

EPICS QT Framework

2.0.0

Generated by Doxygen 1.7.4

Mon Nov 12 2012 08:18:27

Contents

1	QE framework - EPICS aware Qt Widgets and data access classes	1
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
2	GNU General Public License	5
3	INSTALL	7
4	ASgui screen shots	9
5	other applications using epicsqt widgets	15
6	Qt Designer	17
7	Qt Creator	19
8	Class Index	21
8.1	Class Hierarchy	21
9	Class Index	25
9.1	Class List	25

10 Class Documentation	29
10.1 _Field Class Reference	29
10.2 _Item Class Reference	30
10.3 _QDialogItem Class Reference	30
10.4 _QDialogLogin Class Reference	30
10.5 _QPushButtonGroup Class Reference	31
10.6 _QTableWidgetFileBrowser Class Reference	31
10.7 _QTableWidgetLog Class Reference	32
10.8 _QTableWidgetScript Class Reference	32
10.9 QEAnalogIndicator::Band Struct Reference	32
10.10 QEAnalogIndicator::BandList Class Reference	33
10.11 qcastatemachine::ConnectionQCaStateMachine Class Reference	33
10.12 ContainerProfile Class Reference	33
10.13 contextMenu Class Reference	35
10.14 contextMenuObject Class Reference	37
10.15 QEPeriodic::elementInfoStruct Struct Reference	37
10.16 flipRotateMenu Class Reference	38
10.17 imageContextMenu Class Reference	38
10.18 imageMarkup Class Reference	39
10.19 localEnumerationItem Class Reference	40
10.20 managePixmaps Class Reference	41
10.21 markupBeam Class Reference	41
10.22 markupHLine Class Reference	42
10.23 markupItem Class Reference	43
10.24 markupLine Class Reference	44
10.25 markupRegion Class Reference	45
10.26 markupTarget Class Reference	45
10.27 markupText Class Reference	46
10.28 markupVLine Class Reference	47
10.29 PeriodicDialog Class Reference	47
10.30 PeriodicElementSetupForm Class Reference	48
10.31 PeriodicSetupDialog Class Reference	48
10.32 QEStripChart::PrivateData Class Reference	48
10.33 QEStripChartItem::PrivateData Class Reference	49

10.34	profilePlot Class Reference	49
10.35	PushButtonSpecifications Struct Reference	49
10.36	QBitStatus Class Reference	50
10.37	QBitStatusManager Class Reference	51
10.38	QCaAlarmInfo Class Reference	52
10.39	QCaConnectionInfo Class Reference	52
10.40	QCaDataPoint Struct Reference	53
10.41	QCaDataPointList Class Reference	53
10.42	QCaDateTime Class Reference	53
10.43	QCaEventFilter Class Reference	54
10.44	QCaEventItem Class Reference	54
10.45	QCaEventUpdate Class Reference	54
10.46	QCaInstalledFiltersListItem Class Reference	55
10.47	qcaobject::QCaObject Class Reference	55
10.48	qcastatemachine::QCaStateMachine Class Reference	57
10.49	QCaVariableNamePropertyManager Class Reference	57
10.50	QEAnalogIndicator Class Reference	58
10.51	QEAnalogIndicatorManager Class Reference	60
10.52	QEAnalogProgressBar Class Reference	61
10.52.1	Member Function Documentation	64
10.52.1.1	requestEnabled	64
10.52.2	Property Documentation	64
10.52.2.1	addUnits	64
10.52.2.2	allowDrop	64
10.52.2.3	arrayAction	64
10.52.2.4	enabled	65
10.52.2.5	format	65
10.52.2.6	int	65
10.52.2.7	leadingZero	65
10.52.2.8	localEnumeration	65
10.52.2.9	notation	65
10.52.2.10	precision	66
10.52.2.11	trailingZeros	66
10.52.2.12	useDbPrecision	66

10.52.2.13	userLevelEnabled	66
10.52.2.14	userLevelEngineerStyle	66
10.52.2.15	userLevelScientistStyle	66
10.52.2.16	userLevelUserStyle	67
10.52.2.17	userLevelVisibility	67
10.52.2.18	variable	67
10.52.2.19	variableAsToolTip	67
10.52.2.20	variableSubstitutions	67
10.52.2.21	visible	67
10.53	QEAnalogProgressBarManager Class Reference	68
10.54	QEBitStatus Class Reference	68
10.54.1	Member Function Documentation	71
10.54.1.1	requestEnabled	71
10.54.2	Property Documentation	71
10.54.2.1	allowDrop	71
10.54.2.2	enabled	71
10.54.2.3	int	71
10.54.2.4	userLevelEnabled	71
10.54.2.5	userLevelEngineerStyle	72
10.54.2.6	userLevelScientistStyle	72
10.54.2.7	userLevelUserStyle	72
10.54.2.8	userLevelVisibility	72
10.54.2.9	variable	72
10.54.2.10	variableAsToolTip	72
10.54.2.11	variableSubstitutions	73
10.54.2.12	visible	73
10.55	QEBitStatusManager Class Reference	73
10.56	QEByteArray Class Reference	73
10.57	QEComboBox Class Reference	74
10.57.1	Member Function Documentation	77
10.57.1.1	requestEnabled	77
10.57.2	Property Documentation	77
10.57.2.1	allowDrop	77
10.57.2.2	enabled	77

10.57.2.3 int	77
10.57.2.4 userLevelEnabled	78
10.57.2.5 userLevelEngineerStyle	78
10.57.2.6 userLevelScientistStyle	78
10.57.2.7 userLevelUserStyle	78
10.57.2.8 userLevelVisibility	78
10.57.2.9 variable	79
10.57.2.10variableAsToolTip	79
10.57.2.11variableSubstitutions	79
10.57.2.12visible	79
10.58QEComboBoxManager Class Reference	79
10.59QEConfiguredLayout Class Reference	80
10.60QEConfiguredLayoutManager Class Reference	82
10.61QEDragDrop Class Reference	82
10.62QEFileBrowser Class Reference	84
10.63QEFileBrowserManager Class Reference	86
10.64QEFloating Class Reference	86
10.65QEFloatingFormatting Class Reference	87
10.66QEForm Class Reference	87
10.67QEFormManager Class Reference	89
10.68QEFrame Class Reference	90
10.68.1 Member Function Documentation	92
10.68.1.1 requestEnabled	92
10.68.2 Property Documentation	92
10.68.2.1 allowDrop	92
10.68.2.2 enabled	92
10.68.2.3 int	92
10.68.2.4 userLevelEnabled	92
10.68.2.5 userLevelEngineerStyle	92
10.68.2.6 userLevelScientistStyle	93
10.68.2.7 userLevelUserStyle	93
10.68.2.8 userLevelVisibility	93
10.68.2.9 variableAsToolTip	93
10.68.2.10visible	93

10.69QEFrameManager Class Reference	94
10.70QEGenericButton Class Reference	94
10.71QEGroupBox Class Reference	97
10.71.1 Member Function Documentation	99
10.71.1.1 requestEnabled	99
10.71.2 Property Documentation	99
10.71.2.1 allowDrop	99
10.71.2.2 enabled	99
10.71.2.3 int	99
10.71.2.4 userLevelEnabled	99
10.71.2.5 userLevelEngineerStyle	99
10.71.2.6 userLevelScientistStyle	100
10.71.2.7 userLevelUserStyle	100
10.71.2.8 userLevelVisibility	100
10.71.2.9 variableAsToolTip	100
10.71.2.10 visible	100
10.72QEGroupBoxManager Class Reference	101
10.73QEImage Class Reference	101
10.73.1 Member Function Documentation	107
10.73.1.1 requestEnabled	107
10.73.2 Property Documentation	107
10.73.2.1 allowDrop	107
10.73.2.2 beamXVariable	107
10.73.2.3 beamYVariable	107
10.73.2.4 clippingHighVariable	107
10.73.2.5 clippingLowVariable	107
10.73.2.6 clippingOnOffVariable	107
10.73.2.7 enabled	108
10.73.2.8 heightVariable	108
10.73.2.9 imageVariable	108
10.73.2.10 int	108
10.73.2.11 regionOfInterestHVariable	108
10.73.2.12 regionOfInterestWVariable	108
10.73.2.13 regionOfInterestXVariable	108

10.73.2.14	regionOfInterestYVariable	108
10.73.2.15	targetTriggerVariable	109
10.73.2.16	targetXVariable	109
10.73.2.17	targetYVariable	109
10.73.2.18	userLevelEnabled	109
10.73.2.19	userLevelEngineerStyle	109
10.73.2.20	userLevelScientistStyle	109
10.73.2.21	userLevelUserStyle	110
10.73.2.22	userLevelVisibility	110
10.73.2.23	variableAsToolTip	110
10.73.2.24	variableSubstitutions	110
10.73.2.25	visible	110
10.73.2.26	widthVariable	110
10.74	QEImageManager Class Reference	111
10.75	QEInteger Class Reference	111
10.76	QEIntegerFormatting Class Reference	112
10.77	QELabel Class Reference	112
10.77.1	Detailed Description	116
10.77.2	Member Enumeration Documentation	117
10.77.2.1	updateOptions	117
10.77.3	Constructor & Destructor Documentation	117
10.77.3.1	QELabel	117
10.77.3.2	QELabel	117
10.77.4	Member Function Documentation	117
10.77.4.1	dbValueChanged	117
10.77.4.2	requestEnabled	117
10.77.5	Property Documentation	117
10.77.5.1	addUnits	117
10.77.5.2	allowDrop	118
10.77.5.3	arrayAction	118
10.77.5.4	enabled	118
10.77.5.5	format	118
10.77.5.6	int	118
10.77.5.7	leadingZero	119

10.77.5.8 localEnumeration	119
10.77.5.9 notation	119
10.77.5.10 pixmap0	119
10.77.5.11 pixmap1	119
10.77.5.12 pixmap2	119
10.77.5.13 pixmap3	119
10.77.5.14 pixmap4	119
10.77.5.15 pixmap5	119
10.77.5.16 pixmap6	120
10.77.5.17 pixmap7	120
10.77.5.18 precision	120
10.77.5.19 trailingZeros	120
10.77.5.20 updateOption	120
10.77.5.21 useDbPrecision	120
10.77.5.22 userLevelEnabled	120
10.77.5.23 userLevelEngineerStyle	121
10.77.5.24 userLevelScientistStyle	121
10.77.5.25 userLevelUserStyle	121
10.77.5.26 userLevelVisibility	121
10.77.5.27 variable	121
10.77.5.28 variableAsToolTip	121
10.77.5.29 variableSubstitutions	122
10.77.5.30 visible	122
10.78 QELabelManager Class Reference	122
10.79 QELineEdit Class Reference	122
10.79.1 Member Function Documentation	126
10.79.1.1 requestEnabled	126
10.79.2 Property Documentation	126
10.79.2.1 addUnits	126
10.79.2.2 allowDrop	126
10.79.2.3 arrayAction	126
10.79.2.4 enabled	127
10.79.2.5 format	127
10.79.2.6 int	127

10.79.2.7 leadingZero	127
10.79.2.8 localEnumeration	127
10.79.2.9 notation	127
10.79.2.10precision	127
10.79.2.11trailingZeros	128
10.79.2.12useDbPrecision	128
10.79.2.13userLevelEnabled	128
10.79.2.14userLevelEngineerStyle	128
10.79.2.15userLevelScientistStyle	128
10.79.2.16userLevelUserStyle	128
10.79.2.17userLevelVisibility	129
10.79.2.18variable	129
10.79.2.19variableAsToolTip	129
10.79.2.20variableSubstitutions	129
10.79.2.21visible	129
10.80QELineEditManager Class Reference	129
10.81QELink Class Reference	130
10.82QELinkManager Class Reference	132
10.83QELog Class Reference	132
10.84QELogin Class Reference	135
10.85QELoginManager Class Reference	137
10.86QELogManager Class Reference	137
10.87QEPeriodic Class Reference	138
10.87.1 Member Function Documentation	141
10.87.1.1 requestEnabled	141
10.87.2 Member Data Documentation	141
10.87.2.1 allowDrop	141
10.87.3 Property Documentation	142
10.87.3.1 enabled	142
10.87.3.2 int	142
10.87.3.3 readbackLabelVariable1	142
10.87.3.4 readbackLabelVariable2	142
10.87.3.5 userLevelEnabled	142
10.87.3.6 userLevelEngineerStyle	142

10.87.3.7 userLevelScientistStyle	143
10.87.3.8 userLevelUserStyle	143
10.87.3.9 userLevelVisibility	143
10.87.3.10variableAsToolTip	143
10.87.3.11variableSubstitutions	143
10.87.3.12visible	143
10.87.3.13writeButtonVariable1	144
10.87.3.14writeButtonVariable2	144
10.88QEPeriodicComponentData Class Reference	144
10.89QEPeriodicManager Class Reference	144
10.90QEPeriodicTaskMenu Class Reference	145
10.91QEPeriodicTaskMenuFactory Class Reference	145
10.92QEpicsPV Class Reference	145
10.93QEPlot Class Reference	146
10.93.1 Member Function Documentation	151
10.93.1.1 requestEnabled	151
10.93.2 Member Data Documentation	151
10.93.2.1 allowDrop	151
10.93.3 Property Documentation	151
10.93.3.1 enabled	151
10.93.3.2 int	151
10.93.3.3 userLevelEnabled	151
10.93.3.4 userLevelEngineerStyle	152
10.93.3.5 userLevelScientistStyle	152
10.93.3.6 userLevelUserStyle	152
10.93.3.7 userLevelVisibility	152
10.93.3.8 variable1	152
10.93.3.9 variable2	152
10.93.3.10variable3	153
10.93.3.11variable4	153
10.93.3.12variableAsToolTip	153
10.93.3.13variableSubstitutions	153
10.93.3.14visible	153
10.94QEPlotManager Class Reference	153

10.95QEPushButton Class Reference	154
10.95.1 Member Function Documentation	158
10.95.1.1 requestEnabled	158
10.95.2 Property Documentation	158
10.95.2.1 allowDrop	158
10.95.2.2 altReadbackVariable	158
10.95.2.3 enabled	158
10.95.2.4 int	158
10.95.2.5 userLevelEnabled	158
10.95.2.6 userLevelEngineerStyle	159
10.95.2.7 userLevelScientistStyle	159
10.95.2.8 userLevelUserStyle	159
10.95.2.9 userLevelVisibility	159
10.95.2.10variable	159
10.95.2.11variableAsToolTip	159
10.95.2.12visible	160
10.96QEPushButtonManager Class Reference	160
10.97QEPvProperties Class Reference	160
10.97.1 Member Function Documentation	163
10.97.1.1 requestEnabled	163
10.97.2 Property Documentation	163
10.97.2.1 allowDrop	163
10.97.2.2 enabled	163
10.97.2.3 int	163
10.97.2.4 userLevelEnabled	163
10.97.2.5 userLevelEngineerStyle	164
10.97.2.6 userLevelScientistStyle	164
10.97.2.7 userLevelUserStyle	164
10.97.2.8 userLevelVisibility	164
10.97.2.9 variable	164
10.97.2.10variableAsToolTip	165
10.97.2.11variableSubstitutions	165
10.97.2.12visible	165
10.98QEPvPropertiesManager Class Reference	165

