interpreted as an integer and used to select one of the pixmaps specified by properties pixmap0 through to pixmap7.

**9.102.5.27 bool QELabel::useDbPrecision [read, write]**

If true (default), format floating point numbers using the precision supplied with the data. If false, the precision property is used.

**9.102.5.28 UserLevels QELabel::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.102.5.29 QString QELabel::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.102.5.30 QString QELabel::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example,

'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.102.5.31 QString QELabel::userLevelUserStyle [read, write]

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.102.5.32 UserLevels QELabel::userLevelVisibility [read, write]

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### 9.102.5.33 QString QELabel::variable [read, write]

EPICS variable name (CA PV)

#### 9.102.5.34 bool QELabel::variableAsToolTip [read, write]

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### 9.102.5.35 QString QELabel::variableSubstitutions [read, write]

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### 9.102.5.36 bool QELabel::visible [read, write]

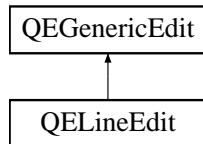
Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELabel/QELabel.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELabel/QELabel.cpp

## 9.103 QELineEdit Class Reference

Inheritance diagram for QELineEdit::



### Public Types

- enum `Formats` {
   
    Default = QEStructFormatting::FORMAT\_DEFAULT,     Float-
   
    ing = QEStructFormatting::FORMAT\_FLOATING,     Integer =
   
    QEStructFormatting::FORMAT\_INTEGER,             UnsignedInteger =
   
    QEStructFormatting::FORMAT\_UNSIGNEDINTEGER,
   
    Time = QEStructFormatting::FORMAT\_TIME,        LocalEnumeration =
   
    QEStructFormatting::FORMAT\_LOCAL\_ENUMERATE }
- enum `Separators` { `NoSeparator` = QEStructFormatting::SEPARATOR\_-
   
    NONE, `Comma` = QEStructFormatting::SEPARATOR\_COMMA, `Under-
   
    score` = QEStructFormatting::SEPARATOR\_UNDERSCORE, `Space` =
   
    QEStructFormatting::SEPARATOR\_SPACE }
- enum `Notations` { `Fixed` = QEStructFormatting::NOTATION\_FIXED, `Sci-
   
    entific` = QEStructFormatting::NOTATION\_SCIENTIFIC, `Automatic` =
   
    QEStructFormatting::NOTATION\_AUTOMATIC }
- enum `ArrayActions` { `Append` = QEStructFormatting::APPEND, `Ascii` =
   
    QEStructFormatting::ASCII, `Index` = QEStructFormatting::INDEX }

### Signals

- void `dbValueChanged` (const QString &out)
- void `dbValueChanged` (const int &out)
- void `dbValueChanged` (const long &out)
- void `dbValueChanged` (const qlonglong &out)
- void `dbValueChanged` (const double &out)
- void `dbValueChanged` (const bool &out)
- void `dbConnectionChanged` (const bool &isConnected)

*Sent when the widget state updated following a channel connection change.*

- void `userChange` (const QString &oldValue, const QString &newValue, const QString &lastValue)

*Internal use only. Used by `QEConfiguredLayout` to be notified when one of its widgets has written something.*

- void `requestResend ()`

*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*

## Public Member Functions

- void `setFormatProperty (Formats format)`

*Access function for `format` property - refer to `format` property for details.*

- `Formats getFormatProperty ()`

*Access function for `format` property - refer to `format` property for details.*

- void `setSeparatorProperty (const Separators notation)`

*Access function for `separator` property - refer to `separator` property for details.*

- `Separators getSeparatorProperty () const`

*Access function for `separator` property - refer to `separator` property for details.*

- void `setNotationProperty (Notations notation)`

*Access function for `notation` property - refer to `notation` property for details.*

- `Notations getNotationProperty ()`

*Access function for `notation` property - refer to `notation` property for details.*

- void `setArrayActionProperty (ArrayActions arrayAction)`

*Access function for `arrayAction` property - refer to `arrayAction` property for details.*

- `ArrayActions getArrayActionProperty ()`

*Access function for `arrayAction` property - refer to `arrayAction` property for details.*

- `QELLineEdit (QWidget *parent=0)`

- `QELLineEdit (const QString &variableName, QWidget *parent=0)`

## Properties

- int `precision`
- bool `useDbPrecision`
- bool `leadingZero`
- bool `trailingZeros`
- bool `addUnits`
- `QString localEnumeration`
- `Formats format`
- int `radix`
- `Separators separator`
- `Notations notation`
- `ArrayActions arrayAction`

### 9.103.1 Member Enumeration Documentation

#### 9.103.1.1 enum QELineEdit::ArrayActions

User friendly enumerations for arrayAction property - refer to [QEStringFormatting::arrayActions](#) for details.

##### Enumerator:

*Append* Refer to [QEStringFormatting::APPEND](#) for details.

*Ascii* Refer to [QEStringFormatting::ASCII](#) for details.

*Index* Refer to [QEStringFormatting::INDEX](#) for details.

#### 9.103.1.2 enum QELineEdit::Formats

User friendly enumerations for format property - refer to [QEStringFormatting::formats](#) for details.

##### Enumerator:

*Default* Format as best appropriate for the data type.

*Floating* Format as a floating point number.

*Integer* Format as an integer.

*UnsignedInteger* Format as an unsigned integer.

*Time* Format as a time.

*LocalEnumeration* Format as a selection from the [localEnumeration](#) property.

#### 9.103.1.3 enum QELineEdit::Notations

User friendly enumerations for notation property - refer to [QEStringFormatting::notations](#) for details.

##### Enumerator:

*Fixed* Refer to [QEStringFormatting::NOTATION\\_FIXED](#) for details.

*Scientific* Refer to [QEStringFormatting::NOTATION\\_SCIENTIFIC](#) for details.

*Automatic* Refer to [QEStringFormatting::NOTATION\\_AUTOMATIC](#) for details.

#### 9.103.1.4 enum QELineEdit::Separators

User friendly enumerations for separator property - refer to [QEStringFormatting::formats](#) for details.

**Enumerator:**

*NoSeparator* Use no separator.

*Comma* Use ',' as separator.

*Underscore* Use '\_' as separator.

*Space* Use ' ' as separator.

## 9.103.2 Constructor & Destructor Documentation

### 9.103.2.1 QELineEdit::QELineEdit (QWidget \**parent* = 0)

Create without a variable. Use setVariableNameProperty() and setSubstitutionsProperty() to define a variable and, optionally, macro substitutions later.

### 9.103.2.2 QELineEdit::QELineEdit (const QString & *variableName*, QWidget \**parent* = 0)

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

## 9.103.3 Member Function Documentation

### 9.103.3.1 void QELineEdit::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change. Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

## 9.103.4 Property Documentation

### 9.103.4.1 bool QELineEdit::addUnits [read, write]

If true (default), add engineering units supplied with the data.

### 9.103.4.2 ArrayActions QELineEdit::arrayAction [read, write]

Text formatting option for array data. Default is ASCII. Options are:

- ASCII - treat array as a single text string. For example an array of three characters 'a' 'b' 'c' will be formatted as 'abc'.
- APPEND - treat array as an array of numbers and format a string containing them all with a space between each. For example, an array of three numbers 10, 11 and 12 will be formatted as '10 11 12'.

- INDEX - Extract a single item from the array. The item is then formatted as any other non array data would be. The item selected is determined by the arrayIndex property. For example, if arrayIndex property is 1, an array of three numbers 10, 11 and 12 will be formatted as '11'.

#### 9.103.4.3 Formats QLineEdit::format [read, write]

Format to apply to data. Default is 'Default' in which case the data type supplied with the data determines how the data is formatted. For all other options, an attempt is made to format the data as requested (whatever its native form).

#### 9.103.4.4 bool QLineEdit::leadingZero [read, write]

If true (default), always add a leading zero when formatting numbers.

#### 9.103.4.5 QString QLineEdit::localEnumeration [read, write]

An enumeration list used to data values. Used only when the formatting option is 'local enumeration'. Value is converted to an integer and used to select a string from this list.

Format is:

[[<|<=|=|=|>|=|>]value1]\* : string1 , [[<|<=|=|=|>|=|>]value2]\* : string2 ,  
[[<|<=|=|=|>|=|>]value3]\* : string3 , ...

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
>= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm"  
<2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than  
2" 3:"Beamline Available", \*:"" "Pump Off":"OH NO!, the pump is OFF!","Pump  
On":"It's OK, the pump is on"

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10:""'

A range of numbers can be covered by a pair of values as in the following example:  
>=4:"Between 4 and 8",<=8:"Between 4 and 8"

**9.103.4.6 Notations QELineEdit::notation [read, write]**

Notation used for numerical formatting. Default is fixed.

**9.103.4.7 int QELineEdit::precision [read, write]**

Precision used when formatting floating point numbers. The default is 4. This is only used if useDbPrecision is false.

**9.103.4.8 int QELineEdit::radix [read, write]**

Base used for when formatting integers. Default is 10 (duh!)

**9.103.4.9 Separators QELineEdit::separator [read, write]**

Separators used for integer and fixed point formatting. Default is None.

**9.103.4.10 bool QELineEdit::trailingZeros [read, write]**

If true (default), always remove any trailing zeros when formatting numbers.

**9.103.4.11 bool QELineEdit::useDbPrecision [read, write]**

If true (default), format floating point numbers using the precision supplied with the data. If false, the precision property is used.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELineEdit/QELineEdit.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELineEdit/QELineEdit.cpp

## 9.104 QELineEditManager Class Reference

### Public Member Functions

- **QELineEditManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELineEdit/QELineEditManager.h

## 9.105 QELink Class Reference

### Public Types

- enum **conditions** {
   
    **CONDITION\_EQ**, **CONDITION\_NE**, **CONDITION\_GT**, **CONDITION\_GE**,
   
    **CONDITION\_LT**, **CONDITION\_LE** }
- enum **ConditionNames** {
   
    **Equal** = QELink::**CONDITION\_EQ**, **NotEqual** = QELink::**CONDITION\_NE**,
 **GreaterThan** = QELink::**CONDITION\_GT**, **GreaterThanOrEqual** =
 QELink::**CONDITION\_GE**,
   
    **LessThan** = QELink::**CONDITION\_LT**, **LessThanOrEqual** =
 QELink::**CONDITION\_LE** }

### Public Slots

- void **in** (const bool &in)
- void **in** (const int &in)
- void **in** (const long &in)
- void **in** (const qlonglong &in)
- void **in** (const double &in)
- void **in** (const QString &in)
- void **autoFillBackground** (const bool &enable)

### Signals

- void **out** (const bool &out)
- void **out** (const int &out)
- void **out** (const long &out)
- void **out** (const qlonglong &out)
- void **out** (const double &out)
- void **out** (const QString &out)

### Public Member Functions

- **QELink** (QWidget \*parent=0)
- void **setCondition** (conditions conditionIn)
- conditions **getCondition** ()
- void **setComparisonValue** (QString comparisonValue)
- QString **getComparisonValue** ()
- void **setSignalTrue** (bool signalTrue)
- bool **getSignalTrue** ()
- void **setSignalFalse** (bool signalFalse)

- bool **getSignalFalse ()**
- void **setOutTrueValue (QString outTrueValue)**
- QString **getOutTrueValue ()**
- void **setOutFalseValue (QString outFalseValue)**
- QString **getOutFalseValue ()**
- void **setConditionProperty (ConditionNames condition)**
- ConditionNames **getConditionProperty ()**

## Protected Attributes

- conditions **condition**
- QVariant **comparisonValue**
- bool **signalTrue**
- bool **signalFalse**
- QVariant **outTrueValue**
- QVariant **outFalseValue**

## Properties

- ConditionNames **condition**
- QString **comparisonValue**
- QString **outTrueValue**
- QString **outFalseValue**
- bool **runVisible**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELink/QELink.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELink/QELink.cpp

## 9.106 QELocalEnumeration Class Reference

```
#include <QELocalEnumeration.h>
```

### Classes

- class **localEnumerationItem**

### Public Member Functions

- [QELocalEnumeration \(\)](#)
- [QELocalEnumeration \(const QString &localEnumeration\)](#)
- void [setLocalEnumeration \(const QString &localEnumeration\)](#)
- QString [getLocalEnumeration \(\) const](#)
- bool [isDefined \(\) const](#)
- QVariant [valueToText \(const QVariant &value, bool &match\) const](#)
- QVariant [textToValue \(const QString &text, bool &ok\) const](#)
- int [textToInt \(const QString &text, bool &ok\) const](#)
- double [texttoDouble \(const QString &text, bool &ok\) const](#)

#### 9.106.1 Detailed Description

This class allows a user defined two-way value to enumeration map. The map is define using a single string, typically a widget property string. This may then be used to replace the enumeration values provided by EPICS and/or provide an enueration set of more than 16 values. See [setLocalEnumeration\(\)](#) for the use of 'localEnumeration'.

This functionality that this class provided was formerly embedded within [QEString-Formatting](#).

#### 9.106.2 Constructor & Destructor Documentation

##### 9.106.2.1 QELocalEnumeration::QELocalEnumeration ()

Constructors

##### 9.106.2.2 QELocalEnumeration::QELocalEnumeration (const QString & *localEnumeration*)

Constructor with localEnumeration

### 9.106.3 Member Function Documentation

#### 9.106.3.1 `QString QELocalEnumeration::getLocalEnumeration () const`

Get the local enumeration strings. See [setLocalEnumeration\(\)](#) for the use of 'localEnumeration'.

#### 9.106.3.2 `bool QELocalEnumeration::isDefined () const`

Evaluates: `getLocalEnumeration.count() > 0`

#### 9.106.3.3 `void QELocalEnumeration::setLocalEnumeration (const QString & localEnumeration)`

Parse the local enumeration string.

Format is:

`[[<|<=|=|=|>|=|>]value1|*] : string1 , [[<|<=|=|=|>|=|>]value2|*] : string2 , [[<|<=|=|=|>|=|>]value3|*] : string3 , ...`

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
 >= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

`0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm" <2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than 2" 3:"Beamline Available", *:"" "Pump Off":"OH NO!, the pump is OFF!","Pump On":"It's OK, the pump is on"`

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10:""'

A range of numbers can be covered by a pair of values as in the following example:  
`>=4:"Between 4 and 8",<=8:"Between 4 and 8"`

Will completely re-initialises the object.

#### 9.106.3.4 `double QELocalEnumeration::text.ToDouble (const QString & text, bool & ok) const`

Generate a double value given a string, using formatting defined within this class. If the value can be formatted the formatted value is returned and 'ok' is true. If the value

can't be formatted then 0.0 is returned and 'ok' is false.

**9.106.3.5 int QELocalEnumeration::textToInt (const QString & *text*, bool & *ok*) const**

Generate an integer value given a string, using formatting defined within this class. If the value can be formatted the formatted value is returned and 'ok' is true. If the value can't be formatted then 0 is returned and 'ok' is false.

**9.106.3.6 QVariant QELocalEnumeration::textToValue (const QString & *text*, bool & *ok*) const**

Generate a value given a string, using formatting defined within this class. If the value can be formatted the formatted value is returned and 'ok' is true. If the value can't be formatted an error string is returned and 'ok' is false

**9.106.3.7 QString QELocalEnumeration::valueToText (const QVariant & *value*, bool & *match*) const**

Format a variant value using local enumeration list. If the value is numeric, then the value is compared to the numeric interpretation of the enumeration values, if the value is textual, then the value is compared to the textual enumeration values.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QELocalEnumeration.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QELocalEnumeration.cpp

## 9.107 QELog Class Reference

### Public Types

- enum **optionsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }
- enum **MessageFilterOptions** { **Any** = UserMessage::MESSAGE\_FILTER\_ - ANY, **Match** = UserMessage::MESSAGE\_FILTER\_MATCH, **None** = UserMessage::MESSAGE\_FILTER\_NONE }
- enum **UserLevels** { **User** = userLevelTypes::USERLEVEL\_USER, **Scientist** = userLevelTypes::USERLEVEL\_SCIENTIST, **Engineer** = userLevelTypes::USERLEVEL\_ENGINEER }
- enum **DisplayAlarmStateOptions** { **Never** = standardProperties::DISPLAY\_ - ALARM\_STATE\_NEVER, **Always** = standardProperties::DISPLAY\_ - ALARM\_STATE\_ALWAYS, **WhenInAlarm** = standardProperties::DISPLAY\_ - ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void **setManagedVisible** (bool v)

### Public Member Functions

- **QELog** (QWidget \*parent=0)
- void  **setShowColumnType** (bool pValue)
- bool **getShowColumnType** ()
- void  **setShowColumnMessage** (bool pValue)
- bool **getShowColumnMessage** ()
- void  **setShowMessageFilter** (bool pValue)
- bool **getShowMessageFilter** ()
- void  **setShowClear** (bool pValue)
- bool **getShowClear** ()
- void  **setShowSave** (bool pValue)
- bool **getShowSave** ()
- void  **setOptionsLayout** (int pValue)
- int **getOptionsLayout** ()
- void  **setScrollToBottom** (bool pValue)
- bool **getScrollToBottom** ()
- void  **setInfoColor** (QColor pValue)
- QColor  **getInfoColor** ()
- void  **setWarningColor** (QColor pValue)
- QColor  **getWarningColor** ()
- void  **setErrorColor** (QColor pValue)
- QColor  **getErrorColor** ()

- void **clearLog** ()
- void **addLog** (int pType, QString pMessage)
- void **refreshLog** ()
- void **setOptionsLayoutProperty** (optionsLayoutProperty pOptionsLayout)
- optionsLayoutProperty **getOptionsLayoutProperty** ()
- MessageFilterOptions **getMessageFormFilter** ()
- void **setMessageFormFilter** (MessageFilterOptions messageFormFilter)
- MessageFilterOptions **getMessageSourceFilter** ()
- void **setMessageSourceFilter** (MessageFilterOptions messageSourceFilter)
- **UserLevels getUserLevelVisibilityProperty** ()
 

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void **setUserLevelVisibilityProperty** (**UserLevels** level)
 

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- **UserLevels getUserLevelEnabledProperty** ()
 

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void **setUserLevelEnabledProperty** (**UserLevels** level)
 

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- **DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty** ()
 

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void **setDisplayAlarmStateOptionProperty** (**DisplayAlarmStateOptions** option)
 

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

## Protected Attributes

- **\_QTableWidgetLog \* qTableWidgetLog**
- **QCheckBox \* qCheckBoxInfoMessage**
- **QCheckBox \* qCheckBoxWarningMessage**
- **QCheckBox \* qCheckBoxErrorMessage**
- **QPushButton \* qPushButtonClear**
- **QPushButton \* qPushButtonSave**
- **QColor qColorInfo**
- **QColor qColorWarning**
- **QColor qColorError**
- **bool scrollToBottom**
- **int optionsLayout**

## Properties

- bool **showColumnTime**
- bool **showColumnType**
- bool **showColumnMessage**
- bool **showMessageFilter**
- bool **showClear**
- bool **showSave**
- optionsLayoutProperty **optionsLayout**
- QColor **infoColor**
- QColor **warningColor**
- QColor **errorColor**
- MessageFilterOptions **messageFormFilter**
- MessageFilterOptions **messageSourceFilter**
- bool **variableAsToolTip**
- bool **allowDrop**
- bool **visible**
- unsigned **int**
- QString **styleSheet**
- QString **defaultStyle**
- QString **userLevelUserStyle**
- QString **userLevelScientistStyle**
- QString **userLevelEngineerStyle**
- UserLevels **userLevelVisibility**
- UserLevels **userLevelEnabled**
- bool **displayAlarmState**
- DisplayAlarmStateOptions **displayAlarmStateOption**

### 9.107.1 Member Enumeration Documentation

#### 9.107.1.1 enum QELog::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

**Never** Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

**Always** Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

**WhenInAlarm** Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

### 9.107.1.2 enum QELog::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

#### Enumerator:

*User* Refer to `USERLEVEL_USER` for details.

*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.

*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

## 9.107.2 Member Function Documentation

### 9.107.2.1 void QELog::setManagedVisible (bool *v*) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.107.3 Property Documentation

### 9.107.3.1 bool QELog::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

### 9.107.3.2 QString QELog::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

### 9.107.3.3 bool QELog::displayAlarmState [read, write]

DEPRECATED. USE `displayAlarmStateOption` INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

### 9.107.3.4 DisplayAlarmStateOptions QELog::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm' If 'Never' widget will never indicate the alarm state

of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.107.3.5 `unsigned QELog::int [read, write]`

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

#### 9.107.3.6 `QString QELog::styleSheet [read, write]`

Hide style sheet from designer as style calculation by the styleManager and not directly settable per se. This also stops transient styles being saved to the ui file.

#### 9.107.3.7 `UserLevels QELog::userLevelEnabled [read, write]`

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmaticaly through `setUserLevel()`. Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

#### 9.107.3.8 `QString QELog::userLevelEngineerStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.107.3.9 `QString QELog::userLevelScientistStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.107.3.10 QString QELog::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.107.3.11 UserLevels QELog::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.107.3.12 bool QELog::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.107.3.13 bool QELog::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELog/QELog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELog/QELog.cpp

## 9.108 QELogin Class Reference

### Signals

- void **login** ()

### Public Member Functions

- **QELogin** (QWidget \*pParent=0)
- bool **login** (userLevelTypes::userLevels level, QString password)
- QString **getPriorityUserPassword** ()
- QString **getPriorityScientistPassword** ()
- QString **getPriorityEngineerPassword** ()
- void **setUserPassword** (QString pValue)
- QString **getUserPassword** ()
- void **setScientistPassword** (QString pValue)
- QString **getScientistPassword** ()
- void **setEngineerPassword** (QString pValue)
- QString **getEngineerPassword** ()
- void **setCompactStyle** (bool compactStyle)
- bool **getCompactStyle** ()
- void **setStatusOnly** (bool statusOnlyIn)
- bool **getStatusOnly** ()
- QString **getUserTypeName** (userLevelTypes::userLevels type)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELogin/QELogin.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELogin/QELogin.cpp

## 9.109 QELoginDialog Class Reference

### Public Member Functions

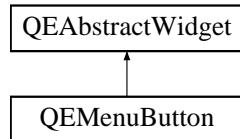
- **QELoginDialog** ([QELogin](#) \*ownerIn)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELogin/QELogin.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QELogin/QELogin.cpp

## 9.110 QEMenuBar Class Reference

#include <QEMenuBar.h> Inheritance diagram for QEMenuBar::



### Signals

- void **newGui** (const QEActionRequests &request)  
*Internal use only. Request a new GUI is created. Typically, this is caught by the QEGui application.*

### Public Member Functions

- **QEMenuBar** (QWidget \*parent=0)
- void **setSubstitutionsProperty** (const QString &substitutions)
- QString **getSubstitutionsProperty** () const
- void **setMenuString** (const QString &s)
- QString **getMenuString** () const

### Protected Member Functions

- QSize **sizeHint** () const

### Protected Attributes

- QMenu \* **buttonMainMenu**
- QString **theMenuString**

### Properties

- bool **allowDrop**
- bool **variableAsToolTip**
- DisplayAlarmStateOptions **displayAlarmStateOption**
- QString **defaultSubstitutions**
- QString **menuEntries**

*Specifies the menu entry values, encoded and an XML string.*

### 9.110.1 Detailed Description

This widget provides a menu attached to a button. Each menu\_submenu can trigger same items as a [QEPushButton](#), i.e: a) write a value to a PV (click only); and/or b) run a nominated program; and/or c) open a specified ui file.

### 9.110.2 Property Documentation

#### 9.110.2.1 bool QEMenuButton::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

Reimplemented from [QEAbstractWidget](#).

#### 9.110.2.2 QString QEMenuButton::defaultSubstitutions [read, write]

Default macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to program names and arguments etc.

#### 9.110.2.3 DisplayAlarmStateOptions QEMenuButton::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm' If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

Reimplemented from [QEAbstractWidget](#).

#### 9.110.2.4 bool QEMenuButton::variableAsToolTip [read, write]

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

Reimplemented from [QEAbstractWidget](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButton.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButton.cpp

## 9.111 QEMenuButtonData Class Reference

### Public Member Functions

- `QVariant toVariant () const`
- `bool setValue (const QVariant &data)`

### Static Public Member Functions

- `static QString psoToString (const applicationLauncher::programStartupOptions value)`
- `static applicationLauncher::programStartupOptions stringToPso (const QString &image)`
- `static QString optionToString (const QEActionRequests::Options value)`
- `static QEActionRequests::Options stringToOption (const QString &image)`
- `static QString formatToString (const QEStringFormatting::formats value)`
- `static QEStringFormatting::formats stringToFormat (const QString &image)`
- `static QString join (const QStringList &x)`
- `static QStringList split (const QString &x)`

### Public Attributes

- `bool separator`
- `QString programName`
- `QStringList programArguments`
- `applicationLauncher::programStartupOptions programStartupOption`
- `QString uiFilename`
- `QString prioritySubstitutions`
- `QEActionRequests::Options creationOption`
- `QString customisationName`
- `QString variable`
- `QString variableValue`
- `QEStringFormatting::formats format`

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButtonData.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButtonData.cpp`

## 9.112 QEMenuButtonItem Class Reference

```
#include <QEMenuButtonItem.h>
```

### Public Member Functions

- **QEMenuButtonItem** (const QString &name, const bool isSubMenuContainer, **QEMenuButtonItem** \*parent=NULL)
- **QString getName () const**
- **bool getIsSubMenuContainer () const**
- **QDomElement createDomElement (QDomDocument &doc) const**
- **bool extractFromDomElement (const QDomElement &element)**
- **QAction \* constructAction (QMenu \*parent)**
- **QMenu \* constructMenu (QMenu \*parent)**
- **int columnCount () const**
- **QEMenuButtonItem \* getChild (const int position) const**
- **QEMenuButtonItem \* getParent () const**
- **int childCount () const**
- **int childPosition () const**
- **QVariant getData (const int column) const**
- **bool insertChild (const int position, **QEMenuButtonItem** \*child)**
- **bool removeChildren (const int position, const int count)**

### Public Attributes

- **QString name**
- **bool isSubMenuContainer**
- **QEMenuButtonData data**

#### 9.112.1 Detailed Description

This class is based on the TreeItem example specified in:  
<http://qt-project.org/doc/qt-4.8/itemviews-editabletreemodel.html>

Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies). Contact:  
<http://www.qt-project.org/legal>

Differences: there is no itemData variant array - these values calculated as an when needed. Also some function name changes such as parent => getParent and some changes just to follow QE prefered style.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButt
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButt

## 9.113 QE\_MenuButtonManager Class Reference

### Public Member Functions

- **QE\_MenuButtonManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

### Protected Attributes

- bool **initialized**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QE\_MenuButton/QE\_MenuButtonManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QE\_MenuButton/QE\_MenuButtonManager.c

## 9.114 QE\_MenuButtonModel Class Reference

### Public Member Functions

- **QE\_MenuButtonModel** (QObject \*parent=0)
- bool **parseXml** (const QString &xml)
- QString **serialiseXml** () const
- bool **constructMenu** (QMenu \*rootMenu)
- **QE\_MenuButtonItem** \* **indexToItem** (const QModelIndex &index) const
- QModelIndex **getIndex** (const **QE\_MenuButtonItem** \*item) const
- bool **addItemToModel** (**QE\_MenuButtonItem** \*item, **QE\_MenuButtonItem** \*parentItem, const int position=-1)
- bool **removeItemFromModel** (**QE\_MenuButtonItem** \*item)
- void **modelUpdated** ()
- void **itemUpdated** (const **QE\_MenuButtonItem** \*item)

### Static Public Member Functions

- static QString **emptyXml** ()

### Protected Member Functions

- QVariant **data** (const QModelIndex &index, int role) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const
- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const
- QModelIndex **parent** (const QModelIndex &child) const
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const
- Qt::ItemFlags **flags** (const QModelIndex &index) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)
- bool **setHeaderData** (int section, Qt::Orientation orientation, const QVariant &value, int role=Qt::EditRole)
- bool **insertRows** (int position, int rows, const QModelIndex &parent=QModelIndex())
- bool **removeRows** (int position, int rows, const QModelIndex &parent=QModelIndex())
- Qt::DropActions **supportedDropActions** () const
- QStringList **mimeTypes** () const
- QMimeData \* **mimeData** (const QModelIndexList &indexes) const
- bool **dropMimeData** (const QMimeData \*data, Qt::DropAction action, int row, int column, const QModelIndex &parent)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuBar/QEMenuBarModel.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuBar/QEMenuBarModel.cpp

## 9.115 QEMenuButtonSetupDialog Class Reference

### Public Member Functions

- **QEMenuButtonSetupDialog** ([QEMenuButton](#) \*owner, [QWidget](#) \*parent)

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

- [/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButton.h](#)
- [/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButton.cpp](#)

## 9.116 QEMenuBarTaskMenu Class Reference

### Public Member Functions

- **QEMenuBarTaskMenu** ([QEMenuBar](#) \*menuButton, [QObject](#) \*parent)
- **QAction \*** **preferredEditAction** () const
- **QList< QAction \* >** **taskActions** () const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuBar/QEMenuBarManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuBar/QEMenuBarManager.c

## 9.117 **QEMenuButtonTaskMenuFactory** Class Reference

### Public Member Functions

- **QEMenuButtonTaskMenuFactory** (QExtensionManager \*parent=0)

### Protected Member Functions

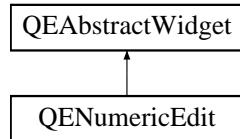
- QObject \* **createExtension** (QObject \*object, const QString &iid, QObject \*parent) const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButt
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEMenuButton/QEMenuButt

## 9.118 QENumericEdit Class Reference

Inheritance diagram for QENumericEdit::



### Public Slots

- void [setDefaultStyle](#) (const QString &style)  
*Update the default style applied to this widget.*

### Signals

- void [dbValueChanged](#) (const QString &out)
- void [dbValueChanged](#) (const int &out)
- void [dbValueChanged](#) (const long &out)
- void [dbValueChanged](#) (const qlonglong &out)
- void [dbValueChanged](#) (const double &out)
- void [dbValueChanged](#) (const bool &out)
- void [dbConnectionChanged](#) (const bool &isConnected)

### Public Member Functions

- [QENumericEdit](#) (QWidget \*parent=0)
- [QENumericEdit](#) (const QString &variableName, QWidget \*parent=0)
- virtual ~[QENumericEdit](#) ()  
*Destruction.*
- void [writeNow](#) ()
- double [getValue](#) () const
- void [setValue](#) (const double value, const bool isUserUpdate=false)
- double [getNumericValue](#) () const
- void [setNumericValue](#) (const double value, const bool isUserUpdate=false)
- void [setAutoScale](#) (const bool)
- bool [getAutoScale](#) () const
- void [setAddUnits](#) (const bool)
- bool [getAddUnits](#) () const
- void [setWriteOnLoseFocus](#) (const bool)
- bool [getWriteOnLoseFocus](#) () const

- void **setWriteOnEnter** (const bool)
- bool **getWriteOnEnter** () const
- void **setWriteOnFinish** (const bool)
- bool **getWriteOnFinish** () const
- void **setWriteOnChange** (const bool)
- bool **getWriteOnChange** () const
- void **setConfirmWrite** (const bool)
- bool **getConfirmWrite** () const
- void **setAllowFocusUpdate** (const bool)
- bool **getAllowFocusUpdate** () const
- void **setLeadingZeros** (const int value)
- int **getLeadingZeros** () const
- void **setPrecision** (const int value)
- int **getPrecision** () const
- void **setMinimum** (const double value)
- double **getMinimum** () const
- void **setMaximum** (const double value)
- double **getMaximum** () const
- QString **getCleanText** () const

## Protected Member Functions

- void **fontChange** (const QFont &font)
- void **focusInEvent** (QFocusEvent \*event)
- QMenu \* **getDefaultContextMenu** ()
- void **activated** ()
- void **establishConnection** (unsigned int variableIndex)
- [qcaobject::QCaObject](#) \* **createQcaItem** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)

## Properties

- QString **variable**
- QString **variableSubstitutions**
- int **arrayIndex**
- bool **frame**
- Qt::Alignment **alignment**
- QString **cleanText**
- bool **autoScale**
- [QNumericEdit::Notations](#) **notation**
- QEFixedPointRadix::Radicies **radix**

- QEFixedPointRadix::Separators [separator](#)
- int [leadingZeros](#)
- int [precision](#)
- double [minimum](#)
- double [maximum](#)
- bool [addUnits](#)
- bool [writeOnLoseFocus](#)
- bool [writeOnEnter](#)
- bool [writeOnFinish](#)
- bool [writeOnChange](#)
- bool [confirmWrite](#)
- bool [allowFocusUpdate](#)

## 9.118.1 Constructor & Destructor Documentation

### 9.118.1.1 QENumericEdit::QENumericEdit (QWidget \* *parent* = 0) [explicit]

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

### 9.118.1.2 QENumericEdit::QENumericEdit (const QString & *variableName*, QWidget \* *parent* = 0) [explicit]

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

## 9.118.2 Member Function Documentation

### 9.118.2.1 void QENumericEdit::dbConnectionChanged (const bool & *isConnected*) [signal]

Sent when the widget state updated following a channel connection change Applied to provary varible.

### 9.118.2.2 void QENumericEdit::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

### 9.118.3 Property Documentation

#### 9.118.3.1 bool QENumericEdit::addUnits [read, write]

If true (default), add engineering units supplied with the data.

#### 9.118.3.2 Qt::Alignment QENumericEdit::alignment [read, write]

This property holds the alignment of the numeric edit. Both horizontal and vertical alignment is allowed here, Qt::AlignJustify will map to Qt::AlignLeft. By default, this property contains a combination of Qt::AlignRight and Qt::AlignVCenter.

#### 9.118.3.3 bool QENumericEdit::allowFocusUpdate [read, write]

Allow updated while widget has focus - defaults to false.

#### 9.118.3.4 int QENumericEdit::arrayIndex [read, write]

Index used to select a single item of data for processing. The default is 0.

#### 9.118.3.5 bool QENumericEdit::autoScale [read, write]

If true (default), display and editing of numbers using the PV's precision and control limits supplied with the data. If false, the precision, leadingZeros, minimum and maximum properties values are used.

#### 9.118.3.6 QString QENumericEdit::cleanText [read]

This property holds the displayed text. Not a property available to designer.

#### 9.118.3.7 bool QENumericEdit::confirmWrite [read, write]

Sets if this widget will ask for confirmation (using a dialog box) prior to writing data. Default is 'false' (will not ask for confirmation (using a dialog box) prior to writing data). Note: writeOnChange and confirmWrite are mutually exclusive.

#### 9.118.3.8 bool QENumericEdit::frame [read, write]

This property holds whether the numeric edit draws itself with a frame. If enabled (the default) the numeric edit draws itself inside a frame, otherwise the line edit draws itself without any frame.

**9.118.3.9 int QENumericEdit::leadingZeros [read, write]**

Specifies the number of leading zeros. The default is 3. This is only used if autoScale is false. When autoScale is true the PV's control range is used to determine the number of required leading zeros. Strictly speaking, this should be an unsigned int, but designer properties editor much 'nicer' with ints.

**9.118.3.10 double QENumericEdit::maximum [read, write]**

Specifies the maximum allowed value. This is only used if autoScale is false.

**9.118.3.11 double QENumericEdit::minimum [read, write]**

Specifies the mimimum allowed value. This is only used if autoScale is false.

**9.118.3.12 QNumericEdit::Notations QENumericEdit::notation [read, write]**

Notation used for formatting/editing. Default is fixed.

**9.118.3.13 int QENumericEdit::precision [read, write]**

Precision used for the display and editing of numbers. The default is 2. This is only used if autoScale is false. When autoScale is true the PV's precision is used. Strictly speaking, this should be an unsigned int, but designer properties editor much 'nicer' with ints.

**9.118.3.14 QEFixedPointRadix::Radicies QENumericEdit::radix [read, write]**

Specify radix, default is Decimal.

**9.118.3.15 QEFixedPointRadix::Separators QENumericEdit::separator [read, write]**

Specify digit 'thousands' separator character, default is none.

**9.118.3.16 QString QENumericEdit::variable [read, write]**

EPICS variable name (CA PV)

**9.118.3.17 QString QENumericEdit::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

**9.118.3.18 bool QENumericEdit::writeOnChange [read, write]**

If true the widget writes to the PV as value are changes. If false (default) a write only occurs when as per writeOnLoseFocus, writeOnEnter and/or writeOnFinish values. Note: writeOnChange and confirmWrite are mutually exclusive.

**9.118.3.19 bool QENumericEdit::writeOnEnter [read, write]**

Sets if this widget writes any changes when the user presses 'enter'. Note, the current value will be written even if the user has not changed it. Default is 'true' (writes any changes when the user presses 'enter').

**9.118.3.20 bool QENumericEdit::writeOnFinish [read, write]**

Sets if this widget writes any changes when the user finished editing (the underlying QLineEdit 'editingFinished' signal is emitted). No writing occurs if no changes were made. Default is 'true' (writes any changes when the QLineEdit 'editingFinished' signal is emitted).

**9.118.3.21 bool QENumericEdit::writeOnLoseFocus [read, write]**

Sets if this widget automatically writes any changes when it loses focus. Default is 'false' (does not write any changes when it loses focus).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QENumericEdit/QENumericEdit.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QENumericEdit/QENumericEdit.cpp

## 9.119 QENumericEditManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QENumericEdit/QENumericEditManager.h

## 9.120 QEPeriodic Class Reference

### Classes

- struct [elementInfoStruct](#)
- struct [userInfoStructArray](#)

### Public Types

- enum **variableTypes** {
   
**VARIABLE\_TYPE\_NUMBER**, **VARIABLE\_TYPE\_ATOMIC\_WEIGHT**,
 **VARIABLE\_TYPE\_MELTING\_POINT**, **VARIABLE\_TYPE\_BOILING\_POINT**,
   
**VARIABLE\_TYPE\_DENSITY**, **VARIABLE\_TYPE\_GROUP**,
 **VARIABLE\_TYPE\_IONIZATION\_ENERGY**, **VARIABLE\_TYPE\_USER\_VALUE\_1**,
   
**VARIABLE\_TYPE\_USER\_VALUE\_2** }
- enum **presentationOptions** { **PRESENTATION\_BUTTON\_AND\_LABEL**,
 **PRESENTATION\_BUTTON\_ONLY**, **PRESENTATION\_LABEL\_ONLY** }
- enum **userInfoSourceOptions** { **USER\_INFO\_SOURCE\_TEXT**, **USER\_INFO\_SOURCE\_FILE** }
- enum **UserLevels** { **User** = userLevelTypes::USERLEVEL\_USER,
 **Scientist** = userLevelTypes::USERLEVEL\_SCIENTIST, **Engineer** =
 userLevelTypes::USERLEVEL\_ENGINEER }
- enum **DisplayAlarmStateOptions** { **Never** = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER,
 **Always** = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS,
 **WhenInAlarm** = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }
- enum **PresentationOptions** { **buttonAndLabel** =
 QEPeriodic::PRESENTATION\_BUTTON\_AND\_LABEL, **buttonOnly** =
 QEPeriodic::PRESENTATION\_BUTTON\_ONLY, **labelOnly** =
 QEPeriodic::PRESENTATION\_LABEL\_ONLY }
- enum **VariableTypes** {
   
**Number** = QEPeriodic::VARIABLE\_TYPE\_NUMBER, **atomicWeight** =
 QEPeriodic::VARIABLE\_TYPE\_ATOMIC\_WEIGHT, **meltingPoint** =
 QEPeriodic::VARIABLE\_TYPE\_MELTING\_POINT, **boilingPoint** =
 QEPeriodic::VARIABLE\_TYPE\_BOILING\_POINT,
   
**density** = QEPeriodic::VARIABLE\_TYPE\_DENSITY, **group** =
 QEPeriodic::VARIABLE\_TYPE\_GROUP, **ionizationEnergy** =
 QEPeriodic::VARIABLE\_TYPE\_IONIZATION\_ENERGY, **userValue1** =
 QEPeriodic::VARIABLE\_TYPE\_USER\_VALUE\_1,
   
**userValue2** = QEPeriodic::VARIABLE\_TYPE\_USER\_VALUE\_2 }
- enum **UserInfoSourceOptions** { **userInfoSourceText** = QEPeriodic::USER\_INFO\_SOURCE\_TEXT,
 **userInfoSourceFile** = QEPeriodic::USER\_INFO\_SOURCE\_FILE }

## Public Slots

- void **setElement** (const QString symbol)

## Signals

- void **userElementChanged** (const QString &symbol)  
*Sent when the element is changed by the user selecting an element.*
- void **dbValueChanged** (const double &out)
- void **dbElementChanged** (const QString &out)
- void **requestResend** ()  
*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*

## Public Member Functions

- **QEPeriodic** (QWidget \*parent=0)
- **QEPeriodic** (const QString &variableName, QWidget \*parent=0)
- void **setSubscribe** (bool subscribe)
- bool **getSubscribe** ()
- void **setPresentationOption** (presentationOptions presentationOptionIn)
- presentationOptions **getPresentationOption** ()
- void **setVariableType1** (variableTypes variableType1In)
- variableTypes **getVariableType1** ()
- void **setVariableType2** (variableTypes variableType2In)
- variableTypes **getVariableType2** ()
- void **setVariableTolerance1** (double variableTolerance1In)
- double **getVariableTolerance1** ()
- void **setVariableTolerance2** (double variableTolerance2In)
- double **getVariableTolerance2** ()
- void **setUserInfo** (QString userInfo)
- QString **getUserInfo** ()
- void **setUserInfoText** (QString userInfo)
- QString **getUserInfoText** ()
- void **setUserInfoFile** (QString userInfoFileIn)
- QString **getUserInfoFile** ()
- void **setUserInfoSourceOption** (userInfoSourceOptions userInfoSourceOptionIn)
- userInfoSourceOptions **getUserInfoSourceOption** ()
- void **updateUserInfoSource** ()
- bool **getElementValues** (QString symbol, double \*value1, double \*value2)
- QString **getSelectedSymbol** ()
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)

*Property access function for [variableSubstitutions](#) property. This has special behaviour to work well within designer.*

- **QString getVariableNameSubstitutionsProperty ()**

*Property access function for [variableSubstitutions](#) property. This has special behaviour to work well within designer.*

- **UserLevels getUserLevelVisibilityProperty ()**

*Access function for [userLevelVisibility](#) property - refer to [userLevelVisibility](#) property for details.*

- **void setUserLevelVisibilityProperty (UserLevels level)**

*Access function for [userLevelVisibility](#) property - refer to [userLevelVisibility](#) property for details.*

- **UserLevels getUserLevelEnabledProperty ()**

*Access function for [userLevelEnabled](#) property - refer to [userLevelEnabled](#) property for details.*

- **void setUserLevelEnabledProperty (UserLevels level)**

*Access function for [userLevelEnabled](#) property - refer to [userLevelEnabled](#) property for details.*

- **DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()**

*Access function for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property for details.*

- **void setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)**

*Access function for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property for details.*

- **void setPresentationOptionProperty (PresentationOptions presentationOption)**

- **PresentationOptions getPresentationOptionProperty ()**

- **void setVariableType1Property (VariableTypes variableType)**

- **void setVariableType2Property (VariableTypes variableType)**

- **VariableTypes getVariableType1Property ()**

- **VariableTypes getVariableType2Property ()**

- **void setUserInfoSourceOptionProperty (UserInfoSourceOptions userInfoSourceOption)**

- **UserInfoSourceOptions getUserInfoSourceOptionProperty ()**

## Public Attributes

- **userInfoStruct userInfo [NUM\_ELEMENTS]**

## Static Public Attributes

- static `elementInfoStruct elementInfo [NUM_ELEMENTS]`

## Protected Types

- enum `variableIndexes {`
- `WRITE_VARIABLE_1, WRITE_VARIABLE_2, READ_VARIABLE_1,`
- `READ_VARIABLE_2,`
- `QEPPERIODIC_NUM_VARIABLES }`

## Protected Member Functions

- void `establishConnection (unsigned int variableIndex)`
- void `dragEnterEvent (QDragEnterEvent *event)`
- void `dropEvent (QDropEvent *event)`
- void `mousePressEvent (QMouseEvent *event)`
- void `setDrop (QVariant drop)`
- `QVariant getDrop ()`
- `QString copyVariable ()`
- `QVariant copyData ()`
- void `paste (QVariant s)`

## Protected Attributes

- `QEFloatingFormatting floatingFormatting`
- bool `localEnabled`
- bool `allowDrop`
- variableTypes `variableType1`
- variableTypes `variableType2`
- double `variableTolerance1`
- double `variableTolerance2`

## Properties

- `QString writeButtonVariable1`
- `QString writeButtonVariable2`
- `QString readbackLabelVariable1`
- `QString readbackLabelVariable2`
- `QString variableSubstitutions`
- bool `subscribe`
- bool `variableAsToolTip`
- bool `visible`
- unsigned `int`

- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- `PresentationOptions presentationOption`
- `VariableTypes variableType1`
- `VariableTypes variableType2`
- `QString userInfo`
- `UserInfoSourceOptions userInfoSourceOption`

### 9.120.1 Member Enumeration Documentation

#### 9.120.1.1 enum QEPeriodic::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

**Enumerator:**

*Never* Refer to `DISPLAY_ALARM_STATE_NEVER` for details.