10.99QERadioButton Class Reference	166
10.99.1 Member Function Documentation	169
10.99.1.1 requestEnabled	169
10.99.2 Property Documentation	169
10.99.2.1 allowDrop	169
10.99.2.2 enabled	170
10.99.2.3 int	170
10.99.2.4 userLevelEnabled	170
10.99.2.5 userLevelEngineerStyle	170
10.99.2.6 userLevelScientistStyle	170
10.99.2.7 userLevelUserStyle	171
10.99.2.8 userLevelVisibility	171
10.99.2.9 variable	171
10.99.2.10variableAsToolTip	171
10.99.2.11variableSubstitutions	171
10.99.2.12visible	171
10.10QERadioButtonManager Class Reference	172
10.10QERecipe Class Reference	172
10.10QERecipeManager Class Reference	175
10.10QEScript Class Reference	176
10.10QEScriptManager Class Reference	178
10.10QEShape Class Reference	178
10.105.1 Member Function Documentation	185
10.105.1.1requestEnabled	185
10.105.2 Property Documentation	185
10.105.2.1allowDrop	185
10.105.2.2enabled	185
10.105.2.3int	185
10.105.2.4userLevelEnabled	185
10.105.2.5userLevelEngineerStyle	185
10.105.2.6userLevelScientistStyle	186
10.105.2.7userLevelUserStyle	186
10.105.2.8userLevelVisibility	186
10.105.2.9variable1	186

10.105.2.10Variable2	186
10.105.2.11Variable3	187
10.105.2.12Variable4	187
10.105.2.13Variable5	187
10.105.2.14Variable6	187
10.105.2.15VariableAsToolTip	187
10.105.2.16VariableSubstitutions	187
10.105.2.17Visible	187
10.106QEShapeManager Class Reference	188
10.107QESlider Class Reference	188
10.107.1Member Function Documentation	191
10.107.1.1requestEnabled	191
10.107.2Property Documentation	191
10.107.2.1allowDrop	191
10.107.2.2enabled	191
10.107.2.3int	191
10.107.2.4userLevelEnabled	192
10.107.2.5userLevelEngineerStyle	192
10.107.2.6userLevelScientistStyle	192
10.107.2.7userLevelUserStyle	192
10.107.2.8userLevelVisibility	192
10.107.2.9variable	193
10.107.2.10VariableAsToolTip	193
10.107.2.11VariableSubstitutions	193
10.107.2.12Visible	193
10.108QESliderManager Class Reference	193
10.109QESpinBox Class Reference	194
10.109.1Member Function Documentation	196
10.109.1.1requestEnabled	196
10.109.2Property Documentation	196
10.109.2.1allowDrop	196
10.109.2.2enabled	197
10.109.2.3int	197
10.109.2.4userLevelEnabled	197

10.109.2.5userLevelEngineerStyle	197
10.109.2.6userLevelScientistStyle	197
10.109.2.7userLevelUserStyle	198
10.109.2.8userLevelVisibility	198
10.109.2.9variable	198
10.109.2.10variableAsToolTip	198
10.109.2.11variableSubstitutions	198
10.109.2.12visible	198
10.110QESpinBoxManager Class Reference	199
10.111QString Class Reference	199
10.112QStringFormatting Class Reference	200
10.113QStringFormattingMethods Class Reference	201
10.114QEStripChart Class Reference	202
10.115QEStripChartItem Class Reference	204
10.116QEStripChartItemDialog Class Reference	205
10.117QEStripChartManager Class Reference	205
10.118QEStripChartTimeDialog Class Reference	205
10.119QESubstitutedLabel Class Reference	206
10.119.1Member Data Documentation	207
10.119.1.1labelText	207
10.119.2Property Documentation	207
10.119.2.1textSubstitutions	207
10.120QESubstitutedLabelManager Class Reference	207
10.121QEToolTip Class Reference	208
10.122QEWidget Class Reference	209
10.123QEWidgets Class Reference	210
10.124QCastStateMachine::ReadQCaStateMachine Class Reference	211
10.125RecordSpec Class Reference	211
10.126RecordSpecList Class Reference	211
10.127SelectMenu Class Reference	212
10.128StandardProperties Class Reference	212
10.129StateMachineTemplate Class Reference	214
10.130StyleManager Class Reference	214
10.130.1Detailed Description	216

10.130.2	Member Function Documentation	216
10.130.2.1	getStyleEngineer	216
10.130.2.2	getStyleScientist	217
10.130.2.3	getStyleUser	217
10.130.2.4	setStyleEngineer	217
10.130.2.5	setStyleScientist	217
10.130.2.6	setStyleUser	217
10.130.2.7	styleUserLevelChanged	217
10.130.2.8	updateDataStyle	217
10.130.2.9	updateStatusStyle	217
10.131	QcstateMachine::SubscriptionQCaStateMachine Class Reference . . .	218
10.132	QcstateMachine::Trace Class Reference	218
10.133	QcstateMachine::TrackRange Class Reference	219
10.134	QcstateMachine::UserInfoStruct Class Reference	219
10.135	QcstateMachine::UserPeriodic::userInfoStructArray Struct Reference . . .	219
10.136	QcstateMachine::UserLevelSignal Class Reference	219
10.137	QcstateMachine::UserLevelSlot Class Reference	220
10.138	QcstateMachine::UserMessage Class Reference	220
10.139	QcstateMachine::UserMessageSignal Class Reference	222
10.140	QcstateMachine::UserMessageSlot Class Reference	222
10.141	QcstateMachine::VariableNameManager Class Reference	223
10.142	QcstateMachine::VideoWidget Class Reference	224
10.143	QcstateMachine::WidgetRef Class Reference	225
10.144	QcstateMachine::WriteQCaStateMachine Class Reference	225
10.145	QcstateMachine::ZoomMenu Class Reference	226

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 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
```

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.

Symbolic links to the resultant library libQEPlugin.so will need to set up according to how it is to be used - see QE_GettingStarted.pdf for details.

Any Qt specific queries? start at [Qt developer documentation](#)

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>

Chapter 2

GNU General Public License

The EPICS QT Framework is free software: you can redistribute it and/or modify it under the terms of the GNU 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

INSTALL

Chapter 4

ASgui screen shots

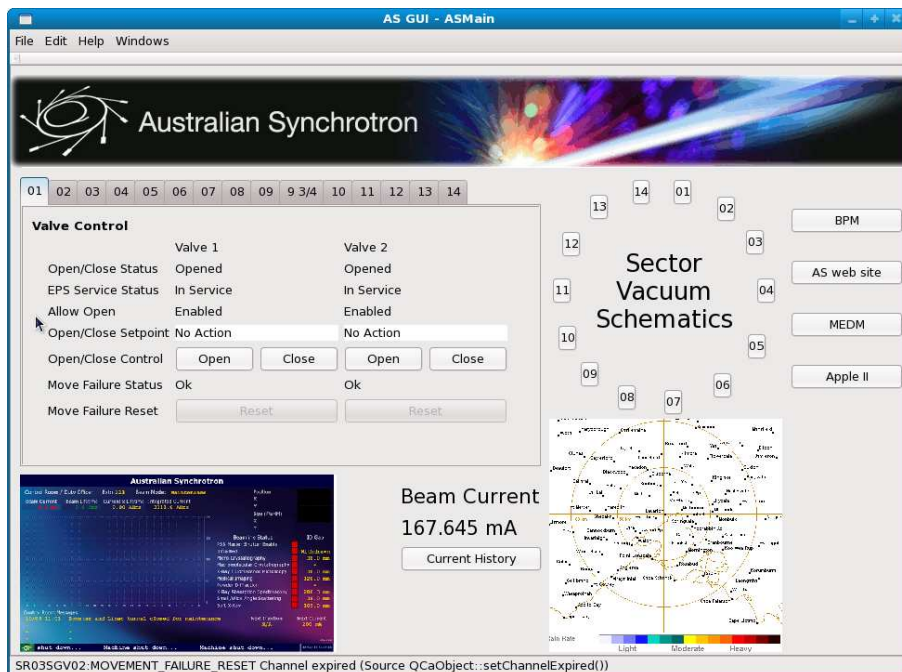


Figure 4.1: Australian Synchrotron mock up

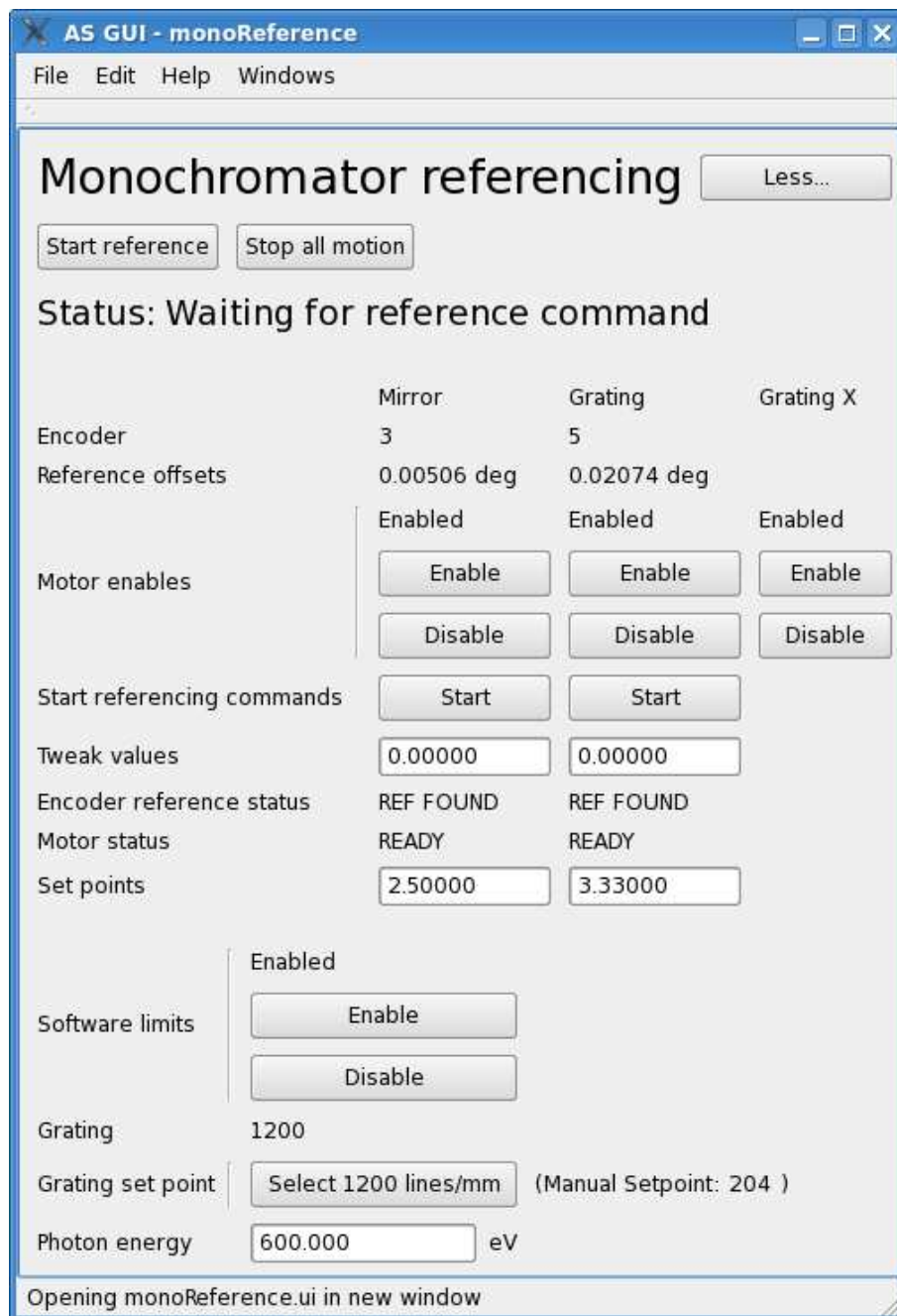


Figure 4.2: Monochromator referencing

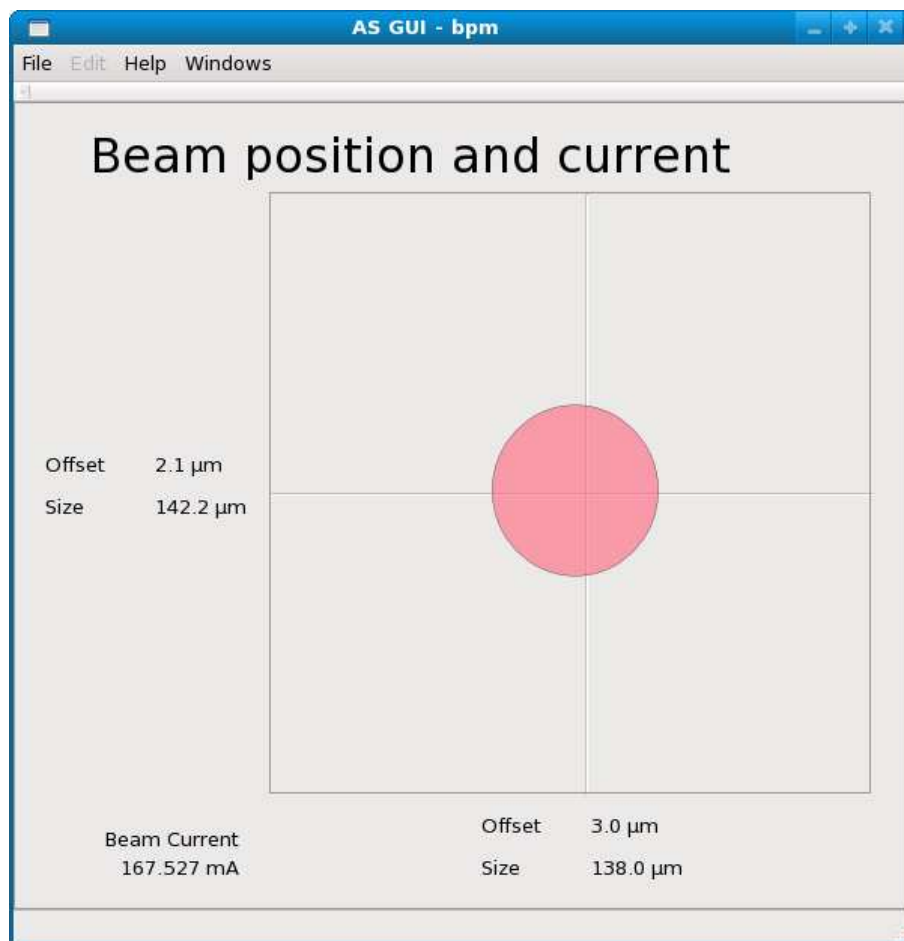


Figure 4.3: Beam position monitor

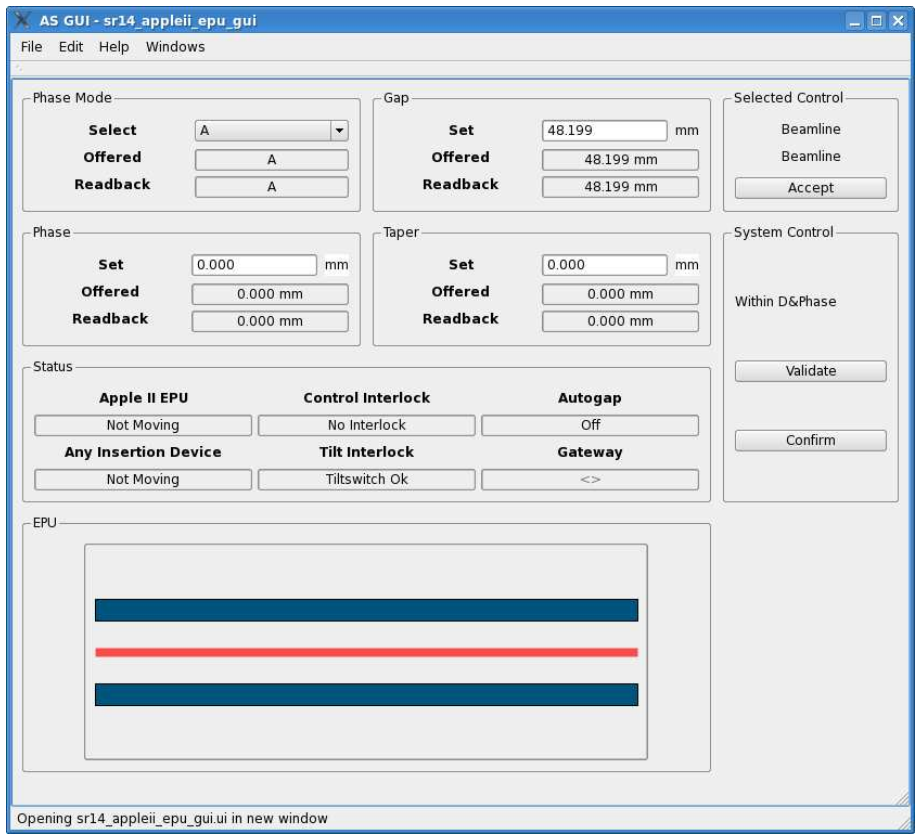


Figure 4.4: Insertion device

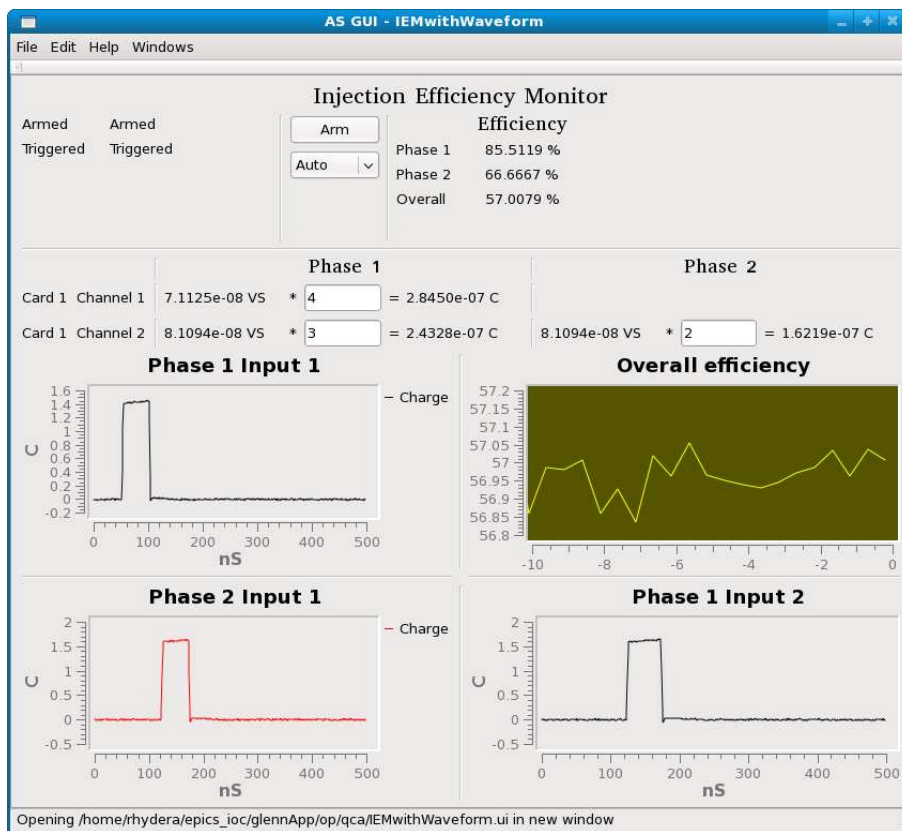


Figure 4.5: Injection efficiency monitor

Chapter 5

other applications using epicsqt widgets

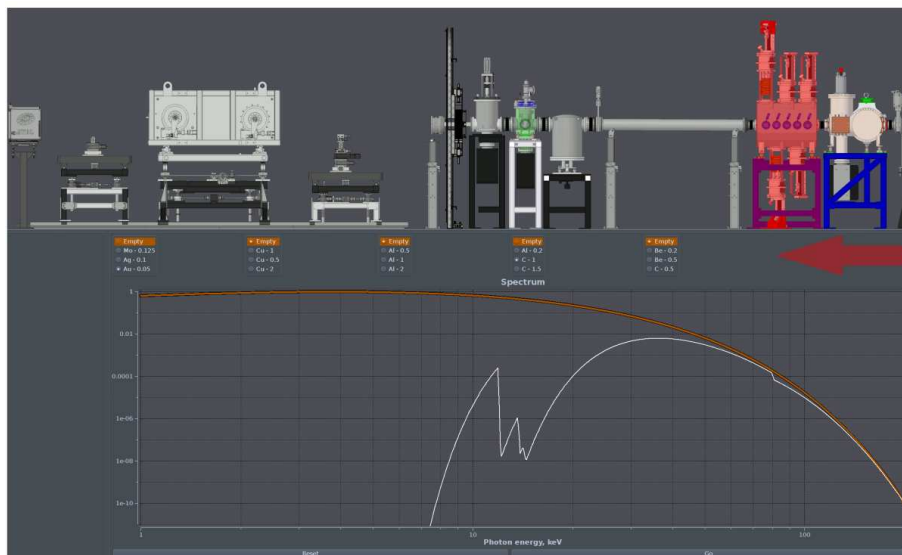


Figure 5.1: Medical Imaging beamline

2B SampleTable Z <@SR08ID01OPI01>

PV name: SR08ID01:MTR32B View mode: Macro

Description: 2B SampleTable Z

Precision: 5 Units: mm

Message: Connection established. clean

User: 6mm Move absolutely Raw: -80932

JOG< UNDO >JOG

LIMIT< Move relatively >LIMIT

< 1mm >

step/10 step/2 step*2 step*10

User: 6mm = Hi limit

Resolution: 0.0001mm/step * 42271.2068mm

Raw: -80932 + Lo Limit

Offset: -2.0932mm -42275.3932mm

Speed Acceleration

Maximum: 1.2mm/s

Normal: 0.8mm/s 1s

Backlash: 0mm/s 1s

log: 1mm/s 10s

Backlash: 0mm

Figure 5.2: Motor controller

MotorMx <@SR08ID01OPI01>

- ▲ ▼	DEI Theta Mono	109.5mm	<	0.1	>	UNDO
- ▲ ▼	DEI Mono Z	-0.3mm	<	0.1	>	UNDO
- ▲ ▼	2B Sample Table Y	0mm	<	1	>	UNDO
- ▲ ▼	2B SampleTable Z	6mm	<	1	>	UNDO
- ▲ ▼	2B Detector Table Z	42mm	<	5	>	UNDO
- ▲ ▼	2B Sample Rotate	-1deg	<	1	>	UNDO
- ▲ ▼	2B Detector Table Y	13.9025mm	<	1	>	UNDO
- ▲ ▼	SETUP	0	<	relative	>	STOP
- ▲ ▼	SLW01:LEFT	9.99975mm	<	3456	>	UNDO

Add motor

Figure 5.3: Motor controller

Chapter 6

Qt Designer

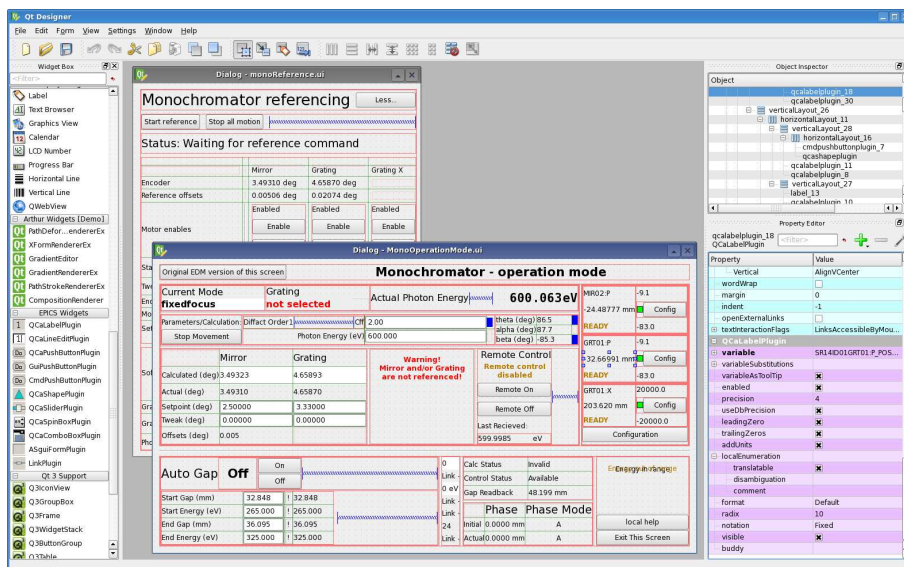


Figure 6.1: Editing multiple GUIs

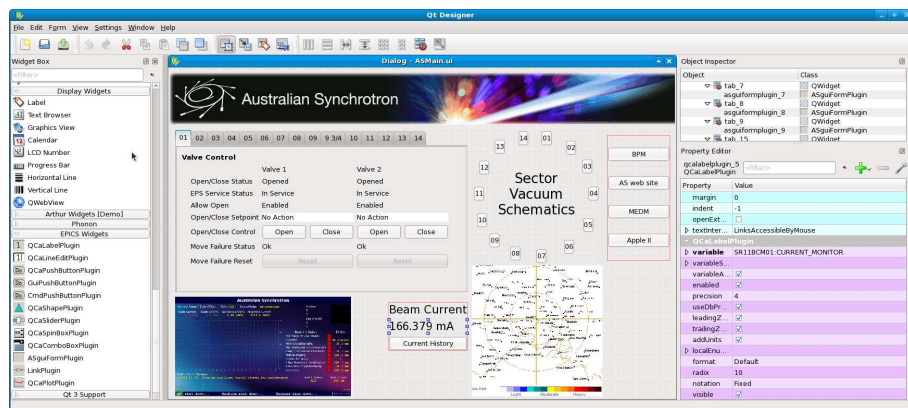


Figure 6.2: Editing a GUI

Chapter 7

Qt Creator

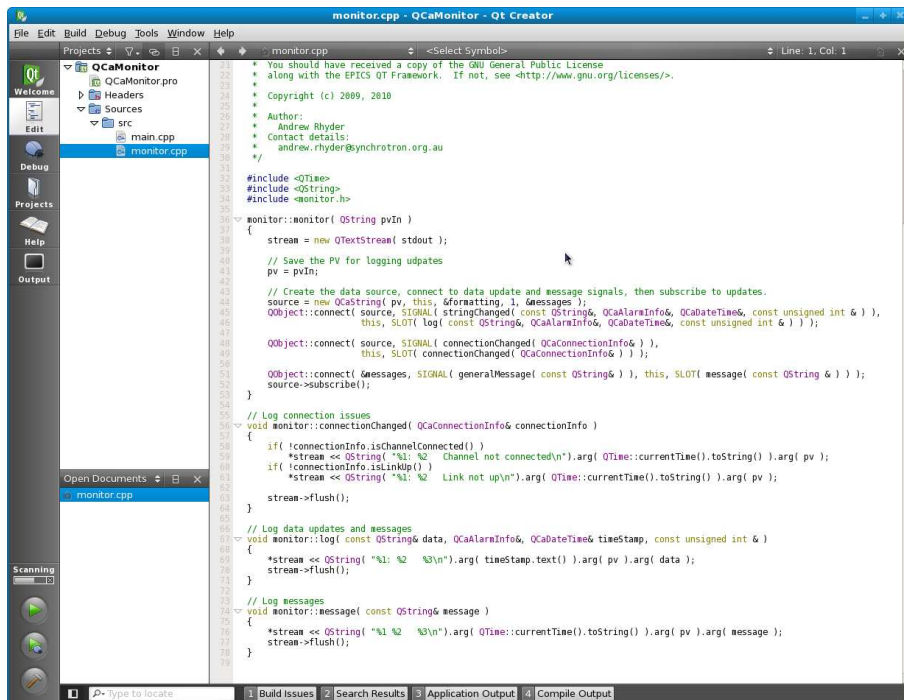


Figure 7.1: Application using epicsqt data source classes

Chapter 8

Class Index

8.1 Class Hierarchy

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

_Field	29
_Item	30
_QDialogItem	30
_QDialogLogin	30
_QPushButtonGroup	31
_QTableWidgetFileBrowser	31
_QTableWidgetLog	32
_QTableWidgetScript	32
QEEAnalogIndicator::Band	32
QEEAnalogIndicator::BandList	33
ContainerProfile	33
QEEWidget	209
QEEAnalogProgressBar	61
QEEBitStatus	68
QEEComboBox	74
QEEConfiguredLayout	80
QEEFileBrowser	84
QEEForm	87
QEEFrame	90
QEEGenericButton	94
QEEPUSHBUTTON	154
QEERadioBUTTON	166
QEEGroupBox	97
QEEImage	101
QEELabel	112
QEELineEdit	122
QEELink	130
QEELog	132
QEELogin	135

QEPeriodic	138
QEPlot	146
QEPvProperties	160
QERecipe	172
QEScript	176
QEShape	178
QESlider	188
QESpinBox	194
QEStripChart	202
QESubstitutedLabel	206
contextMenu	35
QEWidget	209
contextMenuObject	37
QEPeriodic::elementInfoStruct	37
flipRotateMenu	38
imageContextMenu	38
imageMarkup	39
VideoWidget	224
localEnumerationItem	40
managePixmap	41
QEGenericButton	94
QELabel	112
markupItem	43
markupBeam	41
markupHLine	42
markupLine	44
markupRegion	45
markupTarget	45
markupText	46
markupVLine	47
PeriodicDialog	47
PeriodicElementSetupForm	48
PeriodicSetupDialog	48
QEStripChart::PrivateData	48
QEStripChartItem::PrivateData	49
profilePlot	49
PushButtonSpecifications	49
QBitStatus	50
QEBitStatus	68
QBitStatusManager	51
QCaAlarmInfo	52
QCaConnectionInfo	52
QCaDataPoint	53
QCaDataPointList	53
QCaDateTime	53
QCaEventFilter	54
QCaEventItem	54
QCaEventUpdate	54

QCaInstalledFiltersListItem	55
qcaobject::QCaObject	55
QEByteArray	73
QEFloating	86
QEInteger	111
QEString	199
QCaVariableNamePropertyManager	57
QEAnalogIndicator	58
QEAnalogProgressBar	61
QEAnalogIndicatorManager	60
QEAnalogProgressBarManager	68
QEBitStatusManager	73
QEComboBoxManager	79
QEConfiguredLayoutManager	82
QEDragDrop	82
QEWidget	209
QEFileBrowserManager	86
QEFloatingFormatting	87
QEFormManager	89
QEFrameManager	94
QEGroupBoxManager	101
QEImageManager	111
QEIntegerFormatting	112
QELabelManager	122
QELineEditManager	129
QELinkManager	132
QELoginManager	137
QELogManager	137
QEPeriodicComponentData	144
QEPeriodicManager	144
QEPeriodicTaskMenu	145
QEPeriodicTaskMenuFactory	145
QEpicsPV	145
QEPlotManager	153
QEPushButtonManager	160
QEPvPropertiesManager	165
QERadioButtonManager	172
QERecipeManager	175
QEScriptManager	178
QEShapeManager	188
QESliderManager	193
QESpinBoxManager	199
QEStringFormatting	200
QEStringFormattingMethods	201
QEAnalogProgressBar	61
QEGenericButton	94
QELabel	112
QELineEdit	122
QEStriptChartItem	204

QESTripChartItemDialog	205
QESTripChartManager	205
QESTripChartTimeDialog	205
QESubstitutedLabelManager	207
QEToolTip	208
QEWWidget	209
QEWWidgets	210
RecordSpec	211
RecordSpecList	211
selectMenu	212
standardProperties	212
QEWWidget	209
StateMachineTemplate	214
qcastatemachine::QCaStateMachine	57
qcastatemachine::ConnectionQCaStateMachine	33
qcastatemachine::ReadQCaStateMachine	211
qcastatemachine::SubscriptionQCaStateMachine	218
qcastatemachine::WriteQCaStateMachine	225
styleManager	214
QEWWidget	209
trace	218
TrackRange	219
userInfoStruct	219
QEPERiodic::userInfoStructArray	219
userLevelSignal	219
userLevelSlot	220
UserMessage	220
QEWWidget	209
UserMessageSignal	222
UserMessageSlot	222
VariableNameManager	223
QEWWidget	209
WidgetRef	225
zoomMenu	226

Chapter 9

Class Index

9.1 Class List

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

_Field	29
_Item	30
_QDialogItem	30
_QDialogLogin	30
_QPushButtonGroup	31
_QTableWidgetFileBrowser	31
_QTableWidgetLog	32
_QTableWidgetScript	32
QEAnalogIndicator::Band	32
QEAnalogIndicator::BandList	33
qcastatemachine::ConnectionQCaStateMachine	33
ContainerProfile	33
contextMenu	35
contextMenuObject	37
QEPeriodic::elementInfoStruct	37
flipRotateMenu	38
imageContextMenu	38
imageMarkup	39
localEnumerationItem	40
managePixmaps	41
markupBeam	41
markupHLine	42
markupItem	43
markupLine	44
markupRegion	45
markupTarget	45
markupText	46
markupVLine	47
PeriodicDialog	47

PeriodicElementSetupForm	48
PeriodicSetupDialog	48
QEStripChart::PrivateData	48
QEStripChartItem::PrivateData	49
profilePlot	49
PushButtonSpecifications	49
QBitStatus	50
QBitStatusManager	51
QCaAlarmInfo	52
QCaConnectionInfo	52
QCaDataPoint	53
QCaDataPointList	53
QCaDateTime	53
QCaEventFilter	54
QCaEventItem	54
QCaEventUpdate	54
QCaInstalledFiltersListItem	55
qcaobject::QCaObject	55
qcastatemachine::QCaStateMachine	57
QCaVariableNamePropertyManager	57
QEAnalogIndicator	58
QEAnalogIndicatorManager	60
QEAnalogProgressBar	61
QEAnalogProgressBarManager	68
QEBitStatus	68
QEBitStatusManager	73
QEByteArray	73
QEComboBox	74
QEComboBoxManager	79
QEConfiguredLayout	80
QEConfiguredLayoutManager	82
QEDragDrop	82
QEFileBrowser	84
QEFileBrowserManager	86
QEFloating	86
QEFloatingFormatting	87
QEForm	87
QEFormManager	89
QEFrame	90
QEFrameManager	94
QEGenericButton	94
QEGroupBox	97
QEGroupBoxManager	101
QEImage	101
QEImageManager	111
QEInteger	111
QEIntegerFormatting	112
QELabel	112
QELabelManager	122
QELineEdit	122

QELineEditManager	129
QELink	130
QELinkManager	132
QELog	132
QELogin	135
QELoginManager	137
QELogManager	137
QEPeriodic	138
QEPeriodicComponentData	144
QEPeriodicManager	144
QEPeriodicTaskMenu	145
QEPeriodicTaskMenuFactory	145
QEpicsPV	145
QEPlot	146
QEPlotManager	153
QEPushButton	154
QEPushButtonManager	160
QEPvProperties	160
QEPvPropertiesManager	165
QERadioButton	166
QERadioButtonManager	172
QERecipe	172
QERecipeManager	175
QEScript	176
QEScriptManager	178
QEShape	178
QEShapeManager	188
QESlider	188
QESliderManager	193
QESpinBox	194
QESpinBoxManager	199
QEString	199
QEStringFormatting	200
QEStringFormattingMethods	201
QEStripChart	202
QEStripChartItem	204
QEStripChartItemDialog	205
QEStripChartManager	205
QEStripChartTimeDialog	205
QESubstitutedLabel	206
QESubstitutedLabelManager	207
QEToolTip	208
QEWidget	209
QEWidgets	210
qcastatemachine::ReadQCaStateMachine	211
RecordSpec	211
RecordSpecList	211
selectMenu	212
standardProperties	212
StateMachineTemplate	214

styleManager	214
qcastatemachine::SubscriptionQCaStateMachine	218
trace	218
TrackRange	219
userInfoStruct	219
QEPeiodic::userInfoStructArray	219
userLevelSignal	219
userLevelSlot	220
UserMessage	220
UserMessageSignal	222
UserMessageSlot	222
VariableNameManager	223
VideoWidget	224
WidgetRef	225
qcastatemachine::WriteQCaStateMachine	225
zoomMenu	226

Chapter 10

Class Documentation

10.1 _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](#) * **qCaWidget**

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

- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

10.2 `_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/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

10.3 `_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/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

10.4 `_QDialogLogin` Class Reference

Public Member Functions

- **_QDialogLogin** (QWidget *pParent=0, int pUserType=-1, Qt::WindowFlags pF=0)
- void **setCurrentUserType** (int pValue)
- void **setPassword** (QString pValue)

Protected Attributes

- QGridLayout * **qGridLayout**
- QVBoxLayout * **qVBoxLayout**
- QGroupBox * **qGroupBox**
- QRadioButton * **qRadioButtonUser**
- QRadioButton * **qRadioButtonScientist**
- QRadioButton * **qRadioButtonEngineer**
- QLabel * **qLabelType**
- QLineEdit * **qLineEditPassword**
- QPushButton * **qPushButtonOk**
- QPushButton * **qPushButtonCancel**
- int **userType**

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

- /home/rhydera/epicsqt/framework/widgets/QELogin/QELogin.h