*Always* Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.

*WhenInAlarm* Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

#### 9.120.1.2 enum QEPeriodic::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

**Enumerator:**

*User* Refer to `USERLEVEL_USER` for details.

*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.

*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

### 9.120.2 Member Function Documentation

#### 9.120.2.1 void QEPeriodic::dbElementChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.120.2.2 void QEPeriodic::dbValueChanged (const double & *out*) [signal]**

Sent when the widget is updated following a data change. Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.120.3 Member Data Documentation****9.120.3.1 bool QEPeriodic::allowDrop [read, write, protected]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.120.4 Property Documentation****9.120.4.1 bool QEPeriodic::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.120.4.2 DisplayAlarmStateOptions QEPeriodic::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.120.4.3 unsigned QEPeriodic::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.120.4.4 QString QEPeriodic::readbackLabelVariable1 [read, write]**

EPICS variable name (CA PV). This variable is used to read the value to the first of two positioners to determine which (if any) element is currently selected.

**9.120.4.5 QString QEPeriodic::readbackLabelVariable2 [read, write]**

EPICS variable name (CA PV). This variable is used to read the value to the second of two positioners to determine which (if any) element is currently selected.

**9.120.4.6 bool QEPeriodic::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.120.4.7 UserLevels QEPeriodic::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.120.4.8 QString QEPeriodic::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.120.4.9 QString QEPeriodic::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.120.4.10 QString QEPeriodic::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.120.4.11 UserLevels QEPeriodic::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.120.4.12 bool QEPeriodic::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.120.4.13 QString QEPeriodic::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'SAMPLE=SAM1, NAME = "Ref foil"' These substitutions are applied to all the variable names.

**9.120.4.14 bool QEPeriodic::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

**9.120.4.15 QString QEPeriodic::writeButtonVariable1 [read, write]**

EPICS variable name (CA PV). This variable is used to write a value to the first of two positioners that will position the select element.

**9.120.4.16 QString QEPeriodic::writeButtonVariable2 [read, write]**

EPICS variable name (CA PV). This variable is used to write a value to the second of two positioners that will position the select element.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.cpp

## 9.121 QEPeriodicComponentData Class Reference

### Public Attributes

- unsigned int **variableIndex1**
- double **lastData1**
- bool **haveLastData1**
- unsigned int **variableIndex2**
- double **lastData2**
- bool **haveLastData2**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h

## 9.122 QEPeriodicTaskMenu Class Reference

### Public Member Functions

- **QEPeriodicTaskMenu** ([QEPeriodic](#) \*periodic, [QObject](#) \*parent)
- **QAction \*** **preferredEditAction** () const
- **QList< QAction \* >** **taskActions** () const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskMenu.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskMenuExtensi

## 9.123 QEPeriodicTaskMenuFactory Class Reference

### Public Member Functions

- **QEPeriodicTaskMenuFactory** (QExtensionManager \*parent=0)

### Protected Member Functions

- **QObject \* createExtension** (QObject \*object, const QString &iid, QObject \*parent) const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskM
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskM

## 9.124 QEpicsPV Class Reference

### Public Slots

- const QVariant & **set** (QVariant value, int delay=-1)
- void **setPV** (const QString &\_pvName="")

### Signals

- void **connectionChanged** (bool connected)
- void **connected** ()
- void **disconnected** ()
- void **valueChanged** (const QVariant &value)
- void **valueUpdated** (const QVariant &value)
- void **valueInitiated** (const QVariant &value)

### Public Member Functions

- **QEpicsPV** (const QString &\_pvName, QObject \*parent=0)
- **QEpicsPV** (QObject \*parent=0)
- const QVariant & **get** () const
- void **needUpdated** () const
- const QVariant & **getUpdated** (int delay=defaultDelay) const
- bool **isConnected** () const
- const QStringList & **getEnum** () const
- const QString & **pv** () const
- const QVariant & **getReady** (int delay=defaultDelay) const

### Static Public Member Functions

- static void **setDebugLevel** (unsigned level=0)
- static QVariant **get** (const QString &\_pvName, int delay=defaultDelay)
- static QVariant **set** (QString &\_pvName, const QVariant &value, int delay=-1)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/qepicspv.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/qepicspv.cpp

## 9.125 QEPlot Class Reference

### Public Types

- enum **UserLevels** { **User** = userLevelTypes::USERLEVEL\_USER, **Scientist** = userLevelTypes::USERLEVEL\_SCIENTIST, **Engineer** = userLevelTypes::USERLEVEL\_ENGINEER }
- enum **DisplayAlarmStateOptions** { **Never** = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, **Always** = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, **WhenInAlarm** = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }
- enum **TraceStyles** { **Lines** = QwtPlotCurve::Lines, **Sticks** = QwtPlotCurve::Sticks, **Steps** = QwtPlotCurve::Steps, **Dots** = QwtPlotCurve::Dots }

### Public Slots

- void **setManagedVisible** (bool v)

### Signals

- void **mouseMove** (const QPointF &posn)
- void **dbValueChanged** (const double &out)
- void **dbValueChanged** (const QVector< double > &out)

### Public Member Functions

- **QEPlot** (QWidget \*parent=0)
- **QEPlot** (const QString &variableName, QWidget \*parent=0)
- QSize **sizeHint** () const
- void **setYMin** (double yMin)
- double **getYMin** () const
- void **setYMax** (double yMax)
- double **getYMax** () const
- void **setAutoScale** (bool autoScale)
- bool **getAutoScale** () const
- void **setAxisEnableX** (bool axisEnableXIn)
- bool **getAxisEnableX** () const
- void **setAxisEnableY** (bool axisEnableYIn)
- bool **getAxisEnableY** () const
- QString **getTitle** () const
- void **setBackgroundColor** (QColor backgroundColor)
- QColor **getBackgroundColor** () const
- void **setTraceStyle** (QwtPlotCurve::CurveStyle traceStyle, const unsigned int variableIndex)

- QwtPlotCurve::CurveStyle **getTraceStyle** (const unsigned int variableIndex) const
- void **setTraceColor** (QColor traceColor, const unsigned int variableIndex)
- void **setTraceColor1** (QColor traceColor)
- void **setTraceColor2** (QColor traceColor)
- void **setTraceColor3** (QColor traceColor)
- void **setTraceColor4** (QColor traceColor)
- QColor **getTraceColor** (const unsigned int variableIndex) const
- QColor **getTraceColor1** () const
- QColor **getTraceColor2** () const
- QColor **getTraceColor3** () const
- QColor **getTraceColor4** () const
- void **setTraceLegend1** (QString traceLegend)
- void **setTraceLegend2** (QString traceLegend)
- void **setTraceLegend3** (QString traceLegend)
- void **setTraceLegend4** (QString traceLegend)
- QString **getTraceLegend1** () const
- QString **getTraceLegend2** () const
- QString **getTraceLegend3** () const
- QString **getTraceLegend4** () const
- void **setXUnit** (QString xUnit)
- QString **getXUnit** () const
- void **setYUnit** (QString yUnit)
- QString **getYUnit** () const
- void **setGridEnableMajorX** (bool gridEnableMajorXIn)
- void **setGridEnableMajorY** (bool gridEnableMajorYIn)
- void **setGridEnableMinorX** (bool gridEnableMinorXIn)
- void **setGridEnableMinorY** (bool gridEnableMinorYIn)
- bool **getGridEnableMajorX** () const
- bool **getGridEnableMajorY** () const
- bool **getGridEnableMinorX** () const
- bool **getGridEnableMinorY** () const
- void **setGridMajorColor** (QColor gridMajorColorIn)
- void **setGridMinorColor** (QColor gridMinorColorIn)
- QColor **getGridMajorColor** () const
- QColor **getGridMinorColor** () const
- void **setXStart** (double xStart)
- double **getXStart** () const
- void **setXIncrement** (double xIncrement)
- double **getXIncrement** () const
- void **setTimeSpan** (unsigned int timeSpan)
- unsigned int **getTimeSpan** () const
- void **setTickRate** (unsigned int tickRate)
- unsigned int **getTickRate** () const
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)

*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*

- `QString getVariableNameSubstitutionsProperty ()`

*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*

- `UserLevels getUserLevelVisibilityProperty ()`

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- `void setUserLevelVisibilityProperty (UserLevels level)`

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- `UserLevels getUserLevelEnabledProperty ()`

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- `void setUserLevelEnabledProperty (UserLevels level)`

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- `void setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)`

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- `void setTraceStyle1 (TraceStyles traceStyle)`

- `void setTraceStyle2 (TraceStyles traceStyle)`

- `void setTraceStyle3 (TraceStyles traceStyle)`

- `void setTraceStyle4 (TraceStyles traceStyle)`

- `TraceStyles getTraceStyle1 ()`

- `TraceStyles getTraceStyle2 ()`

- `TraceStyles getTraceStyle3 ()`

- `TraceStyles getTraceStyle4 ()`

## Protected Member Functions

- `void canvasMouseMove (QMouseEvent *mouseEvent)`
- `bool eventFilter (QObject *obj, QEvent *event)`
- `void establishConnection (unsigned int variableIndex)`
- `void dragEnterEvent (QDragEnterEvent *event)`
- `void dropEvent (QDropEvent *event)`

- void **mousePressEvent** (QMouseEvent \*event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)

## Protected Attributes

- [QEFloatingFormatting floatingFormatting](#)
- bool **localEnabled**
- bool [allowDrop](#)

## Properties

- QString [variable1](#)
- QString [variable2](#)
- QString [variable3](#)
- QString [variable4](#)
- QString [variableSubstitutions](#)
- bool [variableAsToolTip](#)
- bool [visible](#)
- unsigned [int](#)
- QString [styleSheet](#)
- QString [defaultStyle](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- UserLevels [userLevelVisibility](#)
- UserLevels [userLevelEnabled](#)
- bool [displayAlarmState](#)
- [DisplayAlarmStateOptions displayAlarmStateOption](#)
- QColor [traceColor1](#)
- QColor [traceColor2](#)
- QColor [traceColor3](#)
- QColor [traceColor4](#)
- TraceStyles [traceStyle1](#)
- TraceStyles [traceStyle2](#)
- TraceStyles [traceStyle3](#)
- TraceStyles [traceStyle4](#)
- QString [traceLegend1](#)
- QString [traceLegend2](#)
- QString [traceLegend3](#)
- QString [traceLegend4](#)
- QString [title](#)
- QColor [backgroundColor](#)
- QString [xUnit](#)
- QString [yUnit](#)

## 9.125.1 Member Enumeration Documentation

### 9.125.1.1 enum QEPlot::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and displayAlarmStateOptions enumeration for details.

#### Enumerator:

*Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

### 9.125.1.2 enum QEPlot::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and userLevel enumeration for details.

#### Enumerator:

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

## 9.125.2 Member Function Documentation

### 9.125.2.1 void QEPlot::dbValueChanged (const QVector< double > & out) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

### 9.125.2.2 void QEPlot::dbValueChanged (const double & out) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

### 9.125.2.3 void QEPlot::setManagedVisible (bool v) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

### 9.125.3 Member Data Documentation

#### 9.125.3.1 bool QEPlot::allowDrop [read, write, protected]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

### 9.125.4 Property Documentation

#### 9.125.4.1 QString QEPlot::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

#### 9.125.4.2 bool QEPlot::displayAlarmState [read, write]

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.125.4.3 DisplayAlarmStateOptions QEPlot::displayAlarmStateOption [read, write]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.125.4.4 unsigned QEPlot::int [read, write]

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

#### 9.125.4.5 QString QEPlot::styleSheet [read, write]

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.125.4.6 UserLevels QEPlot::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.125.4.7 QString QEPlot::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the `styleManager` class. Refer to the `styleManager` class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.125.4.8 QString QEPlot::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the `styleManager` class. Refer to the `styleManager` class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.125.4.9 QString QEPlot::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the `styleManager` class. Refer to the `styleManager` class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.125.4.10 UserLevels QEPlot::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.125.4.11 QString QEPlot::variable1 [read, write]**

EPICS variable name (CA PV). This variable is used to read updating values or waveforms for plotting in the first [trace](#).

**9.125.4.12 QString QEPlot::variable2 [read, write]**

EPICS variable name (CA PV). This variable is used to read updating values or waveforms for plotting in the second [trace](#).

**9.125.4.13 QString QEPlot::variable3 [read, write]**

EPICS variable name (CA PV). This variable is used to read updating values or waveforms for plotting in the third [trace](#).

**9.125.4.14 QString QEPlot::variable4 [read, write]**

EPICS variable name (CA PV). This variable is used to read updating values or waveforms for plotting in the fourth [trace](#).

**9.125.4.15 bool QEPlot::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.125.4.16 QString QEPlot::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'SAMPLE=SAM1, NAME = "Ref foil"' These substitutions are applied to all the variable names.

**9.125.4.17 bool QEPlot::visible [read, write]**

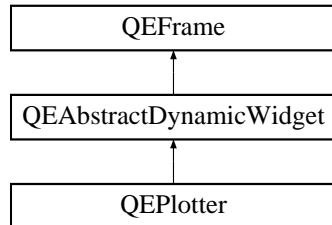
Display the widget. Default is true. Setting this property false is usefull if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlot/QEPlot.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlot/QEPlot.cpp

## 9.126 QEPlotter Class Reference

Inheritance diagram for QEPlotter::



### Classes

- class **DataSets**

### Public Slots

- void **setXYDataPV** (const int slot, const QString &pvName)
- void **setXYAlias** (const int slot, const QString &alias)
- void **setXRange** (const double xMinimum, const double xMaximum)
- void **setYRange** (const double yMinimum, const double yMaximum)
- void **setDataPvNameSet** (const QStringList &pvNameSet)
- void **setAliasNameSet** (const QStringList &aliasNameSet)
- void **setPlotterEntry** (const int slot, const QString &pvName, const QString &alias)

### Signals

- void **crosshairIndexChanged** (int value)
- void **coordinateSelected** (QPointF xyvalue)
- void **xCoordinateSelected** (double xvalue)
- void **yCoordinateSelected** (double yvalue)
- void **pvDataNameSetChanged** (const QStringList &nameSet)
- void **aliasNameSetChanged** (const QStringList &nameSet)
- void **requestAction** (const QEActionRequests &)

### Public Member Functions

- **QEPlotter** (QWidget \*parent=0)
- QSize **sizeHint** () const
- QString **getXYDataPV** (const int slot) const
- void **setXYSizePV** (const int slot, const QString &pvName)
- QString **getXYSizePV** (const int slot) const

- `QString getXYAlias (const int slot) const`
- `void setXYColour (const int slot, const QColor &colour)`
- `QColor getXYColour (const int slot) const`
- `void setXYLineVisible (const int slot, const bool isVisible)`
- `bool getXYLineVisible (const int slot) const`
- `void setXYLineBold (const int slot, const bool isBold)`
- `bool getXYLineBold (const int slot) const`
- `void setXYLineDashed (const int slot, const bool isDashed)`
- `bool getXYLineDashed (const int slot) const`
- `void setXYLineHasDots (const int slot, const bool hasDots)`
- `bool getXYLineHasDots (const int slot) const`
- `void setEnableConextMenu (bool enable)`
- `bool getEnableConextMenu () const`
- `void setMenuEmitText (const QString &text)`
- `QString getMenuEmitText () const`
- `void setToolBarVisible (bool visible)`
- `bool getToolBarVisible () const`
- `void setPvItemsVisible (bool visible)`
- `bool getPvItemsVisible () const`
- `void setStatusVisible (bool visible)`
- `bool getStatusVisible () const`
- `void setAxisEnableX (bool axisEnableX)`
- `bool getAxisEnableX () const`
- `void setAxisEnableY (bool axisEnableY)`
- `bool getAxisEnableY () const`
- `void setXLogarithmic (bool visible)`
- `bool getXLogarithmic () const`
- `void setYLogarithmic (bool visible)`
- `bool getYLogarithmic () const`
- `int addPvName (const QString &pvName)`
- `int getCrosshairIndex () const`
- `QStringList getDataPvNameSet () const`
- `QStringList getAliasNameSet () const`

## Protected Member Functions

- `qcaobject::QCaObject * createQcaItem (unsigned int variableIndex)`
- `void establishConnection (unsigned int variableIndex)`
- `void activated ()`
- `bool eventFilter (QObject *obj, QEvent *event)`
- `void mousePressEvent (QMouseEvent *event)`
- `void dragEnterEvent (QDragEnterEvent *event)`
- `void dropEvent (QDropEvent *event)`
- `QString copyVariable ()`
- `QVariant copyData ()`
- `void saveConfiguration (PersistanceManager *pm)`

- void **restoreConfiguration** (PersistanceManager \*pm, restorePhases re-storePhase)
- int **findSlot** (QObject \*obj)
- QString **getXYExpandedDataPV** (const int slot) const
- QString **getXYExpandedSizePV** (const int slot) const

## Properties

- QString **variableSubstitutions**
- bool **enableContextMenu**
- bool **toolBarIsVisible**
- bool **pvItemsIsVisible**
- bool **statusIsVisible**
- bool **xLogarithmic**
- bool **yLogarithmic**
- bool **axisEnableX**
- bool **axisEnableY**
- QString **contextMenuEmitText**
- QString **DataVariableX**
- QString **DataVariableA**
- QString **DataVariableB**
- QString **DataVariableC**
- QString **DataVariableD**
- QString **DataVariableE**
- QString **DataVariableF**
- QString **DataVariableG**
- QString **DataVariableH**
- QString **DataVariableI**
- QString **DataVariableJ**
- QString **DataVariableK**
- QString **DataVariableL**
- QString **DataVariableM**
- QString **DataVariableN**
- QString **DataVariableO**
- QString **DataVariableP**
- QString **SizeVariableX**
- QString **SizeVariableA**
- QString **SizeVariableB**
- QString **SizeVariableC**
- QString **SizeVariableD**
- QString **SizeVariableE**
- QString **SizeVariableF**
- QString **SizeVariableG**
- QString **SizeVariableH**
- QString **SizeVariableI**
- QString **SizeVariableJ**

- `QString SizeVariableK`
- `QString SizeVariableL`
- `QString SizeVariableM`
- `QString SizeVariableN`
- `QString SizeVariableO`
- `QString SizeVariableP`
- `QString AliasNameX`
- `QString AliasNameA`
- `QString AliasNameB`
- `QString AliasNameC`
- `QString AliasNameD`
- `QString AliasNameE`
- `QString AliasNameF`
- `QString AliasNameG`
- `QString AliasNameH`
- `QString AliasNameI`
- `QString AliasNameJ`
- `QString AliasNameK`
- `QString AliasNameL`
- `QString AliasNameM`
- `QString AliasNameN`
- `QString AliasNameO`
- `QString AliasNameP`
- `QColor ColourA`
- `QColor ColourB`
- `QColor ColourC`
- `QColor ColourD`
- `QColor ColourE`
- `QColor ColourF`
- `QColor ColourG`
- `QColor ColourH`
- `QColor ColourI`
- `QColor ColourJ`
- `QColor ColourK`
- `QColor ColourL`
- `QColor ColourM`
- `QColor ColourN`
- `QColor ColourO`
- `QStringList aliasNames`

## Friends

- class `DataSets`

### 9.126.1 Property Documentation

#### 9.126.1.1 QStringList QEPlotter::aliasNames [read, write]

Allows specification of alias as a 'single property', which is also a slot.

#### 9.126.1.2 QString QEPlotter::variableSubstitutions [read, write]

Default macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'SAMPLE=SAM1, NAME = "Ref foil"' These substitutions are applied to all the variable names.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotter.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotter.cpp

## 9.127 QEPlotterItemDialog Class Reference

### Public Member Functions

- **QEPlotterItemDialog** (QWidget \*parent=0)
- void **setFieldInformation** (const QString dataIn, const QString aliasIn, const QString sizeIn)
- void **getFieldInformation** (QString &dataOut, QString &aliasOut, QString &sizeOut)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterItemDialog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterItemDialog.cpp

## 9.128 QEPlotterManager Class Reference

### Public Member Functions

- **QEPlotterManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterManager.cpp

## 9.129 QEPlotterMenu Class Reference

```
#include <QEPlotterMenu.h>
```

### Signals

- void **selected** (const QEPlotterNames::MenuActions action, const int slot)

### Public Member Functions

- **QEPlotterMenu** (QWidget \*parent=0)
- **QEPlotterMenu** (const int slot, QWidget \*parent=0)
- void **setActionChecked** (const QEPlotterNames::MenuActions action, const bool checked)
- void **setActionEnabled** (const QEPlotterNames::MenuActions action, const bool enabled)
- void **setActionVisible** (const QEPlotterNames::MenuActions action, const bool visible)
- void **setActionText** (const QEPlotterNames::MenuActions action, const QString &caption)
- void **setCheckedStates** (const bool isDisplayed, const bool isBold, const bool isDashed, const bool showDots)

### 9.129.1 Detailed Description

[QEPlotter](#) PV item specific context menu.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterMenu.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterMenu.cpp

## 9.130 QEPlotterNames Class Reference

### Public Types

- enum **MenuActions** {
   
PLOTTER\_FIRST = contextMenu::CM\_SPECIFIC\_WIDGETS\_START\_HERE, PLOTTER\_SHOW\_HIDE\_CROSSHAIRS, PLOTTER\_SHOW\_HIDE\_TOOLBAR, PLOTTER\_SHOW\_HIDE\_pv\_ITEMS,
   
PLOTTER\_SHOW\_HIDE\_STATUS, PLOTTER\_EMIT\_COORDINATES, PLOTTER\_PREV, PLOTTER\_NEXT,
   
PLOTTER\_NORMAL\_VIDEO, PLOTTER\_REVERSE\_VIDEO, PLOTTER\_LINEAR\_Y\_SCALE, PLOTTER\_LOG\_Y\_SCALE,
   
PLOTTER\_MANUAL\_Y\_RANGE, PLOTTER\_CURRENT\_Y\_RANGE, PLOTTER\_DYNAMIC\_Y\_RANGE, PLOTTER\_NORMAILED\_Y\_RANGE,
   
PLOTTER\_FRACTIONAL\_Y\_RANGE, PLOTTER\_LINEAR\_X\_SCALE, PLOTTER\_LOG\_X\_SCALE, PLOTTER\_MANUAL\_X\_RANGE,
   
PLOTTER\_CURRENT\_X\_RANGE, PLOTTER\_DYNAMIC\_X\_RANGE, PLOTTER\_MANUAL\_XY\_RANGE, PLOTTER\_PLAY,
   
PLOTTER\_PAUSE, PLOTTER\_LOAD\_CONFIG, PLOTTER\_SAVE\_CONFIG, PLOTTER\_COPY\_VARIABLE,
   
PLOTTER\_COPY\_DATA, PLOTTER\_PASTE, PLOTTER\_DRAG\_VARIABLE, PLOTTER\_DRAG\_DATA,
   
PLOTTER\_LINE\_BOLD, PLOTTER\_LINE\_DASHED, PLOTTER\_LINE\_DOTS, PLOTTER\_LINE\_VISIBLE,
   
PLOTTER\_LINE\_NO\_MEDIAN\_FILTER, PLOTTER\_LINE\_MEDIAN\_3\_FILTER, PLOTTER\_LINE\_MEDIAN\_5\_FILTER, PLOTTER\_LINE\_COLOUR,
   
PLOTTER\_DATA\_SELECT, PLOTTER\_DATA\_DIALOG, PLOTTER\_DATA\_CLEAR, PLOTTER\_SCALE\_TO\_MIN\_MAX,
   
PLOTTER\_SCALE\_TO\_ZERO\_MAX, PLOTTER\_COPY\_DATA\_VARIABLE, PLOTTER\_COPY\_DATA\_DATA, PLOTTER\_PASTE\_DATA\_pv,
   
PLOTTER\_SHOW\_DATA\_pv\_PROPERTIES, PLOTTER\_ADD\_DATA\_pv\_TO\_STRIPCHART, PLOTTER\_ADD\_DATA\_pv\_TO\_SCRATCH\_PAD, PLOTTER\_GENERAL\_DATA\_pv\_EDIT,
   
PLOTTER\_COPY\_SIZE\_VARIABLE, PLOTTER\_COPY\_SIZE\_DATA, PLOTTER\_PASTE\_SIZE\_pv, PLOTTER\_SHOW\_SIZE\_pv\_PROPERTIES,
   
PLOTTER\_ADD\_SIZE\_pv\_TO\_STRIPCHART, PLOTTER\_ADD\_SIZE\_pv\_TO\_SCRATCH\_PAD, PLOTTER\_GENERAL\_SIZE\_pv\_EDIT, PLOTTER\_LAST }
   
• enum **ScaleModes** { smFixed, smNormalised, smFractional, smDynamic }

## Public Member Functions

- **QEPlotterNames** (QObject \*parent=0)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterNames.h

## 9.131 QEPlotterPushButtonSpecifications Struct Reference

### Public Attributes

- int **gap**
- int **width**
- bool **isIcon**
- const QString **captionOrIcon**
- QEPlotterNames::MenuActions **action**
- const QString **toolTip**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterToolBar.c

## 9.132 QEPlotterState Class Reference

### Public Member Functions

- void **saveConfiguration** (PMElement &parentElement)
- void **restoreConfiguration** (PMElement &parentElement)

### Public Attributes

- bool **isXLogarithmic**
- bool **isYLogarithmic**
- bool **isReverse**
- bool **isPaused**
- double **xMinimum**
- double **xMaximum**
- double **yMinimum**
- double **yMaximum**
- QEPlotterNames::ScaleModes **xScaleMode**
- QEPlotterNames::ScaleModes **yScaleMode**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterState.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterState.cpp

## 9.133 QEPlotterStateList Class Reference

### Public Member Functions

- void **clear** ()
- void **push** (const QEPlotterState &state)
- bool **prev** (QEPlotterState &state)
- bool **next** (QEPlotterState &state)
- bool **prevAvailable** ()
- bool **nextAvailable** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterState.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterState.cpp

## 9.134 QEPlotterToolBar Class Reference

```
#include <QEPlotterToolBar.h>
```

### Signals

- void **selected** (const QEPlotterNames::MenuActions action, const int slot)

### Public Member Functions

- **QEPlotterToolBar** (QWidget \*parent=0)
- void **setEnabled** (const QEPlotterNames::MenuActions action, const bool enabled)

### Static Public Attributes

- static const int **designHeight** = 32

### Protected Member Functions

- void **resizeEvent** (QResizeEvent \*event)

### 9.134.1 Detailed Description

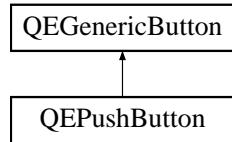
This class holds all the [QEPlotter](#) tool bar widget.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterToolBar.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlotter/QEPlotterToolBar.cpp

## 9.135 QEPushButton Class Reference

Inheritance diagram for QEPushButton::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }
- enum `Formats` {
 `Default` = QEStringFormatting::FORMAT\_DEFAULT, `Floating` = QEStringFormatting::FORMAT\_FLOATING, `Integer` = QEStringFormatting::FORMAT\_INTEGER, `UnsignedInteger` = QEStringFormatting::FORMAT\_UNSIGNEDINTEGER,
 `Time` = QEStringFormatting::FORMAT\_TIME, `LocalEnumeration` = QEStringFormatting::FORMAT\_LOCAL\_ENUMERATE }
- enum `Notations` { `Fixed` = QEStringFormatting::NOTATION\_FIXED, `Scientific` = QEStringFormatting::NOTATION\_SCIENTIFIC, `Automatic` = QEStringFormatting::NOTATION\_AUTOMATIC }
- enum `ArrayActions` { `Append` = QEStringFormatting::APPEND, `Ascii` = QEStringFormatting::ASCII, `Index` = QEStringFormatting::INDEX }
- enum `UpdateOptions` { `Text` = QEGenericButton::UPDATE\_TEXT, `Icon` = QEGenericButton::UPDATE\_ICON, `TextAndIcon` = QEGenericButton::UPDATE\_TEXT\_AND\_ICON, `State` = QEGenericButton::UPDATE\_STATE }

*User friendly enumerations for updateOption property - refer to QEGenericButton::updateOptions for details.*

- enum `ProgramStartupOptionNames` { `None` = applicationLauncher::PSO\_NONE, `Terminal` = applicationLauncher::PSO\_TERMINAL, `LogOutput` = applicationLauncher::PSO\_LOGOUTPUT, `StdOutput` = applicationLauncher::PSO\_STDOUPUT }
- enum `CreationOptionNames` {
 `Open` = QEActionRequests::OptionOpen, `NewTab` = QEActionRequests::OptionNewTab, `NewWindow` = QEActionRequests::OptionNewWindow, `DockTop` = QEActionRequests::OptionTopDockWindow,

```

DockBottom = QEActionRequests::OptionBottomDockWindow, DockLeft = QEActionRequests::OptionLeftDockWindow, DockRight = QEActionRequests::OptionRightDockWindow, DockTopTabbed = QEActionRequests::OptionTopDockWindowTabbed,
DockBottomTabbed = QEActionRequests::OptionBottomDockWindowTabbed,
DockLeftTabbed = QEActionRequests::OptionLeftDockWindowTabbed, DockRightTabbed = QEActionRequests::OptionRightDockWindowTabbed, DockFloating = QEActionRequests::OptionFloatingDockWindow }
```

*Creation options. Used to indicate how to present a GUI when requesting a new GUI be created. Open a new window, open a new tab, or replace the current window.*

## Public Slots

- void **requestAction** (const QEActionRequests &request)
- void **setDefaultStyle** (const QString &style)  
*Update the default style applied to this widget.*
- void **setManagedVisible** (bool v)

## Signals

- void **dbValueChanged** (const QString &out)
- void **dbValueChanged** (const int &out)
- void **dbValueChanged** (const long &out)
- void **dbValueChanged** (const qlonglong &out)
- void **dbValueChanged** (const double &out)
- void **dbValueChanged** (const bool &out)
- void **dbConnectionChanged** (const bool &isConnected)

*Sent when the widget state updated following a channel connection change.*

- void **requestResend** ()  
*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*
- void **newGui** (const QEActionRequests &request)  
*Internal use only. Request a new GUI is created. Typically, this is caught by the QEGui application.*
- void **pressed** (int value)
- void **released** (int value)
- void **clicked** (int value)
- void **programCompleted** ()  
*Program started by button has completed.*

## Public Member Functions

- [QEPushButton \(QWidget \\*parent=0\)](#)
- [QEPushButton \(const QString &variableName, QWidget \\*parent=0\)](#)
- [~QEPushButton \(\)](#)

*Destructor.*

- void [writeNow \(\)](#)

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- void [setVariableNameSubstitutionsProperty \(const QString &substitutions\)](#)

*Access function for `variableNameSubstitutions` property - refer to `variableNameSubstitutions` property for details.*

- void [setAltReadbackProperty \(const QString &variableName\)](#)

*Access function for `altReadback` property - refer to `altReadback` property for details.*

- QString [getAltReadbackProperty \(\) const](#)

*Access function for `altReadback` property - refer to `altReadback` property for details.*

- void [setAltReadbackArrayIndex \(const int arrayIndex\)](#)

*Access function for `altReadbackArrayIndex` property - refer to `altReadbackArrayIndex` property for details.*

- int [getAltReadbackArrayIndex \(\) const](#)

*Access function for `altReadbackArrayIndex` property - refer to `altReadbackArrayIndex` property for details.*

- UserLevels [getUserLevelVisibilityProperty \(\)](#)

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- void [setUserLevelVisibilityProperty \(UserLevels level\)](#)

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- UserLevels [getUserLevelEnabledProperty \(\)](#)

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- void [setUserLevelEnabledProperty \(UserLevels level\)](#)

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- DisplayAlarmStateOptions [getDisplayAlarmStateOptionProperty \(\)](#)

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- void [setDisplayAlarmStateOptionProperty \(DisplayAlarmStateOptions option\)](#)

*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

- void [setFormatProperty \(Formats format\)](#)

*Access function for `format` property - refer to `format` property for details.*

- Formats [getFormatProperty \(\)](#)

*Access function for `format` property - refer to `format` property for details.*

- void [setNotationProperty \(Notations notation\)](#)

*Access function for `notation` property - refer to `notation` property for details.*

- Notations [getNotationProperty \(\)](#)

*Access function for [notation](#) property - refer to [notation](#) property for details.*

- void [setArrayActionProperty](#) ([ArrayActions](#) arrayAction)

*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*

- [ArrayActions getArrayActionProperty](#) ()

*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*

## Properties

- [QString variable](#)
- [QString variableSubstitutions](#)
- [int arrayIndex](#)
- [QString altReadbackVariable](#)
- [int altReadbackArrayIndex](#)
- [bool subscribe](#)
- [bool variableAsToolTip](#)
- [bool allowDrop](#)
- [bool visible](#)
- [unsigned int](#)
- [QString styleSheet](#)
- [QString defaultStyle](#)
- [QString userLevelUserStyle](#)
- [QString userLevelScientistStyle](#)
- [QString userLevelEngineerStyle](#)
- [UserLevels userLevelVisibility](#)
- [UserLevels userLevelEnabled](#)
- [bool displayAlarmState](#)
- [DisplayAlarmStateOptions displayAlarmStateOption](#)
- [int precision](#)
- [bool useDbPrecision](#)
- [bool leadingZero](#)
- [bool trailingZeros](#)
- [bool addUnits](#)
- [QString localEnumeration](#)
- [Formats format](#)
- [Notations notation](#)
- [ArrayActions arrayAction](#)
- [QEWidgetProperties::DisabledRecordPolicy disabledRecordPolicy](#)
- [Qt::Alignment alignment](#)
- [UpdateOptions updateOption](#)
- [QPixmap pixmap0](#)
- [QPixmap pixmap1](#)
- [QPixmap pixmap2](#)
- [QPixmap pixmap3](#)

- `QPixmap pixmap4`
- `QPixmap pixmap5`
- `QPixmap pixmap6`
- `QPixmap pixmap7`
- `QString password`
- `bool confirmAction`
- `QString confirmText`
- `bool writeOnPress`
- `bool writeOnRelease`
- `bool writeOnClick`
- `QString pressText`
- `QString releaseText`
- `QString clickText`
- `QString clickCheckedText`
- `QString labelText`
- `QString program`
- `QStringList arguments`
- `ProgramStartupOptionNames programStartupOption`
- `QString guiFile`
- `CreationOptionNames creationOption`
- `QString prioritySubstitutions`
- `QString customisationName`

### 9.135.1 Member Enumeration Documentation

#### 9.135.1.1 enum QEPushButton::ArrayActions

User friendly enumerations for arrayAction property - refer to `QEStringFormatting::arrayActions` for details.

##### Enumerator:

*Append* Refer to `QEStringFormatting::APPEND` for details.

*Ascii* Refer to `QEStringFormatting::ASCII` for details.

*Index* Refer to `QEStringFormatting::INDEX` for details.

#### 9.135.1.2 enum QEPushButton::CreationOptionNames

Creation options. Used to indicate how to present a GUI when requesting a new GUI be created. Open a new window, open a new tab, or replace the current window.

##### Enumerator:

*Open* Replace the current GUI with the new GUI.

*NewTab* Open new GUI in a new tab.

*NewWindow* Open new GUI in a new window.

**DockTop** Open new GUI in a top dock window.

**DockBottom** Open new GUI in a bottom dock window.

**DockLeft** Open new GUI in a left dock window.

**DockRight** Open new GUI in a right dock window.

**DockTopTabbed** Open new GUI in a top dock window (tabbed with any existing dock in that area).

**DockBottomTabbed** Open new GUI in a bottom dock window (tabbed with any existing dock in that area).

**DockLeftTabbed** Open new GUI in a left dock window (tabbed with any existing dock in that area).

**DockRightTabbed** Open new GUI in a right dock window (tabbed with any existing dock in that area).

**DockFloating** Open new GUI in a floating dock window.

#### 9.135.1.3 enum QEPushButton::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

**Never** Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

**Always** Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

**WhenInAlarm** Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.135.1.4 enum QEPushButton::Formats

User friendly enumerations for format property - refer to [QEStringFormatting::formats](#) for details.

##### Enumerator:

**Default** Format as best appropriate for the data type.

**Floating** Format as a floating point number.

**Integer** Format as an integer.

**UnsignedInteger** Format as an unsigned integer.

**Time** Format as a time.

**LocalEnumeration** Format as a selection from the [localEnumeration](#) property.

### 9.135.1.5 enum QEPushButton::Notations

User friendly enumerations for notation property - refer to [QEStringFormatting::notations](#) for details.

#### Enumerator:

**Fixed** Refer to [QEStringFormatting::NOTATION\\_FIXED](#) for details.

**Scientific** Refer to [QEStringFormatting::NOTATION\\_SCIENTIFIC](#) for details.

**Automatic** Refer to [QEStringFormatting::NOTATION\\_AUTOMATIC](#) for details.

### 9.135.1.6 enum QEPushButton::ProgramStartupOptionNames

Startup options. Just run the command, run the command within a terminal, or display the output in QE message system.

#### Enumerator:

**None** Just run the program.

**Terminal** Run the program in a terminal (in Windows a command interpreter will also be started, so the program may be a built-in command like 'dir').

**LogOutput** Run the program, and log the output in the QE message system.

**StdOutput** Run the program, and send output to standard output and standard error.

### 9.135.1.7 enum QEPushButton::UpdateOptions

User friendly enumerations for updateOption property - refer to [QEGenericButton::updateOptions](#) for details.

#### Enumerator:

**Text** Data updates will update the button text.

**Icon** Data updates will update the button icon.

**TextAndIcon** Data updates will update the button text and icon.

**State** Data updates will update the button state (checked or unchecked).

### 9.135.1.8 enum QEPushButton::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and userLevel enumeration for details.

#### Enumerator:

**User** Refer to [USERLEVEL\\_USER](#) for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

## 9.135.2 Constructor & Destructor Documentation

### 9.135.2.1 QEPushButton::QEPushButton (QWidget \* *parent* = 0)

Create without a variable. Use setVariableNameProperty() and setSubstitutionsProperty() to define a variable and, optionally, macro substitutions later.

### 9.135.2.2 QEPushButton::QEPushButton (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

## 9.135.3 Member Function Documentation

### 9.135.3.1 void QEPushButton::clicked (int *value*) [signal]

Button has been Clicked. The value emitted is the integer interpretation of the clickText property (or the clickCheckedText property if the button was checked)

### 9.135.3.2 void QEPushButton::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

### 9.135.3.3 void QEPushButton::pressed (int *value*) [signal]

Button has been Pressed. The value emitted is the integer interpretation of the pressText property

### 9.135.3.4 void QEPushButton::released (int *value*) [signal]

Button has been Released The value emitted is the integer interpretation of the releaseText property

**9.135.3.5 void QEPushButton::requestAction (const QEActionRequests & request) [inline, slot]**

Default slot used to create a new GUI if there is no slot indicated in the ContainerProfile class. This slot is typically used when the button is pressed within the Designer preview window to allow the operation of the button to be tested. If an application does not specify a slot to use for creating new windows (through the ContainerProfile class) a window will still be created through this slot, but it will not respect the window creation options or any other window related application constraints. For example, the QEgui application does provide a slot for creating new GUIs in the ContainerProfile class which respects the creation options, knows how to add tabs in the application, and extend the application's window menu in the menu bar.

**9.135.3.6 void QEPushButton::setManagedVisible (bool v) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.135.4 Property Documentation

**9.135.4.1 bool QEPushButton::addUnits [read, write]**

If true (default), add engineering units supplied with the data.

**9.135.4.2 Qt::Alignment QEPushButton::alignment [read, write]**

Set the buttons text alignment. Left justification is particularly useful when displaying quickly changing numeric data updates.

**9.135.4.3 bool QEPushButton::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.135.4.4 QString QEPushButton::altReadbackVariable [read, write]**

EPICS variable name (CA PV). This variable is used to provide a readback value when different to the variable written to by a button press.

**9.135.4.5 QStringList QEPushButton::arguments [read, write]**

Arguments for program specified in the 'program' property.

**9.135.4.6 ArrayActions QEPushButton::arrayAction [read, write]**

Text formatting option for array data. Default is ASCII. Options are:

- ASCII - treat array as a single text string. For example an array of three characters 'a' 'b' 'c' will be formatted as 'abc'.
- APPEND - treat array as an array of numbers and format a string containing them all with a space between each. For example, an array of three numbers 10, 11 and 12 will be formatted as '10 11 12'.
- INDEX - Extract a single item from the array. The item is then formatted as any other non array data would be. The item selected is determined by the arrayIndex property. For example, if arrayIndex property is 1, an array of three numbers 10, 11 and 12 will be formatted as '11'.

**9.135.4.7 int QEPushButton::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.135.4.8 QString QEPushButton::clickCheckedText [read, write]**

Text used to compare with text written or read to determine if push button should be marked as checked. Note, must be an exact match following formatting of data updates. When writing values, the 'pressText', 'ReleaseText', or 'clickedtext' must match this property to cause the button to be checked when the write occurs.

Good example: formatting set to diaplay a data value of '1' as 'On', clickCheckedText is 'On', clickText is 'On'. In this example, the push button will be checked when a data update occurs with a value of 1 or when the button is clicked.

Bad example: formatting set to diaplay a data value of '1' as 'On', clickCheckedText is 'On', clickText is '1'. In this example, the push button will be checked when a data update occurs with a value of 1 but, although a valid value will be written when clicked, the button will not be checked when clicked as '1' is not the same as 'On'.

Reimplemented from [QEGenericButton](#).

**9.135.4.9 QString QEPushButton::clickText [read, write]**

Value written when user clicks button if 'writeOnClick' property is true

Reimplemented from [QEGenericButton](#).

**9.135.4.10 bool QEPushButton::confirmAction [read, write]**

If true, a dialog will be presented asking the user to confirm if the button action should be carried out

**9.135.4.11 QString QEPushButton::confirmText [read, write]**

Text used to confirm action if confirmation dialog is presented

Reimplemented from [QEGenericButton](#).

**9.135.4.12 CreationOptionNames QEPushButton::creationOption [read, write]**

Creation options when opening a new GUI. Open a new window, open a new tab, or replace the current window. The creation option is supplied when the button generates a newGui signal. Application code connected to this signal should honour this request if possible. When used within the QEGui application, the QEGui application creates a new window, new tab, or replaces the current window as appropriate.

Reimplemented from [QEGenericButton](#).

**9.135.4.13 QString QEPushButton::customisationName [read, write]**

Window customisation name. This name will be used to select a set of window customisations including menu items and tool bar buttons. Applications such as QEGui can load .xml files containing named sets of window customisations. This property is used to select a set loaded from these files. The selected set of customisations will be applied to the main window containing the new GUI. Customisations are not applied if the GUI is opened as a dock.

Reimplemented from [QEGenericButton](#).