- /home/rhydera/epicsqt/framework/widgets/QELogin/QELogin.cpp

10.5 _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/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

10.6 _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/rhydera/epicsqt/framework/widgets/QFileBrowser/QFileBrowser.h
- /home/rhydera/epicsqt/framework/widgets/QFileBrowser/QFileBrowser.cpp

10.7 **_QTableWidgetLog Class Reference**

Public Member Functions

- **_QTableWidgetLog** (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/rhydera/epicsqt/framework/widgets/QELog/QELog.h
- /home/rhydera/epicsqt/framework/widgets/QELog/QELog.cpp

10.8 **_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/rhydera/epicsqt/framework/widgets/QEScript/QEScript.h
- /home/rhydera/epicsqt/framework/widgets/QEScript/QEScript.cpp

10.9 **QEAnalogIndicator::Band Struct Reference**

Public Attributes

- double **lower**

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

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

- /home/rhydera/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalogIndicator.h

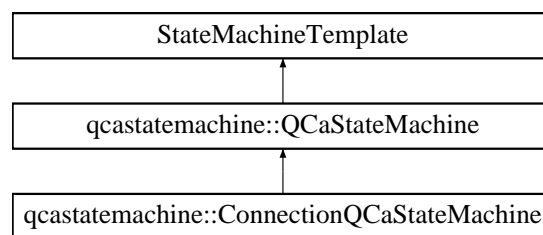
10.10 QEAnalogIndicator::BandList Class Reference

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

- /home/rhydera/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalogIndicator.h

10.11 qcastatemachine::ConnectionQCaStateMachine Class Reference

Inheritance diagram for qcastatemachine::ConnectionQCaStateMachine:



Public Member Functions

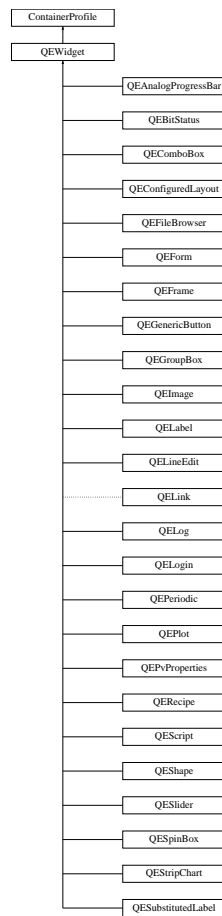
- **ConnectionQCaStateMachine** (void *parent)
- bool **process** (int requestedState)

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

- /home/rhydera/epicsqt/framework/data/include/QCaStateMachine.h
- /home/rhydera/epicsqt/framework/data/src/QCaStateMachine.cpp

10.12 ContainerProfile Class Reference

Inheritance diagram for ContainerProfile:



Public Member Functions

- void **takeLocalCopy** ()
- void **setupProfile** (QObject *guiLaunchConsumerIn, QString pathIn, QString parentPathIn, QString macroSubstitutionsIn)
- void **setupLocalProfile** (QObject *guiLaunchConsumerIn, QString pathIn, QString parentPathIn, QString macroSubstitutionsIn)
- void **updateConsumers** (QObject *guiLaunchConsumerIn)
- QObject * **replaceGuiLaunchConsumer** (QObject *newGuiLaunchConsumerIn)
- void **addMacroSubstitutions** (QString macroSubstitutionsIn)
- void **removeMacroSubstitutions** ()
- QObject * **getGuiLaunchConsumer** ()
- QString **getPath** ()
- QString **getParentPath** ()
- void **setPublishedParentPath** (QString publishedParentPathIn)
- QString **getMacroSubstitutions** ()
- bool **isProfileDefined** ()

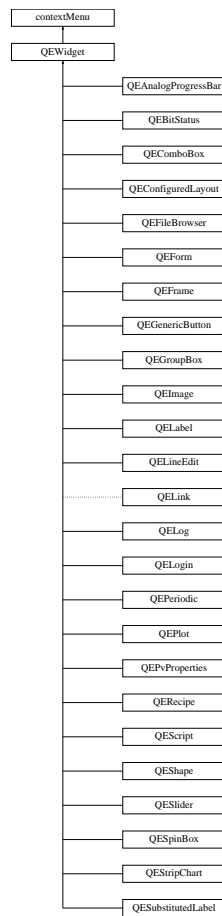
- void **addContainedWidget** ([QEWWidget](#) *containedWidget)
- [QEWWidget](#) * **getNextContainedWidget** ()
- void **removeContainedWidget** ([QEWWidget](#) *containedWidget)
- unsigned int **getMessageFormId** ()
- unsigned int **getPublishedMessageFormId** ()
- void **setPublishedMessageFormId** (unsigned int publishedMessageFormIdIn)
- void **releaseProfile** ()
- void **publishOwnProfile** ()
- void **setUserLevel** (userLevels level)
- userLevels **getUserLevel** ()
- virtual void **userLevelChanged** (userLevels)

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

- /home/rhydera/epicsqt/framework/widgets/include/ContainerProfile.h
- /home/rhydera/epicsqt/framework/widgets/src/ContainerProfile.cpp

10.13 contextMenu Class Reference

Inheritance diagram for contextMenu:



Public Types

- enum **contextMenuOptions** {
CM_NONE, **CM_COPY_VARIABLE**, **CM_COPY_DATA**, **CM_PASTE**,
CM_DRAG_VARIABLE, **CM_DRAG_DATA**, **CM_SPECIFIC_WIDGETS_START_ - HERE** }

Public Member Functions

- void **addContextMenuToWidget** (QWidget *w)
- bool **isDraggingVariable** ()
- QMenu * **getContextMenu** ()
- virtual QString **copyVariable** ()
- virtual QVariant **copyData** ()
- virtual void **paste** (QVariant)

Friends

- class [contextMenuObject](#)

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

- /home/rhydera/epicsqt/framework/widgets/include/contextMenu.h
- /home/rhydera/epicsqt/framework/widgets/src/contextMenu.cpp

10.14 contextMenuObject Class Reference

Public Slots

- void **contextMenuTriggered** (QAction *selectedItem)
- void **showContextMenu** (const QPoint &pos)
- void **setChecked** ()

Public Member Functions

- void **addContextMenuToWidget** (QWidget *w)
- void **manageChecked** (bool draggingVariable)
- void **setMenu** ([contextMenu](#) *menuIn)
- bool **isDraggingVariable** ()

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

- /home/rhydera/epicsqt/framework/widgets/include/contextMenu.h
- /home/rhydera/epicsqt/framework/widgets/src/contextMenu.cpp

10.15 QEPPeriodic::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/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h

10.16 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/rhydera/epicsqt/framework/widgets/QEImage/flipRotateMenu.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/flipRotateMenu.cpp

10.17 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_ENABLE_CONTRAST_REVERSAL**, **ICM_-**
ENABLE_PAN, **ICM_ENABLE_VERT**,
ICM_ENABLE_HOZ, **ICM_ENABLE_AREA**, **ICM_ENABLE_LINE**, **ICM_ENABLE_-**
TARGET,
ICM_DISPLAY_BUTTON_BAR, **ICM_ZOOM_SELECTED**, **ICM_ZOOM_FIT**, **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_HSLICE,
ICM_SELECT_VSLICE, **ICM_SELECT_AREA**, **ICM_SELECT_PROFILE**, **ICM_-**
SELECT_TARGET,
ICM_SELECT_BEAM }

Public Member Functions

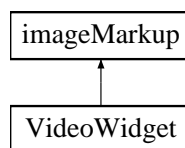
- **imageContextMenu** (QWidget *parent=0)
- void **getContextMenuOption** (const QPoint &, imageContextMenuOptions *option, bool *checked)
- void **addMenuItem** (const QString &title, const bool checkable, const bool checked, const imageContextMenuOptions option)
- void **addOptionMenuItem** (const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

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageContextMenu.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageContextMenu.cpp

10.18 imageMarkup Class Reference

Inheritance diagram for imageMarkup:



Public Types

- enum **markupIds** {
MARKUP_ID_REGION, **MARKUP_ID_H_SLICE**, **MARKUP_ID_V_SLICE**, **MARKUP_ID_LINE**,
MARKUP_ID_TARGET, **MARKUP_ID_BEAM**, **MARKUP_ID_TIMESTAMP**, **MARKUP_ID_COUNT**,
MARKUP_ID_NONE }

Public Member Functions

- void **markupMousePressEvent** (QMouseEvent *event)
- void **markupMouseReleaseEvent** (QMouseEvent *event)
- void **markupMouseMoveEvent** (QMouseEvent *event)
- void **setShowTime** (bool visibleIn)
- bool **getShowTime** ()
- markupIds **getMode** ()
- void **setMode** (markupIds modeIn)

- `QVector< QRect > &getMarkupAreas ()`
- `bool anyVisibleMarkups ()`
- `QCursor getDefaultMarkupCursor ()`
- `void setMarkupTime (QCaDateTime &time)`
- `void setMarkupColor (markupIds mode, QColor markupColorIn)`
- `QColor getMarkupColor (markupIds mode)`
- `QCursor getCircleCursor ()`
- `QCursor getTargetCursor ()`
- `virtual void markupSetCursor (QCursor cursor)=0`

Public Attributes

- `QImage * markupImage`
- `QVector< markupItem * > items`
- `QPoint grabOffset`
- `bool markupAreasStale`

Protected Member Functions

- `void markupResize (QSize newSize)`
- `virtual void markupChange (QImage &markups, QVector< QRect > &changedAreas)=0`
- `virtual void markupAction (markupIds mode, QPoint point1, QPoint point2)=0`

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

- `/home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h`
- `/home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp`

10.19 localEnumerationItem Class Reference

Public Types

- `enum operations {
LESS, LESS_EQUAL, EQUAL, NOT_EQUAL,
GREATER_EQUAL, GREATER, ALWAYS, UNKNOWN }`

Public Attributes

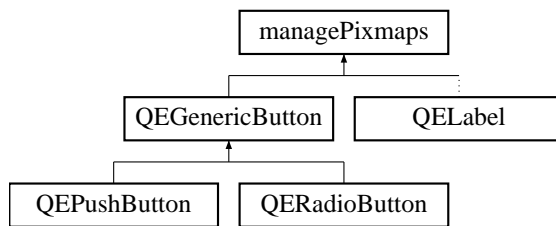
- `double dValue`
- `QString sValue`
- `operations op`
- `QString text`

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

- /home/rhydera/epicsqt/framework/data/include/QEStringFormatting.h

10.20 managePixmaps Class Reference

Inheritance diagram for managePixmaps:



Public Member Functions

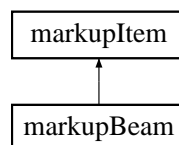
- void **setDataPixmap** (const QPixmap &Pixmap, const unsigned int index)
- QPixmap **getDataPixmap** (const unsigned int index)
- QPixmap **getDataPixmap** (const QString value)

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

- /home/rhydera/epicsqt/framework/widgets/include/managePixmaps.h
- /home/rhydera/epicsqt/framework/widgets/src/managePixmaps.cpp

10.21 markupBeam Class Reference

Inheritance diagram for markupBeam:



Public Member Functions

- **markupBeam** (imageMarkup *ownerIn, bool interactiveIn, bool reportOnMoveIn)
- void **startDrawing** (QPoint pos)

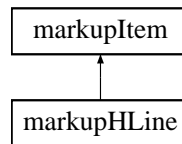
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (QPoint pos)
- bool **isOver** (QPoint point, QCursor *cursor)
- QPoint **origin** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.22 markupHLine Class Reference

Inheritance diagram for markupHLine:



Public Member Functions

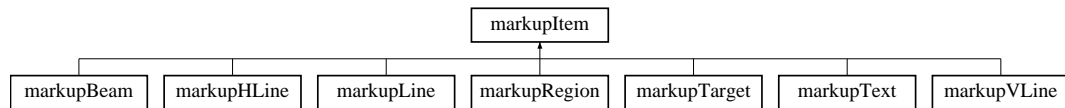
- **markupHLine** ([imageMarkup](#) *ownerIn, bool interactiveIn, bool reportOnMoveIn)
- void **startDrawing** (QPoint pos)
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (QPoint pos)
- bool **isOver** (QPoint point, QCursor *cursor)
- QPoint **origin** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.23 markupItem Class Reference

Inheritance diagram for markupItem:



Public Types

- enum **isOverOptions** { OVER_LINE, OVER_BORDER, OVER_AREA }
- enum **markupHandles** {
 MARKUP_HANDLE_NONE, MARKUP_HANDLE_START, MARKUP_HANDLE_END, 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

- **markupItem** ([imageMarkup](#) *ownerIn, isOverOptions over, bool interactiveIn, bool reportOnMoveIn)
- virtual void **setArea** ()=0
- virtual QPoint **origin** ()=0
- virtual void **moveTo** (QPoint pos)=0
- void **erase** ()
- virtual void **drawMarkup** (QPainter &p)=0
- virtual void **startDrawing** (QPoint pos)=0
- virtual bool **isOver** (QPoint point, QCursor *cursor)=0
- virtual QPoint **getPoint1** ()=0
- virtual QPoint **getPoint2** ()=0
- virtual QCursor **defaultCursor** ()=0
- bool **pointsNear** (QPoint p1, QPoint p)
- void **drawMarkupIn** ()
- void **drawMarkupOut** ()
- void **setColor** (QColor colorIn)
- QColor **getColor** ()

Public Attributes

- markupHandles **activeHandle**
- isOverOptions **isOverType**
- QRect **area**

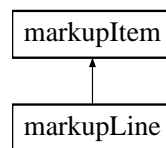
- bool **visible**
- bool **interactive**
- bool **reportOnMove**
- bool **highlighted**
- int **highlightMargin**
- QColor **color**
- [imageMarkup](#) * **owner**

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.24 markupLine Class Reference

Inheritance diagram for markupLine:



Public Member Functions

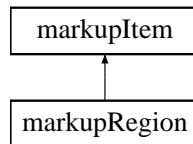
- **markupLine** ([imageMarkup](#) *ownerIn, bool interactiveIn, bool reportOnMoveIn)
- void **startDrawing** (QPoint pos)
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (QPoint pos)
- bool **isOver** (QPoint point, QCursor *cursor)
- QPoint **origin** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.25 markupRegion Class Reference

Inheritance diagram for markupRegion:



Public Member Functions

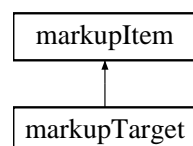
- **markupRegion** ([imageMarkup](#) *ownerIn, bool interactiveIn, bool reportOnMoveIn)
- void **startDrawing** (QPoint pos)
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (QPoint pos)
- bool **isOver** (QPoint point, QCursor *cursor)
- QPoint **origin** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.26 markupTarget Class Reference

Inheritance diagram for markupTarget:



Public Member Functions

- **markupTarget** ([imageMarkup](#) *ownerIn, bool interactiveIn, bool reportOnMoveIn)

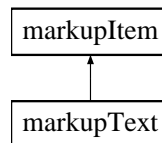
- void **startDrawing** (QPoint pos)
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (QPoint pos)
- bool **isOver** (QPoint point, QCursor *cursor)
- QPoint **origin** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.27 markupText Class Reference

Inheritance diagram for markupText:



Public Member Functions

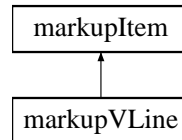
- **markupText** ([imageMarkup](#) *ownerIn, bool interactiveIn, bool reportOnMoveIn)
- void **setText** (QString textIn, bool draw)
- void **startDrawing** (QPoint pos)
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (QPoint pos)
- bool **isOver** (QPoint point, QCursor *cursor)
- QPoint **origin** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.28 markupVLine Class Reference

Inheritance diagram for markupVLine:



Public Member Functions

- **markupVLine** ([imageMarkup](#) *ownerIn, bool interactiveIn, bool reportOnMoveIn)
- void **startDrawing** (QPoint pos)
- void **setArea** ()
- void **drawMarkup** (QPainter &p)
- void **moveTo** (QPoint pos)
- bool **isOver** (QPoint point, QCursor *cursor)
- QPoint **origin** ()
- QPoint **getPoint1** ()
- QPoint **getPoint2** ()
- QCursor **defaultCursor** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/imageMarkup.cpp

10.29 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/rhydera/epicsqt/framework/widgets/QEPeriodic/PeriodicDialog.h
- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/PeriodicDialog.cpp

10.30 PeriodicElementSetupForm Class Reference

Public Member Functions

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

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

- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/PeriodicElementSetupForm.h
- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/PeriodicElementSetupForm.cpp

10.31 PeriodicSetupDialog Class Reference

Public Member Functions

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

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

- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/PeriodicSetupDialog.h
- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/PeriodicSetupDialog.cpp

10.32 QEStripChart::PrivateData Class Reference

Public Member Functions

- **PrivateData** (QEStripChart *chartIn)
- QEStripChartItem * **getItem** (unsigned int slot)
- QwtPlotCurve * **allocateCurve** ()
- void **calcDisplayMinMax** ()
- void **plotData** ()
- void **setReadOut** (QString text)

Public Attributes

- enum ChartYScale **chartYScale**
- enum ChartTimeMode **chartTimeMode**

Protected Member Functions

- bool **eventFilter** (QObject *obj, QEvent *event)

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

- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChart.cpp

10.33 QEStripChartItem::PrivateData Class Reference

Public Attributes

- [QEStripChart](#) * **chart**
- QLabel * **pvName**
- [QELabel](#) * **caLabel**

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

- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartItem.cpp

10.34 profilePlot Class Reference

Public Member Functions

- **profilePlot** (QWidget *parent=0)
- void **setScale** (int scaleIn)
- void **setProfile** (QVector< QPointF > &profile, double minX, double maxX, double minY, double maxY)
- void **setCursor** (int cursorIn)

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/profilePlot.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/profilePlot.cpp

10.35 PushButtonSpecifications Struct Reference

Public Attributes

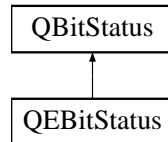
- int **width**
- const QString **caption**
- const QString **iconName**
- const QString **toolTip**
- const char * **member**

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

- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChart.cpp

10.36 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** ()

Properties

- int **value**
- int **numberOfBits**
- 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**

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

- /home/rhydera/epicsqt/framework/widgets/QEBitStatus/QBitStatus.h
- /home/rhydera/epicsqt/framework/widgets/QEBitStatus/QBitStatus.cpp

10.37 QBitStatusManager Class Reference

Public Member Functions

- **QBitStatusManager** (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/rhydera/epicsqt/framework/widgets/QEBitStatus/QEBitStatusManager.h
- /home/rhydera/epicsqt/framework/widgets/QEBitStatus/QEBitStatusManager.cpp

10.38 QCaAlarmInfo Class Reference

Public Member Functions

- **QCaAlarmInfo** (unsigned short statusIn, unsigned short severityIn)
- QString **statusName** ()
- QString **severityName** ()
- bool **isInAlarm** ()
- bool **isMinor** ()
- bool **isMajor** ()
- bool **isInvalid** ()
- QString **style** ()
- QString **getColorName** ()
- QCAALARMINFO_SEVERITY **getSeverity** ()

Static Public Member Functions

- static QCAALARMINFO_SEVERITY **getInvalidSeverity** ()

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

- /home/rhydera/epicsqt/framework/data/include/QCaAlarmInfo.h
- /home/rhydera/epicsqt/framework/data/src/QCaAlarmInfo.cpp

10.39 QCaConnectionInfo Class Reference

Public Member Functions

- **QCaConnectionInfo** (unsigned short channelStateIn, unsigned short linkStateIn)
- bool **isChannelConnected** ()

- bool **isLinkUp** ()

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

- /home/rhydera/epicsqt/framework/data/include/QCaConnectionInfo.h
- /home/rhydera/epicsqt/framework/data/src/QCaConnectionInfo.cpp

10.40 QCaDataPoint Struct Reference

Public Attributes

- double **value**
- [QCaDateTime](#) **datetime**
- [QCaAlarmInfo](#) **alarm**

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

- /home/rhydera/epicsqt/framework/data/include/QCaDataPoint.h

10.41 QCaDataPointList Class Reference

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

- /home/rhydera/epicsqt/framework/data/include/QCaDataPoint.h

10.42 QCaDateTime Class Reference

Public Member Functions

- **QCaDateTime** (QDateTime dt)
- void **operator=** (const [QCaDateTime](#) &other)
- **QCaDateTime** (unsigned long seconds, unsigned long nanoseconds)
- QString **text** ()
- double **floating** (QDateTime base)

Public Attributes

- unsigned long **nSec**

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

- /home/rhydera/epicsqt/framework/data/include/QCaDateTime.h
- /home/rhydera/epicsqt/framework/data/src/QCaDateTime.cpp

10.43 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/rhydera/epicsqt/framework/data/include/QCaEventFilter.h
- /home/rhydera/epicsqt/framework/data/src/QCaEventFilter.cpp

10.44 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/rhydera/epicsqt/framework/data/include/QCaEventUpdate.h

10.45 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/rhydera/epicsqt/framework/data/include/QCaEventUpdate.h
- /home/rhydera/epicsqt/framework/data/src/QCaEventUpdate.cpp

10.46 QCalInstalledFiltersListItem Class Reference

Public Member Functions

- **QCalInstalledFiltersListItem** (QObject *eventObjectIn)

Public Attributes

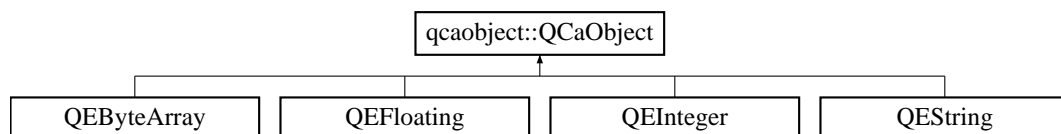
- QObject * **eventObject**
- long **referenceCount**

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

- /home/rhydera/epicsqt/framework/data/include/QCaEventFilter.h

10.47 qcaobject::QCaObject Class Reference

Inheritance diagram for qcaobject::QCaObject:



Public Slots

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

Signals

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

Public Member Functions

- **QCaObject** (const QString &recordName, QObject *eventObject, unsigned char signalsToSendIn=SIG_VARIANT)
- **QCaObject** (const QString &recordName, QObject *eventObject, [UserMessage](#) *userMessageIn, unsigned char signalsToSendIn=SIG_VARIANT)
- bool **subscribe** ()
- bool **singleShotRead** ()
- bool **dataTypeKnown** ()
- 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** ()
- QString **getEgu** ()
- QStringList **getEnumerations** ()
- unsigned int **getPrecision** ()
- double **getDisplayLimitUpper** ()
- double **getDisplayLimitLower** ()