**9.135.4.14 QString QEPushButton::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.135.4.15 QEWidgetProperties::DisabledRecordPolicy  
QEPushButton::disabledRecordPolicy [read, write]**

Set the widget's disabled record policy, i.e. the action to be taken when the underlying record is disabled, i.e. when the associated record's DISA and DISV field values are equal. Note: this is only applicable IOC process variables. When the policy is ignore, then no special action is taken. This is the default policy. When the policy is grayout, the widget's style is set as if disconnected when the record is disabled.

Reimplemented from [QEGenericButton](#).

**9.135.4.16 bool QEPushButton::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never

indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.135.4.17 **DisplayAlarmStateOptions** `QEPushButton::displayAlarmStateOption` [`read`, `write`]

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### 9.135.4.18 **Formats** `QEPushButton::format` [`read`, `write`]

Format to apply to data. Default is 'Default' in which case the data type supplied with the data determines how the data is formatted. For all other options, an attempt is made to format the data as requested (whatever its native form).

#### 9.135.4.19 **QString** `QEPushButton::guiFile` [`read`, `write`]

File name of GUI to be presented on button click. File name can be absolute, relative to the path of the QEform in which the `QEPushButton` is located, relative to the any path in the path list published in the ContainerProfile class, or relative to the current path. See `QEWidget::openQEFile()` in `QEWidget.cpp` for details.

#### 9.135.4.20 **unsigned** `QEPushButton::int` [`read`, `write`]

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a `QELog` widget may be set up to only log messages from a select set of widgets.

Base used for when formatting integers. Default is 10 (duh!)

#### 9.135.4.21 **QString** `QEPushButton::labelText` [`read`, `write`]

Button label text (prior to substitution). Macro substitutions will be applied to this text and the result will be set as the button text. Used when data updates are not being represented in the button text. IF NOT LEFT EMPTY, THIS TEXT WILL TAKE PRIORITY OVER THE PUSH BUTTON 'text' PROPERTY! For example, a button in a sub form may have a 'labelText' property of 'Turn Pump On'. When the sub form is used twice in a main form with substitutions PUMPNUM=1 and PUMPNUM=2

respectively, the two identical buttons in the sub forms will have the labels 'Turn Pump 1 On' and 'Turn Pump 2 On' respectively.

Reimplemented from [QEGenericButton](#).

#### **9.135.4.22 bool QEPushButton::leadingZero [read, write]**

If true (default), always add a leading zero when formatting numbers.

#### **9.135.4.23 QString QEPushButton::localEnumeration [read, write]**

An enumeration list used to data values. Used only when the formatting option is 'local enumeration'. Value is converted to an integer and used to select a string from this list.

Format is:

[[<|<=|=|=|>=|>]value1|\*] : string1 , [[<|<=|=|=|>=|>]value2|\*] : string2 ,  
[[<|<=|=|=|>=|>]value3|\*] : string3 , ...

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
>= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm"  
<2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than 2" 3:"Beamline Available", \*:"" "Pump Off":"OH NO!, the pump is OFF!","Pump On":"It's OK, the pump is on"

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10:""'

A range of numbers can be covered by a pair of values as in the following example:  
>=4:"Between 4 and 8",<=8:"Between 4 and 8"

#### **9.135.4.24 Notations QEPushButton::notation [read, write]**

Notation used for numerical formatting. Default is fixed.

#### **9.135.4.25 QString QEPushButton::password [read, write]**

Password user will need to enter before any action is taken

Reimplemented from [QEGenericButton](#).

**9.135.4.26 QPixmap QEPushButton:: pixmap0 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 0

**9.135.4.27 QPixmap QEPushButton:: pixmap1 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 1

**9.135.4.28 QPixmap QEPushButton:: pixmap2 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 2

**9.135.4.29 QPixmap QEPushButton:: pixmap3 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 3

**9.135.4.30 QPixmap QEPushButton:: pixmap4 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 4

**9.135.4.31 QPixmap QEPushButton:: pixmap5 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 5

**9.135.4.32 QPixmap QEPushButton:: pixmap6 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 6

**9.135.4.33 QPixmap QEPushButton:: pixmap7 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 7

**9.135.4.34 int QEPushButton:: precision [read, write]**

Precision used when formatting floating point numbers. The default is 4. This is only used if useDbPrecision is false.

**9.135.4.35 QString QEPushButton::pressText [read, write]**

Value written when user presses button if 'writeOnPress' property is true

Reimplemented from [QEGenericButton](#).

**9.135.4.36 QString QEPushButton::prioritySubstitutions [read, write]**

Overriding macro substitutions. These macro substitions take precedence over any existing macro substitutions defined by the variableSubstitutions property, any parent forms, or the application containing the button. These macro substitutions are particularly usefull when the button's function is to reload the same form but with different macro substitutions. The variableSubstitutions property cannot be used for this since, although they are added to the list of macro substitions applied to the new form, they are appended to the list and the existing macro substitutions take precedence.

Reimplemented from [QEGenericButton](#).

**9.135.4.37 QString QEPushButton::program [read, write]**

Program to run when the button is clicked. No attempt to run a program is made if this property is empty. Example: firefox

**9.135.4.38 ProgramStartupOptionNames QEPushButton::programStartupOption [read, write]**

Startup options. Just run the command, run the command within a terminal, or display the output in QE message system.

**9.135.4.39 QString QEPushButton::releaseText [read, write]**

Value written when user releases button if 'writeOnRelease' property is true

Reimplemented from [QEGenericButton](#).

**9.135.4.40 QString QEPushButton::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.135.4.41 bool QEPushButton::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.135.4.42 bool QEPushButton::trailingZeros [read, write]**

If true (default), always remove any trailing zeros when formatting numbers.

**9.135.4.43 UpdateOptions QEPushButton::updateOption [read, write]**

Update options (text, pixmap, both, or state (checked or unchecked)

Reimplemented from [QEGenericButton](#).

**9.135.4.44 bool QEPushButton::useDbPrecision [read, write]**

If true (default), format floating point numbers using the precision supplied with the data. If false, the precision property is used.

**9.135.4.45 UserLevels QEPushButton::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.135.4.46 QString QEPushButton::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.135.4.47 QString QEPushButton::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.135.4.48 QString QEPushButton::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.135.4.49 UserLevels QEPushButton::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.135.4.50 QString QEPushButton::variable [read, write]**

EPICS variable name (CA PV)

**9.135.4.51 bool QEPushButton::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.135.4.52 QString QEPushButton::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

**9.135.4.53 bool QEPushButton::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

**9.135.4.54 bool QEPushButton::writeOnClick [read, write]**

If true, the 'clickText' property is written when the button is clicked. Default is true  
Reimplemented from [QEGenericButton](#).

**9.135.4.55 bool QEPushButton::writeOnPress [read, write]**

If true, the 'pressText' property is written when the button is pressed. Default is false

Reimplemented from [QEGenericButton](#).

**9.135.4.56 bool QEPushButton::writeOnRelease [read, write]**

If true, the 'releaseText' property is written when the button is released. Default is false

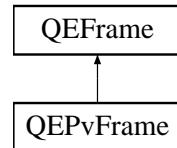
Reimplemented from [QEGenericButton](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QEPushButton.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QEPushButton.cpp

## 9.136 QEPvFrame Class Reference

Inheritance diagram for QEPvFrame::



### Signals

- void **dbValueChanged** (const QString &out)
- void **dbValueChanged** (const int &out)
- void **dbValueChanged** (const long &out)
- void **dbValueChanged** (const qlonglong &out)
- void **dbValueChanged** (const double &out)
- void **dbValueChanged** (const bool &out)
- void **dbConnectionChanged** (const bool &isConnected)

### Public Member Functions

- **QEPvFrame** (QWidget \*parent=0)
- **QEPvFrame** (const QString &variableName, QWidget \*parent=0)

### Protected Member Functions

- **qcaobject::QCaObject \* createQcaItem** (unsigned int variableIndex)
- void **establishConnection** (unsigned int variableIndex)
- QString **copyVariable** ()
- QVariant **copyData** ()

### Properties

- QString **variable**
- QString **variableSubstitutions**
- int **arrayIndex**

### 9.136.1 Member Function Documentation

#### 9.136.1.1 void QEPvFrame::dbConnectionChanged (const bool & isConnected) [signal]

Sent when the widget state updated following a channel connection change Applied to provary varible.

**9.136.1.2 void QEPvFrame::dbValueChanged (const QString & *out*)  
[signal]**

Sent when the widget is updated following a data change. Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.136.2 Property Documentation****9.136.2.1 int QEPvFrame::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.136.2.2 QString QEPvFrame::variable [read, write]**

EPICS variable name (CA PV)

**9.136.2.3 QString QEPvFrame::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFrame/QEPvFrame.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFrame/QEPvFrame.cpp

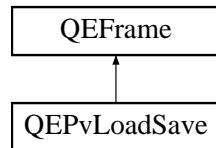
## 9.137 QEPvFrameManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEFframe/QEPvFrameManager.h

## 9.138 QEPvLoadSave Class Reference

Inheritance diagram for QEPvLoadSave::



### Classes

- class **Halves**

### Signals

- void **requestAction** (const QEActionRequests &)

### Public Member Functions

- [QEPvLoadSave](#) (QWidget \*parent=0)
- virtual [~QEPvLoadSave](#) ()  
*Destruction.*
- virtual QSize [sizeHint](#) () const  
*Size hint.*
- void [setConfigurationFileLeft](#) (QString configurationFile)
- QString [getConfigurationFileLeft](#) ()
- void [setConfigurationFileRight](#) (QString configurationFile)
- QString [getConfigurationFileRight](#) ()
- void [setSubstitutions](#) (QString configFileSubstitutions)
- QString [getSubstitutions](#) ()
- void [setDefaultDir](#) (const QString &defaultDir)
- QString [getDefaultDir](#) () const
- void [setConfirmAction](#) (bool confirmRequiredIn)
- bool [getConfirmAction](#) () const

### Static Public Attributes

- static const int **NumberOfButtons** = 15

## Protected Member Functions

- `qcaobject::QCaObject * createQcaItem (unsigned int variableIndex)`
- `void establishConnection (unsigned int variableIndex)`
- `void resizeEvent (QResizeEvent *)`
- `bool eventFilter (QObject *obj, QEvent *event)`

## Properties

- `QString configurationFileLeft`
- `QString configurationFileRight`
- `QString defaultSubstitutions`
- `QString defaultDir`
- `bool confirmAction`

## Friends

- class `QEPvLoadSaveCompare`
- class `QEPvLoadSaveModel`

### 9.138.1 Constructor & Destructor Documentation

#### 9.138.1.1 `QEPvLoadSave::QEPvLoadSave (QWidget *parent = 0)`

Create without a nominated config file.

### 9.138.2 Property Documentation

#### 9.138.2.1 `QString QEPvLoadSave::configurationFileLeft [read, write]`

configurationFile

#### 9.138.2.2 `bool QEPvLoadSave::confirmAction [read, write]`

If true, a dialog will be presented asking the user to confirm if the PV write actions should be carried out. Defaults to true.

#### 9.138.2.3 `QString QEPvLoadSave::defaultDir [read, write]`

Default directory used for loading/saving files. Default to null string which is interpreted as the current directory.

**9.138.2.4 QString QEPvLoadSave::defaultSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSave.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSave.cpp

## 9.139 QEPvLoadSaveCommon Class Reference

### Public Types

- enum **ActionKinds** { **Apply**, **Extract**, **ReadArchive** }
- typedef QMap< QString, double > **PvNameValueMaps**

### Static Public Member Functions

- static PvNameValueMaps **merge** (const PvNameValueMaps &a, const PvNameValueMaps &b)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSa
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSa

## 9.140 QEPvLoadSaveCompare Class Reference

```
#include <QEPvLoadSaveCompare.h>
```

### Classes

- struct **ValuePairs**

### Public Member Functions

- **QEPvLoadSaveCompare** ([QEPvLoadSave](#) \*owner, const int side, QWidget \*parent=0)
- void **processModelData** ()

#### 9.140.1 Detailed Description

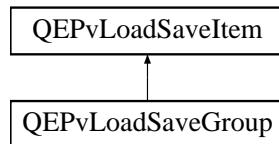
This widget displays graphically the difference between two PV data sets from the left/right hand side of a [QEPvLoadSave](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveCompare.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveCompare.cpp

## 9.141 QEPvLoadSaveGroup Class Reference

Inheritance diagram for QEPvLoadSaveGroup::



### Public Member Functions

- **QEPvLoadSaveGroup** (const QString &groupName, [QEPvLoadSaveItem](#) \*parent=0)
- bool **getIsGroup** () const
- [QEPvLoadSaveItem](#) \* **clone** ([QEPvLoadSaveItem](#) \*parent)
- void **actionConnect** (QObject \*actionCompleteObject, const char \*actionCompleteSlot)
- void **extractPVData** ()
- void **applyPVData** ()
- void **readArchiveData** (const [QCaDateTime](#) &dateTime)
- int **leafCount** () const
- [QEPvLoadSaveCommon::PvNameValueMaps](#) **getPvNameValueMap** () const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSave.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSave.cpp

## 9.142 QEPvLoadSaveGroupNameDialog Class Reference

### Public Member Functions

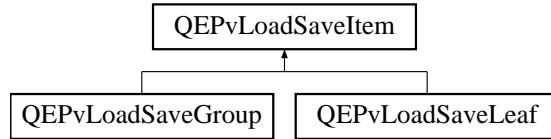
- **QEPvLoadSaveGroupNameDialog** (QWidget \*parent=0)
- void **setGroupName** (QString pvNameIn)
- QString **getGroupName** ()
- bool **isClear** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveGroupNameD...
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveGroupNameD...

## 9.143 QEPvLoadSaveItem Class Reference

#include <QEPvLoadSaveItem.h>Inheritance diagram for QEPvLoadSaveItem::



### Signals

- void **reportActionComplete** (const [QEPvLoadSaveItem](#) \*item, [QEPvLoadSaveCommon::ActionKinds](#) action, bool actionSuccessful)

### Public Member Functions

- int **columnCount** () const
- [QEPvLoadSaveItem](#) \* **getChild** (int position) const
- [QEPvLoadSaveItem](#) \* **getParent** () const
- int **childCount** () const
- int **childPosition** () const
- QVariant **getData** (int column) const
- bool **insertChild** (int position, [QEPvLoadSaveItem](#) \*get Child)
- bool **removeChildren** (int position, int count)
- void **appendChild** ([QEPvLoadSaveItem](#) \*get Child)
- [QEPvLoadSaveItem](#) \* **getNamedChild** (const QString &searchName)
- virtual [QEPvLoadSaveItem](#) \* **clone** ([QEPvLoadSaveItem](#) \*parent)
- virtual void **actionConnect** (QObject \*actionCompleteObject, const char \*actionCompleteSlot)
- QStringList **getNodePath** ()
- virtual void **setNodeName** (const QString &nodeName)
- QString **getNodeName** () const
- void **setnodeValue** (const QVariant &value)
- QVariant **getNodeValue** () const
- int **getElementCount** () const
- virtual bool **getIsPV** () const
- virtual bool **getIsGroup** () const
- virtual void **extractPVData** ()
- virtual void **applyPVData** ()
- virtual void **readArchiveData** (const [QCaDateTime](#) &dateTime)
- virtual int **leafCount** () const
- virtual [QEPvLoadSaveCommon::PvNameValueMaps](#) **getPvNameValueMap** () const

## Protected Member Functions

- **QEPvLoadSaveItem** (const QString &nodeName, const QVariant &value, QEPvLoadSaveItem \*parent=0)

## Protected Attributes

- QList< QEPvLoadSaveItem \* > **childItems**
- **QEPvLoadSaveItem \* parentItem**
- QVariant **value**

### 9.143.1 Detailed Description

This class is based on the TreeItem example specified in:  
<http://qt-project.org/doc/qt-4.8/itemviews-editabletreemodel.html>

Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies). Contact:  
<http://www.qt-project.org/legal>

A major difference is that it is derived from QObject because each leaf item is associated with a PV and needs slots to receive value data. A consequence of this is that this class must be exposed in a header file even though it is essentially a QEPvLoadSave private class. Also there is no itemData variant array - these values calculated as an when needed. Lastly some function name changes such as parent => getParent (as parent already inherited from QObject) and some changes just to follow my preferred style.

Note: although QObjects provide a parent/child linking, this class manages its own parent child relationships as per the TreeItem example.

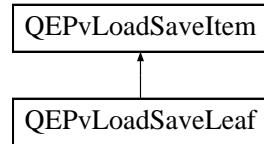
QEPvLoadSaveItem are created in one of two flavours: a/ node - used for groups b/ leaf - used for PVs. both of which inherited from the QEPvLoadSaveItem base class

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveItem.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveItem.cpp

## 9.144 QEPvLoadSaveLeaf Class Reference

Inheritance diagram for QEPvLoadSaveLeaf::



### Public Member Functions

- **QEPvLoadSaveLeaf** (const QString &setPointPvName, const QString &readBackPvName, const QString &archiverPvName, const QVariant &value, [QEPvLoadSaveItem](#) \*parent=0)
- void **setSetPointPvName** (const QString &pvName)
- QString **getSetPointPvName** () const
- void **setReadBackPvName** (const QString &pvName)
- QString **getReadBackPvName** () const
- void **setArchiverPvName** (const QString &pvName)
- QString **getArchiverPvName** () const
- void **setNodeName** (const QString &nodeName)
- bool **getIsPV** () const
- [QEPvLoadSaveItem](#) \* **clone** ([QEPvLoadSaveItem](#) \*parent)
- QEPvLoadSaveCommon::PvNameValueMaps **getPvNameValueMap** () const
- void **actionConnect** (QObject \*actionCompleteObject, const char \*actionCompleteSlot)
- void **extractPVData** ()
- void **applyPVData** ()
- void **readArchiveData** (const [QCaDateTime](#) &dateTime)
- int **leafCount** () const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSave.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSave.cpp

## 9.145 QEPvLoadSaveManager Class Reference

### Public Member Functions

- **QEPvLoadSaveManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveManager.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveManager.cpp

## 9.146 QEPvLoadSaveModel Class Reference

```
#include <QEPvLoadSaveModel.h>
```

### Signals

- void **reportActionComplete** (QEPvLoadSaveCommon::ActionKinds, bool)

### Public Member Functions

- **QEPvLoadSaveModel** (QTreeView \*treeView, QEPvLoadSave \*parent)
- QVariant **data** (const QModelIndex &index, int role) const
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const
- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const
- Qt::ItemFlags **flags** (const QModelIndex &index) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)
- bool **setHeaderData** (int section, Qt::Orientation orientation, const QVariant &value, int role=Qt::EditRole)
- bool **insertRows** (int position, int rows, const QModelIndex &parent=QModelIndex())
- bool **removeRows** (int position, int rows, const QModelIndex &parent=QModelIndex())
- void **setHeading** (const QString &heading)
- QString **getHeading** () const
- void **setupModelData** (QEPvLoadSaveItem \*rootItem, const QString &heading)
- void **modelUpdated** ()
- void **itemUpdated** (const QEPvLoadSaveItem \*item)
- bool **addItemToModel** (QEPvLoadSaveItem \*item, QEPvLoadSaveItem \*parentItem)
- bool **removeItemFromModel** (QEPvLoadSaveItem \*item)
- bool **mergeItemInToModel** (QEPvLoadSaveItem \*item)
- void **extractPVData** ()
- void **applyPVData** ()
- void **readArchiveData** (const QCaDateTime &dateTime)
- int **leafCount** () const
- QEPvLoadSaveCommon::PvNameValueMaps **getPvNameValueMap** () const
- QEPvLoadSaveItem \* **getRootItem** ()
- QEPvLoadSaveItem \* **getSelectedItem** ()
- QModelIndex **getRootIndex** ()
- QEPvLoadSaveItem \* **indexToItem** (const QModelIndex &index) const

## Protected Member Functions

- bool **eventFilter** (QObject \*obj, QEvent \*event)

### 9.146.1 Detailed Description

This class is based on the TreeModel example specified in:

<http://qt-project.org/doc/qt-4.8/itemviews-editabletreemodel.html>

Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies). Contact:

<http://www.qt-project.org/legal>

Note on naming: the example's root item that provide header info is referred to the core item. It only ever has one child which is the visible (on the QTreeView) which is the user root item, named "ROOT".

Note: we only re-size the number of rows (children). The number of columns is fixed.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveModel.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveModel.cpp

## 9.147 QEPvLoadSaveTimeDialog Class Reference

### Public Member Functions

- **QEPvLoadSaveTimeDialog** (QWidget \*parent=0)
- void **setMaximumDateTime** (const QDateTime &datetime)
- void **setDateTime** (const QDateTime &datetime)
- QDateTime **getDateTime** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSa
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSa

## 9.148 QEPvLoadSaveUtilities Class Reference

### Static Public Member Functions

- static [QEPvLoadSaveItem](#) \* **readTree** (const QString &filename, const QString &macroString)
- static bool **writeTree** (const QString &filename, const [QEPvLoadSaveItem](#) \*root)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveUtilities.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSaveUtilities.cpp

## 9.149 QEPvLoadSaveValueEditDialog Class Reference

### Public Member Functions

- **QEPvLoadSaveValueEditDialog** (QWidget \*parent=0)
- void **setPvName** (const QString &pvName)
- void **setValue** (const QVariant &valueList)
- QVariant **getValue** () const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSa
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvLoadSave/QEPvLoadSa

## 9.150 QEPVNameLists Class Reference

### Public Member Functions

- void **prependOrMoveToFirst** (const QString &item)
- void **saveConfiguration** (PMElement &parentElement)
- void **restoreConfiguration** (PMElement &parentElement)

### Static Public Member Functions

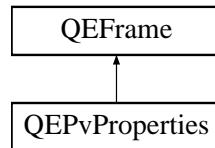
- static void **constructor** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChart.cpp

## 9.151 QEPvProperties Class Reference

Inheritance diagram for QEPvProperties::



### Public Types

- enum **OwnContextMenuOptions** { **PVPROP\_NONE** = CM\_SPECIFIC\_WIDGETS\_START\_HERE, **PVPROP\_SORT\_FIELD\_NAMES**, **PVPROP\_RESET\_FIELD\_NAMES**, **PVPROP\_SUB\_CLASS\_WIDGETS\_START\_HERE** }

### Signals

- void **setCurrentBoxIndex** (int index)

### Public Member Functions

- void **setVariableNameProperty** (QString variableName)
 

*Property access function for `variable` property. This has special behaviour to work well within designer.*
- QString **getVariableNameProperty** ()
 

*Property access function for `variable` property. This has special behaviour to work well within designer.*
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)
 

*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*
- QString **getVariableNameSubstitutionsProperty** ()
 

*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*
- **QEPvProperties** (QWidget \*parent=0)
- **QEPvProperties** (const QString &variableName, QWidget \*parent=0)
- QSize **sizeHint** () const

## Protected Member Functions

- void **resizeEvent** (QResizeEvent \*event)
- QMenu \* **buildContextMenu** ()
- void **contextMenuTriggered** (int selectedItemNum)
- bool **itemLessThan** (const int a, const int b, QObject \*context=NULL) const
- void **swapItems** (const int a, const int b, QObject \*context=NULL)
- void **establishConnection** (unsigned int variableIndex)
- **qcaobject::QCaObject** \* **createQcaItem** (unsigned int variableIndex)
- void **mousePressEvent** (QMouseEvent \*event)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- void **saveConfiguration** (PersistanceManager \*pm)
- void **restoreConfiguration** (PersistanceManager \*pm, restorePhases restorePhase)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)

## Properties

- QString **variable**
- QString **variableSubstitutions**

### 9.151.1 Property Documentation

#### 9.151.1.1 QString QEPvProperties::variable [read, write]

EPICS variable name (CA PV)

#### 9.151.1.2 QString QEPvProperties::variableSubstitutions [read, write]

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.cpp

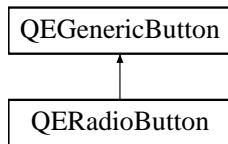
## 9.152 QEPvPropertiesManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.h

## 9.153 QERadioButton Class Reference

Inheritance diagram for QERadioButton::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }
- enum `Formats` {
 `Default` = QEStringFormatting::FORMAT\_DEFAULT, `Floating` = QEStringFormatting::FORMAT\_FLOATING, `Integer` = QEStringFormatting::FORMAT\_INTEGER, `UnsignedInteger` = QEStringFormatting::FORMAT\_UNSIGNEDINTEGER,
 `Time` = QEStringFormatting::FORMAT\_TIME, `LocalEnumeration` = QEStringFormatting::FORMAT\_LOCAL\_ENUMERATE
 }
- enum `Separators` { `NoSeparator` = QEStringFormatting::SEPARATOR\_NONE, `Comma` = QEStringFormatting::SEPARATOR\_COMMA, `Under_score` = QEStringFormatting::SEPARATOR\_UNDERSCORE, `Space` = QEStringFormatting::SEPARATOR\_SPACE }
- enum `Notations` { `Fixed` = QEStringFormatting::NOTATION\_FIXED, `Scientific` = QEStringFormatting::NOTATION\_SCIENTIFIC, `Automatic` = QEStringFormatting::NOTATION\_AUTOMATIC }
- enum `ArrayActions` { `Append` = QEStringFormatting::APPEND, `Ascii` = QEStringFormatting::ASCII, `Index` = QEStringFormatting::INDEX }
- enum `UpdateOptions` { `Text` = QEGenericButton::UPDATE\_TEXT, `Icon` = QEGenericButton::UPDATE\_ICON, `TextAndIcon` = QEGenericButton::UPDATE\_TEXT\_AND\_ICON, `State` = QEGenericButton::UPDATE\_STATE }

*User friendly enumerations for updateOption property - refer to QEGenericButton::updateOptions for details.*

- enum `ProgramStartupOptionNames` { `None` = applicationLauncher::PSO\_NONE, `Terminal` = applicationLauncher::PSO\_TERMINAL, `LogOutput` = applicationLauncher::PSO\_LOGOUTPUT, `StdOutput` = applicationLauncher::PSO\_STDOUPUT }

- enum `CreationOptionNames` {
   
`Open` = QEActionRequests::OptionOpen,     `NewTab` = QEActionRequests::OptionNewTab,     `NewWindow` = QEActionRequests::OptionNewWindow,     `DockTop` = QEActionRequests::OptionTopDockWindow,
   
`DockBottom` = QEActionRequests::OptionBottomDockWindow,     `DockLeft` = QEActionRequests::OptionLeftDockWindow,     `DockRight` = QEActionRequests::OptionRightDockWindow,     `DockTopTabbed` = QEActionRequests::OptionTopDockWindowTabbed,
   
`DockBottomTabbed` = QEActionRequests::OptionBottomDockWindowTabbed,     `DockLeftTabbed` = QEActionRequests::OptionLeftDockWindowTabbed,     `DockRightTabbed` = QEActionRequests::OptionRightDockWindowTabbed,     `DockFloating` = QEActionRequests::OptionFloatingDockWindow }

*Creation options. Used to indicate how to present a GUI when requesting a new GUI be created. Open a new window, open a new tab, or replace the current window.*

## Public Slots

- void `requestAction` (const QEActionRequests &request)
- void `setDefaultStyle` (const QString &style)
   
*Update the default style applied to this widget.*
- void `setManagedVisible` (bool v)

## Signals

- void `dbValueChanged` (const QString &out)
- void `dbValueChanged` (const int &out)
- void `dbValueChanged` (const long &out)
- void `dbValueChanged` (const qlonglong &out)
- void `dbValueChanged` (const double &out)
- void `dbValueChanged` (const bool &out)
- void `dbConnectionChanged` (const bool &isConnected)
   
*Sent when the widget state updated following a channel connection change.*
- void `requestResend` ()
   
*Internal use only. Used when changing a property value to force a re-display to reflect the new property value.*
- void `newGui` (const QEActionRequests &request)
   
*Internal use only. Request a new GUI is created. Typically, this is caught by the QEGui application.*
- void `pressed` (int value)
- void `released` (int value)

- void [clicked](#) (int value)
- void [programCompleted](#) ()

*Program started by button has completed.*

## Public Member Functions

- [QERadioButton](#) (QWidget \*parent=0)
- [QERadioButton](#) (const QString &variableName, QWidget \*parent=0)
- void [writeNow](#) ()
- [UserLevels getUserLevelVisibilityProperty](#) ()  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- void [setUserLevelVisibilityProperty](#) (UserLevels level)  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- [UserLevels getUserLevelEnabledProperty](#) ()  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- void [setUserLevelEnabledProperty](#) (UserLevels level)  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- [DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty](#) ()  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void [setDisplayAlarmStateOptionProperty](#) (DisplayAlarmStateOptions option)  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void [setFormatProperty](#) (Formats format)  
*Access function for format property - refer to format property for details.*
- [Formats getFormatProperty](#) ()  
*Access function for format property - refer to format property for details.*
- void [setSeparatorProperty](#) (const Separators notation)  
*Access function for separator property - refer to separator property for details.*
- [Separators getSeparatorProperty](#) () const  
*Access function for separator property - refer to separator property for details.*
- void [setNotationProperty](#) (Notations notation)

*Access function for [notation](#) property - refer to [notation](#) property for details.*

- [Notations getNotationProperty \(\)](#)

*Access function for [notation](#) property - refer to [notation](#) property for details.*

- void [setArrayActionProperty \(ArrayActions arrayAction\)](#)

*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*

- [ArrayActions getArrayActionProperty \(\)](#)

*Access function for [arrayAction](#) property - refer to [arrayAction](#) property for details.*

## Properties

- [QString variable](#)
- [QString variableSubstitutions](#)
- int [arrayIndex](#)
- bool [subscribe](#)
- bool [variableAsToolTip](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- [QString styleSheet](#)
- [QString defaultStyle](#)
- [QString userLevelUserStyle](#)
- [QString userLevelScientistStyle](#)
- [QString userLevelEngineerStyle](#)
- [UserLevels userLevelVisibility](#)
- [UserLevels userLevelEnabled](#)
- bool [displayAlarmState](#)
- [DisplayAlarmStateOptions displayAlarmStateOption](#)
- int [precision](#)
- bool [useDbPrecision](#)
- bool [leadingZero](#)
- bool [trailingZeros](#)
- bool [addUnits](#)
- [QString localEnumeration](#)
- [Formats format](#)
- int [radix](#)
- [Separators separator](#)
- [Notations notation](#)
- [ArrayActions arrayAction](#)
- [QEWidgetProperties::DisabledRecordPolicy disabledRecordPolicy](#)
- [Qt::Alignment alignment](#)
- [UpdateOptions updateOption](#)
- [QPixmap pixmap0](#)

- QPixmap `pixmap1`
- QPixmap `pixmap2`
- QPixmap `pixmap3`
- QPixmap `pixmap4`
- QPixmap `pixmap5`
- QPixmap `pixmap6`
- QPixmap `pixmap7`
- QString `password`
- bool `confirmAction`
- QString `confirmText`
- bool `writeOnPress`
- bool `writeOnRelease`
- bool `writeOnClick`
- QString `pressText`
- QString `releaseText`
- QString `clickText`
- QString `clickCheckedText`
- QString `labelText`
- QString `program`
- QStringList `arguments`
- ProgramStartupOptionNames `programStartupOption`
- QString `guiFile`
- CreationOptionNames `creationOption`
- QString `prioritySubstitutions`
- QString `customisationName`

### 9.153.1 Member Enumeration Documentation

#### 9.153.1.1 enum QERadioButton::ArrayActions

User friendly enumerations for arrayAction property - refer to [QEStringFormatting::arrayActions](#) for details.

**Enumerator:**

*Append* Refer to [QEStringFormatting::APPEND](#) for details.

*Ascii* Refer to [QEStringFormatting::ASCII](#) for details.

*Index* Refer to [QEStringFormatting::INDEX](#) for details.

#### 9.153.1.2 enum QERadioButton::CreationOptionNames

Creation options. Used to indicate how to present a GUI when requesting a new GUI be created. Open a new window, open a new tab, or replace the current window.

**Enumerator:**

*Open* Replace the current GUI with the new GUI.

**NewTab** Open new GUI in a new tab.

**NewWindow** Open new GUI in a new window.

**DockTop** Open new GUI in a top dock window.

**DockBottom** Open new GUI in a bottom dock window.

**DockLeft** Open new GUI in a left dock window.

**DockRight** Open new GUI in a right dock window.

**DockTopTabbed** Open new GUI in a top dock window (tabbed with any existing dock in that area).

**DockBottomTabbed** Open new GUI in a bottom dock window (tabbed with any existing dock in that area).

**DockLeftTabbed** Open new GUI in a left dock window (tabbed with any existing dock in that area).

**DockRightTabbed** Open new GUI in a right dock window (tabbed with any existing dock in that area).

**DockFloating** Open new GUI in a floating dock window.

#### 9.153.1.3 enum QERadioButton::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

**Never** Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

**Always** Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

**WhenInAlarm** Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.153.1.4 enum QERadioButton::Formats

User friendly enumerations for format property - refer to [QEStringFormatting::formats](#) for details.

##### Enumerator:

**Default** Format as best appropriate for the data type.

**Floating** Format as a floating point number.

**Integer** Format as an integer.

**UnsignedInteger** Format as an unsigned integer.

**Time** Format as a time.

**LocalEnumeration** Format as a selection from the [localEnumeration](#) property.

### 9.153.1.5 enum QERadioButton::Notations

User friendly enumerations for notation property - refer to [QEStringFormatting::notations](#) for details.

#### Enumerator:

*Fixed* Refer to [QEStringFormatting::NOTATION\\_FIXED](#) for details.

*Scientific* Refer to [QEStringFormatting::NOTATION\\_SCIENTIFIC](#) for details.

*Automatic* Refer to [QEStringFormatting::NOTATION\\_AUTOMATIC](#) for details.

### 9.153.1.6 enum QERadioButton::ProgramStartupOptionNames

Startup options. Just run the command, run the command within a terminal, or display the output in QE message system.

#### Enumerator:

*None* Just run the program.

*Terminal* Run the program in a terminal (in Windows a command interpreter will also be started, so the program may be a built-in command like 'dir').

*LogOutput* Run the program, and log the output in the QE message system.

*StdOutput* Run the program, and send output to standard output and standard error.

### 9.153.1.7 enum QERadioButton::Separators

User friendly enumerations for separator property - refer to [QEStringFormatting::formats](#) for details.

#### Enumerator:

*NoSeparator* Use no separator.

*Comma* Use ',' as separator.

*Underscore* Use '\_' as separator.

*Space* Use ' ' as separator.

### 9.153.1.8 enum QERadioButton::UpdateOptions

User friendly enumerations for updateOption property - refer to [QEGenericButton::updateOptions](#) for details.

#### Enumerator:

*Text* Data updates will update the button text.

**Icon** Data updates will update the button icon.

**TextAndIcon** Data updates will update the button text and icon.

**State** Data updates will update the button state (checked or unchecked).

### 9.153.1.9 enum QERadioButton::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

#### Enumerator:

**User** Refer to `USERLEVEL_USER` for details.

**Scientist** Refer to `USERLEVEL_SCIENTIST` for details.

**Engineer** Refer to `USERLEVEL_ENGINEER` for details.

## 9.153.2 Constructor & Destructor Documentation

### 9.153.2.1 QERadioButton::QERadioButton (QWidget \* *parent* = 0)

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

### 9.153.2.2 QERadioButton::QERadioButton (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

## 9.153.3 Member Function Documentation

### 9.153.3.1 void QERadioButton::clicked (int *value*) [signal]

Button has been Clicked. The value emitted is the integer interpretation of the `clickText` property (or the `clickCheckedText` property if the button was checked)

### 9.153.3.2 void QERadioButton::dbValueChanged (const QString & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.153.3.3 void QERadioButton::pressed (int *value*) [signal]**

Button has been Pressed. The value emitted is the integer interpretation of the pressText property

**9.153.3.4 void QERadioButton::released (int *value*) [signal]**

Button has been Released The value emitted is the integer interpretation of the releaseText property

**9.153.3.5 void QERadioButton::requestAction (const QEActionRequests & *request*) [inline, slot]**

Default slot used to create a new GUI if there is no slot indicated in the ContainerProfile class. This slot is typically used when the button is pressed within the Designer preview window to allow the operation of the button to be tested. If an application does not specify a slot to use for creating new windows (through the ContainerProfile class) a window will still be created through this slot, but it will not respect the window creation options or any other window related application constraints. For example, the QEgui application does provide a slot for creating new GUIs in the ContainerProfile class which respects the creation options, knows how to add tabs in the application, and extend the application's window menu in the menu bar.

**9.153.3.6 void QERadioButton::setManagedVisible (bool *v*) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.153.4 Property Documentation

**9.153.4.1 bool QERadioButton::addUnits [read, write]**

If true (default), add engineering units supplied with the data.

**9.153.4.2 Qt::Alignment QERadioButton::alignment [read, write]**

Set the buttons text alignment. Left justification is particularly useful when displaying quickly changing numeric data updates.

**9.153.4.3 bool QERadioButton::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.153.4.4 QStringList QERadioButton::arguments [read, write]**

Arguments for program specified in the 'program' property.

**9.153.4.5 ArrayActions QERadioButton::arrayAction [read, write]**

Text formatting option for array data. Default is ASCII. Options are:

- ASCII - treat array as a single text string. For example an array of three characters 'a' 'b' 'c' will be formatted as 'abc'.
- APPEND - treat array as an array of numbers and format a string containing them all with a space between each. For example, an array of three numbers 10, 11 and 12 will be formatted as '10 11 12'.
- INDEX - Extract a single item from the array. The item is then formatted as any other non array data would be. The item selected is determined by the arrayIndex property. For example, if arrayIndex property is 1, an array of three numbers 10, 11 and 12 will be formatted as '11'.

**9.153.4.6 int QERadioButton::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.153.4.7 QString QERadioButton::clickCheckedText [read, write]**

Text used to compare with text written or read to determine if push button should be marked as checked. Note, must be an exact match following formatting of data updates. When writing values, the 'pressText', 'ReleaseText', or 'clickedtext' must match this property to cause the button to be checked when the write occurs.

Good example: formatting set to display a data value of '1' as 'On', clickCheckedText is 'On', clickText is 'On'. In this example, the push button will be checked when a data update occurs with a value of 1 or when the button is clicked.

Bad example: formatting set to display a data value of '1' as 'On', clickCheckedText is 'On', clickText is '1'. In this example, the push button will be checked when a data update occurs with a value of 1 but, although a valid value will be written when clicked, the button will not be checked when clicked as '1' is not the same as 'On'.

Reimplemented from [QEGenericButton](#).

**9.153.4.8 QString QERadioButton::clickText [read, write]**

Value written when user clicks button if 'writeOnClick' property is true

Reimplemented from [QEGenericButton](#).

**9.153.4.9 bool QERadioButton::confirmAction [read, write]**

If true, a dialog will be presented asking the user to confirm if the button action should be carried out

**9.153.4.10 QString QERadioButton::confirmText [read, write]**

Text used to confirm action if confirmation dialog is presented

Reimplemented from [QEGenericButton](#).

**9.153.4.11 CreationOptionNames QERadioButton::creationOption [read, write]**

Creation options when opening a new GUI. Open a new window, open a new tab, or replace the current window. The creation option is supplied when the button generates a newGui signal. Application code connected to this signal should honour this request if possible. When used within the QEgui application, the QEgui application creates a new window, new tab, or replaces the current window as appropriate.

Reimplemented from [QEGenericButton](#).

**9.153.4.12 QString QERadioButton::customisationName [read, write]**

Window customisation name. This name will be used to select a set of window customisations including menu items and tool bar buttons. Applications such as QEgui can load .xml files containing named sets of window customisations. This property is used to select a set loaded from these files. The selected set of customisations will be applied to the main window containing the new GUI. Customisations are not applied if the GUI is opened as a dock.

Reimplemented from [QEGenericButton](#).

**9.153.4.13 QString QERadioButton::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.153.4.14 QEWidgetProperties::DisabledRecordPolicy  
QERadioButton::disabledRecordPolicy [read, write]**

Set the widget's disabled record policy, i.e. the action to be taken when the under lying record is disabled, i.e. when the assiociated record's DISA and DISV field values are equal. Note: this is only applicable IOC process variables. When the policy is ignore, then no special action is taken. This is the default policy. When the policy is grayout, the widget is style is set as if disconnected when the record is disabled.

Reimplemented from [QEGenericButton](#).

**9.153.4.15 bool QERadioButton::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.153.4.16 DisplayAlarmStateOptions QERadioButton::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.153.4.17 Formats QERadioButton::format [read, write]**

Format to apply to data. Default is 'Default' in which case the data type supplied with the data determines how the data is formatted. For all other options, an attempt is made to format the data as requested (whatever its native form).

**9.153.4.18 QString QERadioButton::guiFile [read, write]**

File name of GUI to be presented on button click. File name can be absolute, relative to the path of the QEform in which the [QEPushButton](#) is located, relative to the any path in the path list published in the ContainerProfile class, or relative to the current path. See [QEWidget::openQEFfile\(\)](#) in [QEWidget.cpp](#) for details.

**9.153.4.19 unsigned QERadioButton::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.153.4.20 QString QERadioButton::labelText [read, write]**

Button label text (prior to substitution). Macro substitutions will be applied to this text and the result will be set as the button text. Used when data updates are not being represented in the button text. IF NOT LEFT EMPTY, THIS TEXT WILL TAKE

PRIORITY OVER THE PUSH BUTTON 'text' PROPERTY! For example, a button in a sub form may have a 'labelText' property of 'Turn Pump On'. When the sub form is used twice in a main form with substitutions PUMPNUM=1 and PUMPNUM=2 respectively, the two identical buttons in the sub forms will have the labels 'Turn Pump 1 On' and 'Turn Pump 2 On' respectively.

Reimplemented from [QEGenericButton](#).

#### **9.153.4.21 bool QERadioButton::leadingZero [read, write]**

If true (default), always add a leading zero when formatting numbers.

#### **9.153.4.22 QString QERadioButton::localEnumeration [read, write]**

An enumeration list used to data values. Used only when the formatting option is 'local enumeration'. Value is converted to an integer and used to select a string from this list.

Format is:

[[<|<=|=|=|>|=|>]value1]\*] : string1 , [[<|<=|=|=|>|=|>]value2]\*] : string2 ,  
[[<|<=|=|=|>|=|>]value3]\*] : string3 , ...

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
>= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm"  
<2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than  
2" 3:"Beamline Available", \*:"" "Pump Off":"OH NO!, the pump is OFF!","Pump  
On":"It's OK, the pump is on"

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10:""'

A range of numbers can be covered by a pair of values as in the following example:  
>=4:"Between 4 and 8",<=8:"Between 4 and 8"

#### **9.153.4.23 Notations QERadioButton::notation [read, write]**

Notation used for numerical formatting. Default is fixed.

**9.153.4.24 QString QERadioButton::password [read, write]**

Password user will need to enter before any action is taken

Reimplemented from [QEGenericButton](#).

**9.153.4.25 QPixmap QERadioButton:: pixmap0 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 0

**9.153.4.26 QPixmap QERadioButton:: pixmap1 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 1

**9.153.4.27 QPixmap QERadioButton:: pixmap2 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 2

**9.153.4.28 QPixmap QERadioButton:: pixmap3 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 3

**9.153.4.29 QPixmap QERadioButton:: pixmap4 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 4

**9.153.4.30 QPixmap QERadioButton:: pixmap5 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 5

**9.153.4.31 QPixmap QERadioButton:: pixmap6 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 6

**9.153.4.32 QPixmap QERadioButton:: pixmap7 [read, write]**

Pixmap to display if updateOption is Icon or TextAndIcon and data value translates to an index of 7

**9.153.4.33 int QERadioButton::precision [read, write]**

Precision used when formatting floating point numbers. The default is 4. This is only used if useDbPrecision is false.

**9.153.4.34 QString QERadioButton::pressText [read, write]**

Value written when user presses button if 'writeOnPress' property is true

Reimplemented from [QEGenericButton](#).