- double **getAlarmLimitUpper** ()
- double **getAlarmLimitLower** ()
- double **getWarningLimitUpper** ()
- double **getWarningLimitLower** ()
- double **getControlLimitUpper** ()
- double **getControlLimitLower** ()
- generic::generic_types **getDataType** ()

Static Public Member Functions

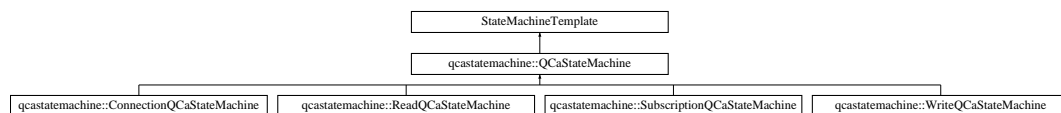
- static void **processEventStatic** (QCaEventUpdate *dataUpdateEvent)

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

- /home/rhydera/epicsqt/framework/data/include/QCaObject.h
- /home/rhydera/epicsqt/framework/data/src/QCaObject.cpp

10.48 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/rhydera/epicsqt/framework/data/include/QCaStateMachine.h
- /home/rhydera/epicsqt/framework/data/src/QCaStateMachine.cpp

10.49 QCaVariableNamePropertyManager Class Reference

Signals

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

Public Member Functions

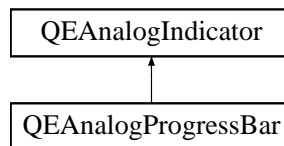
- QString **getVariableNameProperty** ()
- void **setVariableNameProperty** (QString variableNamePropertyIn)
- QString **getSubstitutionsProperty** ()
- void **setSubstitutionsProperty** (QString substitutionsPropertyIn)
- void **setVariableIndex** (unsigned int variableIndexIn)

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

- /home/rhydera/epicsqt/framework/data/include/QCaVariableNamePropertyManager.h
- /home/rhydera/epicsqt/framework/data/src/QCaVariableNamePropertyManager.cpp

10.50 QEAnalogIndicator Class Reference

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)
- virtual QSize **sizeHint** () const
- double **getValue** ()
- void **setMinimum** (const double value)
- double **getMinimum** ()
- void **setMaximum** (const double value)
- double **getMaximum** ()
- void **setOrientation** (const enum Orientations value)
- enum Orientations **getOrientation** ()
- void **setMode** (const enum Modes value)
- enum Modes **getMode** ()
- void **setCentreAngle** (const int value)
- int **getCentreAngle** ()
- void **setSpanAngle** (const int value)
- int **getSpanAngle** ()
- void **setMinorInterval** (const double value)
- double **getMinorInterval** ()
- void **setMajorInterval** (const double value)
- double **getMajorInterval** ()
- void **setLogScaleInterval** (const int value)
- int **getLogScaleInterval** ()
- void **setBorderColour** (const QColor value)
- QColor **getBorderColour** ()
- void **setForegroundColour** (const QColor value)
- QColor **getForegroundColour** ()
- void **setBackgroundColour** (const QColor value)
- QColor **getBackgroundColour** ()
- void **setFontColour** (const QColor value)
- QColor **getFontColour** ()
- void **setShowText** (const bool value)
- bool **getShowText** ()
- void **setShowScale** (const bool value)
- bool **getShowScale** ()
- void **setLogScale** (const bool value)
- bool **getLogScale** ()

Protected Member Functions

- virtual QString **getTextImage** ()
- virtual [BandList](#) **getBandList** ()

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**

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

- /home/rhydera/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalogIndicator.h
- /home/rhydera/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalogIndicator.cpp

10.51 QEAnalogIndicatorManager Class Reference

Public Member Functions

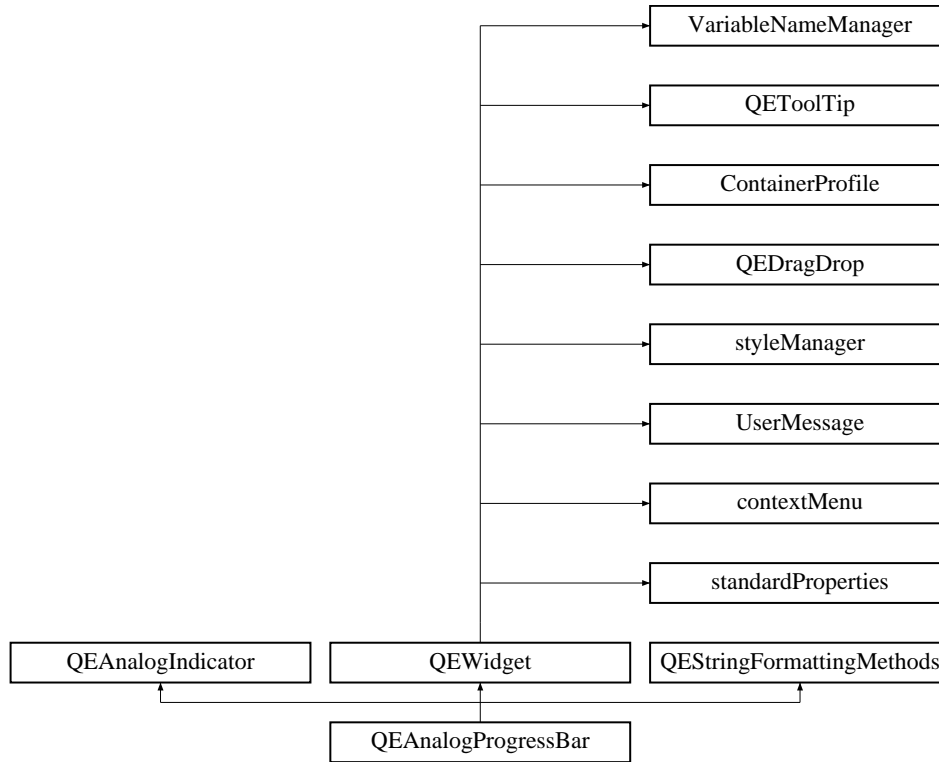
- **QEAnalogIndicatorManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalogIndicatorManager.h
- /home/rhydera/epicsqt/framework/widgets/QEAnalogIndicator/QEAnalogIndicatorManager.cpp

10.52 QEAnalogProgressBar Class Reference

Inheritance diagram for QEAnalogProgressBar:



Public Types

- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }
User friendly enumerations for userLevelVisibility and userLevelEnabled properties - refer to userLevelVisibility and userLevelEnabled properties and userLevel enumeration for details.
- 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 }
User friendly enumerations for format property - refer to QQStringFormatting::formats for details.
- enum **Notations** { **Fixed** = QQStringFormatting::NOTATION_FIXED, **Scientific** = QQStringFormatting::NOTATION_SCIENTIFIC, **Automatic** = QQStringFormatting::NOTATION_AUTOMATIC }

User friendly enumerations for notation property - refer to `QStringFormatting::notations` for details.

- enum `ArrayActions` { **Append** = `QStringFormatting::APPEND`, **Ascii** = `QStringFormatting::ASCII`, **Index** = `QStringFormatting::INDEX` }

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

- enum **AlarmSeverityDisplayModes** { **none**, **foreground**, **background** }

Public Slots

- void `requestEnabled` (const bool &state)

Signals

- void **dbValueChanged** (const double &out)
- void **requestResend** ()

Public Member Functions

- bool `isEnabled` () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void `setEnabled` (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- 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.

- **QEAnalogProgressBar** (QWidget *parent=0)
- **QEAnalogProgressBar** (const QString &variableName, QWidget *parent=0)
- void **setVariableNameAndSubstitutions** (QString variableNameIn, QString variableNameSubstitutionsIn, unsigned int variableIndex)
- void **setUseDbDisplayLimits** (bool useDbDisplayLimitsIn)
- bool **getUseDbDisplayLimits** ()
- void **setAlarmSeverityDisplayMode** (AlarmSeverityDisplayModes value)
- AlarmSeverityDisplayModes **getAlarmSeverityDisplayMode** ()

Protected Member Functions

- QString **getTextImage** ()
- [BandList](#) **getBandList** ()
- 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** ()

Protected Attributes

- [QEFloatingFormatting](#) **floatingFormatting**

Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned int
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- int [precision](#)

- bool [useDbPrecision](#)
- bool [leadingZero](#)
- bool [trailingZeros](#)
- bool [addUnits](#)
- QString [localEnumeration](#)
- [Formats](#) [format](#)
- [Notations](#) [notation](#)
- [ArrayActions](#) [arrayAction](#)
- bool **useDbDisplayLimits**
- AlarmSeverityDisplayModes **alarmSeverityDisplayMode**

10.52.1 Member Function Documentation

10.52.1.1 void `QEAAnalogProgressBar::requestEnabled (const bool & state)` [`inline`, `slot`]

Similar to standard `setEnabled` slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.52.2 Property Documentation

10.52.2.1 bool `QEAAnalogProgressBar::addUnits` [`read`, `write`]

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

10.52.2.2 bool `QEAAnalogProgressBar::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 [QEDragDrop](#).

10.52.2.3 [ArrayActions](#) `QEAAnalogProgressBar::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'.

10.52.2.4 `bool QEAnalogProgressBar::enabled` [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.52.2.5 `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).

10.52.2.6 `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.

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

Index used to select a single item of data for formatting from an array of data. Default is 0. Only used when the `arrayAction` property is INDEX. Refer to the `arrayAction` property for more details.

10.52.2.7 `bool QEAnalogProgressBar::leadingZero` [read, write]

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

10.52.2.8 `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.

10.52.2.9 `Notations QEAnalogProgressBar::notation` [read, write]

Notation used for numerical formatting. Default is fixed.

10.52.2.10 `int QEAnalogProgressBar::precision` `[read, write]`

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

10.52.2.11 `bool QEAnalogProgressBar::trailingZeros` `[read, write]`

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

10.52.2.12 `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.

10.52.2.13 `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 through the [QELogin](#) widget, or programatically 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'.

10.52.2.14 `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.

10.52.2.15 `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.

10.52.2.16 `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.

10.52.2.17 `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 through the [QELogin](#) widget, or programatically 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'.

10.52.2.18 `QString QEAnalogProgressBar::variable` [read, write]

EPICS variable name (CA PV)

10.52.2.19 `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.

Reimplemented from [QEWidget](#).

10.52.2.20 `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 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.

10.52.2.21 `bool QEAnalogProgressBar::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/rhydera/epicsqt/framework/widgets/QEAnalogProgressBar/QEAnalogProgressBar.h`
- `/home/rhydera/epicsqt/framework/widgets/QEAnalogProgressBar/QEAnalogProgressBar.cpp`

10.53 QEAnalogProgressBarManager Class Reference

Public Member Functions

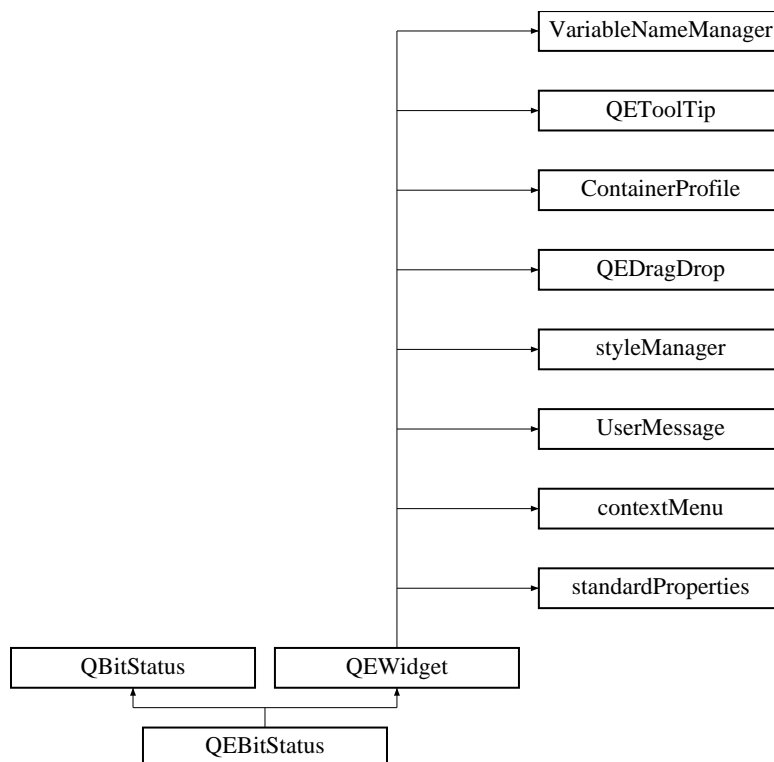
- **QEAnalogProgressBarManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEAnalogProgressBar/QEAnalogProgressBarManager.h
- /home/rhydera/epicsqt/framework/widgets/QEAnalogProgressBar/QEAnalogProgressBarManager.cpp

10.54 QEBitStatus Class Reference

Inheritance diagram for QEBitStatus:



Public Types

- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

Public Slots

- void **requestEnabled** (const bool &state)

Signals

- void **dbValueChanged** (const long &out)

Public Member Functions

- bool **isEnabled** () const

Access function for 'enabled' property - refer to 'enabled' property for details.

- void [setEnabled](#) (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- **QEBitStatus** (QWidget *parent=0)
- **QEBitStatus** (const QString &variableName, QWidget *parent=0)