**9.153.4.35 QString QERadioButton::prioritySubstitutions [read, write]**

Overriding macro substitutions. These macro substitions take precedence over any existing macro substitutions defined by the variableSubstitutions property, any parent forms, or the application containing the button. These macro substitutions are particularly usefull when the button's function is to reload the same form but with different macro substitutions. The variableSubstitutions property cannot be used for this since, although they are added to the list of macro substitions applied to the new form, they are appended to the list and the existing macro substitutions take precedence.

Reimplemented from [QEGenericButton](#).

**9.153.4.36 QString QERadioButton::program [read, write]**

Program to run when the button is clicked. No attempt to run a program is made if this property is empty. Example: firefox

**9.153.4.37 ProgramStartupOptionNames QERadioButton::programStartupOption [read, write]**

Startup options. Just run the command, run the command within a terminal, or display the output in QE message system.

**9.153.4.38 int QERadioButton::radix [read, write]**

Base used for when formatting integers. Default is 10 (duh!)

**9.153.4.39 QString QERadioButton::releaseText [read, write]**

Value written when user releases button if 'writeOnRelease' property is true

Reimplemented from [QEGenericButton](#).

**9.153.4.40 Separators QERadioButton::separator [read, write]**

Separators used for integer and fixed point formatting. Default is None.

**9.153.4.41 QString QERadioButton::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.153.4.42 bool QERadioButton::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.153.4.43 bool QERadioButton::trailingZeros [read, write]**

If true (default), always remove any trailing zeros when formatting numbers.

**9.153.4.44 UpdateOptions QERadioButton::updateOption [read, write]**

Update options (text, pixmap, both, or state (checked or unchecked)

Reimplemented from [QEGenericButton](#).

**9.153.4.45 bool QERadioButton::useDbPrecision [read, write]**

If true (default), format floating point numbers using the precision supplied with the data. If false, the precision property is used.

**9.153.4.46 UserLevels QERadioButton::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.153.4.47 QString QERadioButton::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager

class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.153.4.48 QString QERadioButton::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.153.4.49 QString QERadioButton::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.153.4.50 UserLevels QERadioButton::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.153.4.51 QString QERadioButton::variable [read, write]**

EPICS variable name (CA PV)

#### **9.153.4.52 bool QERadioButton::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### **9.153.4.53 QString QERadioButton::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For exam-

ple, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### **9.153.4.54 bool QERadioButton::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

#### **9.153.4.55 bool QERadioButton::writeOnClick [read, write]**

If true, the 'clickText' property is written when the button is clicked. Default is true  
Reimplemented from [QEGenericButton](#).

#### **9.153.4.56 bool QERadioButton::writeOnPress [read, write]**

If true, the 'pressText' property is written when the button is pressed. Default is false  
Reimplemented from [QEGenericButton](#).

#### **9.153.4.57 bool QERadioButton::writeOnRelease [read, write]**

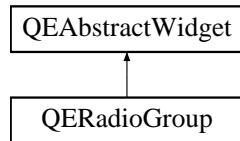
If true, the 'releaseText' property is written when the button is released. Default is false  
Reimplemented from [QEGenericButton](#).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QERadioButton.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEButton/QERadioButton.cpp

## 9.154 QERadioGroup Class Reference

Inheritance diagram for QERadioGroup::



### Public Slots

- void [setDefaultStyle](#) (const QString &style)  
*Update the default style applied to this widget.*

### Signals

- void [dbValueChanged](#) (const QString &out)
- void [dbValueChanged](#) (const int &out)
- void [dbValueChanged](#) (const long &out)
- void [dbValueChanged](#) (const qlonglong &out)
- void [dbValueChanged](#) (const double &out)
- void [dbValueChanged](#) (const bool &out)
- void [dbConnectionChanged](#) (const bool &isConnected)

### Public Member Functions

- [QERadioGroup](#) (QWidget \*parent=0)
- [QERadioGroup](#) (const QString &variableName, QWidget \*parent=0)
- [QERadioGroup](#) (const QString &title, const QString &variableName, QWidget \*parent=0)
- virtual [~QERadioGroup](#) ()  
*Destruction.*
- int [getCurrentIndex](#) () const
- void [setVariableNameSubstitutionsProperty](#) (const QString &substitutions)
- void [setSubstitutedTitleProperty](#) (const QString &substitutedTitle)
- QString [getSubstitutedTitleProperty](#) () const
- void [setUseDbEnumerations](#) (bool useDbEnumerations)
- bool [getUseDbEnumerations](#) () const
- void [setLocalEnumerations](#) (const QString &localEnumerations)
- QString [getLocalEnumerations](#) () const

## Protected Member Functions

- QSize **sizeHint** () const
- void **fontChange** (const QFont &font)
- void **activated** ()
- void **establishConnection** (unsigned int variableIndex)
- **qcaobject::QCaObject** \* **createQcaItem** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- void **mousePressEvent** (QMouseEvent \*event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)

## Properties

- QString **variable**
- QString **variableSubstitutions**
- int **arrayIndex**
- QString **substitutedTitle**
- QString **title**
- int **columns**
- int **spacing**
- bool **useDbEnumerations**
- QString **localEnumeration**
- **QRadioButton::ButtonStyles** **buttonStyle**
- **QRadioButton::ButtonOrders** **buttonOrder**

### 9.154.1 Constructor & Destructor Documentation

#### 9.154.1.1 **QERadioGroup::QERadioGroup (QWidget \* *parent* = 0) [explicit]**

Create without a variable. Use `setVariableNameProperty()` and `setSubstitutionsProperty()` to define a variable and, optionally, macro substitutions later.

#### 9.154.1.2 **QERadioGroup::QERadioGroup (const QString & *variableName*, QWidget \* *parent* = 0) [explicit]**

Create with a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

**9.154.1.3 QERadioGroup::QERadioGroup (const QString & *title*, const QString & *variableName*, QWidget \* *parent* = 0) [explicit]**

Create with a group title and a variable. A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

## 9.154.2 Member Function Documentation

**9.154.2.1 void QERadioGroup::dbConnectionChanged (const bool & *isConnected*) [signal]**

Sent when the widget state updated following a channel connection change Applied to provary varible.

**9.154.2.2 void QERadioGroup::dbValueChanged (const QString & *out*) [signal]**

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

## 9.154.3 Property Documentation

**9.154.3.1 int QERadioGroup::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.154.3.2 QRadioGroup::ButtonOrders QERadioGroup::buttonOrder [read, write]**

Allows selection of buttom order (rowMajor (default) or colMajor)

**9.154.3.3 QRadioGroup::ButtonStyles QERadioGroup::buttonStyle [read, write]**

Allows selection of buttom style (Radio (default) or Push)

**9.154.3.4 int QERadioGroup::columns [read, write]**

Number of colums - defaults to two.

**9.154.3.5 QString QERadioGroup::localEnumeration [read, write]**

Enumerations values used when useDbEnumerations is false.

**9.154.3.6 int QERadioGroup::spacing [read, write]**

Interay layout margins and spacing - defaults to 4.

**9.154.3.7 QString QERadioGroup::substitutedTitle [read, write]**

Group box title text to be substituted. This text will be copied to the group box title text after applying any macro substitutions from the variableSubstitutions property. The former is deprecated and included for backward compatibility only and not presented on designer.

**9.154.3.8 bool QERadioGroup::useDbEnumerations [read, write]**

Use database enumerations - defaults to true. False implies use of local enumeration.

**9.154.3.9 QString QERadioGroup::variable [read, write]**

EPICS variable name (CA PV)

**9.154.3.10 QString QERadioGroup::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QERadioGroup/QERadioGroup.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QERadioGroup/QERadioGroup.cpp

## 9.155 QERadioGroupManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QERadioGroup/QERadioGroupManager.h

## 9.156 QEReipe Class Reference

### Public Types

- enum **configurationTypesProperty** { **File** = FROM\_FILE, **Text** = FROM\_TEXT }
- enum **optionsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }
- enum **userTypesProperty** { **User** = userLevelTypes::USERLEVEL\_USER, **Scientist** = userLevelTypes::USERLEVEL\_SCIENTIST, **Engineer** = userLevelTypes::USERLEVEL\_ENGINEER }

### Public Member Functions

- **QEReipe** (QWidget \*pParent=0)
- void **setRecipeDescription** (QString pValue)
- QString **getRecipeDescription** ()
- void **setShowRecipeList** (bool pValue)
- bool **getShowRecipeList** ()
- void **setShowNew** (bool pValue)
- bool **getShowNew** ()
- void **setShowSave** (bool pValue)
- bool **getShowSave** ()
- void **setShowDelete** (bool pValue)
- bool **getShowDelete** ()
- void **setShowApply** (bool pValue)
- bool **getShowApply** ()
- void **setShowRead** (bool pValue)
- bool **getShowRead** ()
- void **setShowFields** (bool pValue)
- bool **getShowFields** ()
- void **setConfigurationType** (int pValue)
- int **getConfigurationType** ()
- void **setConfigurationFile** (QString pValue)
- QString **getConfigurationFile** ()
- void **setRecipeFile** (QString pValue)
- QString **getRecipeFile** ()
- void **setConfigurationText** (QString pValue)
- QString **getConfigurationText** ()
- void **setOptionsLayout** (int pValue)
- int **getOptionsLayout** ()
- void **setCurrentUserType** (int pValue)
- int **getCurrentUserType** ()
- bool **saveRecipeList** ()
- void **refreshRecipeList** ()
- void **refreshButton** ()

- void **userLevelChanged** (userLevelTypes::userLevels pValue)
- void **setConfigurationTypeProperty** (configurationTypesProperty pConfigurationType)
- configurationTypesProperty **getConfigurationTypeProperty** ()
- void **setOptionsLayoutProperty** (optionsLayoutProperty pOptionsLayout)
- optionsLayoutProperty **getOptionsLayoutProperty** ()
- void **setCurrentUserTypeProperty** (userTypesProperty pUserType)
- userTypesProperty **getCurrentUserTypeProperty** ()

## Protected Attributes

- QLabel \* **qLabelRecipeDescription**
- QComboBox \* **qComboBoxRecipeList**
- QPushButton \* **qPushButtonNew**
- QPushButton \* **qPushButtonSave**
- QPushButton \* **qPushButtonDelete**
- QPushButton \* **qPushButtonApply**
- QPushButton \* **qPushButtonRead**
- QEConfiguredLayout \* **qEConfiguredLayoutRecipeFields**
- QDomDocument **document**
- QString **recipeFile**
- QString **filename**
- int **optionsLayout**
- int **currentUserType**

## Properties

- QString **recipeDescription**
- bool **showRecipeList**
- bool **showNew**
- bool **showSave**
- bool **showDelete**
- bool **showApply**
- bool **showRead**
- bool **showFields**
- configurationTypesProperty **configurationType**
- QString **configurationFile**
- QString **configurationText**
- optionsLayoutProperty **optionsLayout**
- userTypesProperty **currentUserType**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEReipe/QEReipe.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEReipe/QEReipe.cpp

## 9.157 QERecordSpec Class Reference

### Public Member Functions

- **QERecordSpec** (const QString recordType)
- **QString getRecordType () const**
- **QString getFieldName (const int index) const**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.cpp

## 9.158 QERecordSpecList Class Reference

### Public Member Functions

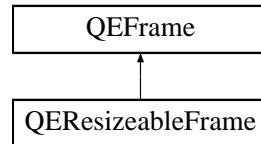
- `QERecordSpec * find (const QString recordType) const`
- `void appendOrReplace (QERecordSpec *recordSpec)`
- `bool processRecordSpecFile (const QString &filename)`

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvProperties/QEPvPropertiesUtilities.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPvProperties/QEPvPropertiesUtilities.cpp`

## 9.159 QEResizableFrame Class Reference

#include <QEResizableFrame.h>  
Inheritance diagram for QEResizableFrame::



### Public Types

- enum **GrabbingEdges** { **TopEdge**, **LeftEdge**, **BottomEdge**, **RightEdge** }

### Public Member Functions

- **QEResizableFrame** (QWidget \*parent=0)
- **QEResizableFrame** (GrabbingEdges grabbingEdge, int minimum, int maximum, QWidget \*parent=0)  
*Construct widget specifying min and max allowed heights.*
- QWidget \* **widget** () const  
*Returns a ref to the resizable frame's widget, or 0 if there is none.*
- void **setWidget** (QWidget \*widget)
- QWidget \* **takeWidget** ()  
*Removes the resizable frame's widget, and passes ownership management of the widget to the caller.*
- void **setGrabberToolTip** (const QString &tip)  
*Set the tool tip for the internal grabber object.*
- void **setAllowedMinimum** (const int minimum)  
*(Re)set allowed limits.*
- int **getAllowedMinimum** () const
- void **setAllowedMaximum** (const int maximum)
- int **getAllowedMaximum** () const
- void **setGrabbingEdge** (const GrabbingEdges edge)
- GrabbingEdges **getGrabbingEdge** () const

### Protected Member Functions

- bool **eventFilter** (QObject \*obj, QEvent \*event)

## Properties

- GrabbingEdges [grabbingEdge](#)
- int [allowedMinimum](#)
- int [allowedMaximum](#)

### 9.159.1 Detailed Description

The [QEResizableFrame](#) provides a frame capable of holding another widget together with a grabber widget that allows the frame to be re-sized, and hence contained widget to be resized. The class currently only supports vertical or horizontal resizing, but not both.

### 9.159.2 Member Function Documentation

#### 9.159.2.1 void QEResizableFrame::setWidget (QWidget \* *widget*)

Sets the resizeable frame's widget. The widget becomes a child of the resizable frame, and will be destroyed when the resizable frame is deleted or when a new widget is set. Any existing widget is deleted - use takeWidget first if needs be.

### 9.159.3 Property Documentation

#### 9.159.3.1 int QEResizableFrame::allowedMaximum [read, write]

Set the maximum allowed size (defaults to 100).

#### 9.159.3.2 int QEResizableFrame::allowedMinimum [read, write]

Set the minimum allowed size (defaults to 10).

#### 9.159.3.3 GrabbingEdges QEResizableFrame::grabbingEdge [read, write]

Nominated edge for the grabbing location. Defaults to BottomEdge.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEResizableFrame/QEResizableFrame.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEResizableFrame/QEResizableFrame.cpp

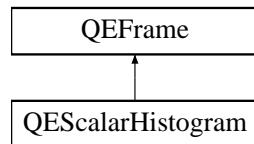
## 9.160 QEResizableFrameManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEResizableFrame/QEResiz

## 9.161 QEScalarHistogram Class Reference

Inheritance diagram for QEScalarHistogram::



### Public Types

- enum [ScaleModes](#) { [Manual](#), [Auto](#), [OperationalRange](#) }

### Signals

- void **mouseIndexChanged** (const int index)
- void **mouseIndexPressed** (const int index, const Qt::MouseButton button)

### Public Member Functions

- **QEScalarHistogram** (QWidget \*parent=0)
- void **setScaleMode** (const [ScaleModes](#) scaleMode)
- [ScaleModes](#) **getScaleMode** () const

### Protected Member Functions

- [qcaobject::QCaObject](#) \* **createQcaItem** (unsigned int variableIndex)
- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- void **mousePressEvent** (QMouseEvent \*event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant v)
- void **addPvName** (const QString &pvName)

### Properties

- QString [variableSubstitutions](#)
- bool [autoBarGapWidths](#)
- int [barWidth](#)
- int [gap](#)
- int [margin](#)

- **ScaleModes scaleMode**
- double **minimum**
- double **maximum**
- double **baseLine**
- bool **drawAxes**
- bool **showScale**
- bool **showGrid**
- bool **logScale**
- QColor **backgroundColour**
- QColor **barColour**
- bool **drawBorder**
- Qt::Orientation **orientation**
- QString **variable1**

*EPICS variable names (CA PV).*

- QString **variable2**
- QString **variable3**
- QString **variable4**
- QString **variable5**
- QString **variable6**
- QString **variable7**
- QString **variable8**
- QString **variable9**
- QString **variable10**
- QString **variable11**
- QString **variable12**
- QString **variable13**
- QString **variable14**
- QString **variable15**
- QString **variable16**
- QString **variable17**
- QString **variable18**
- QString **variable19**
- QString **variable20**
- QString **variable21**
- QString **variable22**
- QString **variable23**
- QString **variable24**
- QString **variable25**
- QString **variable26**
- QString **variable27**
- QString **variable28**
- QString **variable29**
- QString **variable30**
- QString **variable31**
- QString **variable32**

- **QString variable33**
- **QString variable34**
- **QString variable35**
- **QString variable36**
- **QString variable37**
- **QString variable38**
- **QString variable39**
- **QString variable40**
- **QString variable41**
- **QString variable42**
- **QString variable43**
- **QString variable44**
- **QString variable45**
- **QString variable46**
- **QString variable47**
- **QString variable48**
- **QString variable49**
- **QString variable50**
- **QString variable51**
- **QString variable52**
- **QString variable53**
- **QString variable54**
- **QString variable55**
- **QString variable56**
- **QString variable57**
- **QString variable58**
- **QString variable59**
- **QString variable60**
- **QString variable61**
- **QString variable62**
- **QString variable63**
- **QString variable64**
- **QString variable65**
- **QString variable66**
- **QString variable67**
- **QString variable68**
- **QString variable69**
- **QString variable70**
- **QString variable71**
- **QString variable72**
- **QString variable73**
- **QString variable74**
- **QString variable75**
- **QString variable76**
- **QString variable77**
- **QString variable78**

- `QString variable79`
- `QString variable80`
- `QString variable81`
- `QString variable82`
- `QString variable83`
- `QString variable84`
- `QString variable85`
- `QString variable86`
- `QString variable87`
- `QString variable88`
- `QString variable89`
- `QString variable90`
- `QString variable91`
- `QString variable92`
- `QString variable93`
- `QString variable94`
- `QString variable95`
- `QString variable96`
- `QString variable97`
- `QString variable98`
- `QString variable99`
- `QString variable100`
- `QString variable101`
- `QString variable102`
- `QString variable103`
- `QString variable104`
- `QString variable105`
- `QString variable106`
- `QString variable107`
- `QString variable108`
- `QString variable109`
- `QString variable110`
- `QString variable111`
- `QString variable112`
- `QString variable113`
- `QString variable114`
- `QString variable115`
- `QString variable116`
- `QString variable117`
- `QString variable118`
- `QString variable119`
- `QString variable120`

### 9.161.1 Member Enumeration Documentation

#### 9.161.1.1 enum QEScalarHistogram::ScaleModes

Enumerator:

*Manual* Use property minimum/maximum to scale [histogram](#).

*Auto* Dynamically scale based on minimum/maximum displayed value.

*OperationalRange* Use process variable operational range (LOPR/HOPR).

### 9.161.2 Property Documentation

#### 9.161.2.1 QString QEScalarHistogram::variableSubstitutions [read, write]

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'SAMPLE=SAM1, NAME = "Ref foil"' These substitutions are applied to all the variable names.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEScalarHistogram.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEScalarHistogram.cpp

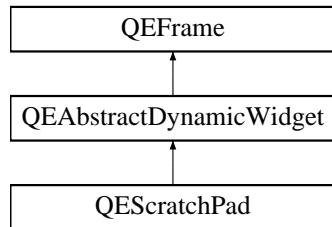
## 9.162 QEScalarHistogramManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEScalarHistog

## 9.163 QEScratchPad Class Reference

#include <QEScratchPad.h> Inheritance diagram for QEScratchPad::



### Classes

- class **DataSets**

### Public Slots

- void **setSelection** (int value)
- void **setPvNameSet** (const QStringList &pvNameSet)

### Signals

- void **selectionChanged** (int value)
- void **pvNameSetChanged** (const QStringList &nameSet)

### Public Member Functions

- **QEScratchPad** (QWidget \*parent=0)
- QSize **sizeHint** () const
- void **setPvName** (const int slot, const QString &pvName)
- QString **getPvName** (const int slot) const
- int **getSelection** () const
- QStringList **getPvNameSet** () const

### Static Public Attributes

- static const int **NUMBER\_OF\_ITEMS** = 72

## Protected Member Functions

- void **resizeEvent** (QResizeEvent \*event)
- void **activated** ()
- QMenu \* **buildContextMenu** ()
- void **contextMenuTriggered** (int selectedItemNum)
- bool **itemLessThan** (const int a, const int b, QObject \*context=NULL) const
- void **swapItems** (const int a, const int b, QObject \*context=NULL)
- void **mousePressEvent** (QMouseEvent \*event)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dragMoveEvent** (QDragMoveEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)
- void **saveConfiguration** (PersistanceManager \*pm)
- void **restoreConfiguration** (PersistanceManager \*pm, restorePhases restorePhase)
- void **showEvent** (QShowEvent \*event)
- void **keyPressEvent** (QKeyEvent \*event)
- bool **eventFilter** (QObject \*obj, QEvent \*event)
- int **findSlot** (QObject \*obj) const

### 9.163.1 Detailed Description

This class provides a flexible scratch pad form, to which any Process Variable may be added. It displays the PV Name, the Description, i.e. the content of the DESC field together with the value of PV itself.

This class is a direct re-implementation of TScratch\_Pad\_Form out of the Delphi OPI framework.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScratchPad/QEScratchPad.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScratchPad/QEScratchPad.cpp

## 9.164 QEScratchPadManager Class Reference

### Public Member Functions

- **QEScratchPadManager** (QObject \*parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget \* **createWidget** (QWidget \*parent)
- void **initialize** (QDesignerFormEditorInterface \*core)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScratchPad/QEScratchPadManager.h

## 9.165 QEScratchPadMenu Class Reference

### Public Types

- enum **ContextMenuOptions** {
   
**SCRATCHPAD\_NONE** = QEAbstractDynamicWidget::ADWCM\_SUB\_CLASS\_WIDGETS\_START\_HERE, **SCRATCHPAD\_SORT\_PV\_NAMES**,  
**SCRATCHPAD\_CLEAR\_ALL**, **SCRATCHPAD\_ADD\_PV\_NAME**,  
**SCRATCHPAD\_PASTE\_PV\_NAME**, **SCRATCHPAD\_EDIT\_PV\_NAME**,  
**SCRATCHPAD\_DATA\_CLEAR** }

### Signals

- void **contextMenuItemSelected** (const int, const QEScratchPadMenu::ContextMenuOptions)

### Public Member Functions

- **QEScratchPadMenu** (const int slot, QWidget \*parent=0)
- void **setActionChecked** (const ContextMenuOptions option, const bool visible)
  
- void **setActionEnabled** (const ContextMenuOptions option, const bool visible)
- void **setActionVisible** (const ContextMenuOptions option, const bool visible)
- void **setIsInUse** (const bool isInUse)

### Static Public Attributes

- static const ContextMenuOptions **ContextMenuFirst** = SCRATCHPAD\_ADD\_PV\_NAME
- static const ContextMenuOptions **ContextMenuLast** = SCRATCHPAD\_DATA\_CLEAR
- static const int **NumberContextMenuItems** = ContextMenuItemLast - ContextMenuItemFirst + 1

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScratchPad/QEScratchPad.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScratchPad/QEScratchPad.cpp

## 9.166 QEScript Class Reference

```
#include <QEScript.h>
```

### Public Types

- enum **scriptTypesProperty** { **File** = FROM\_FILE, **Text** = FROM\_TEXT }
- enum **optionsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }
- enum **UserLevels** { **User** = userLevelTypes::USERLEVEL\_USER, **Scientist** = userLevelTypes::USERLEVEL\_SCIENTIST, **Engineer** = userLevelTypes::USERLEVEL\_ENGINEER }
- enum **DisplayAlarmStateOptions** { **Never** = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, **Always** = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, **WhenInAlarm** = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void **setManagedVisible** (bool v)

### Signals

- void **selected** (QString pFilename)

### Public Member Functions

- **QEScript** (QWidget \*pParent=0)
- void **setShowScriptList** (bool pValue)
- bool **getShowScriptList** ()
- void **setShowNew** (bool pValue)
- bool **getShowNew** ()
- void **setShowSave** (bool pValue)
- bool **getShowSave** ()
- void **setShowDelete** (bool pValue)
- bool **getShowDelete** ()
- void **setShowExecute** (bool pValue)
- bool **getShowExecute** ()
- void **setShowAbort** (bool pValue)
- bool **getShowAbort** ()
- void **setEditableTable** (bool pValue)
- bool **getEditableTable** ()
- void **setShowTable** (bool pValue)
- bool **getShowTable** ()
- void **setShowTableControl** (bool pValue)

- bool **getShowTableControl** ()
- void **setShowColumnNumber** (bool pValue)
- bool **getShowColumnNumber** ()
- void **setShowColumnEnable** (bool pValue)
- bool **getShowColumnEnable** ()
- void **setShowColumnProgram** (bool pValue)
- bool **getShowColumnProgram** ()
- void **setShowColumnParameters** (bool pValue)
- bool **getShowColumnParameters** ()
- void **setShowColumnWorkingDirectory** (bool pValue)
- bool **getShowColumnWorkingDirectory** ()
- void **setShowColumnTimeout** (bool pValue)
- bool **getShowColumnTimeout** ()
- void **setShowColumnStop** (bool pValue)
- bool **getShowColumnStop** ()
- void **setShowColumnLog** (bool pValue)
- bool **getShowColumnLog** ()
- void **setScriptType** (int pValue)
- int **getScriptType** ()
- void **setScriptFile** (QString pValue)
- QString **getScriptFile** ()
- void **setScriptText** (QString pValue)
- QString **getScriptText** ()
- void **setScriptDefault** (QString pValue)
- QString **getScriptDefault** ()
- void **setExecuteText** (QString pValue)
- QString **getExecuteText** ()
- void **setOptionsLayout** (int pValue)
- int **getOptionsLayout** ()
- void **insertRow** (bool pEnable, QString pProgram, QString pParameter, QString pWorkingDirectory, int pTimeOut, bool pStop, bool pLog)
- bool **saveScriptList** ()
- void **refreshScriptList** ()
- void **refreshWidgets** ()
- void **setScriptTypeProperty** (scriptTypesProperty pScriptType)
- scriptTypesProperty **getScriptTypeProperty** ()
- void **setOptionsLayoutProperty** (optionsLayoutProperty pOptionsLayout)
- optionsLayoutProperty **getOptionsLayoutProperty** ()
- UserLevels **getUserLevelVisibilityProperty** ()  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void **setUserLevelVisibilityProperty** (UserLevels level)  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- UserLevels **getUserLevelEnabledProperty** ()

*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*

- void `setUserLevelEnabledProperty` (`UserLevels` level)  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- `DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()`  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void `setDisplayAlarmStateOptionProperty` (`DisplayAlarmStateOptions` option)  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

## Protected Attributes

- `QComboBox * qComboBoxScriptList`
- `QPushButton * qPushButtonNew`
- `QPushButton * qPushButtonSave`
- `QPushButton * qPushButtonDelete`
- `QPushButton * qPushButtonExecute`
- `QPushButton * qPushButtonAbort`
- `QPushButton * qPushButtonAdd`
- `QPushButton * qPushButtonRemove`
- `QPushButton * qPushButtonUp`
- `QPushButton * qPushButtonDown`
- `QPushButton * qPushButtonCopy`
- `QPushButton * qPushButtonPaste`
- `_QTableWidgetScript * qTableWidgetScript`
- `QString scriptFile`

*Define the file where to save the scripts (if not defined then the scripts will be saved in a file named "QEScript.xml").*

- `QString scriptText`  
*Define the XML text that contains the scripts.*
- `QString scriptDefault`  
*Define the script (previously saved by the user) that will be load as the default script when the widget starts.*
- `int scriptType`
- `int optionsLayout`
- `QDomDocument document`
- `QString filename`
- `QList< _CopyPaste * > copyPasteList`

- bool **editableTable**  
*Enable/disable table edition.*
- bool **isExecuting**

## Properties

- bool **showScriptList**  
*Show/hide combobox that contains the list of existing scripts created by the user.*
- bool **showNew**  
*Show/hide button to reset (initialize) the table that contains the sequence of programs to be executed.*
- bool **showSave**  
*Show/hide button to save/overwrite a new/existing script.*
- bool **showDelete**  
*Show/hide button to delete an existing script.*
- bool **showExecute**  
*Show/hide button to execute a sequence of programs.*
- bool **showAbort**  
*Show/hide button to abort the execution of a sequence of programs.*
- bool **showTable**  
*Show/hide table that contains a sequence of programs to be executed.*
- bool **showTableControl**  
*Show/hide the controls of the table that contains a sequence of programs to be executed.*
- bool **showColumnNumber**  
*Show/hide the column '#' that displays the sequential number of programs.*
- bool **showColumnEnable**  
*Show/hide the column 'Enable' that enables the execution of programs.*
- bool **showColumnProgram**  
*Show/hide the column 'Program' that contains the external programs to be executed.*
- bool **showColumnParameters**  
*Show/hide the column 'Parameters' that contains the parameters that are passed to external programs to be executed.*

- bool [showColumnWorkingDirectory](#)

*Show/hide the column 'Directory' that defines the working directory to be used when external programs are executed.*

- bool [showColumnTimeout](#)

*Show/hide the column 'Timeout' that defines a time out period in seconds (if equal to 0 then the program runs until it finishes; otherwise if greater than 0 then the program will only run during this amount of seconds and will be aborted beyond this time).*

- bool [showColumnStop](#)

*Show/hide the column 'Stop' that enables stopping the execution of subsequent programs when the current one exited with an error code different from 0.*

- bool [showColumnLog](#)

*Show/hide the column 'Log' that enables the generation of log messages (these messages may be displayed using the [QELog](#) widget).*

- scriptTypesProperty [scriptType](#)

*Select if the scripts are to be loaded/saved from an XML file or from an XML text.*

- QString [executeText](#)

*Define the caption of the button responsible for starting the execution of external programs (if not defined then the caption will be "Execute").*

- optionsLayoutProperty [optionsLayout](#)

*Change the order of the widgets. Valid orders are: TOP, BOTTOM, LEFT and RIGHT.*

- bool [variableAsToolTip](#)

- bool [allowDrop](#)

- bool [visible](#)

- unsigned [int](#)

- QString [styleSheet](#)

- QString [defaultStyle](#)

- QString [userLevelUserStyle](#)

- QString [userLevelScientistStyle](#)

- QString [userLevelEngineerStyle](#)

- UserLevels [userLevelVisibility](#)

- UserLevels [userLevelEnabled](#)

- bool [displayAlarmState](#)

- DisplayAlarmStateOptions [displayAlarmStateOption](#)

## 9.166.1 Detailed Description

This class is a EPICS aware widget. The [QEScript](#) widget allows the user to define a certain sequence of external programs to be executed. This sequence may be saved, modified or loaded for future usage.

## 9.166.2 Member Enumeration Documentation

### 9.166.2.1 enum QEScript::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and displayAlarmStateOptions enumeration for details.

#### Enumerator:

*Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

### 9.166.2.2 enum QEScript::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and userLevel enumeration for details.

#### Enumerator:

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

## 9.166.3 Member Function Documentation

### 9.166.3.1 void QEScript::setManagedVisible (bool v) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.166.4 Property Documentation

### 9.166.4.1 bool QEScript::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

### 9.166.4.2 QString QEScript::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.166.4.3 bool QEScript::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.166.4.4 DisplayAlarmStateOptions QEScript::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.166.4.5 unsigned QEScript::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.166.4.6 QString QEScript::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.166.4.7 UserLevels QEScript::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmaticaly through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.166.4.8 QString QEScript::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example,

'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.166.4.9 QString QEScript::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.166.4.10 QString QEScript::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.166.4.11 UserLevels QEScript::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.166.4.12 bool QEScript::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### **9.166.4.13 bool QEScript::visible [read, write]**

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScript/QEScript.h

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEScript/QEScript.cpp

## 9.167 QEShape Class Reference

```
#include <QEShape.h>
```

### Public Types

- enum `shapeOptions` {
 **Line, Points, Polyline, Polygon,**  
**Rect, RoundedRect, Ellipse, Arc,**  
**Chord, Pie, Path** }
- enum `animationOptions` {
 **Width, Height, X, Y,**  
**Transparency, Rotation, ColourHue, ColourSaturation,**  
**ColourValue, ColourIndex, Penwidth** }
- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER,  
`Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` =  
userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_-  
ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_-  
ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_-  
ALARM\_STATE\_WHEN\_IN\_ALARM } }

### Public Slots

- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged1` (const qulonglong &out)
- void `dbValueChanged2` (const qulonglong &out)
- void `dbValueChanged3` (const qulonglong &out)
- void `dbValueChanged4` (const qulonglong &out)
- void `dbValueChanged5` (const qulonglong &out)
- void `dbValueChanged6` (const qulonglong &out)

### Public Member Functions

- `QEShape` (QWidget \*parent=0)
- `QEShape` (const QString &variableName, QWidget \*parent=0)
- void `scaleBy` (const int m, const int d)  
*Scale the widgets my m/d.*
- void `setAnimation` (`animationOptions` animation, const int index)

*Access function for animation' properties - refer to animation' properties for details.*

- **animationOptions getAnimation (const int index)**

*Access function for animation' properties - refer to animation' properties for details.*

- **void setScale (const double scale, const int index)**

*Access function for scale' properties - refer to scale' properties for details.*

- **double getScale (const int index)**

*Access function for scale' properties - refer to scale' properties for details.*

- **void setOffset (const double offset, const int index)**

*Access function for offset' properties - refer to offset' properties for details.*

- **double getOffset (const int index)**

*Access function for offset' properties - refer to offset' properties for details.*

- **void setBorder (const bool border)**

*Access function for border' properties - refer to border' properties for details.*

- **bool getBorder ()**

*Access function for border' properties - refer to border' properties for details.*

- **void setFill (const bool fill)**

*Access function for fill' properties - refer to fill' properties for details.*

- **bool getFill ()**

*Access function for fill' properties - refer to fill' properties for details.*

- **void setShape (shapeOptions shape)**

*Access function for shape' properties - refer to shape' properties for details.*

- **shapeOptions getShape ()**

*Access function for shape' properties - refer to shape' properties for details.*

- **void setNumPoints (const unsigned int numPoints)**

*Access function for number of points' properties - refer to number of points' properties for details.*

- **unsigned int getNumPoints ()**

*Access function for number of points' properties - refer to number of points' properties for details.*

- **void setOriginTranslation (const QPoint originTranslation)**

*Access function for origin translation' properties - refer to origin translation' properties for details.*

- **QPoint getOriginTranslation ()**  
*Access function for origin translation' properties - refer to origin translation' properties for details.*
- **void setPoint (const QPoint point, const int index)**  
*Access function for point' properties - refer to point' properties for details.*
- **QPoint getPoint (const int index)**  
*Access function for point' properties - refer to point' properties for details.*
- **void setColor (const QColor color, const int index)**  
*Access function for colour' properties - refer to colour' properties for details.*
- **QColor getColor (const int index)**  
*Access function for colour' properties - refer to colour' properties for details.*
- **void setDrawBorder (const bool drawBorder)**  
*Access function for draw border' properties - refer to draw border' properties for details.*
- **bool getDrawBorder ()**  
*Access function for draw border' properties - refer to draw border' properties for details.*
- **void setLineWidth (const unsigned int lineWidth)**  
*Access function for line width' properties - refer to line width' properties for details.*
- **unsigned int getLineWidth ()**  
*Access function for line width' properties - refer to line width' properties for details.*
- **void setStartAngle (const double startAngle)**  
*Access function for start angle' properties - refer to start angle' properties for details.*
- **double getStartAngle ()**  
*Access function for start angle' properties - refer to start angle' properties for details.*
- **void setRotation (const double rotation)**  
*Access function for rotation' properties - refer to rotation' properties for details.*
- **double getRotation ()**  
*Access function for rotation' properties - refer to rotation' properties for details.*
- **void setArcLength (const double arcLength)**  
*Access function for arc length' properties - refer to arc length' properties for details.*
- **double getArcLength ()**

*Access function for arc length' properties - refer to arc length' properties for details.*

- void [setVariableNameSubstitutionsProperty](#) (QString variableNameSubstitutions)  
*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*
- QString [getVariableNameSubstitutionsProperty](#) ()  
*Property access function for `variableSubstitutions` property. This has special behaviour to work well within designer.*
- UserLevels [getUserLevelVisibilityProperty](#) ()  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void [setUserLevelVisibilityProperty](#) (UserLevels level)  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- UserLevels [getUserLevelEnabledProperty](#) ()  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void [setUserLevelEnabledProperty](#) (UserLevels level)  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- DisplayAlarmStateOptions [getDisplayAlarmStateOptionProperty](#) ()  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void [setDisplayAlarmStateOptionProperty](#) (DisplayAlarmStateOptions option)  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

## Properties

- QString `variable1`
- QString `variable2`
- QString `variable3`
- QString `variable4`
- QString `variable5`
- QString `variable6`
- QString `variableSubstitutions`
- bool `variableAsToolTip`
- bool `allowDrop`

- bool `visible`
- unsigned `int`
- QString `styleSheet`
- QString `defaultStyle`
- QString `userLevelUserStyle`
- QString `userLevelScientistStyle`
- QString `userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- bool `displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- `animationOptions animation1`
- `animationOptions animation2`
- `animationOptions animation3`
- `animationOptions animation4`
- `animationOptions animation5`
- `animationOptions animation6`
- double `scale1`

*Scale factor applied to data from the 1st variable before it is used to animate the shape.*

- double `scale2`
- double `scale3`
- double `scale4`
- double `scale5`
- double `scale6`
- double `offset1`
- double `offset2`
- double `offset3`
- double `offset4`
- double `offset5`
- double `offset6`
- QPoint `point1`
- QPoint `point2`
- QPoint `point3`
- QPoint `point4`
- QPoint `point5`
- QPoint `point6`
- QPoint `point7`
- QPoint `point8`
- QPoint `point9`
- QPoint `point10`
- QColor `color1`
- QColor `color2`
- QColor `color3`
- QColor `color4`

- QColor [color5](#)
- QColor [color6](#)
- QColor [color7](#)
- QColor [color8](#)
- QColor [color9](#)
- QColor [color10](#)

### 9.167.1 Detailed Description

This class is a EPICS aware shape widget based on the Qt widget. One of several shapes can be drawn within the widget, and up to 6 variables can be used to animate various attributes of the shape. For example to represent beam positino and size, an ellipse can be drawn with four variables animating its vertical and horizontal size and position. It is tighly integrated with the base class QEWidget which provides generic support such as macro substitutions, drag/drop, and standard properties.

### 9.167.2 Member Enumeration Documentation

#### 9.167.2.1 enum QEShape::animationOptions

Options for how a variable will animate the shape.

#### 9.167.2.2 enum QEShape::DisplayAlarmStateOptions

User friendly enumerations for [displayAlarmStateOption](#) property - refer to [displayAlarmStateOption](#) property and [displayAlarmStateOptions](#) enumeration for details.

##### Enumerator:

**Never** Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

**Always** Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

**WhenInAlarm** Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

#### 9.167.2.3 enum QEShape::shapeOptions

Options for the type of shape.

#### 9.167.2.4 enum QEShape::UserLevels

User friendly enumerations for [userLevelVisibility](#) and [userLevelEnabled](#) properties - refer to [userLevelVisibility](#) and [userLevelEnabled](#) properties and [userLevel](#) enumeration for details.

**Enumerator:**

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

### 9.167.3 Constructor & Destructor Documentation

#### 9.167.3.1 QEShape::QEShape (QWidget \* *parent* = 0)

Create without a variable. Use setVariableNameProperty() and setSubstitutionsProperty() to define a variable and, optionally, macro substitutions later.

#### 9.167.3.2 QEShape::QEShape (const QString & *variableName*, QWidget \* *parent* = 0)

Create with a single variable. (Note, the [QEShape](#) widget can use up to 6 variables) A connection is automatically established. If macro substitutions are required, create without a variable and set the variable and macro substitutions after creation.

### 9.167.4 Member Function Documentation

#### 9.167.4.1 void QEShape::dbValueChanged1 (const qulonglong & *out*) [signal]

Sent when the widget is updated following a data change for the first variable Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

#### 9.167.4.2 void QEShape::dbValueChanged2 (const qulonglong & *out*) [signal]

Sent when the widget is updated following a data change for the second variable Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

#### 9.167.4.3 void QEShape::dbValueChanged3 (const qulonglong & *out*) [signal]

Sent when the widget is updated following a data change for the third variable Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.167.4.4 void QEShape::dbValueChanged4 (const qlonglong & *out*)  
[signal]**

Sent when the widget is updated following a data change for the fourth variable Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.167.4.5 void QEShape::dbValueChanged5 (const qlonglong & *out*)  
[signal]**

Sent when the widget is updated following a data change for the fifth variable Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.167.4.6 void QEShape::dbValueChanged6 (const qlonglong & *out*)  
[signal]**

Sent when the widget is updated following a data change for the sixth variable Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.167.4.7 void QEShape::setManagedVisible (bool *v*) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

## 9.167.5 Property Documentation

**9.167.5.1 bool QEShape::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.167.5.2 animationOptions QEShape::animation1 [read, write]**

Animation to be effected by the 1st variable. This is used to select what the effect changing data for the 1st variable will have on the shape.

**9.167.5.3 animationOptions QEShape::animation2 [read, write]**

Animation to be effected by the 2nd variable. This is used to select what the effect changing data for the 2nd variable will have on the shape.

**9.167.5.4 animationOptions QEShape::animation3 [read, write]**

Animation to be effected by the 3rd variable. This is used to select what the effect changing data for the 3rd variable will have on the shape.

**9.167.5.5 animationOptions QEShape::animation4 [read, write]**

Animation to be effected by the 4th variable. This is used to select what the effect changing data for the 4th variable will have on the shape.

**9.167.5.6 animationOptions QEShape::animation5 [read, write]**

Animation to be effected by the 5th variable. This is used to select what the effect changing data for the 5th variable will have on the shape.

**9.167.5.7 animationOptions QEShape::animation6 [read, write]**

Animation to be effected by the 6th variable. This is used to select what the effect changing data for the 6th variable will have on the shape.

**9.167.5.8 QColor QEShape::color1 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.9 QColor QEShape::color10 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.10 QColor QEShape::color2 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.11 QColor QEShape::color3 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.12 QColor QEShape::color4 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.13 QColor QEShape::color5 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.14 QColor QEShape::color6 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.15 QColor QEShape::color7 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.16 QColor QEShape::color8 [read, write]**

Used by the color animation to determine the color based on a data value. The scaled and offset data is used as an index to select color properties 'color1' to 'color10'.

**9.167.5.17 QString QEShape::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.167.5.18 bool QEShape::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.167.5.20 DisplayAlarmStateOptions QEShape::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm

state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

#### **9.167.5.21 unsigned QEShape::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

The number of points to use when drawing shapes that are defined by a variable number of points, such as polyline, polygon, path, and series of points.

Sets the width of the pen. Used for the following shapes: Line, Points, Polyline, Polygon, Rect, RoundedRect, Ellipse, Arc, Chord, Pie, Path

#### **9.167.5.22 double QEShape::offset1 [read, write]**

Offset applied to data from the 1st variable before it is used to animate the shape

#### **9.167.5.23 double QEShape::offset2 [read, write]**

Offset applied to data from the 2nd variable before it is used to animate the shape

#### **9.167.5.24 double QEShape::offset3 [read, write]**

Offset applied to data from the 3rd variable before it is used to animate the shape

#### **9.167.5.25 double QEShape::offset4 [read, write]**

Offset applied to data from the 4th variable before it is used to animate the shape

#### **9.167.5.26 double QEShape::offset5 [read, write]**

Offset applied to data from the 5th variable before it is used to animate the shape

#### **9.167.5.27 double QEShape::offset6 [read, write]**

Offset applied to data from the 6th variable before it is used to animate the shape

**9.167.5.28 QPoint QEShape::point1 [read, write]**

1st coordinate used when drawing the shape. Used for the following shapes: Line, Points, Polyline, Polygon, Rect, RoundedRectangle, Ellipse, Arc, Chord, Pie, Path, Text,Pixmap

**9.167.5.29 QPoint QEShape::point10 [read, write]**

10th coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.30 QPoint QEShape::point2 [read, write]**

2nd coordinate used when drawing the shape. Used for the following shapes: Line, Points, Polyline, Polygon, Rect, RoundedRectangle, Ellipse, Arc, Chord, Pie, Path,Pixmap

**9.167.5.31 QPoint QEShape::point3 [read, write]**

3rd coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.32 QPoint QEShape::point4 [read, write]**

4th coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.33 QPoint QEShape::point5 [read, write]**

5th coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.34 QPoint QEShape::point6 [read, write]**

6th coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.35 QPoint QEShape::point7 [read, write]**

7th coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.36 QPoint QEShape::point8 [read, write]**

8th coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.37 QPoint QEShape::point9 [read, write]**

9th coordinate used when drawing the shape. Used for the following shapes: Points, Polyline, Polygon, Path

**9.167.5.38 double QEShape::scale2 [read, write]**

Scale factor applied to data from the 2nd variable before it is used to animate the shape

**9.167.5.39 double QEShape::scale3 [read, write]**

Scale factor applied to data from the 3rd variable before it is used to animate the shape

**9.167.5.40 double QEShape::scale4 [read, write]**

Scale factor applied to data from the 4th variable before it is used to animate the shape

**9.167.5.41 double QEShape::scale5 [read, write]**

Scale factor applied to data from the 5th variable before it is used to animate the shape

**9.167.5.42 double QEShape::scale6 [read, write]**

Scale factor applied to data from the 6th variable before it is used to animate the shape

**9.167.5.43 QString QEShape::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.167.5.44 UserLevels QEShape::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode.

The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

#### 9.167.5.45 `QString QEShape::userLevelEngineerStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the `styleManager` class. Refer to the `styleManager` class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.167.5.46 `QString QEShape::userLevelScientistStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the `styleManager` class. Refer to the `styleManager` class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.167.5.47 `QString QEShape::userLevelUserStyle [read, write]`

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the `styleManager` class. Refer to the `styleManager` class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### 9.167.5.48 `UserLevels QEShape::userLevelVisibility [read, write]`

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through `setUserLevel()`. Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.167.5.49 QString QEShape::variable1 [read, write]**

EPICS variable name (CA PV). This variable is read and used to animate an attribute of the shape. The value read is first scaled and offset by properties scale1 and offset1 then the attribute selected for animation is selected by the property animation1.

**9.167.5.50 QString QEShape::variable2 [read, write]**

EPICS variable name (CA PV). This variable is read and used to animate an attribute of the shape. The value read is first scaled and offset by properties scale2 and offset2 then the attribute selected for animation is selected by the property animation2.

**9.167.5.51 QString QEShape::variable3 [read, write]**

EPICS variable name (CA PV). This variable is read and used to animate an attribute of the shape. The value read is first scaled and offset by properties scale3 and offset3 then the attribute selected for animation is selected by the property animation3.

**9.167.5.52 QString QEShape::variable4 [read, write]**

EPICS variable name (CA PV). This variable is read and used to animate an attribute of the shape. The value read is first scaled and offset by properties scale4 and offset4 then the attribute selected for animation is selected by the property animation4.

**9.167.5.53 QString QEShape::variable5 [read, write]**

EPICS variable name (CA PV). This variable is read and used to animate an attribute of the shape. The value read is first scaled and offset by properties scale5 and offset5 then the attribute selected for animation is selected by the property animation5.

**9.167.5.54 QString QEShape::variable6 [read, write]**

EPICS variable name (CA PV). This variable is read and used to animate an attribute of the shape. The value read is first scaled and offset by properties scale6 and offset6 then the attribute selected for animation is selected by the property animation6.

**9.167.5.55 bool QEShape::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.167.5.56 QString QEShape::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For ex-

ample, 'SAMPLE=SAM1, NAME = "Ref foil"' These substitutions are applied to all the variable names.

#### 9.167.5.57 bool QEShape::visible [read, write]

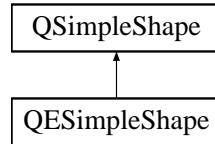
Display the widget. Default is true. Setting this property false is usefull if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEShape/QEShape.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEShape/QEShape.cpp

## 9.168 QE.SimpleShape Class Reference

#include <QE.SimpleShape.h> Inheritance diagram for QE.SimpleShape::



### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const QString &out)
- void `dbValueChanged` (const int &out)
- void `dbValueChanged` (const long &out)
- void `dbValueChanged` (const qlonglong &out)
- void `dbValueChanged` (const double &out)
- void `dbValueChanged` (const bool &out)
- void `dbConnectionChanged` (const bool &isConnected)

### Public Member Functions

- `UserLevels getUserLevelVisibilityProperty ()`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void `setUserLevelVisibilityProperty (UserLevels level)`  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- `UserLevels getUserLevelEnabledProperty ()`

*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*

- void [setUserLevelEnabledProperty \(UserLevels level\)](#)

*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*

- DisplayAlarmStateOptions [getDisplayAlarmStateOptionProperty \(\)](#)

*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*

- void [setDisplayAlarmStateOptionProperty \(DisplayAlarmStateOptions option\)](#)

*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*

- [QESimpleShape \(QWidget \\*parent=0\)](#)
- [QESimpleShape \(const QString &variableName, QWidget \\*parent=0\)](#)
- void [setVariableNameSubstitutionsProperty \(const QString &substitutions\)](#)
- void [setEdgeVariableNameProperty \(const QString &variableName\)](#)
- QString [getEdgeVariableNameProperty \(\) const](#)
- void [setEdgeArrayIndex \(const int arrayIndex\)](#)
- int [getEdgeArrayIndex \(\) const](#)
- DisplayAlarmStateOptions [getEdgeAlarmStateOptionProperty \(\)](#)

*Access function for edgeAlarmStateOption property - refer to edgeAlarmStateOption property for details.*

- void [setEdgeAlarmStateOptionProperty \(DisplayAlarmStateOptions option\)](#)

*Access function for edgeAlarmStateOption property - refer to edgeAlarmStateOption property for details.*

## Protected Member Functions

- void [activated \(\)](#)
  - qcaobject::QCaObject \* [createQcaItem \(unsigned int variableIndex\)](#)
  - void [establishConnection \(unsigned int variableIndex\)](#)
  - void [dragEnterEvent \(QDragEnterEvent \\*event\)](#)
  - void [dropEvent \(QDropEvent \\*event\)](#)
  - void [mousePressEvent \(QMouseEvent \\*event\)](#)
  - QString [copyVariable \(\)](#)
  - QVariant [copyData \(\)](#)
  - void [paste \(QVariant v\)](#)
  - QString [getItemText \(\)](#)
  - QColor [getItemColour \(\)](#)
- returns "" unless overridden.*
- void [stringFormattingChange \(\)](#)

## Properties

- `QString variable`
- `QString variableSubstitutions`
- `int arrayIndex`
- `QString edgeVariable`
- `int edgeArrayIndex`
- `DisplayAlarmStateOptions edgeAlarmStateOption`
- `bool addUnits`
- `QString localEnumeration`
- `int value`  
`Shape value - range 0 to 15 - default is zero.`
- `bool isActive`  
`When inactive, the widgers is grayed-out, as if disabled. The default is true.`
- `bool variableAsToolTip`
- `bool allowDrop`
- `bool visible`
- `unsigned int`
- `QString styleSheet`
- `QString defaultStyle`
- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`

### 9.168.1 Detailed Description

This class is an EPICS aware simple shape. The shape is selected at design time using the shape property - refer to Shapes for details, or may be set by the setShape function. The colour is based on either the value of or the alarm state of a single PV. When the (standard) displayAlarmState property is false, the PV value (modulo 16) is used to select one of the colours defined by the colour0 to colour15 properties. When the displayAlarmState property is true, the colour is based on the alarm state, i.e. green when no alarm, yellow for minor alarm etc. This class extends `QSimpleShape` in order to make it EPICS aware.

### 9.168.2 Member Enumeration Documentation

#### 9.168.2.1 enum QESimpleShape::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

**Enumerator:**

*Never* Refer to DISPLAY\_ALARM\_STATE\_NEVER for details.

*Always* Refer to DISPLAY\_ALARM\_STATE\_ALWAYS for details.

*WhenInAlarm* Refer to DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM for details.

**9.168.2.2 enum QEShape::UserLevels**

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and userLevel enumeration for details.

**Enumerator:**

*User* Refer to USERLEVEL\_USER for details.

*Scientist* Refer to USERLEVEL\_SCIENTIST for details.

*Engineer* Refer to USERLEVEL\_ENGINEER for details.

**9.168.3 Member Function Documentation****9.168.3.1 void QEShape::dbConnectionChanged (const bool & isConnected) [signal]**

Sent when the widget state updated following a channel connection change Applied to provary varible.

**9.168.3.2 void QEShape::dbValueChanged (const QString & out) [signal]**

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.168.3.3 void QEShape::setManagedVisible (bool v) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a calll to this slot if the user level allows.

**9.168.4 Property Documentation****9.168.4.1 bool QEShape::addUnits [read, write]**

If true (default), add engineering units supplied with the data.

**9.168.4.2 bool QESimpleShape::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.168.4.3 int QESimpleShape::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.168.4.4 QString QESimpleShape::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.168.4.5 bool QESimpleShape::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.168.4.6 DisplayAlarmStateOptions QESimpleShape::displayAlarmStateOption [read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.168.4.7 unsigned QESimpleShape::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.168.4.8 QString QESimpleShape::localEnumeration [read, write]**

An enumeration list used to data values. Used only when the formatting option is 'local enumeration'. Value is converted to an integer and used to select a string from this list.

Format is:

```
[[<|<=|=|=|>|=]value1|*] : string1 , [[<|<=|=|=|>|=]value2|*] : string2 ,
[[<|<=|=|=|>|=]value3|*] : string3 , ...
```

Where: < Less than <= Less than or equal = Equal (default if no operator specified)  
>= Greater than or equal > Greater than Always match (used to specify default text)

Values may be numeric or textual Values do not have to be in any order, but first match wins Values may be quoted Strings may be quoted Consecutive values do not have to be present. Operator is assumed to be equality if not present. White space is ignored except within quoted strings.

may be included in a string to indicate a line break

Examples are:

```
0:Off,1:On 0 : "Pump Running", 1 : "Pump not running" 0:"", 1:"Warning!\nAlarm"
<2:"Value is less than two", =2:"Value is equal to two", >2:"Value is grater than
2" 3:"Beamline Available", *:"" "Pump Off":"OH NO!, the pump is OFF!","Pump
On":"It's OK, the pump is on"
```

The data value is converted to a string if no enumeration for that value is available. For example, if the local enumeration is '0:off,1:on', and a value of 10 is processed, the text generated is '10'. If a blank string is required, this should be explicit. for example, '0:off,1:on,10:""'

A range of numbers can be covered by a pair of values as in the following example:  
>=4:"Between 4 and 8",<=8:"Between 4 and 8"

#### **9.168.4.9 QString QESimpleShape::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

#### **9.168.4.10 UserLevels QESimpleShape::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

#### **9.168.4.11 QString QESimpleShape::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string

will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.168.4.12 QString QEShape::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.168.4.13 QString QEShape::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

#### **9.168.4.14 UserLevels QEShape::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel() Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

#### **9.168.4.15 QString QEShape::variable [read, write]**

EPICS variable name (CA PV)

#### **9.168.4.16 bool QEShape::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

#### **9.168.4.17 QString QEShape::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For exam-

ple, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

#### 9.168.4.18 bool QESimpleShape::visible [read, write]

Display the widget. Default is true. Setting this property false is useful if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESimpleShape/QESimpleShape.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESimpleShape/QESimpleShape.cpp

## 9.169 QEShapeManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEShapeManager/QEShapeManager.h

## 9.170 QESlider Class Reference

### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setDefaultStyle` (const QString &style)  
*Update the default style applied to this widget.*
- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const qlonglong &out)

### Public Member Functions

- `QESlider` (QWidget \*parent=0)
- `QESlider` (const QString &variableName, QWidget \*parent=0)
- void `setWriteOnChange` (bool `writeOnChange`)
- bool `getWriteOnChange` () const
- void `setSubscribe` (bool subscribe)
- bool `getSubscribe` () const
- void `setScale` (double scaleIn)
- double `getScale` () const
- void `setOffset` (double offsetIn)
- double `getOffset` () const
- void `setAllowFocusUpdate` (bool allowFocusUpdate)
- bool `getAllowFocusUpdate` () const
- void `writeNow` ()
- `UserLevels getUserLevelVisibilityProperty` ()  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*
- void  `setUserLevelVisibilityProperty` (`UserLevels` level)  
*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- **UserLevels getUserLevelEnabledProperty ()**  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- void **setUserLevelEnabledProperty (UserLevels level)**  
*Access function for `userLevelEnabled` property - refer to `userLevelEnabled` property for details.*
- **DisplayAlarmStateOptions getDisplayAlarmStateOptionProperty ()**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*
- void **setDisplayAlarmStateOptionProperty (DisplayAlarmStateOptions option)**  
*Access function for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property for details.*

## Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)

## Protected Attributes

- **QEFloatingFormatting floatingFormatting**
- bool **writeOnChange**

## Properties

- QString **variable**
- QString **variableSubstitutions**
- int **arrayIndex**
- bool **subscribe**
- bool **allowFocusUpdate**
- bool **variableAsToolTip**
- bool **allowDrop**
- bool **visible**
- unsigned **int**
- QString **styleSheet**
- QString **defaultStyle**

- `QString userLevelUserStyle`
- `QString userLevelScientistStyle`
- `QString userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- `bool displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- `double value`
- `int sliderPosition`

## 9.170.1 Member Enumeration Documentation

### 9.170.1.1 enum QESlider::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

#### Enumerator:

*Never* Refer to `DISPLAY_ALARM_STATE_NEVER` for details.

*Always* Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.

*WhenInAlarm* Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

### 9.170.1.2 enum QESlider::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

#### Enumerator:

*User* Refer to `USERLEVEL_USER` for details.

*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.

*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

## 9.170.2 Member Function Documentation

### 9.170.2.1 void QESlider::dbValueChanged (const qulonglong & *out*) [signal]

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

**9.170.2.2 void QESlider::setManagedVisible (bool v) [inline, slot]**

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a call to this slot if the user level allows.

**9.170.3 Member Data Documentation****9.170.3.1 bool QESlider::writeOnChange [read, write, protected]**

Sets if this widget writes any changes as the user moves the slider (the QSlider 'valueChanged' signal is emitted). Default is 'true' (writes any changes when the QSlider 'valueChanged' signal is emitted).

**9.170.4 Property Documentation****9.170.4.1 bool QESlider::allowDrop [read, write]**

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

**9.170.4.2 bool QESlider::allowFocusUpdate [read, write]**

Allow updated while widget has focus - defaults to false

**9.170.4.3 int QESlider::arrayIndex [read, write]**

Index used to select a single item of data for processing. The default is 0.

**9.170.4.4 QString QESlider::defaultStyle [read, write]**

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

**9.170.4.5 bool QESlider::displayAlarmState [read, write]**

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.170.4.6 DisplayAlarmStateOptions QESlider::displayAlarmStateOption  
[read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.170.4.7 unsigned QESlider::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.170.4.8 QString QESlider::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.170.4.9 bool QESlider::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.170.4.10 UserLevels QESlider::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.170.4.11 QString QESlider::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.170.4.12 QString QESlider::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.170.4.13 QString QESlider::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.170.4.14 UserLevels QESlider::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.170.4.15 QString QESlider::variable [read, write]**

EPICS variable name (CA PV)

**9.170.4.16 bool QESlider::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.170.4.17 QString QESlider::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

**9.170.4.18 bool QESlider::visible [read, write]**

Display the widget. Default is true. Setting this property false is usefull if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESlider/QESlider.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESlider/QESlider.cpp

## 9.171 QESpinBox Class Reference

### Public Types

- enum `UserLevels` { `User` = userLevelTypes::USERLEVEL\_USER, `Scientist` = userLevelTypes::USERLEVEL\_SCIENTIST, `Engineer` = userLevelTypes::USERLEVEL\_ENGINEER }
- enum `DisplayAlarmStateOptions` { `Never` = standardProperties::DISPLAY\_ALARM\_STATE\_NEVER, `Always` = standardProperties::DISPLAY\_ALARM\_STATE\_ALWAYS, `WhenInAlarm` = standardProperties::DISPLAY\_ALARM\_STATE\_WHEN\_IN\_ALARM }

### Public Slots

- void `setDefaultStyle` (const QString &style)  
*Update the default style applied to this widget.*
- void `setManagedVisible` (bool v)

### Signals

- void `dbValueChanged` (const double &out)
- void `userChange` (const QString &oldValue, const QString &newValue, const QString &lastValue)  
*Internal use only. Used by `QEConfiguredLayout` to be notified when one of its widgets has written something.*

### Public Member Functions

- `QESpinBox` (QWidget \*parent=0)
- `QESpinBox` (const QString &variableName, QWidget \*parent=0)
- void `setWriteOnChange` (bool writeOnChangeIn)
- bool `getWriteOnChange` () const
- void `setSubscribe` (bool subscribe)
- bool `getSubscribe` () const
- void `setAddUnitsAsSuffix` (bool addUnitsAsSuffixIn)
- bool `getAddUnitsAsSuffix` () const
- void `setUseDbPrecisionForDecimals` (bool useDbPrecisionForDecimalIn)
- bool `getUseDbPrecisionForDecimals` () const
- void `setAllowFocusUpdate` (bool allowFocusUpdate)
- bool `getAllowFocusUpdate` () const
- void `writeNow` ()
- `UserLevels getUserLevelVisibilityProperty` ()

*Access function for `userLevelVisibility` property - refer to `userLevelVisibility` property for details.*

- void [setUserLevelVisibilityProperty \(UserLevels level\)](#)  
*Access function for userLevelVisibility property - refer to userLevelVisibility property for details.*
- UserLevels [getUserLevelEnabledProperty \(\)](#)  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- void [setUserLevelEnabledProperty \(UserLevels level\)](#)  
*Access function for userLevelEnabled property - refer to userLevelEnabled property for details.*
- DisplayAlarmStateOptions [getDisplayAlarmStateOptionProperty \(\)](#)  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*
- void [setDisplayAlarmStateOptionProperty \(DisplayAlarmStateOptions option\)](#)  
*Access function for displayAlarmStateOption property - refer to displayAlarmStateOption property for details.*

## Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)
- QMenu \* **getDefaultContextMenu** ()

## Protected Attributes

- [QEFloatingFormatting floatingFormatting](#)
- bool **writeOnChange**
- bool **addUnitsAsSuffix**
- bool **useDbPrecisionForDecimal**

## Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- int [arrayIndex](#)

- bool `variableAsToolTip`
- bool `allowDrop`
- bool `visible`
- unsigned `int`
- QString `styleSheet`
- QString `defaultStyle`
- QString `userLevelUserStyle`
- QString `userLevelScientistStyle`
- QString `userLevelEngineerStyle`
- `UserLevels userLevelVisibility`
- `UserLevels userLevelEnabled`
- bool `displayAlarmState`
- `DisplayAlarmStateOptions displayAlarmStateOption`
- bool `subscribe`
- bool `allowFocusUpdate`
- bool `useDbPrecision`
- bool `addUnits`
- double `value`

## 9.171.1 Member Enumeration Documentation

### 9.171.1.1 enum QESpinBox::DisplayAlarmStateOptions

User friendly enumerations for `displayAlarmStateOption` property - refer to `displayAlarmStateOption` property and `displayAlarmStateOptions` enumeration for details.

#### Enumerator:

*Never* Refer to `DISPLAY_ALARM_STATE_NEVER` for details.

*Always* Refer to `DISPLAY_ALARM_STATE_ALWAYS` for details.

*WhenInAlarm* Refer to `DISPLAY_ALARM_STATE_WHEN_IN_ALARM` for details.

### 9.171.1.2 enum QESpinBox::UserLevels

User friendly enumerations for `userLevelVisibility` and `userLevelEnabled` properties - refer to `userLevelVisibility` and `userLevelEnabled` properties and `userLevel` enumeration for details.

#### Enumerator:

*User* Refer to `USERLEVEL_USER` for details.

*Scientist* Refer to `USERLEVEL_SCIENTIST` for details.

*Engineer* Refer to `USERLEVEL_ENGINEER` for details.

## 9.171.2 Member Function Documentation

### 9.171.2.1 void QESpinBox::dbValueChanged (const double & *out*) [signal]

Sent when the widget is updated following a data change. Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget.

### 9.171.2.2 void QESpinBox::setManagedVisible (bool *v*) [inline, slot]

Slot to set the visibility of a QE widget, taking into account the user level. Widget will be hidden if hidden by a call this slot, by will only be made visible by a calll to this slot if the user level allows.

## 9.171.3 Property Documentation

### 9.171.3.1 bool QESpinBox::allowDrop [read, write]

Allow drag/drops operations to this widget. Default is false. Any dropped text will be used as a new variable name.

### 9.171.3.2 bool QESpinBox::allowFocusUpdate [read, write]

Allow updated while widget has focus - defaults to false

### 9.171.3.3 int QESpinBox::arrayIndex [read, write]

Index used to select a single item of data for processing. The default is 0.

### 9.171.3.4 QString QESpinBox::defaultStyle [read, write]

Style Sheet string to be applied before, i.e. lower priority than, any other style, e.g. alarm style and/or user level style. Default is an empty string.

### 9.171.3.5 bool QESpinBox::displayAlarmState [read, write]

DEPRECATED. USE displayAlarmStateOption INSTEAD. If set (default) widget will indicate the alarm state of any variable data it is displaying. If clear widget will never indicate the alarm state of any variable data it is displaying. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.171.3.6 DisplayAlarmStateOptions QESpinBox::displayAlarmStateOption  
[read, write]**

If 'Always' (default) widget will indicate the alarm state of any variable data it is displaying, including 'No Alarm'. If 'Never' widget will never indicate the alarm state of any variable data it is displaying. If 'WhenInAlarm' widget only indicate the alarm state of any variable data it is displaying if it is 'in alarm'. Typically the background colour is set to indicate the alarm state. Note, this property is included in the set of standard properties as it applies to most widgets. It will do nothing for widgets that don't display data.

**9.171.3.7 unsigned QESpinBox::int [read, write]**

Set the ID used by the message filtering system. Default is zero. Widgets or applications that use messages from the framework have the option of filtering on this ID. For example, by using a unique message source ID a [QELog](#) widget may be set up to only log messages from a select set of widgets.

**9.171.3.8 QString QESpinBox::styleSheet [read, write]**

Hide style sheet from designer as style calculation by the styleManager and not directly setable per se. This also stops transient styles being saved to the ui file.

**9.171.3.9 bool QESpinBox::subscribe [read, write]**

Sets if this widget subscribes for data updates and displays current data. Default is 'true' (subscribes for and displays data updates)

**9.171.3.10 UserLevels QESpinBox::userLevelEnabled [read, write]**

Lowest user level at which the widget is enabled. Default is 'User'. Used when designing GUIs that allow access to more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always accessible should be visible at 'User'. Widgets that are only accessible to scientists managing the facility should be visible at 'Scientist'. Widgets that are only accessible to engineers maintaining the facility should be visible at 'Engineer'.

**9.171.3.11 QString QESpinBox::userLevelEngineerStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.171.3.12 QString QESpinBox::userLevelScientistStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.171.3.13 QString QESpinBox::userLevelUserStyle [read, write]**

Style Sheet string to be applied when the widget is displayed in 'User' mode. Default is an empty string. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red' This Style Sheet string will be applied by the styleManager class. Refer to the styleManager class for details about how this Style Sheet string will be merged with any pre-existing Style Sheet string and any Style Sheet strings generated during the display of data.

**9.171.3.14 UserLevels QESpinBox::userLevelVisibility [read, write]**

Lowest user level at which the widget is visible. Default is 'User'. Used when designing GUIs that display more and more detail according to the user mode. The user mode is set application wide through the [QELogin](#) widget, or programmatically through setUserLevel(). Widgets that are always visible should be visible at 'User'. Widgets that are only used by scientists managing the facility should be visible at 'Scientist'. Widgets that are only used by engineers maintaining the facility should be visible at 'Engineer'.

**9.171.3.15 QString QESpinBox::variable [read, write]**

EPICS variable name (CA PV)

**9.171.3.16 bool QESpinBox::variableAsToolTip [read, write]**

Use the variable as the tool tip. Default is true. Tool tip property will be overwritten by the variable name.

**9.171.3.17 QString QESpinBox::variableSubstitutions [read, write]**

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some widgets are also used for other purposes.

**9.171.3.18 bool QESpinBox::visible [read, write]**

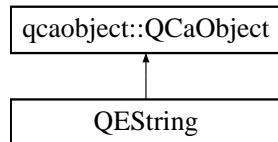
Display the widget. Default is true. Setting this property false is usefull if widget is only used to provide a signal - for example, when supplying data to a [QELink](#) widget. Note, when false the widget will still be visible in Qt Designer.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESpinBox/QESpinBox.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESpinBox/QESpinBox.cpp

## 9.172 QEString Class Reference

Inheritance diagram for QEString::



### Public Slots

- void **writeString** (const QString &data)
- void **writeStringElement** (const QString &data)
- void **writeString** (const QVector<QString> &data)

### Signals

- void **stringConnectionChanged** (QCAConnectionInfo &connectionInfo, const unsigned int &variableIndex)
- void **stringChanged** (const QString &value, QCAlarmInfo &alarmInfo, QCADateTime &timeStamp, const unsigned int &variableIndex)
- void **stringArrayChanged** (const QVector<QString> &values, QCAlarmInfo &alarmInfo, QCADateTime &timeStamp, const unsigned int &variableIndex)

### Public Member Functions

- **QEString** (QString recordName, QObject \*eventObject, QEStringFormatting \*stringFormattingIn, unsigned int variableIndexIn)
- **QEString** (QString recordName, QObject \*eventObject, QEStringFormatting \*stringFormattingIn, unsigned int variableIndexIn, UserMessage \*userMessageIn)
- bool **writeString** (const QString &data, QString &message)
- bool **writeStringElement** (const QString &data, QString &message)
- bool **writeString** (const QVector<QString> &data, QString &message)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEString.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEString.cpp

## 9.173 QEStringFormatting Class Reference

### Public Types

- enum **formats** {
   
    **FORMAT\_DEFAULT**,   **FORMAT\_FLOATING**,   **FORMAT\_INTEGER**,
   
    **FORMAT\_UNSIGNEDINTEGER**,
   
    **FORMAT\_TIME**, **FORMAT\_LOCAL\_ENUMERATE**, **FORMAT\_STRING** }
- enum **notations** { **NOTATION\_FIXED** = QTextStream::FixedNotation,
   
**NOTATION\_SCIENTIFIC** = QTextStream::ScientificNotation, **NOTATION\_AUTOMATIC** = QTextStream::SmartNotation }
- enum **separators** { **SEPARATOR\_NONE** = 0, **SEPARATOR\_COMMA**,
   
**SEPARATOR\_UNDERSCORE**, **SEPARATOR\_SPACE** }
- enum **arrayActions** { **APPEND**, **ASCII**, **INDEX** }

### Public Member Functions

- **QString formatString** (const **QVariant** &value, int arrayIndex=0) const
- **QVariant formatValue** (const **QString** &text, bool &ok)
- **QVariant formatValue** (const **QVector<QString>** &text, bool &ok)
- void **setDbEgu** (**QString** egu)
- void **setDbEnumerations** (**QStringList** enumerations)
- void **setDbPrecision** (unsigned int dbPrecisionIn)
- void **setPrecision** (int precision)
- void **setUseDbPrecision** (bool useDbPrecision)
- void **setLeadingZero** (bool leadingZero)
- void **setTrailingZeros** (bool trailingZeros)
- void **setFormat** (**formats** format)
- void **setSeparator** (const **separators** separator)
- void **setRadix** (const int radix)
- void **setNotation** (**notations** notation)
- void **setArrayAction** (**arrayActions** arrayActionIn)
- void **setAddUnits** (bool addUnits)
- void **setLocalEnumeration** (**QString** localEnumerationIn)
- int **getPrecision** () const
- bool **getUseDbPrecision** () const
- bool **getLeadingZero** () const
- bool **getTrailingZeros** () const
- **formats** **getFormat** () const
- **separators** **getSeparator** () const
- unsigned int **getRadix** () const
- **notations** **getNotation** () const
- **arrayActions** **getArrayAction** () const
- bool **getAddUnits** () const
- **QString** **getLocalEnumeration** () const
- **QELocalEnumeration** **getLocalEnumerationObject** () const

### 9.173.1 Member Enumeration Documentation

#### 9.173.1.1 enum QEStringFormatting::arrayActions

What action to take when formatting array data

**Enumerator:**

**APPEND** Interpret each element in the array as an unsigned integer and append string representations of each element from the array with a space in between each.

**ASCII** Interpret each element from the array as a character in a string. Translate all non printing characters to '?' except for trailing zeros (ignore them).

**INDEX** Interpret the element selected by setArrayIndex() as an unsigned integer.

#### 9.173.1.2 enum QEStringFormatting::formats

Formatting options

**Enumerator:**

**FORMAT\_DEFAULT** Format according to the EPICS database record type.

**FORMAT\_FLOATING** Format as a floating point number.

**FORMAT\_INTEGER** Format as an integer.

**FORMAT\_UNSIGNEDINTEGER** Format as an unsigned integer.

**FORMAT\_TIME** Format as a time.

**FORMAT\_LOCAL\_ENUMERATE** Format as a selection from the local enumerations set by setLocalEnumeration().

**FORMAT\_STRING** Format as a string.

#### 9.173.1.3 enum QEStringFormatting::notations

Notations when formatting a floating point number

**Enumerator:**

**NOTATION\_FIXED** Standard floating point 123456.789.

**NOTATION\_SCIENTIFIC** Scientific representation 1.23456789e6.

**NOTATION\_AUTOMATIC** Automatic choice of standard or scientific notation.

#### 9.173.1.4 enum QEStringFormatting::separators

separators Defines the digit 'thousands' separator to be used.

**Enumerator:**

***SEPARATOR\_NONE*** Use no separator, e.g. 123456.123456789.

***SEPARATOR\_COMMA*** Use ',' as separator, e.g. 123,456.123,456,789.

***SEPARATOR\_UNDERSCORE*** Use '\_' as separator, e.g. 123\_456.123\_456\_-789.

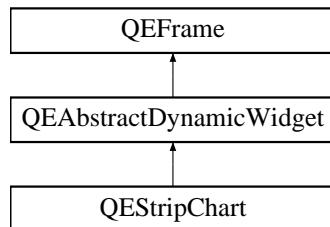
***SEPARATOR\_SPACE*** Use ' ' as separator, e.g. 123 456.123 456 789.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEStringFormatting.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QEStringFormatting.cpp

## 9.174 QEStripChart Class Reference

Inheritance diagram for QEStripChart::



### Public Types

- enum **PropertyChartYRanges** { **manual** = QEStripChartNames::manual, **dynamic** = QEStripChartNames::dynamic }
- enum **Constants** { **NUMBER\_OF\_PVS** = 16 }

### Public Slots

- void **videoModeSelected** (const QEStripChartNames::VideoModes mode)
- void **yRangeSelected** (const QEStripChartNames::ChartYRanges scale)
- void **yScaleModeSelected** (const QEStripChartNames::YScaleModes mode)

### Public Member Functions

- **QEStripChart** (QWidget \*parent=0)
- QSize **sizeHint** () const
- QDateTime **getStartTime** () const
- QDateTime **getEndTime** () const
- void **setEndTime** (QDateTime endTimeIn)
- int **getDuration** () const
- void **setDuration** (int durationIn)
- double **getYMinimum** () const
- void **setYMinimum** (const double yMinimumIn)
- double **getYMaximum** () const
- void **setYMaximum** (const double yMaximumIn)
- void **setYRange** (const double yMinimumIn, const double yMaximumIn)
- void **setPvName** (const int slot, const QString &pvName)
- QString **getPvName** (const int slot) const
- int **addPvName** (const QString &pvName)
- PropertyChartYRanges **getYRangeMode** () const
- void **setYRangeMode** (const PropertyChartYRanges scale)
- QEStripChartNames::VideoModes **getVideoMode** () const

- QEStripChartNames::YScaleModes **getYScaleMode** () const
- void **updateItemMenu** (const int slot, QAction \*action, const bool inUseMenu)

## Protected Member Functions

- bool **eventFilter** (QObject \*obj, QEvent \*event)
- void **mousePressEvent** (QMouseEvent \*event)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- qcaobject::QCaObject \* **createQcaItem** (unsigned int variableIndex)
- void **establishConnection** (unsigned int variableIndex)
- void **saveConfiguration** (PersistanceManager \*pm)
- void **restoreConfiguration** (PersistanceManager \*pm, restorePhases restorePhase)
- void **addToPredefinedList** (const QString &pvName)
- QStringList **getPredefinedPVNameList** () const
- QString **getPredefinedItem** (int i) const
- void **setRecalcIsRequired** ()
- void **setReplotIsRequired** ()
- void **evaluateAllowDrop** ()

## Properties

- int **duration**
- double **yMinimum**
- double **yMaximum**
- QEStripChartNames::VideoModes **videoMode**
- PropertyChartYRanges **chartRange**
- QEStripChartNames::YScaleModes **scaleMode**
- QString **variable1**
- QString **variable2**
- QString **variable3**
- QString **variable4**
- QString **variable5**
- QString **variable6**
- QString **variable7**
- QString **variable8**
- QString **variable9**
- QString **variable10**
- QString **variable11**
- QString **variable12**
- QString **variable13**
- QString **variable14**

- `QString variable15`
- `QString variable16`
- `QString variableSubstitutions`
- `QColor colour1`
- `QColor colour2`
- `QColor colour3`
- `QColor colour4`
- `QColor colour5`
- `QColor colour6`
- `QColor colour7`
- `QColor colour8`
- `QColor colour9`
- `QColor colour10`
- `QColor colour11`
- `QColor colour12`
- `QColor colour13`
- `QColor colour14`
- `QColor colour15`

## Friends

- class [QEStripChartItem](#)

### 9.174.1 Property Documentation

#### 9.174.1.1 `QString QEStripChart::variableSubstitutions [read, write]`

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'SAMPLE=SAM1, NAME = "Ref foil"' These substitutions are applied to all the variable names.

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChart.h`
- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChart.cpp`

## 9.175 QEStripChartAdjustPVDialog Class Reference

### Public Member Functions

- **QEStripChartAdjustPVDialog** (QWidget \*parent=0)
- void **setValueScaling** (const ValueScaling &valueScale)
- ValueScaling **getValueScaling** () const
- void **setSupport** (const double min, const double max, const QEDisplayRanges &loprHopr, const QEDisplayRanges &plotted, const QEDisplayRanges &buffered)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartA
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartA

## 9.176 QEStripChartContextMenu Class Reference

### Signals

- void **contextMenuSelected** (const QEStripChartNames::ContextMenuOptions)

### Public Member Functions

- [QEStripChartContextMenu](#) (bool inUse, QWidget \*parent=0)
- void **setPredefinedNames** (const QStringList &pvList)
- void **setUseReceiveTime** (const bool useReceiveTime)
- void **setArchiveReadHow** (const QEArciveInterface::How how)
- void **setLineDrawMode** (const QEStripChartNames::LineDrawModes mode)
- void **setLinePlotMode** (const QEStripChartNames::LinePlotModes mode)

#### 9.176.1 Constructor & Destructor Documentation

##### 9.176.1.1 QEStripChartContextMenu::QEStripChartContextMenu (bool *inUse*, QWidget \**parent* = 0) [explicit]

Construct strip chart item context menu. This menu item creates all required sub menu items. inUse set true for an inuse slot, i.e. already has a PV allocated. inUse set false for an empty slot.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartContextMenu.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartContextMenu.cpp

## 9.177 QEStripChartDurationDialog Class Reference

### Public Member Functions

- **QEStripChartDurationDialog** (QWidget \*parent=0)
- void **setDuration** (int secs)
- int **getDuration** () const

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartD...
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartD...

## 9.178 QEStripChartItem Class Reference

### Public Slots

- void **setColour** (const QColor &colour)

### Signals

- void **requestAction** (const QEActionRequests &)

### Public Member Functions

- **QEStripChartItem** (QEStripChart \*chart, const int slot, QWidget \*parent)
- bool **isInUse** () const
- bool **isCalculation** () const
- void **setPvName** (const QString &pvName, const QString &substitutions)
- QString **getPvName** () const
- QString **getEgu** () const
- bool **isScaled** () const
- QPointF **dataPointToReal** (const QCaDataPoint &point) const
- bool **getUseReceiveTime** () const
- QEArchiveInterface::How **getArchiveReadHow** () const
- QEStripChartNames::LineDrawModes **getLineDrawMode** () const
- QEStripChartNames::LinePlotModes **getLinePlotMode** () const
- QColor **getColour** ()
- QEDisplayRanges **getLoprHopr** (bool doScale)
- QEDisplayRanges **getDisplayedMinMax** (bool doScale)
- QEDisplayRanges **getBufferedMinMax** (bool doScale)
- QCaDataPointList **determinePlotPoints** ()
- void **readArchive** ()
- void **normalise** ()
- void **plotData** ()
- const QCaDataPoint \* **findNearestPoint** (const QCaDateTime &searchTime) const
- void **saveConfiguration** (PMElement &parentElement)
- void **restoreConfiguration** (PMElement &parentElement)
- void **updateMenu** (QAction \*action, const bool inUseMenu)

### Public Attributes

- QCaVariableNamePropertyManager **pvNamePropertyManager**

## Protected Member Functions

- bool **eventFilter** (QObject \*obj, QEvent \*event)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartItem.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartItem.cpp

## 9.179 QEStripChartNames Class Reference

### Public Types

- enum **ChartTimeModes** { **tmRealTime**, **tmPaused**, **tmHistorical** }
- enum **ChartYRanges** {
 **manual**, **operatingRange**, **plotted**, **buffered**,
 **dynamic**, **normalised** }
- enum **PlayModes** {
 **play**, **pause**, **forward**, **backward**,
 **selectTimes** }
- enum **StateModes** { **previous**, **next** }
- enum **VideoModes** { **normal**, **reverse** }
- enum **YScaleModes** { **linear**, **log** }
- enum **LineDrawModes** { **ldmHide**, **ldmRegular**, **ldmBold** }
- enum **LinePlotModes** { **lpmRectangular**, **lpmSmooth** }
- enum **ContextMenuOptions** {
 **SCCM\_NONE** = QEAbstractDynamicWidget::ADWCM\_SUB\_CLASS\_-  
**WIDGETS\_START\_HERE**, **SCCM\_READ\_ARCHIVE**, **SCCM\_SCALE\_-  
CHART\_AUTO**, **SCCM\_SCALE\_CHART\_PLOTTED**,  
**SCCM\_SCALE\_CHART\_BUFFERED**, **SCCM\_SCALE\_PV\_RESET**,  
**SCCM\_SCALE\_PV\_GENERAL**, **SCCM\_SCALE\_PV\_AUTO**,  
**SCCM\_SCALE\_PV\_PLOTTED**, **SCCM\_SCALE\_PV\_BUFFERED**,  
**SCCM\_SCALE\_PV\_CENTRE**, **SCCM\_PLOT\_RECTANGULAR**,  
**SCCM\_PLOT\_SMOOTH**, **SCCM\_PLOT\_SERVER\_TIME**, **SCCM\_-  
PLOT\_CLIENT\_TIME**, **SCCM\_ARCH\_LINEAR**,  
**SCCM\_ARCH\_PLOTBIN**, **SCCM\_ARCH\_RAW**, **SCCM\_ARCH\_SHEET**,  
**SCCM\_ARCH\_AVERAGED**,  
**SCCM\_LINE\_HIDE**, **SCCM\_LINE\_REGULAR**, **SCCM\_LINE\_BOLD**,  
**SCCM\_LINE\_COLOUR**,  
**SCCM\_PV\_EDIT\_NAME**, **SCCM\_ADD\_TO\_PREDEFINED**, **SCCM\_-  
PV\_WRITE\_TRACE**, **SCCM\_PV\_STATS**,  
**SCCM\_PV\_CLEAR**, **SCCM\_PV\_ADD\_NAME**, **SCCM\_PV\_PASTE\_-  
NAME**, **SCCM\_PREDEFINED\_01**,  
**SCCM\_PREDEFINED\_02**, **SCCM\_PREDEFINED\_03**, **SCCM\_-  
PREDEFINED\_04**, **SCCM\_PREDEFINED\_05**,  
**SCCM\_PREDEFINED\_06**, **SCCM\_PREDEFINED\_07**, **SCCM\_-  
PREDEFINED\_08**, **SCCM\_PREDEFINED\_09**,  
**SCCM\_PREDEFINED\_10** }

### Static Public Member Functions

- static QString **chartTimeModeStatus** (const ChartTimeModes mode)
- static QString **chartYRangeStatus** (const ChartYRanges yRange)

## Static Public Attributes

- static const ContextMenuOptions **ContextMenuItemFirst** = SCCM\_READ\_-  
ARCHIVE
- static const ContextMenuOptions **ContextMenuItemLast** = SCCM\_-  
PREDEFINED\_10
- static const int **NumberPrefefinedItems** = (SCCM\_PREDEFINED\_10 -  
SCCM\_PREDEFINED\_01 + 1)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartNa
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartNa

## 9.180 QEStripChartPushButtonSpecifications Struct Reference

### Public Attributes

- int **gap**
- int **width**
- int **value**
- bool **isIcon**
- const QString **captionOrIcon**
- const QString **toolTip**
- const char \* **member**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartToolBar.cpp

## 9.181 QEStripChartRangeDialog Class Reference

### Public Member Functions

- **QEStripChartRangeDialog** (QWidget \*parent=0)
- void **setRange** (const double min, const double max)
- double **getMinimum** ()
- double **getMaximum** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartRa
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartRa

## 9.182 QEStripChartState Class Reference

### Public Member Functions

- void **saveConfiguration** (PMElement &parentElement)
- void **restoreConfiguration** (PMElement &parentElement)

### Public Attributes

- bool **isNormalVideo**
- QEStripChartNames::ChartTimeModes **chartTimeMode**
- QEStripChartNames::YScaleModes **yScaleMode**
- QEStripChartNames::ChartYRanges **chartYScale**
- double **yMinimum**
- double **yMaximum**
- int **duration**
- Qt::TimeSpec **timeZoneSpec**
- QDateTime **endDateTime**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartState.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartState.cpp

## 9.183 QEStripChartStateList Class Reference

### Public Member Functions

- void **clear** ()
- void **push** (const QEStripChartState &state)
- bool **prev** (QEStripChartState &state)
- bool **next** (QEStripChartState &state)
- bool **prevAvailable** ()
- bool **nextAvailable** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartSt
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartSt

## 9.184 QEStripChartStatistics Class Reference

### Public Member Functions

- **QEStripChartStatistics** (const QString &pvName, const QString &egu, const QCaDataPointList &dataList, QEStripChartItem \*owner, QWidget \*parent=0)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartStatistics.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartStatistics.cpp

## 9.185 QEStripChartTimeDialog Class Reference

### Public Member Functions

- **QEStripChartTimeDialog** (QWidget \*parent=0)
- void **setMaximumDateTime** (QDateTime datetime)
- void **setStartTime** (QDateTime datetime)
- QDateTime **getStartTime** ()
- void **setEndTime** (QDateTime datetime)
- QDateTime **getEndTime** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartTi
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartTi

## 9.186 QEStripChartToolBar Class Reference

This class holds all the StripChart tool bar widgets.

```
#include <QEStripChartToolBar.h>
```

### Classes

- class [OwnTabWidget](#)

### Signals

- void **stateSelected** (const QEStripChartNames::StateModes mode)
- void **videoModeSelected** (const QEStripChartNames::VideoModes mode)
- void **yScaleModeSelected** (const QEStripChartNames::YScaleModes mode)
- void **yRangeSelected** (const QEStripChartNames::ChartYRanges scale)
- void **durationSelected** (const int seconds)
- void **selectDuration** ()
- void **timeZoneSelected** (const Qt::TimeSpec timeSpec)
- void **playModeSelected** (const QEStripChartNames::PlayModes mode)
- void **readArchiveSelected** ()
- void **loadSelected** ()
- void **saveAsSelected** ()
- void **loadSelectedFile** (const QString &filename)

### Public Member Functions

- **QEStripChartToolBar** (QWidget \*parent=0)
- void **setYRangeStatus** (const QEStripChartNames::ChartYRanges yRange)
- void  **setTimeStatus** (const QString &timeStatus)
- void **setDurationStatus** (const QString &durationStatus)
- void **setNOARStatus** (const int noar)
- void  **setTimeModeStatus** (const QEStripChartNames::ChartTimeModes timeMode)
- void  **setStateSelectionEnabled** (const QEStripChartNames::StateModes mode, const bool enabled)
- void  **setTimeRefs** (const QDateTime &t1, const QDateTime &t2)
- void  **setValue1Refs** (const double v1, const double v2)
- void  **setValue2Refs** (const double v1, const double v2)

### Static Public Member Functions

- static int **designHeight** ()

## Friends

- class **OwnTabWidget**

### 9.186.1 Detailed Description

This class holds all the StripChart tool bar widgets.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartTc.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartTc.cpp

## 9.187 QESubstitutedLabel Class Reference

### Public Member Functions

- **QESubstitutedLabel** (QWidget \*parent=0)
- void **setLabelTextProperty** (QString labelTextIn)
- QString **getLabelTextProperty** ()
- void **setSubstitutionsProperty** (QString macroSubstitutionsIn)
- QString **getSubstitutionsProperty** ()
- QString **getLabelTextPropertyFormat** ()
- void **setLabelTextPropertyFormat** (QString labelTextIn)

### Protected Attributes

- QString **labelText**

### Properties

- QString **textSubstitutions**

#### 9.187.1 Member Data Documentation

##### 9.187.1.1 QString QESubstitutedLabel::labelText [read, write, protected]

Label text to be substituted. This text will be copied to the label text after applying any macro substitutions from the textSubstitutions property

#### 9.187.2 Property Documentation

##### 9.187.2.1 QString QESubstitutedLabel::textSubstitutions [read, write]

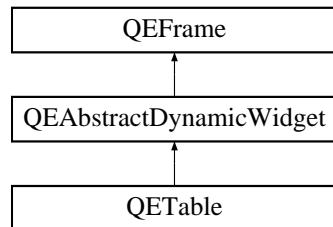
Text substitutions. These substitutions are applied to the 'labelText' property prior to copying it to the label text.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESubstitutedLabel/QESubstitutedLabel.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESubstitutedLabel/QESubstitutedLabel.cpp

## 9.188 QETable Class Reference

#include <QETable.h> Inheritance diagram for QETable::



### Classes

- class **DataSets**

### Public Slots

- void **setVariableName** (const int, const QString &)
- void **setTitles** (const QStringList &titles)
- void **setTitle** (const int slot, const QString &title)
- void **setSelection** (int value)
- void **setPvNameSet** (const QStringList &pvNameSet)
- void **setTableEntry** (const int slot, const QString &pvName, const QString &title)

### Signals

- void **selectionChanged** (int value)
- void **pvNameSetChanged** (const QStringList &pvNameSet)
- void **titlesChanged** (const QStringList &pvNameSet)
- void **dbValueChanged** (const QVector< double > &out)

### Public Member Functions

- [QETable](#) (QWidget \*parent=0)
  - virtual [~QETable](#) ()
- Destruction.*
- QString **getVariableName** (const int) const
  - void **setSubstitutions** (const QString &substitutions)
  - QString **getSubstitutions** () const
  - void **setColumnWidthMinimum** (const int columnWidthMinimum)
  - int **getColumnWidthMinimum** () const

- void **setDisplayMaximum** (const int displayMaximum)
- int **getDisplayMaximum** () const
- void **setOrientation** (const Qt::Orientation orientation)
- Qt::Orientation **getOrientation** () const
- int **addPvName** (const QString &pvName)
- QStringList **getTitles** () const
- int **getSelection** () const
- QStringList **getPvNameSet** () const

## Protected Types

- enum **OwnContextMenuOptions** { **CM\_HORIZONTAL\_TABLE** = ADWCM\_SUB\_CLASS\_WIDGETS\_START\_HERE, **CM\_VERTICAL\_TABLE** }

## Protected Member Functions

- QSize **sizeHint** () const
- void **fontChange** (const QFont &font)
- void **resizeEvent** (QResizeEvent \*event)
- void **establishConnection** (unsigned int variableIndex)
- qcaobject::QCaObject \* **createQcaItem** (unsigned int variableIndex)
- void **activated** ()
- QMenu \* **buildContextMenu** ()
- void **contextMenuTriggered** (int selectedItemNum)
- void **mousePressEvent** (QMouseEvent \*event)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **saveConfiguration** (PersistanceManager \*pm)
- void **restoreConfiguration** (PersistanceManager \*pm, restorePhases restorePhase)

## Properties

- QString **variableName1**  
*EPICS variable names (CA PV).*
- QString **variableName2**
- QString **variableName3**
- QString **variableName4**
- QString **variableName5**
- QString **variableName6**
- QString **variableName7**

- `QString variableName8`
- `QString variableName9`
- `QString variableName10`
- `QString variableName11`
- `QString variableName12`
- `QString variableName13`
- `QString variableName14`
- `QString variableName15`
- `QString variableName16`
- `QString variableName17`
- `QString variableName18`
- `QString variableName19`
- `QString variableName20`
- `QString variableSubstitutions`
- `QStringList titles`
- `int colWidthMinimum`
- `int displayMaximum`
- `Qt::Orientation orientation`
- `bool showGrid`
- `Qt::PenStyle gridStyle`

### 9.188.1 Detailed Description

This class provides an EPICS aware table widget which is capable of displaying up to 20 array PVs in tabular form. When in the default vertical orientation each column displays a consecutive element from an array EPICS variable.