- void **setVariableNameAndSubstitutions** (QString variableNameIn, QString variableNameSubstitutionsIn, unsigned int variableIndex)

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** ()

Protected Attributes

- [QEIntegerFormatting](#) **integerFormatting**

Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)

- [QString](#) [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)

10.54.1 Member Function Documentation

10.54.1.1 `void QEBitStatus::requestEnabled (const bool & state) [inline, slot]`

Similar to standard `setEnabled` slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.54.2 Property Documentation

10.54.2.1 `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.

Reimplemented from [QEDragDrop](#).

10.54.2.2 `bool QEBitStatus::enabled [read, write]`

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.54.2.3 `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.

10.54.2.4 `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 through the [QELogin](#) widget, or programatically 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'.

10.54.2.5 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.

10.54.2.6 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.

10.54.2.7 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.

10.54.2.8 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 through the [QELogin](#) widget, or programatically 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'.

10.54.2.9 QString QEBitStatus::variable [read, write]

EPICS variable name (CA PV)

10.54.2.10 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.

Reimplemented from [QEWidget](#).

10.54.2.11 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.

10.54.2.12 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/rhydera/epicsqt/framework/widgets/QEBitStatus/QEBitStatus.h
- /home/rhydera/epicsqt/framework/widgets/QEBitStatus/QEBitStatus.cpp

10.55 QEBitStatusManager Class Reference

Public Member Functions

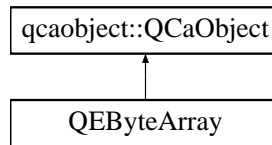
- **QEBitStatusManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEBitStatus/QEBitStatusManager.h
- /home/rhydera/epicsqt/framework/widgets/QEBitStatus/QEBitStatusManager.cpp

10.56 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

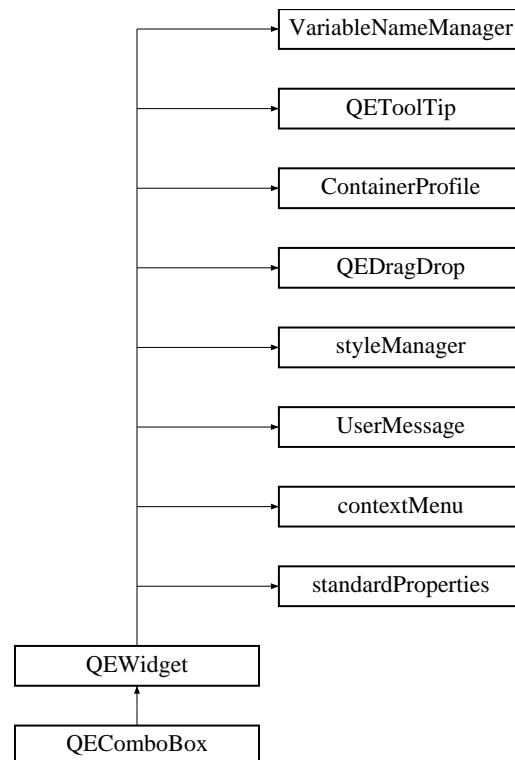
- **QeByteArray** (QString recordName, QObject *eventObject, unsigned int variableIndexIn)
- **QeByteArray** (QString recordName, QObject *eventObject, unsigned int variableIndexIn, [UserMessage](#) *userMessageIn)

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

- /home/rhydera/epicsqt/framework/data/include/QeByteArray.h
- /home/rhydera/epicsqt/framework/data/src/QeByteArray.cpp

10.57 QeComboBox Class Reference

Inheritance diagram for QeComboBox:



Public Types

- enum [UserLevels](#) { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

Public Slots

- void **writeNow** ()
- void [requestEnabled](#) (const bool &state)

Signals

- void **dbValueChanged** (const qlonglong &out)
- void **userChange** (const QString &oldValue, const QString &newValue, const QString &lastValue)

Public Member Functions

- **QEComboBox** (QWidget *parent=0)
- **QEComboBox** (const QString &variableName, QWidget *parent=0)
- void **setWriteOnChange** (bool writeOnChangeIn)
- bool **getWriteOnChange** ()
- void **setSubscribe** (bool subscribe)
- bool **getSubscribe** ()
- void **setUseDbEnumerations** (bool useDbEnumerations)
- bool **getUseDbEnumerations** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.

Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Protected Attributes

- **QEIntegerFormatting integerFormatting**
- bool **useDbEnumerations**
- bool **writeOnChange**

Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- bool **subscribe**
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned int
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- UserLevels [userLevelVisibility](#)
- UserLevels [userLevelEnabled](#)

10.57.1 Member Function Documentation

10.57.1.1 void QEComboBox::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.57.2 Property Documentation

10.57.2.1 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.

Reimplemented from [QEDragDrop](#).

10.57.2.2 bool QEComboBox::enabled [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.57.2.3 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.

10.57.2.4 UserLevels QComboBox::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 through the [QELogin](#) widget, or programatically 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'.

10.57.2.5 QString QComboBox::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.

10.57.2.6 QString QComboBox::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.

10.57.2.7 QString QComboBox::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.

10.57.2.8 UserLevels QComboBox::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 through the [QELogin](#) widget, or programatically 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'.

10.57.2.9 `QString QEComboBox::variable` [read, write]

EPICS variable name (CA PV)

10.57.2.10 `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.

Reimplemented from [QEWidget](#).

10.57.2.11 `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.

10.57.2.12 `bool QEComboBox::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/rhydera/epicsqt/framework/widgets/QEComboBox/QEComboBox.h
- /home/rhydera/epicsqt/framework/widgets/QEComboBox/QEComboBox.cpp

10.58 QEComboBoxManager Class Reference

Public Member Functions

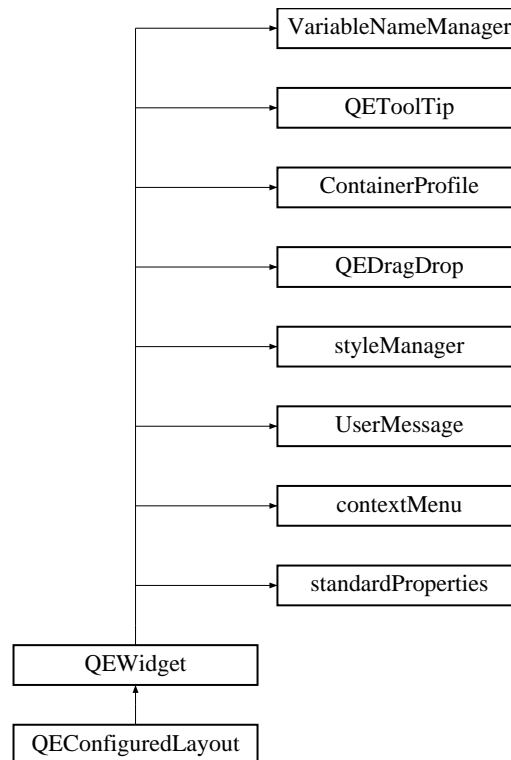
- **QEComboBoxManager** (QObject *parent=0)
- `bool isContainer ()` const
- `bool isInitialized ()` const
- `QIcon icon ()` const
- `QString group ()` const
- `QString includeFile ()` const
- `QString name ()` const
- `QString toolTip ()` const
- `QString whatsThis ()` const
- `QWidget * createWidget (QWidget *parent)`
- `void initialize (QDesignerFormEditorInterface *core)`

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

- /home/rhydera/epicsqt/framework/widgets/QEComboBox/QEComboBoxManager.h
- /home/rhydera/epicsqt/framework/widgets/QEComboBox/QEComboBoxManager.cpp

10.59 QEConfiguredLayout Class Reference

Inheritance diagram for QEConfiguredLayout:



Public Types

- enum **configurationTypesProperty** { **File** = FROM_FILE, **Text** = FROM_TEXT }
- enum **detailsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }
- enum **userTypesProperty** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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 **setDetailsLayout** (int pValue)
- int **getDetailsLayout** ()
- void **setCurrentUserType** (int pValue)
- int **getCurrentUserType** ()
- void **refreshFields** ()
- void **userLevelChanged** (userLevels pValue)
- void **setConfigurationTypeProperty** (configurationTypesProperty pConfigurationType)
- configurationTypesProperty **getConfigurationTypeProperty** ()
- void **setDetailsLayoutProperty** (detailsLayoutProperty pDetailsLayout)
- detailsLayoutProperty **getDetailsLayoutProperty** ()
- void **setCurrentUserTypeProperty** (userTypesProperty pUserType)
- userTypesProperty **getCurrentUserTypeProperty** ()

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 **detailsLayout**
- int **currentUserType**
- bool **subscription**

Properties

- QString **itemDescription**
- bool **showItemList**
- configurationTypesProperty **configurationType**
- detailsLayoutProperty **detailsLayout**
- userTypesProperty **currentUserType**

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

- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.h
- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayout.cpp

10.60 QEConfiguredLayoutManager Class Reference

Public Member Functions

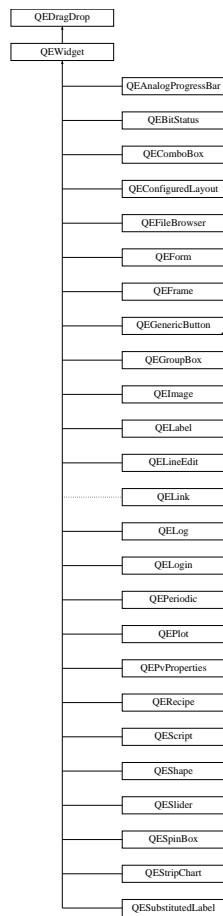
- **QEConfiguredLayoutManager** (QObject *pParent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *pParent)
- void **initialize** (QDesignerFormEditorInterface *pCore)

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

- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayoutManager.h
- /home/rhydera/epicsqt/framework/widgets/QEConfiguredLayout/QEConfiguredLayoutManager.cpp

10.61 QEDragDrop Class Reference

Inheritance diagram for QEDragDrop:



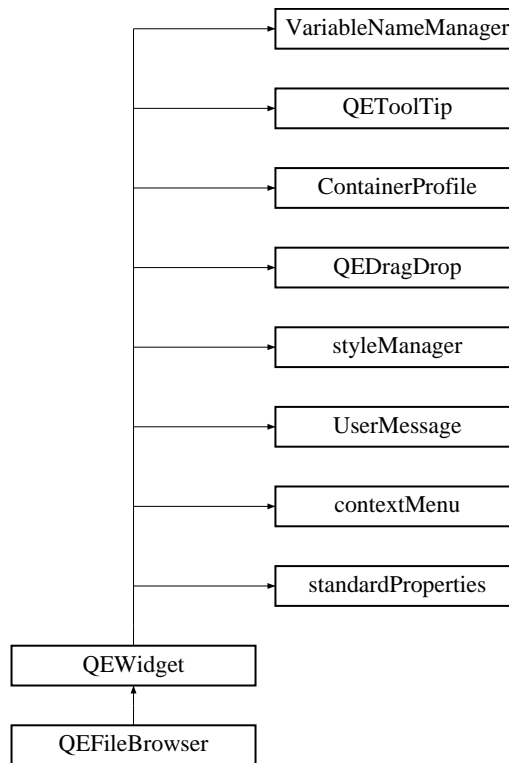
- **QEDragDrop** (QWidget *ownerIn)

- void **qcaDragEnterEvent** (QDragEnterEvent *event)
- void **qcaDropEvent** (QDropEvent *event)
- void **qcaMousePressEvent** (QMouseEvent *event)
- virtual void **setDrop** (QVariant)
- virtual QVariant **getDrop** ()
- void **setAllowDrop** (bool allowDropIn)
- bool **getAllowDrop** ()

- /home/rhydera/epicsqt/framework/widgets/include/QEDragDrop.h
- /home/rhydera/epicsqt/framework/widgets/src/QEDragDrop.cpp

10.62 QFileBrowser Class Reference

Inheritance diagram for QFileBrowser:



Public Types

- enum **detailsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }

Signals

- void **selected** (QString pFilename)

Public Member Functions

- **QFileBrowser** (QWidget *pParent=0)
- 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 **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 **setDetailsLayout** (int pValue)
- int **getDetailsLayout** ()
- void **updateTable** ()
- void **setDetailsLayoutProperty** (detailsLayoutProperty pDetailsLayout)
- detailsLayoutProperty **getDetailsLayoutProperty** ()

Protected Attributes

- QLineEdit * **qlineEditDirectoryPath**
- QPushButton * **qPushButtonDirectoryBrowser**
- QPushButton * **qPushButtonRefresh**
- [_QTableWidgetFileBrowser](#) * **qTableWidgetFileBrowser**
- QString **fileFilter**
- bool **showFileExtension**
- int **detailsLayout**

Properties

- QString **directoryPath**
- bool **showDirectoryPath**
- bool **showDirectoryBrowser**
- bool **showRefresh**
- bool **showColumnTime**
- bool **showColumnSize**
- bool **showColumnFilename**
- detailsLayoutProperty **detailsLayout**

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

- /home/rhydera/epicsqt/framework/widgets/QFileBrowser/QFileBrowser.h