When in horizontal mode, the table and functionality is transposed.

It is tightly integrated, via the [QEAbstractWidget](#), with the base class QEWidget class which provides generic support such as macro substitutions, drag/drop, and standard properties. [QEAbstractWidget](#) provides all standard properties.

### 9.188.2 Constructor & Destructor Documentation

#### 9.188.2.1 `QETable::QETable (QWidget *parent = 0) [explicit]`

Create without a variable(s). Use setVariableName functions.

### 9.188.3 Member Function Documentation

#### 9.188.3.1 `void QETable::dbValueChanged (const QVector< double > & out) [signal]`

Sent when the widget is updated following a data change Can be used to pass on EPICS data (as presented in this widget) to other widgets. For example a QList widget could log updates from this widget. Note: this widget emits the numeric enumeration value as opposed to the associated text.

## 9.188.4 Property Documentation

### 9.188.4.1 int QETable::colWidthMinimum [read, write]

Specified the minimum allow column width. The widget will shrink/expand the width of each column to as to exactly fit the width of the widget. However, columns will not shrink to less than the value provided by this property. Defaults to 80.

### 9.188.4.2 int QETable::displayMaximum [read, write]

The maximum number of array elements that will be displayed irrespective of the number of elements that the EPICS variable contains. Defaults to 4096.

### 9.188.4.3 Qt::PenStyle QETable::gridStyle [read, write]

Sets table grid style. Defaults to SolidLine.

### 9.188.4.4 Qt::Orientation QETable::orientation [read, write]

Determines if the variable values are displayed in rows (orientation is horizontal) or in columns (orientation is vertical). The default is vertical.

### 9.188.4.5 bool QETable::showGrid [read, write]

Controls if table grid is displayed. Default to true.

### 9.188.4.6 QStringList QETable::titles [read, write]

Allows specification of tables titles. If blank, the default, then out-of-the-box QTableWidget heading are used, i.e. 1, 2, etc. If "<>" is specified, then this is replaced by the PV name. This is particularly useful when PV names are specified dynamically or by substitution.

### 9.188.4.7 QString QETable::variableSubstitutions [read, write]

Macro substitutions. The default is no substitutions. The format is NAME1=VALUE1[,] NAME2=VALUE2... Values may be quoted strings. For example, 'PUMP=PMP3, NAME = "My Pump"' These substitutions are applied to variable names for all QE widgets. In some cases they are also used for other purposes.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QETable/QETable.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QETable/QETable.cpp

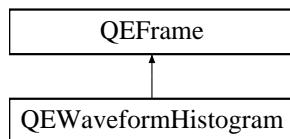
## 9.189 QETableManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QETable/QETableManager.h

## 9.190 QEWaveformHistogram Class Reference

#include <QEWaveformHistogram.h> Inheritance diagram for QEWaveformHistogram::



### Public Types

- enum **ScaleModes** { **Manual**, **Auto**, **OperationalRange** }

### Signals

- void **mouseIndexChanged** (const int index)
- void **mouseIndexPressed** (const int index, const Qt::MouseButton button)

### Public Member Functions

- **QEWaveformHistogram** (QWidget \*parent=0)
- void **setScaleMode** (const **ScaleModes** scaleMode)
- **ScaleModes** **getScaleMode** () const

### Protected Member Functions

- **qcaobject::QCaObject** \* **createQcaItem** (unsigned int variableIndex)
- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent \*event)
- void **dropEvent** (QDropEvent \*event)
- void **mousePressEvent** (QMouseEvent \*event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant v)
- void **setPvName** (const QString &pvName)

### Properties

- QString **variable**
- QString **variableSubstitutions**
- bool **autoBarGapWidths**
- int **barWidth**

- int **gap**
- int **margin**
- **ScaleModes** **scaleMode**
- double **minimum**
- double **maximum**
- double **baseLine**
- bool **drawAxes**
- bool **showScale**
- bool **showGrid**
- bool **logScale**
- QColor **backgroundColour**
- QColor **barColour**
- bool **drawBorder**
- Qt::Orientation **orientation**

### 9.190.1 Detailed Description

The [QEWaveformHistogram](#) class is a EPICS aware [histogram](#) widget. The value of, i.e. the length of each bar of the [histogram](#) is controlled by the corresponding element of an array, e.g. waveform, process variable. When the variable is defined (connected), the bar lengths are updated, and optionally the bar colours set to reflect the variable's severity status. The bar is 'grayed-out' when the variable is disconnected (although the bars retains their last known values/lengths). The [histogram](#) nature of the this widget is provided by a [QEHistogram](#) widget. The [QEWaveformHistogram](#) widget is tightly integrated with the base class QEWidget, via [QEFrame](#), which provides generic support such as macro substitutions, drag/drop, and standard properties.

### 9.190.2 Member Enumeration Documentation

#### 9.190.2.1 enum QEWaveformHistogram::ScaleModes

**Enumerator:**

*Manual* Use property minimum/maximum to scale [histogram](#).

*Auto* Dynamically scale based on minimum/maximum displayed value.

*OperationalRange* Use process variable operational range (LOPR/HOPR).

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEWaveformH
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEWaveformH

## 9.191 QEWaveformHistogramManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEHistogram/QEWaveformHistogramManager.h

## 9.192 QNumericEdit Class Reference

```
#include <QNumericEdit.h>
```

### Public Types

- enum **Notations** { **Fixed**, **Scientific** }

### Public Slots

- void **setValue** (const double value)
- void **setValue** (const int value)

### Signals

- void **valueChanged** (const double value)
- void **valueChanged** (const int value)
- void **returnPressed** ()
- void **editingFinished** ()

### Public Member Functions

- **QNumericEdit** (QWidget \*parent=0)  
*Construction.*
  - virtual **~QNumericEdit** ()  
*Destruction.*
  - double **getValue** () const
  - void **setPrefix** (const QString &prefix)
  - QString **getPrefix** () const
  - void **setSuffix** (const QString &suffix)
  - QString **getSuffix** () const
  - QString **getCleanText** () const
  - void **setLeadingZeros** (const int value)
  - int **getLeadingZeros** () const
  - void **setPrecision** (const int value)
  - int **getPrecision** () const
  - void **setNotation** (const **Notations** notation)
  - **Notations** **getNotation** () const
  - void **setMinimum** (const double value)
  - double **getMinimum** () const
  - void **setMaximum** (const double value)
  - double **getMaximum** () const

- void **setRadix** (const QEFixedPointRadix::Radicies value)
- QEFixedPointRadix::Radicies **getRadix** () const
- void **setSeparator** (const QEFixedPointRadix::Separators value)
- QEFixedPointRadix::Separators **getSeparator** () const

## Protected Member Functions

- void **focusInEvent** (QFocusEvent \*event)
- bool **eventFilter** (QObject \*obj, QEvent \*event)

## Protected Attributes

- QEFixedPointRadix **fpr**

## Properties

- bool **frame**
- QString **suffix**
- QString **prefix**
- Qt::Alignment **alignment**
- QString **cleanText**
- Notations **notation**
- QEFixedPointRadix::Radicies **radix**

*Specify radix, default is Decimal.*

- QEFixedPointRadix::Separators **separator**

*Specify digit 'thousands' separator character, default is none.*

- int **leadingZeros**
- int **precision**
- double **minimum**

*Specify the minimum allowed value.*

- double **maximum**

*Specify the maximum allowed value.*

- double **value**

*Specify the value after min/max.*

## Friends

- class **QENumericEdit**

### 9.192.1 Detailed Description

This class provides a non EPICS aware numeric edit widget, somewhat akin to QDoubleSpinBox. However [QNumericEdit](#) provides a number of advantages over QDoubleSpinBox: a/ the former is restricted to a single spin value, whereas [QNumericEdit](#) allows any digit to be selected and used as the spin value; b/ the former is restricted to decimal representation whereas [QNumericEdit](#) allows for hexadecimal, octal and binary representations; c/ this [QNumericEdit](#) allows a 'thousands' separator character to be specified; and d/ by setting the precision to 0, it effectively becomes akin to a QSpinBox.

### 9.192.2 Member Enumeration Documentation

#### 9.192.2.1 enum [QNumericEdit::Notations](#)

User friendly enumerations for notation property.

**Enumerator:**

*Fixed* Fixed point: +ddd.ddd.

*Scientific* Scientific: +d.ddddde+nn.

### 9.192.3 Property Documentation

#### 9.192.3.1 [Qt::Alignment](#) [QNumericEdit::alignment](#) [[read](#), [write](#)]

This property holds the alignment of the numeric edit. Both horizontal and vertical alignment is allowed here, Qt::AlignJustify will map to Qt::AlignLeft. By default, this property contains a combination of Qt::AlignRight and Qt::AlignVCenter.

#### 9.192.3.2 [QString](#) [QNumericEdit::cleanText](#) [[read](#)]

This property holds the displayed text. Not a property available to designer. It excludes any prefix/suffix.

#### 9.192.3.3 [bool](#) [QNumericEdit::frame](#) [[read](#), [write](#)]

This property holds whether the numeric edit draws itself with a frame. If enabled (the default) the numeric edit draws itself inside a frame, otherwise the line edit draws itself without any frame.

#### 9.192.3.4 [int](#) [QNumericEdit::leadingZeros](#) [[read](#), [write](#)]

Specifies the number of leading zeros. Strictly speaking, this should be an unsigned int, but designer int properties editor much 'nicer'.

**9.192.3.5 Notations QNumericEdit::notation [read, write]**

Notation used for formatting/editing. Default is fixed.

**9.192.3.6 int QNumericEdit::precision [read, write]**

Precision used for the display and editing of numbers. The default is 4. Strictly speaking, this should be an unsigned int, but designer int properties editor much 'nicer'.

**9.192.3.7 QString QNumericEdit::prefix [read, write]**

This property holds any fixed text (default is "") displayed before the numeric value.

**9.192.3.8 QString QNumericEdit::suffix [read, write]**

This property holds any fixed text (default is "") displayed after the numeric value.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QENumericEdit/QNumericEdit.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QENumericEdit/QNumericEdit.cpp

## 9.193 QNumericEditManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QENumericEdit/QNumericEd

## 9.194 QRadioGroup Class Reference

### Public Types

- enum [ButtonStyles](#) { [Radio](#), [Push](#) }
- enum [ButtonOrders](#) { [rowMajor](#), [colMajor](#) }

*Enumrations values used to select the button order.*

### Public Slots

- void [setValue](#) (const int value)

### Signals

- void [valueChanged](#) (const int value)

### Public Member Functions

- [QRadioButton](#) (QWidget \*parent=0)
- [QRadioButton](#) (const QString &title, QWidget \*parent=0)
- virtual ~[QRadioButton](#) ()

*Destruiction.*

- int [getMaximumButtons](#) () const
- int [getValue](#) () const
- void [setStrings](#) (const QStringList &strings)
- QStringList [getStrings](#) () const
- void [setColumns](#) (int columns)
- int [getColumns](#) () const
- void [setSpacing](#) (int spacing)
- int [getSpacing](#) () const
- void [setButtonStyle](#) (const [ButtonStyles](#) buttonStyle)
- [ButtonStyles](#) [getButtonStyle](#) () const
- void [setButtonOrder](#) (const [ButtonOrders](#) buttonOrder)
- [ButtonOrders](#) [getButtonOrder](#) () const

### Protected Member Functions

- QSize [sizeHint](#) () const
- bool [event](#) (QEvent \*event)

## Properties

- [QRadioButton::ButtonStyles buttonStyle](#)
- [QRadioButton::ButtonOrders buttonOrder](#)
- int **columns**
- int **spacing**
- QStringList **strings**
- int **value**

### 9.194.1 Member Enumeration Documentation

#### 9.194.1.1 enum QRadioButton::ButtonOrders

Enumerations values used to select the button order.

**Enumerator:**

*rowMajor* row by row button order - the default

#### 9.194.1.2 enum QRadioButton::ButtonStyles

Enumerations values used to select the button style. Whereas check box buttons can/do work, this option not provided as check boxes are not associated with the radio button, i.e. one and only one selected, paradigm.

**Enumerator:**

*Radio* Use radio buttons - the default.

### 9.194.2 Constructor & Destructor Documentation

#### 9.194.2.1 QRadioButton::QRadioButton (QWidget \* *parent* = 0) [explicit]

Create with default title.

#### 9.194.2.2 QRadioButton::QRadioButton (const QString & *title*, QWidget \* *parent* = 0) [explicit]

Create with a group title.

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QERadioGroup/QRadioGroup.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QERadioGroup/QRadioGroup.cpp

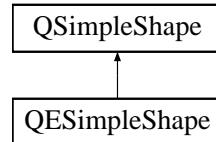
## 9.195 QRadioGroupManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QERadioGroup/QRadioGroupManager.h

## 9.196 QSimpleShape Class Reference

#include <QSimpleShape.h> Inheritance diagram for QSimpleShape::



### Public Types

- enum **Shapes** {
 **circle**, **ellipse**, **rectangle**, **roundRectangle**,  
**roundSquare**, **square**, **triangleUp**, **triangleDown**,  
**triangleLeft**, **triangleRight**, **triangleTopRight**, **triangleBottomRight**,  
**triangleBottomLeft**, **triangleTopLeft**, **diamond**, **equalDiamond**,  
**arrowUp**, **arrowDown**, **arrowLeft**, **arrowRight**,  
**crossHorizontal**, **crossVertical**, **hexagon**, **octagon**,  
**snakeHorizontal**, **snakeVertical** }
- enum **TextFormats** { **FixedText**, **StateSet**, **PvText**, **LocalEnumeration** }

### Public Slots

- void **setValue** (const int value)

### Public Member Functions

- [\*\*QSimpleShape\*\*](#) (QWidget \*parent=0)
- virtual [\*\*~QSimpleShape\*\*](#) ()  
*Destruction.*
- int **getValue** () const
- void **setModulus** (const int value)
- int **getModulus** () const
- void **setStateSet** (const QStringList &stateSet)
- QStringList **getStateSet** () const
- void **setShape** (const Shapes value)
- Shapes **getShape** () const
- void **setEdgeWidth** (const int value)
- int **getEdgeWidth** () const
- void **setSemiCycles** (const int value)
- int **getSemiCycles** () const

- void **setPercentSize** (const int value)
- int **getPercentSize** () const
- void **setEdgeColour** (const QColor value)
- QColor **getEdgeColour** () const
- void **setTextFormat** (const [TextFormats](#) value)
- [TextFormats](#) **getTextFormat** () const
- void **setFixedText** (const QString &value)
- QString **getFixedText** () const
- void **setFlashRate** (const QEScanTimers::ScanRates)
- QEScanTimers::ScanRates **getFlashRate** () const
- void **setFlashOffColour** (const QColor colour)
- QColor **getFlashOffColour** () const
- void **setIsActive** (const bool value)
- bool **getIsActive** () const
- void **setColourProperty** (const int slot, const QColor color)
- QColor **getColourProperty** (const int slot) const
- void **setFlashProperty** (int slot, const bool isFlashing)
- bool **getFlashProperty** (int slot) const

## Protected Member Functions

- virtual QString **getItemText** ()
- virtual QColor **getItemColour** ()

*returns "" unless overriden.*

## Properties

- Shapes [shape](#)
- int [edgeWidth](#)
- int [semiCycles](#)
- int [percentSize](#)
- int [value](#)

*Shape value - range 0 to 15 - default is zero.*

- int [modulus](#)

*Shape value modulus - range 2 to 16 - default is 16.*

- [TextFormats](#) [format](#)
- QString [fixedText](#)

*User specified text - defaults to null string.*

- QStringList [stateSet](#)

*Text values used when format is LocalEnumeration.*

- QEScanTimers::ScanRates **flashRate**
- bool **isActive**

*When inactive, the widgers is grayed-out, as if disabled. The default is true.*

- QColor **edgeColour**

*Edge color - default is black.*

- QColor **flashOffColour**

*Flash offColour - default is clear.*

- QColor **colour0**

*Value (modulo modulus) used to select colour.*

- QColor **colour1**

- QColor **colour2**

- QColor **colour3**

- QColor **colour4**

- QColor **colour5**

- QColor **colour6**

- QColor **colour7**

- QColor **colour8**

- QColor **colour9**

- QColor **colour10**

- QColor **colour11**

- QColor **colour12**

- QColor **colour13**

- QColor **colour14**

- QColor **colour15**

- bool **flash0**

- bool **flash1**

- bool **flash2**

- bool **flash3**

- bool **flash4**

- bool **flash5**

- bool **flash6**

- bool **flash7**

- bool **flash8**

- bool **flash9**

- bool **flash10**

- bool **flash11**

- bool **flash12**

- bool **flash13**

- bool **flash14**

- bool **flash15**

### 9.196.1 Detailed Description

This class is an basic simple shape widget. It has been refactored from [QESimpleShape](#) and will become the bases for [QESimpleShape](#).

### 9.196.2 Member Enumeration Documentation

#### 9.196.2.1 enum QSimpleShape::TextFormats

Enumerator:

*FixedText* Use user specified fixed text (default).

*StateSet* Use one of the stae Set values.

*PvText* Use EPICS value agumented with units if selected.

### 9.196.3 Constructor & Destructor Documentation

#### 9.196.3.1 QSimpleShape::QSimpleShape (QWidget \* *parent* = 0)

Construction

### 9.196.4 Property Documentation

#### 9.196.4.1 int QSimpleShape::edgeWidth [read, write]

Edge width - range 0 to 20 - default is 1. If edge width set to 0, then shape colour used for edge/boarder colour.

#### 9.196.4.2 bool QSimpleShape::flash0 [read, write]

When the widget's state coresponds to N and flashN is set true, the widget will alternate its normal colour with the specified flashOffColour.

#### 9.196.4.3 QEScanTimers::ScanRates QSimpleShape::flashRate [read, write]

Flash rate. The default value is Medium, i.e. 1Hz flash rate.

#### 9.196.4.4 TextFormats QSimpleShape::format [read, write]

Nominated text format

**9.196.4.5 int QSimpleShape::percentSize [read, write]**

Breadth of serpentine line as a percentage of widget height or width depending on wheather horizontal or vertical orientatiion. Range 1 to 50, default 10% Only applies to serpentine items.

**9.196.4.6 int QSimpleShape::semiCycles [read, write]**

Number of semi-cycles - range 1 to 30 - default is 8. Only applies to serpentine items.

**9.196.4.7 Shapes QSimpleShape::shape [read, write]**

Nominated shape

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESimpleShape/QSimpleShape.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESimpleShape/QSimpleShape.cpp

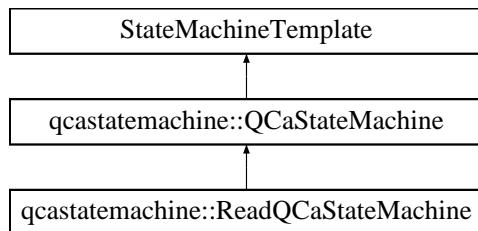
## 9.197 QSimpleShapeManager Class Reference

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QESimpleShape/QSimpleShapeManager.h

## 9.198 qcastatemachine::ReadQCaStateMachine Class Reference

Inheritance diagram for qcastatemachine::ReadQCaStateMachine::



### Public Member Functions

- **ReadQCaStateMachine** (void \*parent)
- bool **process** (int requestedState)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.cpp

## 9.199 recording Class Reference

### Signals

- void **byteArrayChanged** (const QByteArray &value, unsigned long dataSize, QCaAlarmInfo &alarmInfo, QCaDateTime &timeStamp, const unsigned int &variableIndex)
- void **playingBack** (bool playing)

### Public Member Functions

- **recording** (QWidget \*parent=0)
- bool **isRecording** ()
- void **recordImage** (QByteArray image, unsigned long dataSize, QCaAlarmInfo &alarmInfo, QCaDateTime &time)
- void **nextFrameDue** ()

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/recording.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/recording.cpp

## 9.200 imageDisplayProperties::rgbPixel Struct Reference

### Public Attributes

- unsigned char **p** [4]

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/brightnessContrast.h

## 9.201 screenSelectDialog Class Reference

### Public Types

- enum **screens** { **PRIMARY\_SCREEN** = -3, **THIS\_SCREEN** = -2, **ALL\_SCREENS** = -1 }

### Public Member Functions

- **screenSelectDialog** (int numScreens, QWidget \*parent=0)
- int **getScreenNum** ()

### Static Public Member Functions

- static bool **getFullscreenGeometry** (QWidget \*target, QRect &geom)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/screenSelectDialog.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/screenSelectDialog.cpp

## 9.202 selectMenu Class Reference

### Public Member Functions

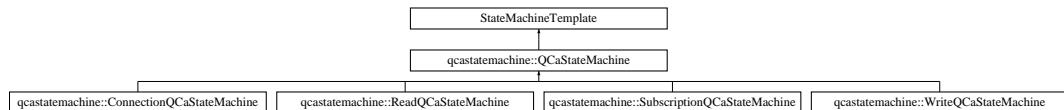
- **selectMenu** (QWidget \*parent=0)
- **imageContextMenu::imageContextMenuOptions** **getSelectOption** (const QPoint &pos)
- **void enable** (imageContextMenu::imageContextMenuOptions option, bool state)
- **bool isEnabled** (imageContextMenu::imageContextMenuOptions option)
- **void setChecked** (const int mode)
- **void setItemText** (imageContextMenu::imageContextMenuOptions option, QString title)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/selectMenu.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/selectMenu.cpp

## 9.203 StateMachineTemplate Class Reference

Inheritance diagram for StateMachineTemplate::



### Public Member Functions

- virtual bool process (int requestedState)=0

### Public Attributes

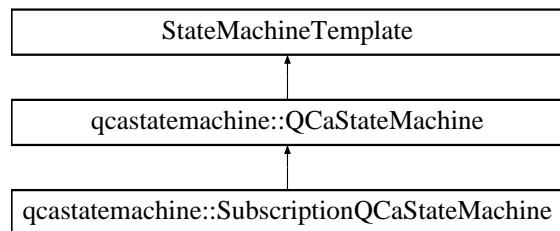
- int **currentState**
- int **requestState**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.h

## 9.204 qcastatemachine::SubscriptionQCaStateMachine Class Reference

Inheritance diagram for qcastatemachine::SubscriptionQCaStateMachine::



### Public Member Functions

- **SubscriptionQCaStateMachine** (void \*parent)
- bool **process** (int requestedState)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.cpp

## 9.205 trace Class Reference

### Public Attributes

- QVector< QCaDateTime > **timeStamps**
- QVector< double > **xdata**
- QVector< double > **ydata**
- QwtPlotCurve \* **curve**
- QColor **color**
- QString **legend**
- bool **waveform**
- QwtPlotCurve::CurveStyle **style**
- bool **hasCurrentPoint**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPlot/QEPlot.h

## 9.206 userInfoStruct Class Reference

### Public Attributes

- bool **enable**
- double **value1**
- double **value2**
- QString **elementText**

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h

## 9.207 QEPeriodic::userInfoStructArray Struct Reference

### Public Attributes

- `userInfoStruct array [NUM_ELEMENTS]`

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

- `/home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h`

## 9.208 ValueScaling Class Reference

### Public Member Functions

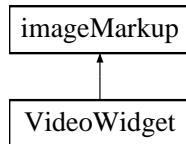
- void **reset** ()
- void **assign** (const ValueScaling &s)
- void **set** (const double dIn, const double mIn, const double cIn)
- void **get** (double &dOut, double &mOut, double &cOut) const
- void **map** (const double fromLower, const double fromUpper, const double toLower, const double toUpper)
- bool **isScaled** () const
- double **value** (const double x) const
- QEDisplayRanges **value** (const QEDisplayRanges &x) const
- void **saveConfiguration** (PMElement &parentElement) const
- void **restoreConfiguration** (PMElement &parentElement)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartU
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEStripChart/QEStripChartU

## 9.209 VideoWidget Class Reference

Inheritance diagram for VideoWidget::



### Signals

- void **userSelection** (imageMarkup::markupIds mode, bool complete, bool clearing, QPoint point1, QPoint point2, unsigned int thickness)
- void **zoomInOut** (int zoomAmount)
- void **currentPixelInfo** (QPoint pos)
- void **pan** (QPoint pos)
- void **redraw** ()

### Public Member Functions

- **VideoWidget** (QWidget \*parent=0)
- void **setNewImage** (QImage image, QCaDateTime &time)
- void **setPanning** (bool panningIn)
- bool **getPanning** ()
- QPoint **scalePoint** (QPoint pnt)
- int **scaleOrdinate** (int ord)
- QPoint **scaleImagePoint** (QPoint pnt)
- QRect **scaleImageRectangle** (QRect r)
- int **scaleImageOrdinate** (int ord)
- QImage **getImage** ()
- QSize **getImageSize** ()
- bool **hasCurrentImage** ()
- void **markupChange** ()

### Protected Member Functions

- void **paintEvent** (QPaintEvent \*)
- void **mousePressEvent** (QMouseEvent \*event)
- void **mouseReleaseEvent** (QMouseEvent \*event)
- void **mouseMoveEvent** (QMouseEvent \*event)
- void **wheelEvent** (QWheelEvent \*event)
- void **keyPressEvent** (QKeyEvent \*event)
- void **markupChange** (QVector< QRect > &changedAreas)
- void **resizeEvent** (QResizeEvent \*event)

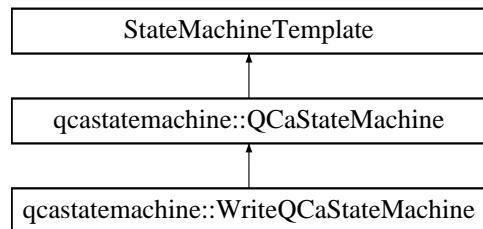
- void **markupSetCursor** (QCursor cursor)
- void **markupAction** (markupIds mode, bool complete, bool clearing, QPoint point1, QPoint point2, unsigned int thickness)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/videowidget.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/videowidget.cpp

## 9.210 qcastatemachine::WriteQCaStateMachine Class Reference

Inheritance diagram for qcastatemachine::WriteQCaStateMachine:::



### Public Member Functions

- **WriteQCaStateMachine** (void \*parent)
- bool **process** (int requestedState)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/data/QCaStateMachine.cpp

## 9.211 zoomMenu Class Reference

### Public Member Functions

- **zoomMenu** (QWidget \*parent=0)
- void **enableAreaSelected** (bool enable)
- imageContextMenu::imageContextMenuOptions **getZoom** (const QPoint &pos)

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

- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/zoomMenu.h
- /home/starritt/p4c/tec/gui/epicsqt/trunk/qeApp/epicsqt/framework/widgets/QEImage/zoomMenu.cpp

# Index

\_CopyPaste, 33  
\_Field, 34  
\_Item, 35  
\_QDialogItem, 36  
\_QPushButtonGroup, 37  
\_QWidgetFileBrowser, 38  
\_QWidgetLog, 39  
\_QWidgetScript, 40

addSeconds  
    QCaDateTime, 105

addUnits  
    QEAnalogProgressBar, 135  
    QCheckBox, 165  
    QELabel, 314  
    QLineEdit, 324  
    QENumericEdit, 354  
    QEPushButton, 400  
    QERadioButton, 441  
    QESimpleShape, 501

alarmSeverityDisplayMode  
    QEAnalogProgressBar, 135

aliasNames  
    QEPlotter, 382

alignment  
    QCheckBox, 165  
    QENumericEdit, 354  
    QEPushButton, 400  
    QERadioButton, 441  
    QNumericEdit, 554

allowDrop  
    QEAbstractWidget, 119  
    QEAnalogProgressBar, 135  
    QEAnalogSlider, 145  
    QEBitStatus, 153  
    QECheckBox, 165  
    QEComboBox, 179  
    QEConfiguredLayout, 186  
    QEFileBrowser, 194  
    QEFileImage, 199  
    QEForm, 210

QEFormGrid, 214  
QEFrame, 221  
QEGenericEdit, 236  
QEGroupBox, 242  
QEImage, 282  
QELabel, 314  
QELog, 336  
QEMenuButton, 342  
QEPeriodic, 363  
QEPlot, 375  
QEPushButton, 400  
QERadioButton, 441  
QEScript, 478  
QEShape, 489  
QESimpleShape, 501  
QESlider, 510  
QESpinBox, 517

allowedMaximum  
    QEResizableFrame, 461

allowedMinimum  
    QEResizableFrame, 461

allowFocusUpdate  
    QEComboBox, 179  
    QENumericEdit, 354  
    QESlider, 510  
    QESpinBox, 517

altReadback Variable  
    QEPushButton, 400

Always  
    QEAbstractWidget, 118  
    QEAnalogProgressBar, 133  
    QEAnalogSlider, 143  
    QEBitStatus, 152  
    QECheckBox, 162  
    QEComboBox, 178  
    QEConfiguredLayout, 185  
    QEFileBrowser, 193  
    QEFileImage, 199  
    QEFrame, 220  
    QEGenericEdit, 233  
    QEGroupBox, 241

QEImage, 278  
 QELabel, 312  
 QELog, 335  
 QEPeriodic, 362  
 QEPlot, 374  
 QEPushButton, 397  
 QERadioButton, 438  
 QEScript, 478  
 QEShape, 487  
 QESimpleShape, 501  
 QESlider, 509  
 QESpinBox, 516  
 animation1  
     QEShape, 489  
 animation2  
     QEShape, 489  
 animation3  
     QEShape, 489  
 animation4  
     QEShape, 490  
 animation5  
     QEShape, 490  
 animation6  
     QEShape, 490  
 animationOptions  
     QEShape, 487  
 APPEND  
     QESTringFormatting, 523  
 Append  
     QEAnalogProgressBar, 132  
     QECheckBox, 161  
     QELabel, 311  
     QELineEdit, 323  
     QEPushButton, 396  
     QERadioButton, 437  
 areaColor  
     QEImage, 282  
 areaInfo, 41  
 arguments  
     QECheckBox, 165  
     QEPushButton, 400  
     QERadioButton, 441  
 arguments1  
     QEImage, 282  
 arguments2  
     QEImage, 282  
 arrayAction  
     QEAnalogProgressBar, 135  
     QECheckBox, 166  
     QELabel, 314  
     QELineEdit, 324  
     QEPushButton, 400  
     QERadioButton, 442  
 ArrayActions  
     QEAnalogProgressBar, 132  
     QECheckBox, 161  
     QELabel, 311  
     QELineEdit, 323  
     QEPushButton, 396  
     QERadioButton, 437  
 arrayActions  
     QESTringFormatting, 523  
 arrayIndex  
     QEAnalogProgressBar, 135  
     QEAnalogSlider, 145  
     QEBitStatus, 153  
     QECheckBox, 166  
     QEComboBox, 179  
     QEGeneralEdit, 226  
     QEGenericEdit, 236  
     QELabel, 314  
     QENumericEdit, 354  
     QEPushButton, 401  
     QEPvFrame, 411  
     QERadioButton, 442  
     QERadioGroup, 453  
     QESimpleShape, 502  
     QESlider, 510  
     QESpinBox, 517  
 ASCII  
     QESTringFormatting, 523  
 Ascii  
     QEAnalogProgressBar, 132  
     QECheckBox, 161  
     QELabel, 311  
     QELineEdit, 323  
     QEPushButton, 396  
     QERadioButton, 437  
 Auto  
     QEScalarHistogram, 467  
     QEWaveformHistogram, 550  
 autoBrightnessContrast  
     QEImage, 282  
 Automatic  
     QEAnalogProgressBar, 133  
     QECheckBox, 163  
     QELabel, 312  
     QELineEdit, 323  
     QEPushButton, 398  
     QERadioButton, 439

autoScale  
QEAnalogSlider, 145  
QENumericEdit, 354  
axisAlarmColours  
QEAnalogSlider, 145  
  
backgroundColour  
QEAnalogIndicator, 126  
Bar  
QEAnalogIndicator, 126  
Bayer  
QEImage, 279  
BayerBG  
QEImage, 279  
BayerGB  
QEImage, 279  
BayerGR  
QEImage, 279  
BayerRG  
QEImage, 279  
beamColor  
QEImage, 283  
beamXVariable  
QEImage, 283  
beamYVariable  
QEImage, 283  
bitDepthVariable  
QEImage, 283  
borderColour  
QEAnalogIndicator, 126  
Bottom\_To\_Top  
QEAnalogIndicator, 126  
BOUNDING\_RECTANGLE  
QEImage, 278  
BoundingRectangle  
QEImage, 278  
briefInfoArea  
QEImage, 283  
buildImageCore  
imagePropertiesCore, 69  
buttonOrder  
QERadioButton, 453  
ButtonOrders  
QRadioButton, 558  
buttonStyle  
QERadioButton, 453  
ButtonStyles  
QRadioButton, 558  
  
CenterAndSize

QEImage, 278  
centreAngle  
QEAnalogIndicator, 126  
cleanText  
QENumericEdit, 354  
QNumericEdit, 554  
clickCheckedText  
QECheckBox, 166  
QEPushButton, 401  
QERadioButton, 442  
clicked  
QECheckBox, 164  
QEPushButton, 399  
QERadioButton, 440  
clickText  
QECheckBox, 166  
QEPushButton, 401  
QERadioButton, 442  
clippingHighVariable  
QEImage, 283  
clippingLowVariable  
QEImage, 283  
clippingOnOffVariable  
QEImage, 283  
colMacroPrefix  
QEFormGrid, 214  
colNumberOffset  
QEFormGrid, 214  
colNumberWidth  
QEFormGrid, 214  
color1  
QEShape, 490  
color10  
QEShape, 490  
color2  
QEShape, 490  
color3  
QEShape, 490  
color4  
QEShape, 490  
color5  
QEShape, 490  
color6  
QEShape, 491  
color7  
QEShape, 491  
color8  
QEShape, 491  
color9  
QEShape, 491

colStrings  
     QEFormGrid, 214  
 columns  
     QEFormGrid, 214  
     QERadioGroup, 453  
 colWidthMinimum  
     QETable, 547  
 Comma  
     QEAnalogProgressBar, 133  
     QECheckBox, 163  
     QELabel, 312  
     QELineEdit, 324  
     QERadioButton, 439  
 configurationFileLeft  
     QEPvLoadSave, 414  
 confirmAction  
     QECheckBox, 166  
     QEPushButton, 401  
     QEPvLoadSave, 414  
     QERadioButton, 442  
 confirmText  
     QECheckBox, 167  
     QEPushButton, 401  
     QERadioButton, 443  
 confirmWrite  
     QEGenericEdit, 236  
     QENumericEdit, 354  
 continuousWrite  
     QEAnalogSlider, 145  
 contrastReversal  
     QEImage, 283  
 creationOption  
     QECheckBox, 167  
     QEPushButton, 402  
     QERadioButton, 443  
 CreationOptionNames  
     QECheckBox, 161  
     QEPushButton, 396  
     QERadioButton, 437  
 customisationName  
     QECheckBox, 167  
     QEPushButton, 402  
     QERadioButton, 443  
 dataTypeVariable  
     QEImage, 284  
 dbConnectionChanged  
     QEAnalogProgressBar, 134  
     QEBitStatus, 152  
     QENumericEdit, 353  
 QEPvFrame, 410  
 QERadioGroup, 453  
 QESimpleShape, 501  
 dbElementChanged  
     QEPeriodic, 362  
 dbValueChanged  
     QEAnalogProgressBar, 134  
     QEAnalogSlider, 144  
     QEBitStatus, 152  
     QECheckBox, 164  
     QEComboBox, 179  
     QEFileImage, 200  
     QEImage, 281  
     QELabel, 314  
     QELineEdit, 324  
     QENumericEdit, 353  
     QEPeriodic, 362  
     QEPlot, 374  
     QEPushButton, 399  
     QEPvFrame, 410  
     QERadioButton, 440  
     QERadioGroup, 453  
     QESimpleShape, 501  
     QESlider, 509  
     QESpinBox, 517  
     QETable, 546  
 dbValueChanged1  
     QEShape, 488  
 dbValueChanged2  
     QEShape, 488  
 dbValueChanged3  
     QEShape, 488  
 dbValueChanged4  
     QEShape, 488  
 dbValueChanged5  
     QEShape, 489  
 dbValueChanged6  
     QEShape, 489  
 Default  
     QEAnalogProgressBar, 133  
     QECheckBox, 162  
     QELabel, 312  
     QELineEdit, 323  
     QEPushButton, 397  
     QERadioButton, 438  
 defaultDir  
     QEAbstractDynamicWidget, 116  
     QEPvLoadSave, 414  
 defaultStyle  
     QEAbstractWidget, 119

QEAnalogProgressBar, 135  
QEAnalogSlider, 145  
QEBitStatus, 153  
QECheckBox, 167  
QEComboBox, 179  
QEConfiguredLayout, 186  
QEFileBrowser, 194  
QEFileImage, 200  
QEFrame, 221  
QEGenericEdit, 236  
QEGroupBox, 242  
QEImage, 284  
QELabel, 315  
QELog, 336  
QEPlot, 375  
QEPushButton, 402  
QERadioButton, 443  
QEScript, 478  
QEShape, 491  
QESimpleShape, 502  
QESlider, 510  
QESpinBox, 517  
defaultSubstitutions  
    QEActionButton, 342  
    QE\_pvLoadSave, 414  
dimension1Variable  
    QEImage, 284  
dimension2Variable  
    QEImage, 284  
dimension3Variable  
    QEImage, 284  
dimensionsVariable  
    QEImage, 284  
disabledRecordPolicy  
    QECheckBox, 167  
    QEPushButton, 402  
    QERadioButton, 443  
displayAlarmState  
    QEAbstractWidget, 119  
    QEAnalogProgressBar, 136  
    QEAnalogSlider, 145  
    QEBitStatus, 153  
    QECheckBox, 167  
    QEComboBox, 180  
    QEConfiguredLayout, 186  
    QEFileBrowser, 194  
    QEFileImage, 200  
    QEFrame, 221  
    QEGenericEdit, 236  
    QEGroupBox, 242  
QEImage, 284  
QELabel, 315  
QELog, 336  
QEPlot, 375  
QEPushButton, 402  
QERadioButton, 443  
QEScript, 478  
QEShape, 491  
QESimpleShape, 502  
QESlider, 510  
QESpinBox, 517  
displayAlarmStateOption  
    QEAbstractWidget, 119  
    QEAnalogProgressBar, 136  
    QEAnalogSlider, 145  
    QEBitStatus, 153  
    QECheckBox, 168  
    QEComboBox, 180  
    QEConfiguredLayout, 186  
    QEFileBrowser, 194  
    QEFileImage, 200  
    QEForm, 210  
    QEFormGrid, 214  
    QEFrame, 221  
    QEGenericEdit, 236  
    QEGroupBox, 242  
    QEImage, 285  
    QELabel, 315  
    QELog, 336  
    QEActionButton, 342  
    QEPeriodic, 363  
    QEPlot, 375  
    QEPushButton, 403  
    QERadioButton, 444  
    QEScript, 479  
    QEShape, 491  
    QESimpleShape, 502  
    QESlider, 510  
    QESpinBox, 517  
DisplayAlarmStateOptions  
    QEAbstractWidget, 118  
    QEAnalogProgressBar, 132  
    QEAnalogSlider, 143  
    QEBitStatus, 152  
    QECheckBox, 162  
    QEComboBox, 178  
    QEConfiguredLayout, 185  
    QEFileBrowser, 193  
    QEFileImage, 199

QEFrame, 220  
 QEGenericEdit, 233  
 QEGroupBox, 241  
 QEImage, 278  
 QELabel, 311  
 QELog, 335  
 QEPeriodic, 362  
 QEPlot, 374  
 QEPushButton, 397  
 QERadioButton, 438  
 QEScript, 478  
 QEShape, 487  
 QESimpleShape, 500  
 QESlider, 509  
 QESpinBox, 516  
 displayArea1Selection  
     QEImage, 285  
 displayArea2Selection  
     QEImage, 285  
 displayArea3Selection  
     QEImage, 285  
 displayArea4Selection  
     QEImage, 285  
 displayBeamSelection  
     QEImage, 285  
 displayButtonBar  
     QEImage, 282  
 displayCursorPixelInfo  
     QEImage, 285  
 displayEllipse  
     QEImage, 285  
 displayHozSlice1Selection  
     QEImage, 286  
 displayHozSlice2Selection  
     QEImage, 286  
 displayHozSlice3Selection  
     QEImage, 286  
 displayHozSlice4Selection  
     QEImage, 286  
 displayHozSlice5Selection  
     QEImage, 286  
 displayMaximum  
     QETable, 547  
 displayProfileSelection  
     QEImage, 286  
 displayTargetSelection  
     QEImage, 286  
 displayVertSlice1Selection  
     QEImage, 286  
 displayVertSlice2Selection  
     QEImage, 286  
 QEImage, 286  
 displayVertSlice3Selection  
     QEImage, 287  
 displayVertSlice4Selection  
     QEImage, 287  
 displayVertSlice5Selection  
     QEImage, 287  
 DockBottom  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DockBottomTabbed  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DockFloating  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DockLeft  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DockLeftTabbed  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DockRight  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DockRightTabbed  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DockTop  
     QECheckBox, 162  
     QEPushButton, 396  
     QERadioButton, 438  
 DockTopTabbed  
     QECheckBox, 162  
     QEPushButton, 397  
     QERadioButton, 438  
 DottedFullCrosshair  
     QEImage, 281  
 drawMarkup  
     markupHLine, 76  
     markupVLine, 83  
 edgeWidth

QSimpleShape, 563  
ellipseColor  
    QEImage, 287  
ellipseHVariable  
    QEImage, 287  
EllipseVariableDefinitions  
    QEImage, 278  
ellipseVariableDefinitions  
    QEImage, 278  
ellipseWVariable  
    QEImage, 287  
ellipseXVariable  
    QEImage, 287  
ellipseYVariable  
    QEImage, 287  
enableArea1Selection  
    QEImage, 287  
enableArea2Selection  
    QEImage, 288  
enableArea3Selection  
    QEImage, 288  
enableArea4Selection  
    QEImage, 288  
enableBeamSelection  
    QEImage, 288  
enableHozSlice1Selection  
    QEImage, 288  
enableHozSlice2Selection  
    QEImage, 288  
enableHozSlice3Selection  
    QEImage, 288  
enableHozSlice4Selection  
    QEImage, 288  
enableHozSlice5Selection  
    QEImage, 289  
enableProfileSelection  
    QEImage, 289  
enableTargetSelection  
    QEImage, 289  
enableVertSlice1Selection  
    QEImage, 289  
enableVertSlice2Selection  
    QEImage, 289  
enableVertSlice3Selection  
    QEImage, 289  
enableVertSlice4Selection  
    QEImage, 289  
enableVertSlice5Selection  
    QEImage, 290  
Engineer

    QEAbstractWidget, 118  
QEAnalogProgressBar, 134  
QEAnalogSlider, 144  
QEBitStatus, 152  
QECheckBox, 164  
QEComboBox, 178  
QEConfiguredLayout, 185  
QEFileBrowser, 193  
QEFileImage, 199  
QEFrame, 220  
QEGenericEdit, 234  
QEGroupBox, 241  
QEImage, 281  
QELabel, 313  
QELog, 336  
QEPeriodic, 362  
QEPlot, 374  
QEPushButton, 399  
QERadioButton, 440  
QEScript, 478  
QEShape, 488  
QESimpleShape, 501  
QESlider, 509  
QESpinBox, 516  
externalControls  
    QEImage, 290

    FFBuffer, 46  
FFThread, 47  
fileFilter  
    QEFileBrowser, 194  
Fit  
    QEImage, 279  
Fixed  
    QEAnalogProgressBar, 133  
QECheckBox, 163  
QELabel, 312  
QELineEdit, 323  
QEPushButton, 398  
QERadioButton, 439  
QNumericEdit, 554  
FixedText  
    QSimpleShape, 563  
flash0  
    QSimpleShape, 563  
flashRate  
    QSimpleShape, 563  
flipRotateMenu, 48  
Floating  
    QEAnalogProgressBar, 133

QECheckBox, 162  
 QELabel, 312  
 QELineEdit, 323  
 QEPushButton, 397  
 QERadioButton, 438  
 floating  
     QCaDateTime, 105  
 fontColour  
     QEAnalogIndicator, 126  
 foregroundColour  
     QEAnalogIndicator, 126  
 format  
     QEAnalogProgressBar, 136  
     QECheckBox, 168  
     QELabel, 315  
     QELineEdit, 325  
     QEPushButton, 403  
     QERadioButton, 444  
     QSimeShape, 563  
 FORMAT\_DEFAULT  
     QStringFormatting, 523  
 FORMAT\_FLOATING  
     QStringFormatting, 523  
 FORMAT\_INTEGER  
     QStringFormatting, 523  
 FORMAT\_LOCAL\_ENUMERATE  
     QStringFormatting, 523  
 FORMAT\_STRING  
     QStringFormatting, 523  
 FORMAT\_TIME  
     QStringFormatting, 523  
 FORMAT\_UNSIGNEDINTEGER  
     QStringFormatting, 523  
 formatInteger  
     QEIntegerFormatting, 306  
 formatIntegerArray  
     QEIntegerFormatting, 306  
 formatOption  
     QEImage, 290  
 FormatOptions  
     QEImage, 278  
 Formats  
     QEAnalogProgressBar, 133  
     QECheckBox, 162  
     QELabel, 312  
     QELineEdit, 323  
     QEPushButton, 397  
     QERadioButton, 438  
 formats  
     QStringFormatting, 523  
 formatValue  
     QEIntegerFormatting, 307  
 formatVariable  
     QEImage, 290  
 frame  
     QENumericEdit, 354  
     QNumericEdit, 554  
 fullScreenWindow, 49  
 getConfirmWrite  
     QEGenericEdit, 234  
 getLocalEnumeration  
     QELocalEnumeration, 331  
 getPixelValueFromData  
     imageProcessor, 65  
 getSubscribe  
     QEGenericEdit, 234  
 getWriteOnEnter  
     QEGenericEdit, 234  
 getWriteOnFinish  
     QEGenericEdit, 234  
 getWriteOnLoseFocus  
     QEGenericEdit, 235  
 grabbingEdge  
     QEResizeableFrame, 461  
 gridOrder  
     QEFormGrid, 214  
 GridOrders  
     QEFormGrid, 213  
 gridStyle  
     QETable, 547  
 guiFile  
     QECheckBox, 168  
     QEPushButton, 403  
     QERadioButton, 444  
 handleGuiLaunchRequests  
     QEForm, 209  
 heightVariable  
     QEImage, 290  
 histogram, 50  
 histogramScroll, 51  
 historicImage, 52  
 horizontalFlip  
     QEImage, 290  
 hozSlice1Color  
     QEImage, 290  
 hozSlice2Color  
     QEImage, 290  
 hozSlice3Color

QEImage, 290  
hozSlice4Color  
    QEImage, 291  
hozSlice5Color  
    QEImage, 291

Icon  
    QECheckBox, 163  
    QEPushButton, 398  
    QERadioButton, 439

imageContextMenu, 53  
imageDisplayProperties, 55  
imageDisplayProperties::rgbPixel, 568  
imageInfo, 57  
imageMarkup, 58  
imageMarkupLegendSetText, 61  
imageProcessor, 62  
    getPixelValueFromData, 65

imageProperties, 66  
    imageProperties, 68  
    ROTATION\_0, 68  
    ROTATION\_180, 68  
    ROTATION\_90\_LEFT, 68  
    ROTATION\_90\_RIGHT, 68  
    rotationOptions, 68

imagePropertiesCore, 69  
    buildImageCore, 69

imageUpdateIndicator, 70

imageVariable  
    QEImage, 291

INDEX  
    QEStringFormatting, 523

Index  
    QEAnalogProgressBar, 132  
    QECheckBox, 161  
    QELabel, 311  
    QELineEdit, 323  
    QEPushButton, 396  
    QERadioButton, 437

initialHosScrollPos  
    QEImage, 291

initialVertScrollPos  
    QEImage, 282

int  
    QEAbstractWidget, 119  
    QEAnalogProgressBar, 136  
    QEAnalogSlider, 146  
    QEBitStatus, 153  
    QECheckBox, 168  
    QEComboBox, 180

QEConfiguredLayout, 186  
QEFileBrowser, 194  
QEFileDialog, 200  
QEForm, 210  
QEFrame, 221  
QEGenericEdit, 237  
QEGroupBox, 242

QEImage, 291  
QELabel, 315  
QELog, 337  
QEPeriodic, 363  
QEPlot, 375  
QEPushButton, 403  
QERadioButton, 444  
QEScript, 479  
QEShape, 492  
QESimpleShape, 502  
QESlider, 511  
QESpinBox, 518

Integer  
    QEAnalogProgressBar, 133  
    QECheckBox, 162  
    QELabel, 312  
    QELineEdit, 323  
    QEPushButton, 397  
    QERadioButton, 438

isDefined  
    QELocalEnumeration, 331

labelText  
    QECheckBox, 168  
    QEPushButton, 403  
    QERadioButton, 444  
    QESubstitutedLabel, 543

leadingZero  
    QEAnalogProgressBar, 136  
    QECheckBox, 169  
    QELabel, 315  
    QELineEdit, 325  
    QEPushButton, 404  
    QERadioButton, 445

leadingZeros  
    QENumericEdit, 354  
    QNumericEdit, 554

Left\_To\_Right  
    QEAnalogIndicator, 126

lineProfileArrayVariable  
    QEImage, 291

lineProfileThicknessVariable  
    QEImage, 291

lineProfileX1Variable  
     QEImage, 291  
 lineProfileX2Variable  
     QEImage, 292  
 lineProfileY1Variable  
     QEImage, 292  
 lineProfileY2Variable  
     QEImage, 292  
 LocalEnumeration  
     QEAnalogProgressBar, 133  
     QECheckBox, 162  
     QELabel, 312  
     QELineEdit, 323  
     QEPushButton, 397  
     QERadioButton, 438  
 localEnumeration  
     QEAnalogProgressBar, 136  
     QECheckBox, 169  
     QEComboBox, 180  
     QELabel, 315  
     QELineEdit, 325  
     QEPushButton, 404  
     QERadioButton, 445  
     QERadioGroup, 453  
     QESimpleShape, 502  
 logBrightness  
     QEImage, 292  
 loginWidget, 71  
 LogOutput  
     QECheckBox, 163  
     QEImage, 279  
     QEPushButton, 398  
     QERadioButton, 439  
 logScale  
     QEAnalogIndicator, 126  
 logScaleInterval  
     QEAnalogIndicator, 127  
 majorInterval  
     QAnalogSlider, 97  
     QEAnalogIndicator, 127  
 Manual  
     QEScalarHistogram, 467  
     QEWaveformHistogram, 550  
 margin  
     QEFormGrid, 215  
 markupCrosshair1, 72  
 markupCrosshair2, 73  
 markupDisplayMenu, 74  
 markupEllipse, 75  
 markupHLine, 76  
     drawMarkup, 76  
 markupItem, 77  
 markupLine, 80  
 markupRegion, 81  
 markupText, 82  
 markupVLine, 83  
     drawMarkup, 83  
 maximum  
     QEAnalogIndicator, 127  
     QENumericEdit, 355  
 messageFormFilter  
     QEForm, 210  
 messageSourceFilter  
     QEForm, 210  
 Meter  
     QEAnalogIndicator, 126  
 minimum  
     QEAnalogIndicator, 127  
     QENumericEdit, 355  
 minorInterval  
     QAnalogSlider, 97  
     QEAnalogIndicator, 127  
 mode  
     QEAnalogIndicator, 127  
 Modes  
     QEAnalogIndicator, 126  
 Mono  
     QEImage, 279  
 mpegSource, 84  
     updateImage, 84  
 mpegSourceObject, 85  
 Never  
     QEAbstractWidget, 118  
     QEAnalogProgressBar, 133  
     QEAnalogSlider, 143  
     QEBitStatus, 152  
     QECheckBox, 162  
     QEComboBox, 178  
     QEConfiguredLayout, 185  
     QEFileBrowser, 193  
     QEFileDialog, 199  
     QEFrame, 220  
     QEGenericEdit, 233  
     QEGroupBox, 241  
     QEImage, 278  
     QELabel, 312  
     QELog, 335  
     QEPeriodic, 362

QEPlot, 374  
QEPushButton, 397  
QERadioButton, 438  
QEScript, 478  
QEShape, 487  
QESimpleShape, 501  
QESlider, 509  
QESpinBox, 516  
NewTab  
    QECheckBox, 161  
    QEPushButton, 396  
    QERadioButton, 437  
NewWindow  
    QECheckBox, 162  
    QEPushButton, 396  
    QERadioButton, 438  
None  
    QECheckBox, 163  
    QEImage, 279  
    QEPushButton, 398  
    QERadioButton, 439  
NoRotation  
    QEImage, 280  
NoSeparator  
    QEAnalogProgressBar, 133  
    QECheckBox, 163  
    QELabel, 312  
    QELlineEdit, 324  
    QERadioButton, 439  
notation  
    QEAnalogProgressBar, 137  
    QECheckBox, 169  
    QELabel, 316  
    QELlineEdit, 325  
    QENumericEdit, 355  
    QEPushButton, 404  
    QERadioButton, 445  
    QNumericEdit, 554  
NOTATION\_AUTOMATIC  
    QEStringFormatting, 523  
NOTATION\_FIXED  
    QEStringFormatting, 523  
NOTATION\_SCIENTIFIC  
    QEStringFormatting, 523  
Notations  
    QEAnalogProgressBar, 133  
    QECheckBox, 162  
    QELabel, 312  
    QELlineEdit, 323  
    QEPushButton, 397  
    QERadioButton, 438  
notations  
    QEStringFormatting, 523  
number  
    QEFormGrid, 215  
offset1  
    QEShape, 492  
offset2  
    QEShape, 492  
offset3  
    QEShape, 492  
offset4  
    QEShape, 492  
offset5  
    QEShape, 492  
offset6  
    QEShape, 492  
Open  
    QECheckBox, 161  
    QEPushButton, 396  
    QERadioButton, 437  
OperationalRange  
    QEScalarHistogram, 467  
    QEWaveformHistogram, 550  
orientation  
    QEAnalogIndicator, 127  
    QEHistogram, 246  
    QETable, 547  
Orientations  
    QEAnalogIndicator, 126  
password  
    QECheckBox, 169  
    QEPushButton, 404  
    QERadioButton, 445  
percentSize  
    QSimpleShape, 563  
PeriodicDialog, 87  
PeriodicElementSetupForm, 88  
PeriodicSetupDialog, 89  
Picture  
    QELabel, 313  
 pixmap  
    QEFrame, 221  
 pixmap0  
    QECheckBox, 170  
    QEFrame, 221  
    QELabel, 316

QEPushButton, 404  
 QERadioButton, 446  
 pixmap1  
     QECheckBox, 170  
     QEFrame, 222  
     QELabel, 316  
     QEPushButton, 405  
     QERadioButton, 446  
 pixmap2  
     QECheckBox, 170  
     QEFrame, 222  
     QELabel, 316  
     QEPushButton, 405  
     QERadioButton, 446  
 pixmap3  
     QECheckBox, 170  
     QEFrame, 222  
     QELabel, 317  
     QEPushButton, 405  
     QERadioButton, 446  
 pixmap4  
     QECheckBox, 170  
     QEFrame, 222  
     QELabel, 317  
     QEPushButton, 405  
     QERadioButton, 446  
 pixmap5  
     QECheckBox, 170  
     QEFrame, 222  
     QELabel, 317  
     QEPushButton, 405  
     QERadioButton, 446  
 pixmap6  
     QECheckBox, 170  
     QEFrame, 222  
     QELabel, 317  
     QEPushButton, 405  
     QERadioButton, 446  
 pixmap7  
     QECheckBox, 170  
     QEFrame, 222  
     QELabel, 317  
     QEPushButton, 405  
     QERadioButton, 446  
 playbackTimer, 90  
 point1  
     QEShape, 492  
 point10  
     QEShape, 493  
 point2

QEShape, 493  
 point3  
     QEShape, 493  
 point4  
     QEShape, 493  
 point5  
     QEShape, 493  
 point6  
     QEShape, 493  
 point7  
     QEShape, 493  
 point8  
     QEShape, 493  
 point9  
     QEShape, 494  
 pointInfo, 91  
 precision  
     QAnalogSlider, 97  
     QEAnalogProgressBar, 137  
     QECheckBox, 170  
     QELabel, 317  
     QELineEdit, 326  
     QENumericEdit, 355  
     QEPushButton, 405  
     QERadioButton, 446  
     QNumericEdit, 555  
 prefix  
     QNumericEdit, 555  
 pressed  
     QECheckBox, 164  
     QEPushButton, 399  
     QERadioButton, 440  
 pressText  
     QECheckBox, 171  
     QEPushButton, 405  
     QERadioButton, 447  
 prioritySubstitutions  
     QECheckBox, 171  
     QEPushButton, 406  
     QERadioButton, 447  
 profileColor  
     QEImage, 292  
 profileHoz1ThicknessVariable  
     QEImage, 292  
 profileHoz1Variable  
     QEImage, 292  
 profileHoz2ThicknessVariable  
     QEImage, 292  
 profileHoz2Variable  
     QEImage, 292

profileHoz3ThicknessVariable  
QEImage, 293  
profileHoz3Variable  
QEImage, 293  
profileHoz4ThicknessVariable  
QEImage, 293  
profileHoz4Variable  
QEImage, 293  
profileHoz5ThicknessVariable  
QEImage, 293  
profileHoz5Variable  
QEImage, 293  
profileHozArrayVariable  
QEImage, 293  
profilePlot, 92  
profileVert1ThicknessVariable  
QEImage, 293  
profileVert1Variable  
QEImage, 294  
profileVert2ThicknessVariable  
QEImage, 294  
profileVert2Variable  
QEImage, 294  
profileVert3ThicknessVariable  
QEImage, 294  
profileVert3Variable  
QEImage, 294  
profileVert4ThicknessVariable  
QEImage, 294  
profileVert4Variable  
QEImage, 294  
profileVert5ThicknessVariable  
QEImage, 294  
profileVert5Variable  
QEImage, 295  
profileVertArrayListVariable  
QEImage, 295  
program  
QECheckBox, 171  
QEPushButton, 406  
QERadioButton, 447  
program1  
QEImage, 295  
program2  
QEImage, 295  
programStartupOption  
QECheckBox, 171  
QEPushButton, 406  
QERadioButton, 447  
programStartupOption1  
QEImage, 295  
programStartupOption2  
QEImage, 295  
ProgramStartupOptionNames  
QECheckBox, 163  
QEImage, 279  
QEPushButton, 398  
QERadioButton, 439  
PushButtonSpecifications, 93  
PvText  
QSimepleShape, 563  
QAnalogSlider, 94  
majorInterval, 97  
minorInterval, 97  
precision, 97  
QAnalogSlider, 97  
tracking, 97  
QAnalogSliderManager, 98  
QBitStatus, 99  
QCaAlarmInfo, 101  
QCaConnectionInfo, 102  
QCaDataPoint, 103  
QCaDataPointList, 104  
QCaDateTime, 105  
addSeconds, 105  
floating, 105  
QCaEventFilter, 106  
QCaEventItem, 107  
QCaEventUpdate, 108  
QCaInstalledFiltersListItem, 109  
qcaobject::QCaObject, 110  
qcastatemachine::ConnectionQCaStateMachine,  
44  
qcastatemachine::QCaStateMachine, 113  
qcastatemachine::ReadQCaStateMachine,  
566  
qcastatemachine::SubscriptionQCaStateMachine,  
572  
qcastatemachine::WriteQCaStateMachine,  
579  
QCaVariableNamePropertyManager, 114  
QEAbstractDynamicWidget, 115  
defaultDir, 116  
QEAbstractWidget, 117  
allowDrop, 119  
Always, 118  
defaultStyle, 119  
displayAlarmState, 119  
displayAlarmStateOption, 119

DisplayAlarmStateOptions, 118  
 Engineer, 118  
 int, 119  
 Never, 118  
 Scientist, 118  
 setManagedVisible, 119  
 styleSheet, 120  
 User, 118  
 userLevelEnabled, 120  
 userLevelEngineerStyle, 120  
 UserLevels, 118  
 userLevelScientistStyle, 120  
 userLevelUserStyle, 120  
 userLevelVisibility, 121  
 variableAsToolTip, 121  
 visible, 121  
 WhenInAlarm, 118  
 QEAnalogIndicator, 122  
 backgroundColour, 126  
 Bar, 126  
 borderColour, 126  
 Bottom\_To\_Top, 126  
 centreAngle, 126  
 fontColour, 126  
 foregroundColour, 126  
 Left\_To\_Right, 126  
 logScale, 126  
 logScaleInterval, 127  
 majorInterval, 127  
 maximum, 127  
 Meter, 126  
 minimum, 127  
 minorInterval, 127  
 mode, 127  
 Modes, 126  
 orientation, 127  
 Orientations, 126  
 Right\_To\_Left, 126  
 Scale, 126  
 showScale, 127  
 showText, 127  
 spanAngle, 127  
 Top\_To\_Bottom, 126  
 value, 127  
 QEAnalogIndicator::Band, 42  
 QEAnalogIndicator::BandList, 43  
 QEAnalogProgressBar, 129  
 addUnits, 135  
 alarmSeverityDisplayStyle, 135  
 allowDrop, 135  
 Always, 133  
 Append, 132  
 arrayAction, 135  
 ArrayActions, 132  
 arrayIndex, 135  
 Ascii, 132  
 Automatic, 133  
 Comma, 133  
 dbConnectionChanged, 134  
 dbValueChanged, 134  
 Default, 133  
 defaultStyle, 135  
 displayAlarmState, 136  
 displayAlarmStateOption, 136  
 DisplayAlarmStateOptions, 132  
 Engineer, 134  
 Fixed, 133  
 Floating, 133  
 format, 136  
 Formats, 133  
 Index, 132  
 int, 136  
 Integer, 133  
 leadingZero, 136  
 LocalEnumeration, 133  
 localEnumeration, 136  
 Never, 133  
 NoSeparator, 133  
 notation, 137  
 Notations, 133  
 precision, 137  
 QEAnalogProgressBar, 134  
 radix, 137  
 Scientific, 133  
 Scientist, 134  
 separator, 137  
 Separators, 133  
 setManagedVisible, 134  
 Space, 134  
 styleSheet, 137  
 Time, 133  
 trailingZeros, 138  
 Underscore, 134  
 UnsignedInteger, 133  
 useDbDisplayLimits, 138  
 useDbPrecision, 138  
 User, 134  
 userLevelEnabled, 138  
 userLevelEngineerStyle, 138  
 UserLevels, 134

userLevelScientistStyle, 138  
userLevelUserStyle, 139  
userLevelVisibility, 139  
value, 139  
variable, 139  
variableAsToolTip, 139  
variableSubstitutions, 139  
visible, 140  
WhenInAlarm, 133

QEAnalogSlider, 141  
allowDrop, 145  
Always, 143  
arrayIndex, 145  
autoScale, 145  
axisAlarmColours, 145  
continuousWrite, 145  
dbValueChanged, 144  
defaultStyle, 145  
displayAlarmState, 145  
displayAlarmStateOption, 145  
DisplayAlarmStateOptions, 143  
Engineer, 144  
int, 146  
Never, 143  
QEAnalogSlider, 144  
Scientist, 144  
setManagedVisible, 144  
styleSheet, 146  
User, 144  
userLevelEnabled, 146  
userLevelEngineerStyle, 146  
UserLevels, 143  
userLevelScientistStyle, 146  
userLevelUserStyle, 147  
userLevelVisibility, 147  
variable, 147  
variableAsToolTip, 147  
variableSubstitutions, 147  
visible, 147  
WhenInAlarm, 143

QEAnalogSliderManager, 149

QEBitStatus, 150  
allowDrop, 153  
Always, 152  
arrayIndex, 153  
dbConnectionChanged, 152  
dbValueChanged, 152  
defaultStyle, 153  
displayAlarmState, 153  
displayAlarmStateOption, 153

DisplayAlarmStateOptions, 152  
Engineer, 152  
int, 153  
Never, 152  
Scientist, 152  
setManagedVisible, 152  
styleSheet, 154  
User, 152  
userLevelEnabled, 154  
userLevelEngineerStyle, 154  
UserLevels, 152  
userLevelScientistStyle, 154  
userLevelUserStyle, 154  
userLevelVisibility, 155  
variable, 155  
variableAsToolTip, 155  
variableSubstitutions, 155  
visible, 155  
WhenInAlarm, 152

QEByteArray, 156

QECheckBox, 157  
addUnits, 165  
alignment, 165  
allowDrop, 165  
Always, 162  
Append, 161  
arguments, 165  
arrayAction, 166  
ArrayActions, 161  
arrayIndex, 166  
Ascii, 161  
Automatic, 163  
clickCheckedText, 166  
clicked, 164  
clickText, 166  
Comma, 163  
confirmAction, 166  
confirmText, 167  
creationOption, 167  
CreationOptionNames, 161  
customisationName, 167  
dbValueChanged, 164  
Default, 162  
defaultStyle, 167  
disabledRecordPolicy, 167  
displayAlarmState, 167  
displayAlarmStateOption, 168  
DisplayAlarmStateOptions, 162  
DockBottom, 162  
DockBottomTabbed, 162

DockFloating, 162  
DockLeft, 162  
DockLeftTabbed, 162  
DockRight, 162  
DockRightTabbed, 162  
DockTop, 162  
DockTopTabbed, 162  
Engineer, 164  
Fixed, 163  
Floating, 162  
format, 168  
Formats, 162  
guiFile, 168  
Icon, 163  
Index, 161  
int, 168  
Integer, 162  
labelText, 168  
leadingZero, 169  
LocalEnumeration, 162  
localEnumeration, 169  
LogOutput, 163  
Never, 162  
NewTab, 161  
NewWindow, 162  
None, 163  
NoSeparator, 163  
notation, 169  
Notations, 162  
Open, 161  
password, 169  
 pixmap0, 170  
 pixmap1, 170  
 pixmap2, 170  
 pixmap3, 170  
 pixmap4, 170  
 pixmap5, 170  
 pixmap6, 170  
 pixmap7, 170  
precision, 170  
pressed, 164  
pressText, 171  
prioritySubstitutions, 171  
program, 171  
programStartupOption, 171  
ProgramStartupOptionNames, 163  
QECheckBox, 164  
radix, 171  
released, 165  
releaseText, 171  
requestAction, 165  
Scientific, 163  
Scientist, 164  
separator, 171  
Separators, 163  
setManagedVisible, 165  
Space, 163  
State, 164  
StdOutput, 163  
styleSheet, 172  
subscribe, 172  
Terminal, 163  
Text, 163  
TextAndIcon, 164  
Time, 162  
trailingZeros, 172  
Underscore, 163  
UnsignedInteger, 162  
updateOption, 172  
UpdateOptions, 163  
useDbPrecision, 172  
User, 164  
userLevelEnabled, 172  
userLevelEngineerStyle, 172  
UserLevels, 164  
userLevelScientistStyle, 173  
userLevelUserStyle, 173  
userLevelVisibility, 173  
variable, 173  
variableAsToolTip, 173  
variableSubstitutions, 173  
visible, 173  
WhenInAlarm, 162  
writeOnClick, 174  
writeOnPress, 174  
writeOnRelease, 174  
QECheckBoxManager, 175  
QEComboBox, 176  
allowDrop, 179  
allowFocusUpdate, 179  
Always, 178  
arrayIndex, 179  
dbValueChanged, 179  
defaultStyle, 179  
displayAlarmState, 180  
displayAlarmStateOption, 180  
DisplayAlarmStateOptions, 178  
Engineer, 178  
int, 180  
localEnumeration, 180

Never, 178  
Scientist, 178  
setManagedVisible, 179  
styleSheet, 180  
subscribe, 180  
useDbEnumerations, 179  
User, 178  
userLevelEnabled, 180  
userLevelEngineerStyle, 181  
UserLevels, 178  
userLevelScientistStyle, 181  
userLevelUserStyle, 181  
userLevelVisibility, 181  
variable, 181  
variableAsToolTip, 182  
variableSubstitutions, 182  
visible, 182  
WhenInAlarm, 178  
writeOnChange, 179

QEConfiguredLayout, 183  
allowDrop, 186  
Always, 185  
defaultStyle, 186  
displayAlarmState, 186  
displayAlarmStateOption, 186  
DisplayAlarmStateOptions, 185  
Engineer, 185  
int, 186  
Never, 185  
Scientist, 185  
setManagedVisible, 186  
styleSheet, 187  
User, 185  
userLevelEnabled, 187  
userLevelEngineerStyle, 187  
UserLevels, 185  
userLevelScientistStyle, 187  
userLevelUserStyle, 187  
userLevelVisibility, 188  
variableAsToolTip, 188  
visible, 188  
WhenInAlarm, 185

QEConfiguredLayoutManager, 189

QEFileBrowser, 190  
allowDrop, 194  
Always, 193  
defaultStyle, 194  
displayAlarmState, 194  
displayAlarmStateOption, 194  
DisplayAlarmStateOptions, 193

Engineer, 193  
fileFilter, 194  
int, 194  
Never, 193  
Scientist, 193  
selected, 193  
setManagedVisible, 193  
styleSheet, 195  
User, 193  
userLevelEnabled, 195  
userLevelEngineerStyle, 195  
UserLevels, 193  
userLevelScientistStyle, 195  
userLevelUserStyle, 195  
userLevelVisibility, 196  
variable, 196  
variableAsToolTip, 196  
variableSubstitutions, 196  
visible, 196  
WhenInAlarm, 193

QEFileDialog, 197  
allowDrop, 200  
Always, 199  
dbValueChanged, 200  
defaultStyle, 200  
displayAlarmState, 200  
displayAlarmStateOption, 200  
DisplayAlarmStateOptions, 199  
Engineer, 199  
int, 200  
Never, 199  
QEFileDialog, 199  
Scientist, 199  
setManagedVisible, 200  
styleSheet, 201  
User, 199  
userLevelEnabled, 201  
userLevelEngineerStyle, 201  
UserLevels, 199  
userLevelScientistStyle, 201  
userLevelUserStyle, 201  
userLevelVisibility, 202  
variable, 202  
variableAsToolTip, 202  
variableSubstitutions, 202  
visible, 202  
WhenInAlarm, 199

QEFileDialogManager, 203

QEFloating, 204

QEFloatingArray, 205

QEFloatingFormatting, 206  
QEForm, 207  
    allowDrop, 210  
    displayAlarmStateOption, 210  
    handleGuiLaunchRequests, 209  
    int, 210  
    messageFormFilter, 210  
    messageSourceFilter, 210  
    resizeContents, 209  
    uiFile, 211  
    variableAsToolTip, 211  
    variableSubstitutions, 211  
QEFormGrid, 212  
    allowDrop, 214  
    colMacroPrefix, 214  
    colNumberOffset, 214  
    colNumberWidth, 214  
    colStrings, 214  
    columns, 214  
    displayAlarmStateOption, 214  
    gridOrder, 214  
    GridOrders, 213  
    margin, 215  
    number, 215  
    QEFormGrid, 214  
    rowMacroPrefix, 215  
    rowNumberOffset, 215  
    rowNumberWidth, 215  
    rowStrings, 215  
    slotMacroPrefix, 215  
    slotNumberOffset, 215  
    slotNumberWidth, 215  
    slotStrings, 215  
    spacing, 216  
    uiFile, 216  
    variableAsToolTip, 216  
QEFormGridManager, 217  
QEFrame, 218  
    allowDrop, 221  
    Always, 220  
    defaultStyle, 221  
    displayAlarmState, 221  
    displayAlarmStateOption, 221  
    DisplayAlarmStateOptions, 220  
    Engineer, 220  
    int, 221  
    Never, 220  
     pixmap, 221  
     pixmap0, 221  
     pixmap1, 222  
     pixmap2, 222  
     pixmap3, 222  
     pixmap4, 222  
     pixmap5, 222  
     pixmap6, 222  
     pixmap7, 222  
     scaledContents, 222  
    Scientist, 220  
    setManagedVisible, 220  
    styleSheet, 222  
    User, 220  
    userLevelEnabled, 223  
    userLevelEngineerStyle, 223  
    UserLevels, 220  
    userLevelScientistStyle, 223  
    userLevelUserStyle, 223  
    userLevelVisibility, 223  
    variableAsToolTip, 224  
    visible, 224  
    WhenInAlarm, 220  
QEGeneralEdit, 225  
    arrayIndex, 226  
QEGeneralEdit, 226  
    variable, 226  
    variableSubstitutions, 226  
QEGeneralEditManager, 227  
QEGenericButton, 228  
QEGenericEdit, 231  
    allowDrop, 236  
    Always, 233  
    arrayIndex, 236  
    confirmWrite, 236  
    defaultStyle, 236  
    displayAlarmState, 236  
    displayAlarmStateOption, 236  
    DisplayAlarmStateOptions, 233  
    Engineer, 234  
    getConfirmWrite, 234  
    getSubscribe, 234  
    getWriteOnEnter, 234  
    getWriteOnFinish, 234  
    getWriteOnLoseFocus, 235  
    int, 237  
    Never, 233  
    QEGenericEdit, 234  
    Scientist, 234  
    setAllowFocusUpdate, 235  
    setConfirmWrite, 235  
    setManagedVisible, 235  
    setSubscribe, 235

setWriteOnEnter, 235  
setWriteOnFinish, 235  
setWriteOnLoseFocus, 235  
styleSheet, 237  
subscribe, 237  
User, 234  
userLevelEnabled, 237  
userLevelEngineerStyle, 237  
UserLevels, 234  
userLevelScientistStyle, 237  
userLevelUserStyle, 238  
userLevelVisibility, 238  
variable, 238  
variableAsToolTip, 238  
variableSubstitutions, 238  
visible, 238  
WhenInAlarm, 234  
writeOnEnter, 238  
writeOnFinish, 239  
writeOnLoseFocus, 239

QEGroupBox, 240  
allowDrop, 242  
Always, 241  
defaultStyle, 242  
displayAlarmState, 242  
displayAlarmStateOption, 242  
DisplayAlarmStateOptions, 241  
Engineer, 241  
int, 242  
Never, 241  
Scientist, 241  
setManagedVisible, 242  
styleSheet, 242  
substitutedTitle, 243  
textSubstitutions, 243  
User, 241  
userLevelEnabled, 243  
userLevelEngineerStyle, 243  
UserLevels, 241  
userLevelScientistStyle, 243  
userLevelUserStyle, 243  
userLevelVisibility, 244  
variableAsToolTip, 244  
visible, 244  
WhenInAlarm, 241

QEHistogram, 245  
    orientation, 246  
QEHistogramManager, 247  
QEImage, 248  
    allowDrop, 282

    Always, 278  
    areaColor, 282  
    arguments1, 282  
    arguments2, 282  
    autoBrightnessContrast, 282  
    Bayer, 279  
    BayerBG, 279  
    BayerGB, 279  
    BayerGR, 279  
    BayerRG, 279  
    beamColor, 283  
    beamXVariable, 283  
    beamYVariable, 283  
    bitDepthVariable, 283  
    BOUNDING\_RECTANGLE, 278  
    BoundingRectangle, 278  
    briefInfoArea, 283  
    CenterAndSize, 278  
    clippingHighVariable, 283  
    clippingLowVariable, 283  
    clippingOnOffVariable, 283  
    contrastReversal, 283  
    dataTypeVariable, 284  
    dbValueChanged, 281  
    defaultStyle, 284  
    dimension1Variable, 284  
    dimension2Variable, 284  
    dimension3Variable, 284  
    dimensionsVariable, 284  
    displayAlarmState, 284  
    displayAlarmStateOption, 285  
    DisplayAlarmStateOptions, 278  
    displayArea1Selection, 285  
    displayArea2Selection, 285  
    displayArea3Selection, 285  
    displayArea4Selection, 285  
    displayBeamSelection, 285  
    displayButtonBar, 282  
    displayCursorPixelInfo, 285  
    displayEllipse, 285  
    displayHozSlice1Selection, 286  
    displayHozSlice2Selection, 286  
    displayHozSlice3Selection, 286  
    displayHozSlice4Selection, 286  
    displayHozSlice5Selection, 286  
    displayProfileSelection, 286  
    displayTargetSelection, 286  
    displayVertSlice1Selection, 286  
    displayVertSlice2Selection, 286  
    displayVertSlice3Selection, 287

displayVertSlice4Selection, 287  
displayVertSlice5Selection, 287  
DottedFullCrosshair, 281  
ellipseColor, 287  
ellipseHVariable, 287  
EllipseVariableDefinitions, 278  
ellipseVariableDefinitions, 278  
ellipseWVariable, 287  
ellipseXVariable, 287  
ellipseYVariable, 287  
enableArea1Selection, 287  
enableArea2Selection, 288  
enableArea3Selection, 288  
enableArea4Selection, 288  
enableBeamSelection, 288  
enableHozSlice1Selection, 288  
enableHozSlice2Selection, 288  
enableHozSlice3Selection, 288  
enableHozSlice4Selection, 288  
enableHozSlice5Selection, 289  
enableProfileSelection, 289  
enableTargetSelection, 289  
enableVertSlice1Selection, 289  
enableVertSlice2Selection, 289  
enableVertSlice3Selection, 289  
enableVertSlice4Selection, 289  
enableVertSlice5Selection, 290  
Engineer, 281  
externalControls, 290  
Fit, 279  
formatOption, 290  
FormatOptions, 278  
formatVariable, 290  
heightVariable, 290  
horizontalFlip, 290  
hozSlice1Color, 290  
hozSlice2Color, 290  
hozSlice3Color, 290  
hozSlice4Color, 291  
hozSlice5Color, 291  
imageVariable, 291  
initialHosScrollPos, 291  
initialVertScrollPos, 282  
int, 291  
lineProfileArrayVariable, 291  
lineProfileThicknessVariable, 291  
lineProfileX1Variable, 291  
lineProfileX2Variable, 292  
lineProfileY1Variable, 292  
lineProfileY2Variable, 292  
logBrightness, 292  
LogOutput, 279  
Mono, 279  
Never, 278  
None, 279  
NoRotation, 280  
profileColor, 292  
profileHoz1ThicknessVariable, 292  
profileHoz1Variable, 292  
profileHoz2ThicknessVariable, 292  
profileHoz2Variable, 292  
profileHoz3ThicknessVariable, 293  
profileHoz3Variable, 293  
profileHoz4ThicknessVariable, 293  
profileHoz4Variable, 293  
profileHoz5ThicknessVariable, 293  
profileHoz5Variable, 293  
profileHozArrayVariable, 293  
profileVert1ThicknessVariable, 293  
profileVert1Variable, 294  
profileVert2ThicknessVariable, 294  
profileVert2Variable, 294  
profileVert3ThicknessVariable, 294  
profileVert3Variable, 294  
profileVert4ThicknessVariable, 294  
profileVert4Variable, 294  
profileVert5ThicknessVariable, 294  
profileVert5Variable, 295  
profileVertArrayVariable, 295  
program1, 295  
program2, 295  
programStartupOption1, 295  
programStartupOption2, 295  
ProgramStartupOptionNames, 279  
QEImage, 281  
regionOfInterest1HVariable, 295  
regionOfInterest1WVariable, 296  
regionOfInterest1XVariable, 296  
regionOfInterest1YVariable, 296  
regionOfInterest2HVariable, 296  
regionOfInterest2WVariable, 296  
regionOfInterest2XVariable, 296  
regionOfInterest2YVariable, 296  
regionOfInterest3HVariable, 296  
regionOfInterest3WVariable, 297  
regionOfInterest3XVariable, 297  
regionOfInterest3YVariable, 297  
regionOfInterest4HVariable, 297  
regionOfInterest4WVariable, 297  
regionOfInterest4XVariable, 297

regionOfInterest4YVariable, 297  
RESIZE\_OPTION\_FIT, 280  
RESIZE\_OPTION\_ZOOM, 280  
resizeOption, 297  
ResizeOptions, 279  
resizeOptions, 279  
rgb1, 279  
rgb2, 279  
rgb3, 279  
Rotate180, 280  
Rotate90Left, 280  
Rotate90Right, 280  
rotation, 298  
RotationOptions, 280  
Scientist, 281  
selectOptions, 280  
setImageFile, 281  
setManagedVisible, 282  
showTime, 298  
SO\_AREA4, 280  
SO\_BEAM, 280  
SO\_HSLICE1, 280  
SO\_HSLICE2, 280  
SO\_HSLICE3, 280  
SO\_HSLICE4, 280  
SO\_HSLICE5, 280  
SO\_NONE, 280  
SO\_PANNING, 280  
SO\_PROFILE, 280  
SO\_TARGET, 280  
SO\_VSLICE1, 280  
SO\_VSLICE2, 280  
SO\_VSLICE3, 280  
SO\_VSLICE4, 280  
SO\_VSLICE5, 280  
SolidSmallCrosshair, 281  
StdOutput, 279  
styleSheet, 298  
targetColor, 298  
TargetOptions, 280  
targetTriggerVariable, 298  
targetXVariable, 298  
targetYVariable, 298  
Terminal, 279  
timeColor, 298  
URL, 298  
useFalseColour, 299  
User, 281  
userLevelEnabled, 299  
userLevelEngineerStyle, 299  
UserLevels, 281  
userLevelScientistStyle, 299  
userLevelUserStyle, 299  
userLevelVisibility, 299  
variableAsToolTip, 300  
variableSubstitutions, 300  
verticalFlip, 300  
vertSlice1Color, 300  
vertSlice2Color, 300  
vertSlice3Color, 300  
vertSlice4Color, 300  
vertSlice5Color, 300  
visible, 301  
WhenInAlarm, 278  
widthVariable, 301  
yuv422, 279  
yuv444, 279  
Zoom, 279  
QEImageMarkupThickness, 302  
QEImageOptionsDialog, 303  
QEInteger, 304  
QEIntegerArray, 305  
QEIntegerFormatting, 306  
formatInteger, 306  
formatIntegerArray, 306  
formatValue, 307  
QELabel, 308  
addUnits, 314  
allowDrop, 314  
Always, 312  
Append, 311  
arrayAction, 314  
ArrayActions, 311  
arrayIndex, 314  
Ascii, 311  
Automatic, 312  
Comma, 312  
dbValueChanged, 314  
Default, 312  
defaultStyle, 315  
displayAlarmState, 315  
displayAlarmStateOption, 315  
DisplayAlarmStateOptions, 311  
Engineer, 313  
Fixed, 312  
Floating, 312  
format, 315  
Formats, 312  
Index, 311  
int, 315

Integer, 312  
leadingZero, 315  
LocalEnumeration, 312  
localEnumeration, 315  
Never, 312  
NoSeparator, 312  
notation, 316  
Notations, 312  
Picture, 313  
pixmap0, 316  
pixmap1, 316  
pixmap2, 316  
pixmap3, 317  
pixmap4, 317  
pixmap5, 317  
pixmap6, 317  
pixmap7, 317  
precision, 317  
QELabel, 313  
radix, 317  
Scientific, 312  
Scientist, 313  
separator, 317  
Separators, 312  
setManagedVisible, 314  
Space, 313  
styleSheet, 317  
Text, 313  
Time, 312  
trailingZeros, 318  
Underscore, 313  
UnsignedInteger, 312  
UPDATE\_PIXMAP, 313  
UPDATE\_TEXT, 313  
updateOption, 318  
UpdateOptions, 313  
updateOptions, 313  
useDbPrecision, 318  
User, 313  
userLevelEnabled, 318  
userLevelEngineerStyle, 318  
UserLevels, 313  
userLevelScientistStyle, 318  
userLevelUserStyle, 319  
userLevelVisibility, 319  
variable, 319  
variableAsToolTip, 319  
variableSubstitutions, 319  
visible, 319  
WhenInAlarm, 312  
QELineEdit, 321  
addUnits, 324  
Append, 323  
arrayAction, 324  
ArrayActions, 323  
Ascii, 323  
Automatic, 323  
Comma, 324  
dbValueChanged, 324  
Default, 323  
Fixed, 323  
Floating, 323  
format, 325  
Formats, 323  
Index, 323  
Integer, 323  
leadingZero, 325  
LocalEnumeration, 323  
localEnumeration, 325  
NoSeparator, 324  
notation, 325  
Notations, 323  
precision, 326  
QELineEdit, 324  
radix, 326  
Scientific, 323  
separator, 326  
Separators, 323  
Space, 324  
Time, 323  
trailingZeros, 326  
Underscore, 324  
UnsignedInteger, 323  
useDbPrecision, 326  
QELineEditManager, 327  
QELink, 328  
QELocalEnumeration, 330  
getLocalEnumeration, 331  
isDefined, 331  
QELocalEnumeration, 330  
setLocalEnumeration, 331  
text.ToDouble, 331  
textToInt, 332  
textToValue, 332  
valueToText, 332  
QELog, 333  
allowDrop, 336  
Always, 335  
defaultStyle, 336  
displayAlarmState, 336

displayAlarmStateOption, 336  
DisplayAlarmStateOptions, 335  
Engineer, 336  
int, 337  
Never, 335  
Scientist, 336  
setManagedVisible, 336  
styleSheet, 337  
User, 336  
userLevelEnabled, 337  
userLevelEngineerStyle, 337  
UserLevels, 335  
userLevelScientistStyle, 337  
userLevelUserStyle, 337  
userLevelVisibility, 338  
variableAsToolTip, 338  
visible, 338  
WhenInAlarm, 335  
QELogin, 339  
QELoginDialog, 340  
QEMenuButton, 341  
allowDrop, 342  
defaultSubstitutions, 342  
displayAlarmStateOption, 342  
variableAsToolTip, 342  
QEMenuButtonData, 343  
QEMenuButtonItem, 344  
QEMenuButtonManager, 345  
QEMenuButtonModel, 346  
QEMenuButtonSetupDialog, 348  
QEMenuButtonTaskMenu, 349  
QEMenuButtonTaskMenuFactory, 350  
QENumericEdit, 351  
addUnits, 354  
alignment, 354  
allowFocusUpdate, 354  
arrayIndex, 354  
autoScale, 354  
cleanText, 354  
confirmWrite, 354  
dbConnectionChanged, 353  
dbValueChanged, 353  
frame, 354  
leadingZeros, 354  
maximum, 355  
minimum, 355  
notation, 355  
precision, 355  
QENumericEdit, 353  
radix, 355  
separator, 355  
variable, 355  
variableSubstitutions, 355  
writeOnChange, 356  
writeOnEnter, 356  
writeOnFinish, 356  
writeOnLoseFocus, 356  
QENumericEditManager, 357  
QEPeriodic, 358  
allowDrop, 363  
Always, 362  
dbElementChanged, 362  
dbValueChanged, 362  
displayAlarmState, 363  
displayAlarmStateOption, 363  
DisplayAlarmStateOptions, 362  
Engineer, 362  
int, 363  
Never, 362  
readbackLabelVariable1, 363  
readbackLabelVariable2, 363  
Scientist, 362  
subscribe, 364  
User, 362  
userLevelEnabled, 364  
userLevelEngineerStyle, 364  
UserLevels, 362  
userLevelScientistStyle, 364  
userLevelUserStyle, 364  
userLevelVisibility, 364  
variableAsToolTip, 365  
variableSubstitutions, 365  
visible, 365  
WhenInAlarm, 362  
writeButtonVariable1, 365  
writeButtonVariable2, 365  
QEPeriodic::elementInfoStruct, 45  
QEPeriodic::userInfoStructArray, 575  
QEPeriodicComponentData, 366  
QEPeriodicTaskMenu, 367  
QEPeriodicTaskMenuFactory, 368  
QEpicsPV, 369  
QEPlot, 370  
allowDrop, 375  
Always, 374  
dbValueChanged, 374  
defaultStyle, 375  
displayAlarmState, 375  
displayAlarmStateOption, 375  
DisplayAlarmStateOptions, 374

Engineer, 374  
int, 375  
Never, 374  
Scientist, 374  
setManagedVisible, 374  
styleSheet, 375  
User, 374  
userLevelEnabled, 375  
userLevelEngineerStyle, 376  
UserLevels, 374  
userLevelScientistStyle, 376  
userLevelUserStyle, 376  
userLevelVisibility, 376  
variable1, 376  
variable2, 377  
variable3, 377  
variable4, 377  
variableAsToolTip, 377  
variableSubstitutions, 377  
visible, 377  
WhenInAlarm, 374  
QEPlotter, 378  
aliasNames, 382  
variableSubstitutions, 382  
QEPlotterItemDialog, 383  
QEPlotterManager, 384  
QEPlotterMenu, 385  
QEPlotterNames, 386  
QEPlotterPushButtonSpecifications, 388  
QEPlotterState, 389  
QEPlotterStateList, 390  
QEPlotterToolBar, 391  
QEPushButton, 392  
    addUnits, 400  
    alignment, 400  
    allowDrop, 400  
    altReadbackVariable, 400  
    Always, 397  
    Append, 396  
    arguments, 400  
    arrayAction, 400  
    ArrayActions, 396  
    arrayIndex, 401  
    Ascii, 396  
    Automatic, 398  
    clickCheckedText, 401  
    clicked, 399  
    clickText, 401  
    confirmAction, 401  
    confirmText, 401  
creationOption, 402  
CreationOptionNames, 396  
customisationName, 402  
dbValueChanged, 399  
Default, 397  
defaultStyle, 402  
disabledRecordPolicy, 402  
displayAlarmState, 402  
displayAlarmStateOption, 403  
DisplayAlarmStateOptions, 397  
DockBottom, 397  
DockBottomTabbed, 397  
DockFloating, 397  
DockLeft, 397  
DockLeftTabbed, 397  
DockRight, 397  
DockRightTabbed, 397  
DockTop, 396  
DockTopTabbed, 397  
Engineer, 399  
Fixed, 398  
Floating, 397  
format, 403  
Formats, 397  
guiFile, 403  
Icon, 398  
Index, 396  
int, 403  
Integer, 397  
labelText, 403  
leadingZero, 404  
LocalEnumeration, 397  
localEnumeration, 404  
LogOutput, 398  
Never, 397  
NewTab, 396  
NewWindow, 396  
None, 398  
notation, 404  
Notations, 397  
Open, 396  
password, 404  
 pixmap0, 404  
 pixmap1, 405  
 pixmap2, 405  
 pixmap3, 405  
 pixmap4, 405  
 pixmap5, 405  
 pixmap6, 405  
 pixmap7, 405

precision, 405  
pressed, 399  
pressText, 405  
prioritySubstitutions, 406  
program, 406  
programStartupOption, 406  
ProgramStartupOptionNames, 398  
QEPushButton, 399  
released, 399  
releaseText, 406  
requestAction, 399  
Scientific, 398  
Scientist, 398  
setManagedVisible, 400  
State, 398  
StdOutput, 398  
styleSheet, 406  
subscribe, 406  
Terminal, 398  
Text, 398  
TextAndIcon, 398  
Time, 397  
trailingZeros, 406  
UnsignedInteger, 397  
updateOption, 407  
UpdateOptions, 398  
useDbPrecision, 407  
User, 398  
userLevelEnabled, 407  
userLevelEngineerStyle, 407  
UserLevels, 398  
userLevelScientistStyle, 407  
userLevelUserStyle, 407  
userLevelVisibility, 408  
variable, 408  
variableAsToolTip, 408  
variableSubstitutions, 408  
visible, 408  
WhenInAlarm, 397  
writeOnClick, 408  
writeOnPress, 408  
writeOnRelease, 409  
QE PvFrame, 410  
arrayIndex, 411  
dbConnectionChanged, 410  
dbValueChanged, 410  
variable, 411  
variableSubstitutions, 411  
QE PvFrameManager, 412  
QE PvLoadSave, 413  
configurationFileLeft, 414  
confirmAction, 414  
defaultDir, 414  
defaultSubstitutions, 414  
QE PvLoadSave, 414  
QE PvLoadSaveCommon, 416  
QE PvLoadSaveCompare, 417  
QE PvLoadSaveGroup, 418  
QE PvLoadSaveGroupNameDialog, 419  
QE PvLoadSaveItem, 420  
QE PvLoadSaveLeaf, 422  
QE PvLoadSaveManager, 423  
QE PvLoadSaveModel, 424  
QE PvLoadSaveTimeDialog, 426  
QE PvLoadSaveUtilities, 427  
QE PvLoadSaveValueEditDialog, 428  
QE PVNameLists, 429  
QE PvProperties, 430  
variable, 431  
variableSubstitutions, 431  
QE PvPropertiesManager, 432  
QE RadioButton, 433  
addUnits, 441  
alignment, 441  
allowDrop, 441  
Always, 438  
Append, 437  
arguments, 441  
arrayAction, 442  
ArrayActions, 437  
arrayIndex, 442  
Ascii, 437  
Automatic, 439  
clickCheckedText, 442  
clicked, 440  
clickText, 442  
Comma, 439  
confirmAction, 442  
confirmText, 443  
creationOption, 443  
CreationOptionNames, 437  
customisationName, 443  
dbValueChanged, 440  
Default, 438  
defaultStyle, 443  
disabledRecordPolicy, 443  
displayAlarmState, 443  
displayAlarmStateOption, 444  
DisplayAlarmStateOptions, 438  
DockBottom, 438

DockBottomTabbed, 438  
DockFloating, 438  
DockLeft, 438  
DockLeftTabbed, 438  
DockRight, 438  
DockRightTabbed, 438  
DockTop, 438  
DockTopTabbed, 438  
Engineer, 440  
Fixed, 439  
Floating, 438  
format, 444  
Formats, 438  
guiFile, 444  
Icon, 439  
Index, 437  
int, 444  
Integer, 438  
labelText, 444  
leadingZero, 445  
LocalEnumeration, 438  
localEnumeration, 445  
LogOutput, 439  
Never, 438  
NewTab, 437  
NewWindow, 438  
None, 439  
NoSeparator, 439  
notation, 445  
Notations, 438  
Open, 437  
password, 445  
pixmap0, 446  
pixmap1, 446  
pixmap2, 446  
pixmap3, 446  
pixmap4, 446  
pixmap5, 446  
pixmap6, 446  
pixmap7, 446  
precision, 446  
pressed, 440  
pressText, 447  
prioritySubstitutions, 447  
program, 447  
programStartupOption, 447  
ProgramStartupOptionNames, 439  
QERadioButton, 440  
radix, 447  
released, 441  
releaseText, 447  
requestAction, 441  
Scientific, 439  
Scientist, 440  
separator, 447  
Separators, 439  
setManagedVisible, 441  
Space, 439  
State, 440  
StdOutput, 439  
styleSheet, 448  
subscribe, 448  
Terminal, 439  
Text, 439  
TextAndIcon, 440  
Time, 438  
trailingZeros, 448  
Underscore, 439  
UnsignedInteger, 438  
updateOption, 448  
UpdateOptions, 439  
useDbPrecision, 448  
User, 440  
userLevelEnabled, 448  
userLevelEngineerStyle, 448  
UserLevels, 440  
userLevelScientistStyle, 449  
userLevelUserStyle, 449  
userLevelVisibility, 449  
variable, 449  
variableAsToolTip, 449  
variableSubstitutions, 449  
visible, 450  
WhenInAlarm, 438  
writeOnClick, 450  
writeOnPress, 450  
writeOnRelease, 450  
QERadioGroup, 451  
arrayIndex, 453  
buttonOrder, 453  
buttonStyle, 453  
columns, 453  
dbConnectionChanged, 453  
dbValueChanged, 453  
localEnumeration, 453  
QERadioGroup, 452  
spacing, 454  
substitutedTitle, 454  
useDbEnumerations, 454  
variable, 454

variableSubstitutions, 454  
QERadioGroupManager, 455  
QERecipe, 456  
QERecordSpec, 458  
QERecordSpecList, 459  
QEResizableFrame, 460  
    allowedMaximum, 461  
    allowedMinimum, 461  
    grabbingEdge, 461  
    setWidget, 461  
QEResizableFrameManager, 462  
QEScalarHistogram, 463  
    Auto, 467  
    Manual, 467  
    OperationalRange, 467  
    ScaleModes, 467  
    variableSubstitutions, 467  
QEScalarHistogramManager, 468  
QEScratchPad, 469  
QEScratchPadManager, 471  
QEScratchPadMenu, 472  
QEScript, 473  
    allowDrop, 478  
    Always, 478  
    defaultStyle, 478  
    displayAlarmState, 478  
    displayAlarmStateOption, 479  
    DisplayAlarmStateOptions, 478  
    Engineer, 478  
    int, 479  
    Never, 478  
    Scientist, 478  
    setManagedVisible, 478  
    styleSheet, 479  
    User, 478  
    userLevelEnabled, 479  
    userLevelEngineerStyle, 479  
    UserLevels, 478  
    userLevelScientistStyle, 480  
    userLevelUserStyle, 480  
    userLevelVisibility, 480  
    variableAsToolTip, 480  
    visible, 480  
    WhenInAlarm, 478  
QEShape, 482  
    allowDrop, 489  
    Always, 487  
    animation1, 489  
    animation2, 489  
    animation3, 489  
        animation4, 490  
        animation5, 490  
        animation6, 490  
        animationOptions, 487  
        color1, 490  
        color10, 490  
        color2, 490  
        color3, 490  
        color4, 490  
        color5, 490  
        color6, 491  
        color7, 491  
        color8, 491  
        color9, 491  
        dbValueChanged1, 488  
        dbValueChanged2, 488  
        dbValueChanged3, 488  
        dbValueChanged4, 488  
        dbValueChanged5, 489  
        dbValueChanged6, 489  
        defaultStyle, 491  
        displayAlarmState, 491  
        displayAlarmStateOption, 491  
        DisplayAlarmStateOptions, 487  
        Engineer, 488  
        int, 492  
        Never, 487  
        offset1, 492  
        offset2, 492  
        offset3, 492  
        offset4, 492  
        offset5, 492  
        offset6, 492  
        point1, 492  
        point10, 493  
        point2, 493  
        point3, 493  
        point4, 493  
        point5, 493  
        point6, 493  
        point7, 493  
        point8, 493  
        point9, 494  
        QEShape, 488  
        scale2, 494  
        scale3, 494  
        scale4, 494  
        scale5, 494  
        scale6, 494  
        Scientist, 488

setManagedVisible, 489  
shapeOptions, 487  
styleSheet, 494  
User, 488  
userLevelEnabled, 494  
userLevelEngineerStyle, 495  
UserLevels, 487  
userLevelScientistStyle, 495  
userLevelUserStyle, 495  
userLevelVisibility, 495  
variable1, 495  
variable2, 496  
variable3, 496  
variable4, 496  
variable5, 496  
variable6, 496  
variableAsToolTip, 496  
variableSubstitutions, 496  
visible, 497  
WhenInAlarm, 487  
QE.SimpleShape, 498  
addUnits, 501  
allowDrop, 501  
Always, 501  
arrayIndex, 502  
dbConnectionChanged, 501  
dbValueChanged, 501  
defaultStyle, 502  
displayAlarmState, 502  
displayAlarmStateOption, 502  
DisplayAlarmStateOptions, 500  
Engineer, 501  
int, 502  
localEnumeration, 502  
Never, 501  
Scientist, 501  
setManagedVisible, 501  
styleSheet, 503  
User, 501  
userLevelEnabled, 503  
userLevelEngineerStyle, 503  
UserLevels, 501  
userLevelScientistStyle, 504  
userLevelUserStyle, 504  
userLevelVisibility, 504  
variable, 504  
variableAsToolTip, 504  
variableSubstitutions, 504  
visible, 505  
WhenInAlarm, 501  
QE.SimpleShapeManager, 506  
QE.Slider, 507  
allowDrop, 510  
allowFocusUpdate, 510  
Always, 509  
arrayIndex, 510  
dbValueChanged, 509  
defaultStyle, 510  
displayAlarmState, 510  
displayAlarmStateOption, 510  
DisplayAlarmStateOptions, 509  
Engineer, 509  
int, 511  
Never, 509  
Scientist, 509  
setManagedVisible, 509  
styleSheet, 511  
subscribe, 511  
User, 509  
userLevelEnabled, 511  
userLevelEngineerStyle, 511  
UserLevels, 509  
userLevelScientistStyle, 511  
userLevelUserStyle, 512  
userLevelVisibility, 512  
variable, 512  
variableAsToolTip, 512  
variableSubstitutions, 512  
visible, 512  
WhenInAlarm, 509  
writeOnChange, 510  
QE.SpinBox, 514  
allowDrop, 517  
allowFocusUpdate, 517  
Always, 516  
arrayIndex, 517  
dbValueChanged, 517  
defaultStyle, 517  
displayAlarmState, 517  
displayAlarmStateOption, 517  
DisplayAlarmStateOptions, 516  
Engineer, 516  
int, 518  
Never, 516  
Scientist, 516  
setManagedVisible, 517  
styleSheet, 518  
subscribe, 518  
User, 516  
userLevelEnabled, 518

userLevelEngineerStyle, 518  
UserLevels, 516  
userLevelScientistStyle, 518  
userLevelUserStyle, 519  
userLevelVisibility, 519  
variable, 519  
variableAsToolTip, 519  
variableSubstitutions, 519  
visible, 519  
WhenInAlarm, 516  
QString, 521  
QStringFormatting, 522  
    APPEND, 523  
    arrayActions, 523  
    ASCII, 523  
    FORMAT\_DEFAULT, 523  
    FORMAT\_FLOATING, 523  
    FORMAT\_INTEGER, 523  
    FORMAT\_LOCAL\_ENUMERATE, 523  
    FORMAT\_STRING, 523  
    FORMAT\_TIME, 523  
    FORMAT\_UNSIGNEDINTEGER, 523  
    formats, 523  
    INDEX, 523  
    NOTATION\_AUTOMATIC, 523  
    NOTATION\_FIXED, 523  
    NOTATION\_SCIENTIFIC, 523  
    notations, 523  
    SEPARATOR\_COMMA, 524  
    SEPARATOR\_NONE, 524  
    SEPARATOR\_SPACE, 524  
    SEPARATOR\_UNDERSCORE, 524  
    separators, 523  
QEStripChart, 525  
    variableSubstitutions, 527  
QEStripChartAdjustPVDialog, 528  
QEStripChartContextMenu, 529  
    QEStripChartContextMenu, 529  
QEStripChartDurationDialog, 530  
QEStripChartItem, 531  
QEStripChartNames, 533  
QEStripChartPushButtonSpecifications, 535  
QEStripChartRangeDialog, 536  
QEStripChartState, 537  
QEStripChartStateList, 538  
QEStripChartStatistics, 539  
QEStripChartTimeDialog, 540  
QEStripChartToolBar, 541  
QEStripChartToolBar::OwnTabWidget, 86  
QESubstitutedLabel, 543  
    labelText, 543  
    textSubstitutions, 543  
QETable, 544  
    colWidthMinimum, 547  
    dbValueChanged, 546  
    displayMaximum, 547  
    gridStyle, 547  
    orientation, 547  
    QETable, 546  
    showGrid, 547  
    titles, 547  
    variableSubstitutions, 547  
QETableManager, 548  
QEWaveformHistogram, 549  
    Auto, 550  
    Manual, 550  
    OperationalRange, 550  
    ScaleModes, 550  
QEWaveformHistogramManager, 551  
QNumericEdit, 552  
    alignment, 554  
    cleanText, 554  
    Fixed, 554  
    frame, 554  
    leadingZeros, 554  
    notation, 554  
    Notations, 554  
    precision, 555  
    prefix, 555  
    Scientific, 554  
    suffix, 555  
QNumericEditManager, 556  
QRadioButton, 557  
    ButtonOrders, 558  
    ButtonStyles, 558  
    QRadioButton, 558  
    Radio, 558  
    rowMajor, 558  
QRadioButtonManager, 559  
QSimpleShape, 560  
    edgeWidth, 563  
    FixedText, 563  
    flash0, 563  
    flashRate, 563  
    format, 563  
    percentSize, 563

PvText, 563  
 QSimpleShape, 563  
 semiCycles, 564  
 shape, 564  
 StateSet, 563  
 TextFormats, 563  
 QSimpleShapeManager, 565

Radio  
 QRadioGroup, 558

radix  
 QEAnalogProgressBar, 137  
 QECheckBox, 171  
 QELabel, 317  
 QELineEdit, 326  
 QENumericEdit, 355  
 QERadioButton, 447

readbackLabelVariable1  
 QEPeriodic, 363

readbackLabelVariable2  
 QEPeriodic, 363

recording, 567

regionOfInterest1HVariable  
 QEImage, 295

regionOfInterest1WVariable  
 QEImage, 296

regionOfInterest1XVariable  
 QEImage, 296

regionOfInterest1YVariable  
 QEImage, 296

regionOfInterest2HVariable  
 QEImage, 296

regionOfInterest2WVariable  
 QEImage, 296

regionOfInterest2XVariable  
 QEImage, 296

regionOfInterest2YVariable  
 QEImage, 296

regionOfInterest3HVariable  
 QEImage, 296

regionOfInterest3WVariable  
 QEImage, 297

regionOfInterest3XVariable  
 QEImage, 297

regionOfInterest3YVariable  
 QEImage, 297

regionOfInterest4HVariable  
 QEImage, 297

regionOfInterest4WVariable  
 QEImage, 297

regionOfInterest4XVariable  
 QEImage, 297

regionOfInterest4YVariable  
 QEImage, 297

released  
 QECheckBox, 165  
 QEPushButton, 399  
 QERadioButton, 441

releaseText  
 QECheckBox, 171  
 QEPushButton, 406  
 QERadioButton, 447

requestAction  
 QECheckBox, 165  
 QEPushButton, 399  
 QERadioButton, 441

RESIZE\_OPTION\_FIT  
 QEImage, 280

RESIZE\_OPTION\_ZOOM  
 QEImage, 280

resizeContents  
 QEForm, 209

resizeOption  
 QEImage, 297

ResizeOptions  
 QEImage, 279

resizeOptions  
 QEImage, 279

rgb1  
 QEImage, 279

rgb2  
 QEImage, 279

rgb3  
 QEImage, 279

Right\_To\_Left  
 QEAnalogIndicator, 126

Rotate180  
 QEImage, 280

Rotate90Left  
 QEImage, 280

Rotate90Right  
 QEImage, 280

rotation  
 QEImage, 298

ROTATION\_0  
 imageProperties, 68

ROTATION\_180  
 imageProperties, 68

ROTATION\_90\_LEFT  
 imageProperties, 68

ROTATION\_90\_RIGHT  
    imageProperties, 68  
RotationOptions  
    QEImage, 280  
rotationOptions  
    imageProperties, 68  
rowMacroPrefix  
    QEFormGrid, 215  
rowMajor  
    QRadioButton, 558  
rowNumberOffset  
    QEFormGrid, 215  
rowNumberWidth  
    QEFormGrid, 215  
rowStrings  
    QEFormGrid, 215

Scale  
    QEAnalogIndicator, 126  
scale2  
    QEShape, 494  
scale3  
    QEShape, 494  
scale4  
    QEShape, 494  
scale5  
    QEShape, 494  
scale6  
    QEShape, 494  
scaledContents  
    QEFrame, 222  
ScaleModes  
    QEScalarHistogram, 467  
    QEWaveformHistogram, 550  
Scientific  
    QEAnalogProgressBar, 133  
    QECheckBox, 163  
    QELabel, 312  
    QELineEdit, 323  
    QEPushButton, 398  
    QERadioButton, 439  
    QNumericEdit, 554  
Scientist  
    QEAbstractWidget, 118  
    QEAnalogProgressBar, 134  
    QEAnalogSlider, 144  
    QEBitStatus, 152  
    QECheckBox, 164  
    QEComboBox, 178  
    QEConfiguredLayout, 185

QEFileBrowser, 193  
QEFileImage, 199  
QEFrame, 220  
QEGenericEdit, 234  
QEGroupBox, 241  
QEImage, 281  
QELabel, 313  
QELog, 336  
QEPeriodic, 362  
QEPlot, 374  
QEPushButton, 398  
QERadioButton, 440  
QEScript, 478  
QEShape, 488  
QESimpleShape, 501  
QESlider, 509  
QESpinBox, 516  
screenSelectDialog, 569  
selected  
    QEFileBrowser, 193  
selectMenu, 570  
selectOptions  
    QEImage, 280  
semiCycles  
    QSimpleShape, 564  
separator  
    QEAnalogProgressBar, 137  
    QECheckBox, 171  
    QELabel, 317  
    QELineEdit, 326  
    QNumericEdit, 355  
    QERadioButton, 447  
SEPARATOR\_COMMA  
    QEStringFormatting, 524  
SEPARATOR\_NONE  
    QEStringFormatting, 524  
SEPARATOR\_SPACE  
    QEStringFormatting, 524  
SEPARATOR\_UNDERSCORE  
    QEStringFormatting, 524  
Separators  
    QEAnalogProgressBar, 133  
    QECheckBox, 163  
    QELabel, 312  
    QELineEdit, 323  
    QERadioButton, 439  
separators  
    QEStringFormatting, 523  
setAllowFocusUpdate  
    QEGenericEdit, 235

setConfirmWrite  
     QEGenericEdit, 235  
 setImageFile  
     QEImage, 281  
 setLocalEnumeration  
     QELocalEnumeration, 331  
 setManagedVisible  
     QEAbstractWidget, 119  
     QEAnalogProgressBar, 134  
     QEAnalogSlider, 144  
     QEBitStatus, 152  
     QECheckBox, 165  
     QEComboBox, 179  
     QEConfiguredLayout, 186  
     QEFileBrowser, 193  
     QEFileImage, 200  
     QEFrame, 220  
     QEGenericEdit, 235  
     QEGroupBox, 242  
     QEImage, 282  
     QELabel, 314  
     QELog, 336  
     QEPlot, 374  
     QEPushButton, 400  
     QERadioButton, 441  
     QEScript, 478  
     QEShape, 489  
     QESimpleShape, 501  
     QESlider, 509  
     QESpinBox, 517  
 setSubscribe  
     QEGenericEdit, 235  
 setWidget  
     QEResizeableFrame, 461  
 setWriteOnEnter  
     QEGenericEdit, 235  
 setWriteOnFinish  
     QEGenericEdit, 235  
 setWriteOnLoseFocus  
     QEGenericEdit, 235  
 shape  
     QSimpleShape, 564  
 shapeOptions  
     QEShape, 487  
 showGrid  
     QETable, 547  
 showScale  
     QEAnalogIndicator, 127  
 showText  
     QEAnalogIndicator, 127  
 showTime  
     QEImage, 298  
 slotMacroPrefix  
     QEFormGrid, 215  
 slotNumberOffset  
     QEFormGrid, 215  
 slotNumberWidth  
     QEFormGrid, 215  
 slotStrings  
     QEFormGrid, 215  
 SO\_AREA4  
     QEImage, 280  
 SO\_BEAM  
     QEImage, 280  
 SO\_HSLICE1  
     QEImage, 280  
 SO\_HSLICE2  
     QEImage, 280  
 SO\_HSLICE3  
     QEImage, 280  
 SO\_HSLICE4  
     QEImage, 280  
 SO\_HSLICE5  
     QEImage, 280  
 SO\_NONE  
     QEImage, 280  
 SO\_PANNING  
     QEImage, 280  
 SO\_PROFILE  
     QEImage, 280  
 SO\_TARGET  
     QEImage, 280  
 SO\_VSLICE1  
     QEImage, 280  
 SO\_VSLICE2  
     QEImage, 280  
 SO\_VSLICE3  
     QEImage, 280  
 SO\_VSLICE4  
     QEImage, 280  
 SO\_VSLICE5  
     QEImage, 280  
 SolidSmallCrosshair  
     QEImage, 281  
 Space  
     QEAnalogProgressBar, 134  
     QECheckBox, 163  
     QELabel, 313  
     QELineEdit, 324  
     QERadioButton, 439

spacing  
QEFormGrid, 216  
QERadioGroup, 454

spanAngle  
QEAnalogIndicator, 127

State  
QECheckBox, 164  
QEPushButton, 398  
QERadioButton, 440

StateMachineTemplate, 571

StateSet  
QSimpleShape, 563

StdOutput  
QECheckBox, 163  
QEImage, 279  
QEPushButton, 398  
QERadioButton, 439

styleSheet  
QEAbstractWidget, 120  
QEAnalogProgressBar, 137  
QEAnalogSlider, 146  
QEBitStatus, 154  
QECheckBox, 172  
QEComboBox, 180  
QEConfiguredLayout, 187  
QEFileBrowser, 195  
QEFileDialog, 201  
QEFrame, 222  
QEGenericEdit, 237  
QEGroupBox, 242  
QEImage, 298  
QELabel, 317  
QELog, 337  
QEPlot, 375  
QEPushButton, 406  
QERadioButton, 448  
QEScript, 479  
QEShape, 494  
QESimpleShape, 503  
QESlider, 511  
QESpinBox, 518

subscribe  
QECheckBox, 172  
QEComboBox, 180  
QEGenericEdit, 237  
QEPeriodic, 364  
QEPushButton, 406  
QERadioButton, 448  
QESlider, 511  
QESpinBox, 518

substitutedTitle  
QEGroupBox, 243  
QERadioGroup, 454

suffix  
QNumericEdit, 555

targetColor  
QEImage, 298

TargetOptions  
QEImage, 280

targetTriggerVariable  
QEImage, 298

targetXVariable  
QEImage, 298

targetYVariable  
QEImage, 298

Terminal  
QECheckBox, 163  
QEImage, 279  
QEPushButton, 398  
QERadioButton, 439

Text  
QECheckBox, 163  
QELabel, 313  
QEPushButton, 398  
QERadioButton, 439

TextAndIcon  
QECheckBox, 164  
QEPushButton, 398  
QERadioButton, 440

TextFormats  
QSimpleShape, 563

textSubstitutions  
QEGroupBox, 243  
QESubstitutedLabel, 543

textToDouble  
QELocalEnumeration, 331

textToInt  
QELocalEnumeration, 332

textToValue  
QELocalEnumeration, 332

Time  
QEAnalogProgressBar, 133  
QECheckBox, 162  
QELabel, 312  
QELineEdit, 323  
QEPushButton, 397  
QERadioButton, 438

timeColor  
QEImage, 298

titles  
     QETable, 547

Top\_To\_Bottom  
     QEAnalogIndicator, 126

trace, 573

tracking  
     QAnalogSlider, 97

trailingZeros  
     QEAnalogProgressBar, 138  
     QECheckBox, 172  
     QELabel, 318  
     QELlineEdit, 326  
     QEPushButton, 407  
     QERadioButton, 448

uiFile  
     QEForm, 211  
     QEFormGrid, 216

Underscore  
     QEAnalogProgressBar, 134  
     QECheckBox, 163  
     QELabel, 313  
     QELlineEdit, 324  
     QERadioButton, 439

UnsignedInteger  
     QEAnalogProgressBar, 133  
     QECheckBox, 162  
     QELabel, 312  
     QELlineEdit, 323  
     QEPushButton, 397  
     QERadioButton, 438

UPDATE\_PIXMAP  
     QELabel, 313

UPDATE\_TEXT  
     QELabel, 313

updateImage  
     mpegSource, 84

updateOption  
     QECheckBox, 172  
     QELabel, 318  
     QEPushButton, 407  
     QERadioButton, 448

UpdateOptions  
     QECheckBox, 163  
     QELabel, 313  
     QEPushButton, 398  
     QERadioButton, 439

updateOptions  
     QELabel, 313

URL

QEImage, 298

useDbDisplayLimits  
     QEAnalogProgressBar, 138

useDbEnumerations  
     QEComboBox, 179  
     QERadioGroup, 454

useDbPrecision  
     QEAnalogProgressBar, 138  
     QECheckBox, 172  
     QELabel, 318  
     QELlineEdit, 326  
     QEPushButton, 407  
     QERadioButton, 448

useFalseColour  
     QEImage, 299

User  
     QEAbstractWidget, 118  
     QEAnalogProgressBar, 134  
     QEAnalogSlider, 144  
     QEBitStatus, 152  
     QECheckBox, 164  
     QEComboBox, 178  
     QEConfiguredLayout, 185  
     QEFileBrowser, 193  
     QEFileImage, 199  
     QEFrame, 220  
     QEGenericEdit, 234  
     QEGroupBox, 241  
     QEImage, 281  
     QELabel, 313  
     QELog, 336  
     QEPeriodic, 362  
     QEPlot, 374  
     QEPushButton, 398  
     QERadioButton, 440  
     QEScript, 478  
     QEShape, 488  
     QESimpleShape, 501  
     QESlider, 509  
     QESpinBox, 516  
     userInfoStruct, 574  
     userLevelEnabled  
         QEAbstractWidget, 120  
         QEAnalogProgressBar, 138  
         QEAnalogSlider, 146  
         QEBitStatus, 154  
         QECheckBox, 172  
         QEComboBox, 180  
         QEConfiguredLayout, 187  
         QEFileBrowser, 195

QEFileDialog, 201  
QEFrame, 223  
QEGenericEdit, 237  
QEGroupBox, 243  
QEImage, 299  
QELabel, 318  
QELog, 337  
QEPeriodic, 364  
QEPlot, 375  
QEPushButton, 407  
QERadioButton, 448  
QEScript, 479  
QEShape, 494  
QESimpleShape, 503  
QESlider, 511  
QESpinBox, 518  
userLevelEngineerStyle  
QEAbstractWidget, 120  
QEAnalogProgressBar, 138  
QEAnalogSlider, 146  
QEBitStatus, 154  
QECheckBox, 172  
QEComboBox, 181  
QEConfiguredLayout, 187  
QEFileBrowser, 195  
QEFileDialog, 201  
QEFrame, 223  
QEGenericEdit, 237  
QEGroupBox, 243  
QEImage, 299  
QELabel, 318  
QELog, 337  
QEPeriodic, 364  
QEPlot, 376  
QEPushButton, 407  
QERadioButton, 448  
QEScript, 479  
QEShape, 495  
QESimpleShape, 503  
QESlider, 511  
QESpinBox, 518  
UserLevels  
QEAbstractWidget, 118  
QEAnalogProgressBar, 134  
QEAnalogSlider, 143  
QEBitStatus, 152  
QECheckBox, 164  
QEComboBox, 178  
QEConfiguredLayout, 185  
QEFileBrowser, 193  
QEFileDialog, 199  
QEFrame, 220  
QEGenericEdit, 234  
QEGroupBox, 241  
QEImage, 281  
QELabel, 313  
QELog, 335  
QEPeriodic, 362  
QEPlot, 374  
QEPushButton, 398  
QERadioButton, 440  
QEScript, 478  
QEShape, 487  
QESimpleShape, 501  
QESlider, 509  
QESpinBox, 516  
userLevelScientistStyle  
QEAbstractWidget, 120  
QEAnalogProgressBar, 138  
QEAnalogSlider, 146  
QEBitStatus, 154  
QECheckBox, 173  
QEComboBox, 181  
QEConfiguredLayout, 187  
QEFileBrowser, 195  
QEFileDialog, 201  
QEFrame, 223  
QEGenericEdit, 237  
QEGroupBox, 243  
QEImage, 299  
QELabel, 318  
QELog, 337  
QEPeriodic, 364  
QEPlot, 376  
QEPushButton, 407  
QERadioButton, 449  
QEScript, 480  
QEShape, 495  
QESimpleShape, 504  
QESlider, 511  
QESpinBox, 518  
userLevelUserStyle  
QEAbstractWidget, 120  
QEAnalogProgressBar, 139  
QEAnalogSlider, 147  
QEBitStatus, 154  
QECheckBox, 173  
QEComboBox, 181  
QEConfiguredLayout, 187  
QEFileBrowser, 195

QEFileDialog, 201  
QEFrame, 223  
QEGenericEdit, 238  
QEGroupBox, 243  
QEImage, 299  
QELabel, 319  
QELog, 337  
QEPeriodic, 364  
QEPlot, 376  
QEPushButton, 407  
QERadioButton, 449  
QEScript, 480  
QEShape, 495  
QESimpleShape, 504  
QESlider, 512  
QESpinBox, 519  
userLevelVisibility  
QEAbstractWidget, 121  
QEAnalogProgressBar, 139  
QEAnalogSlider, 147  
QEBitStatus, 155  
QECheckBox, 173  
QEComboBox, 181  
QEConfiguredLayout, 188  
QEFileBrowser, 196  
QEFileDialog, 202  
QEFrame, 223  
QEGenericEdit, 238  
QEGroupBox, 244  
QEImage, 299  
QELabel, 319  
QELog, 338  
QEPeriodic, 364  
QEPlot, 376  
QEPushButton, 408  
QERadioButton, 449  
QEScript, 480  
QEShape, 495  
QESimpleShape, 504  
QESlider, 512  
QESpinBox, 519  
  
value  
QEAnalogIndicator, 127  
QEAnalogProgressBar, 139  
ValueScaling, 576  
valueToText  
QELocalEnumeration, 332  
variable  
QEAnalogProgressBar, 139  
QEAnalogSlider, 147  
QEBitStatus, 155  
QECheckBox, 173  
QEComboBox, 181  
QEFileDialog, 202  
QEFileImage, 202  
QEGeneralEdit, 226  
QEGenericEdit, 238  
QELabel, 319  
QNumericUpDown, 355  
QEPushButton, 408  
QEPvFrame, 411  
QEPvProperties, 431  
QERadioButton, 449  
QERadioGroup, 454  
QESimpleShape, 504  
QESlider, 512  
QESpinBox, 519  
variable1  
QEPlot, 376  
QEShape, 495  
variable2  
QEPlot, 377  
QEShape, 496  
variable3  
QEPlot, 377  
QEShape, 496  
variable4  
QEPlot, 377  
QEShape, 496  
variable5  
QEShape, 496  
variable6  
QEShape, 496  
variableAsToolTip  
QEAbstractWidget, 121  
QEAnalogProgressBar, 139  
QEAnalogSlider, 147  
QEBitStatus, 155  
QECheckBox, 173  
QEComboBox, 182  
QEConfiguredLayout, 188  
QEFileDialog, 196  
QEFileImage, 202  
QEForm, 211  
QEFormGrid, 216  
QEFrame, 224  
QEGenericEdit, 238  
QEGroupBox, 244  
QEImage, 300

QELabel, 319  
QELog, 338  
QEMenuButton, 342  
QEPeriodic, 365  
QEPlot, 377  
QEPushButton, 408  
QERadioButton, 449  
QEScript, 480  
QEShape, 496  
QESimpleShape, 504  
QESlider, 512  
QESpinBox, 519  
variableSubstitutions  
    QEAnalogProgressBar, 139  
    QEAnalogSlider, 147  
    QEBitStatus, 155  
    QECheckBox, 173  
    QEComboBox, 182  
    QEFileBrowser, 196  
    QEFileImage, 202  
    QEForm, 211  
    QEGeneralEdit, 226  
    QEGenericEdit, 238  
    QEImage, 300  
    QELabel, 319  
    QNumericEdit, 355  
    QEPeriodic, 365  
    QEPlot, 377  
    QEPlotter, 382  
    QEPushButton, 408  
    QE PvFrame, 411  
    QE PvProperties, 431  
    QERadioButton, 449  
    QERadioGroup, 454  
    QEScalarHistogram, 467  
    QEShape, 496  
    QESimpleShape, 504  
    QESlider, 512  
    QESpinBox, 519  
    QEStripChart, 527  
    QETable, 547  
verticalFlip  
    QEImage, 300  
vertSlice1Color  
    QEImage, 300  
vertSlice2Color  
    QEImage, 300  
vertSlice3Color  
    QEImage, 300  
vertSlice4Color  
    QEImage, 300  
vertSlice5Color  
    QEImage, 300  
VideoWidget, 577  
visible  
    QEAbstractWidget, 121  
    QEAnalogProgressBar, 140  
    QEAnalogSlider, 147  
    QEBitStatus, 155  
    QECheckBox, 173  
    QEComboBox, 182  
    QEConfiguredLayout, 188  
    QEFileBrowser, 196  
    QEFileImage, 202  
    QEFrame, 224  
    QEGenericEdit, 238  
    QEGroupBox, 244  
    QEImage, 301  
    QELabel, 319  
    QELog, 338  
    QEPeriodic, 365  
    QEPlot, 377  
    QEPushButton, 408  
    QERadioButton, 450  
    QEScript, 480  
    QEShape, 497  
    QESimpleShape, 505  
    QESlider, 512  
    QESpinBox, 519  
WhenInAlarm  
    QEAbstractWidget, 118  
    QEAnalogProgressBar, 133  
    QEAnalogSlider, 143  
    QEBitStatus, 152  
    QECheckBox, 162  
    QEComboBox, 178  
    QEConfiguredLayout, 185  
    QEFileBrowser, 193  
    QEFileImage, 199  
    QEFrame, 220  
    QEGenericEdit, 234  
    QEGroupBox, 241  
    QEImage, 278  
    QELabel, 312  
    QELog, 335  
    QEPeriodic, 362  
    QEPlot, 374  
    QEPushButton, 397  
    QERadioButton, 438

QEScript, 478  
QEShape, 487  
QESimpleShape, 501  
QESlider, 509  
QESpinBox, 516  
widthVariable  
    QEImage, 301  
writeButtonVariable1  
    QEPeriodic, 365  
writeButtonVariable2  
    QEPeriodic, 365  
writeOnChange  
    QEComboBox, 179  
    QENumericEdit, 356  
    QESlider, 510  
writeOnClick  
    QECheckBox, 174  
    QEPushButton, 408  
    QERadioButton, 450  
writeOnEnter  
    QEGenericEdit, 238  
    QENumericEdit, 356  
writeOnFinish  
    QEGenericEdit, 239  
    QENumericEdit, 356  
writeOnLoseFocus  
    QEGenericEdit, 239  
    QENumericEdit, 356  
writeOnPress  
    QECheckBox, 174  
    QEPushButton, 408  
    QERadioButton, 450  
writeOnRelease  
    QECheckBox, 174  
    QEPushButton, 409  
    QERadioButton, 450  
yuv422  
    QEImage, 279  
yuv444  
    QEImage, 279  
Zoom  
    QEImage, 279  
zoomMenu, 580