- /home/rhydera/epicsqt/framework/widgets/QFileBrowser/QFileBrowser.cpp

10.63 QFileBrowserManager Class Reference

Public Member Functions

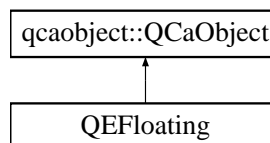
- **QFileBrowserManager** (QObject *pParent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *pParent)
- void **initialize** (QDesignerFormEditorInterface *pCore)

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

- /home/rhydera/epicsqt/framework/widgets/QFileBrowser/QFileBrowserManager.h
- /home/rhydera/epicsqt/framework/widgets/QFileBrowser/QFileBrowserManager.cpp

10.64 QEFloating Class Reference

Inheritance diagram for QEFloating:



Public Slots

- void **writeFloating** (const 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/rhydera/epicsqt/framework/data/include/QEFloating.h
- /home/rhydera/epicsqt/framework/data/src/QEFloating.cpp

10.65 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

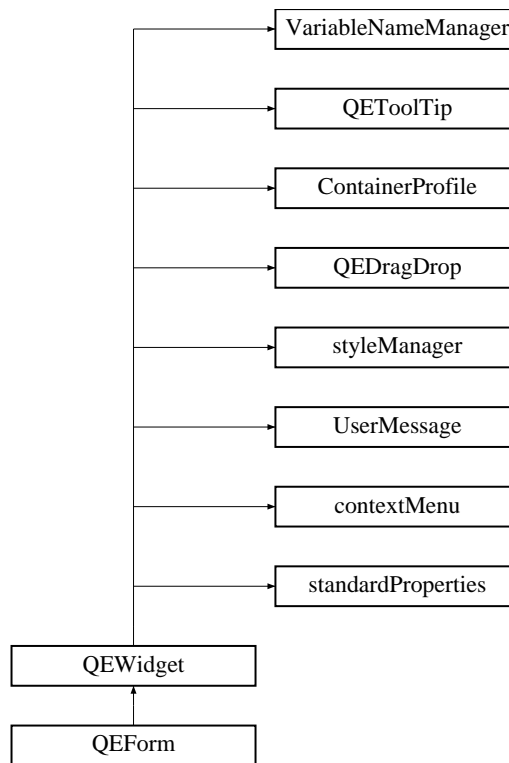
- double **formatFloating** (const QVariant &value)
- QVector< double > **formatFloatingArray** (const QVariant &value)
- QVariant **formatValue** (const double &floatingValue, generic::generic_types valueType)
- void **setPrecision** (unsigned int precision)
- void **setFormat** (formats format)
- unsigned int **getPrecision** ()
- int **getFormat** ()

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

- /home/rhydera/epicsqt/framework/data/include/QEFloatingFormatting.h
- /home/rhydera/epicsqt/framework/data/src/QEFloatingFormatting.cpp

10.66 QEForm Class Reference

Inheritance diagram for QEForm:



Public Types

- enum **creationOptions** { **CREATION_OPTION_OPEN**, **CREATION_OPTION_NEW_TAB**, **CREATION_OPTION_NEW_WINDOW** }
- enum **MessageFilterOptions** { **Match** = UserMessage::MESSAGE_FILTER_MATCH, **None** = UserMessage::MESSAGE_FILTER_NONE }

Public Slots

- bool **readUiFile** ()
- void **launchGui** (QString guiName, QForm::creationOptions createOption)

Public Member Functions

- **QForm** (QWidget *parent=0)
- **QForm** (const QString &uifileNameIn, QWidget *parent=0)
- void **commonInit** (const bool alertIfUINoFoundIn)
- QString **getASGuiTitle** ()
- QString **getFullFileName** ()
- void **setVariableNameAndSubstitutions** (QString variableNameIn, QString variableNameSubstitutionsIn, unsigned int variableIndex)

- void **setUiFileName** (QString uiFile)
- QString **getUiFileName** ()
- void **setHandleGuiLaunchRequests** (bool handleGuiLaunchRequests)
- bool **getHandleGuiLaunchRequests** ()
- void **setResizeContents** (bool resizeContentsIn)
- bool **getResizeContents** ()
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)
- QString **getVariableNameSubstitutionsProperty** ()
- MessageFilterOptions **getMessageFormFilter** ()
- void **setMessageFormFilter** (MessageFilterOptions messageFormFilter)
- MessageFilterOptions **getMessageSourceFilter** ()
- void **setMessageSourceFilter** (MessageFilterOptions messageSourceFilter)

Protected Member Functions

- void **setVariableNameSubstitutions** (QString variableNameSubstitutionsIn)

Protected Attributes

- QString **uiFileName**
- QString **fullUiFileName**
- bool **handleGuiLaunchRequests**
- bool **resizeContents**

Properties

- QString **uiFile**
- QString **variableSubstitutions**
- unsigned int
- MessageFilterOptions **messageFormFilter**
- MessageFilterOptions **messageSourceFilter**

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

- /home/rhydera/epicsqt/framework/widgets/QForm/QForm.h
- /home/rhydera/epicsqt/framework/widgets/QForm/QForm.cpp

10.67 QFormManager Class Reference

Public Member Functions

- **QFormManager** (QObject *parent=0)
- bool **isContainer** () const

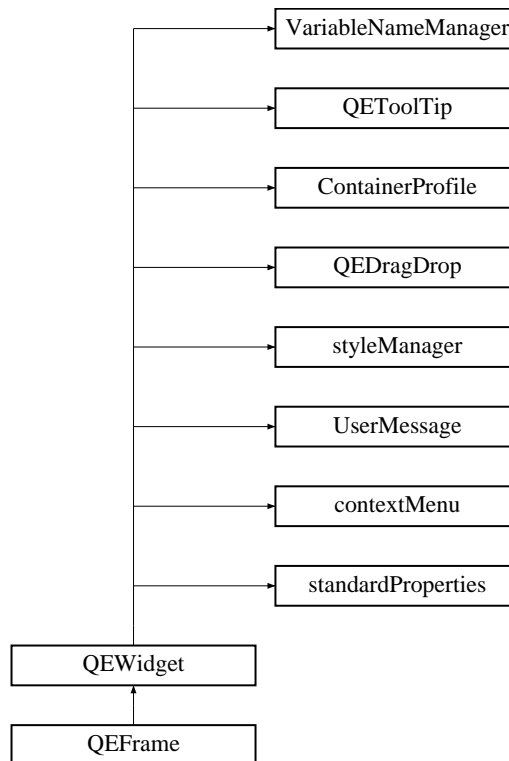
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEForm/QEFormManager.h
- /home/rhydera/epicsqt/framework/widgets/QEForm/QEFormManager.cpp

10.68 QEFrame Class Reference

Inheritance diagram for QEFrame:



Public Types

- enum [UserLevels](#) { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

Public Slots

- void [requestEnabled](#) (const bool &state)

Public Member Functions

- bool [isEnabled](#) () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void [setEnabled](#) (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- QEFrame** (QWidget *parent=0)
- QSize [sizeHint](#) () const

Properties

- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned int
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)

10.68.1 Member Function Documentation

10.68.1.1 void QFrame::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.68.2 Property Documentation

10.68.2.1 bool QFrame::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 [QEDragDrop](#).

10.68.2.2 bool QFrame::enabled [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.68.2.3 unsigned QFrame::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.

10.68.2.4 UserLevels QFrame::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 through the [QELogin](#) widget, or programatically 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'.

10.68.2.5 QString QFrame::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.

10.68.2.6 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.

10.68.2.7 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.

10.68.2.8 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 through the [QELogin](#) widget, or programatically 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'.

10.68.2.9 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 from [QEWidget](#).

10.68.2.10 bool QEFrame::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/rhydera/epicsqt/framework/widgets/QEFrame/QEFrame.h
- /home/rhydera/epicsqt/framework/widgets/QEFrame/QEFrame.cpp

10.69 QEFrameManager Class Reference

Public Member Functions

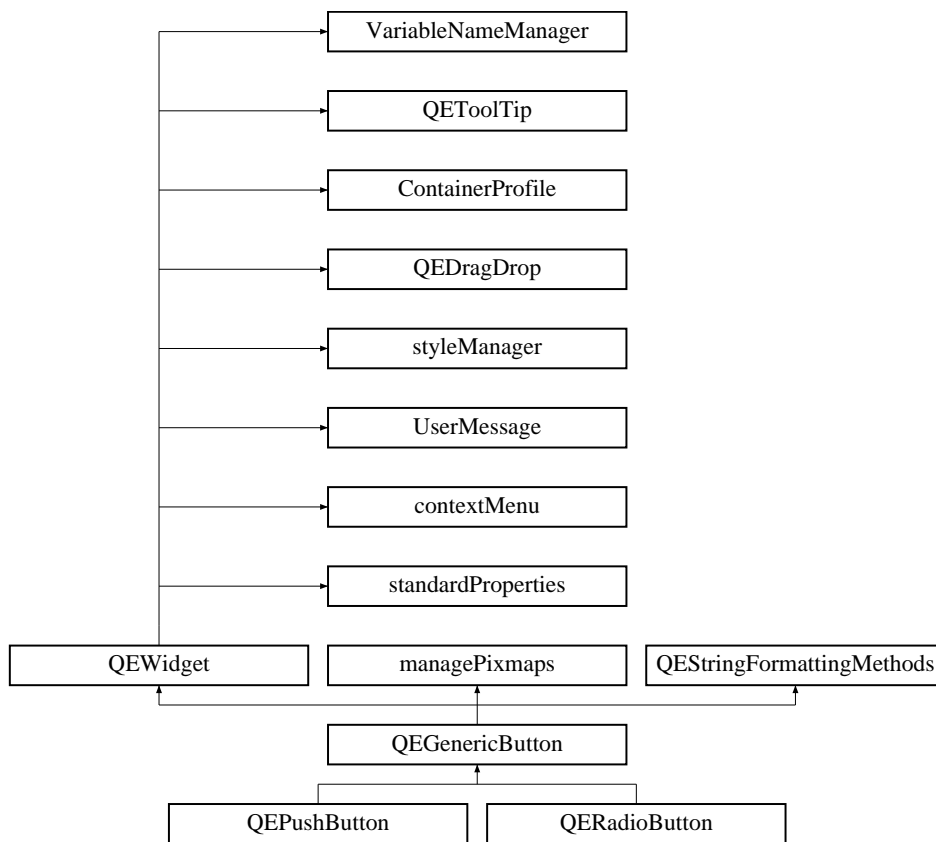
- **QEFrameManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEFrame/QEFrameManager.h
- /home/rhydera/epicsqt/framework/widgets/QEFrame/QEFrameManager.cpp

10.70 QEGenericButton Class Reference

Inheritance diagram for QEGenericButton:



Public Types

- enum **updateOptions** { **UPDATE_TEXT**, **UPDATE_ICON**, **UPDATE_TEXT_AND_ICON**, **UPDATE_STATE** }

Public Member Functions

- **QEGenericButton** (QWidget *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 **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 **setGuiName** (QString guiName)
- QString **getGuiName** ()
- void **setCreationOption** (QForm::creationOptions creationOption)
- QForm::creationOptions **getCreationOption** ()
- void **setLabelTextProperty** (QString labelTextIn)
- QString **getLabelTextProperty** ()
- void **onGeneralMessage** (QString message)

Protected Member Functions

- void **connectionChanged** ([QCaConnectionInfo](#) &connectionInfo)
- void **setGenericButtonText** (const QString &text, [QCaAlarmInfo](#) &alarmInfo, [QCaDateTime](#) &, const unsigned int &variableIndex)
- void **userPressed** ()
- void **userReleased** ()
- void **userClicked** (bool checked)
- void **launchGui** (QString guiName, QForm::creationOptions creationOption)
- virtual updateOptions **getDefaultUpdateOption** ()=0
- void **establishConnection** (unsigned int variableIndex)
- bool **getIsConnected** ()
- void **setup** ()

Protected Attributes

- Qt::Alignment **textAlignment**
- QString **password**
- bool **writeOnPress**
- bool **writeOnRelease**
- bool **writeOnClick**
- QString **releaseText**

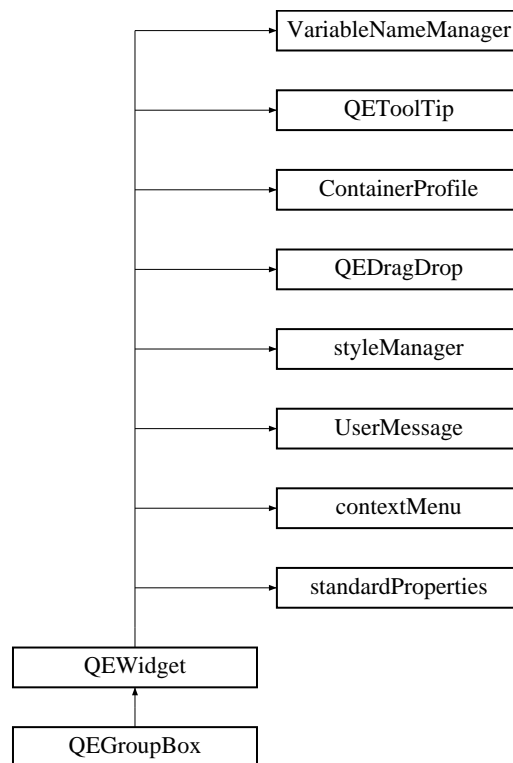
- QString **pressText**
- QString **clickText**
- QString **clickCheckedText**
- QString **program**
- QStringList **arguments**
- QString **guiName**
- QForm::creationOptions **creationOption**
- bool **localEnabled**
- updateOptions **updateOption**
- QString **labelText**

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

- /home/rhydera/epicsqt/framework/widgets/QEButton/QEGenericButton.h
- /home/rhydera/epicsqt/framework/widgets/QEButton/QEGenericButton.cpp

10.71 QEGroupBox Class Reference

Inheritance diagram for QEGroupBox:



Public Types

- enum [UserLevels](#) { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

Public Slots

- void [requestEnabled](#) (const bool &state)

Public Member Functions

- bool [isEnabled](#) () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void [setEnabled](#) (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- **QGroupBox** (QWidget *parent=0)
- QSize **sizeHint** () const

Properties

- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)

10.71.1 Member Function Documentation

10.71.1.1 `void QEGroupBox::requestEnabled (const bool & state) [inline, slot]`

Similar to standard `setEnabled` slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.71.2 Property Documentation

10.71.2.1 `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.

Reimplemented from [QEDragDrop](#).

10.71.2.2 `bool QEGroupBox::enabled [read, write]`

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.71.2.3 `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.

10.71.2.4 `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 through the [QELogin](#) widget, or programatically 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'.

10.71.2.5 `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.

10.71.2.6 QString QGroupBox::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.

10.71.2.7 QString QGroupBox::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.

10.71.2.8 UserLevels QGroupBox::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 through the [QELogin](#) widget, or programatically 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'.

10.71.2.9 bool QGroupBox::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 [QEWidget](#).

10.71.2.10 bool QGroupBox::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/rhydera/epicsqt/framework/widgets/QEGroupBox/QEGroupBox.h
- /home/rhydera/epicsqt/framework/widgets/QEGroupBox/QEGroupBox.cpp

10.72 QEGroupBoxManager Class Reference

Public Member Functions

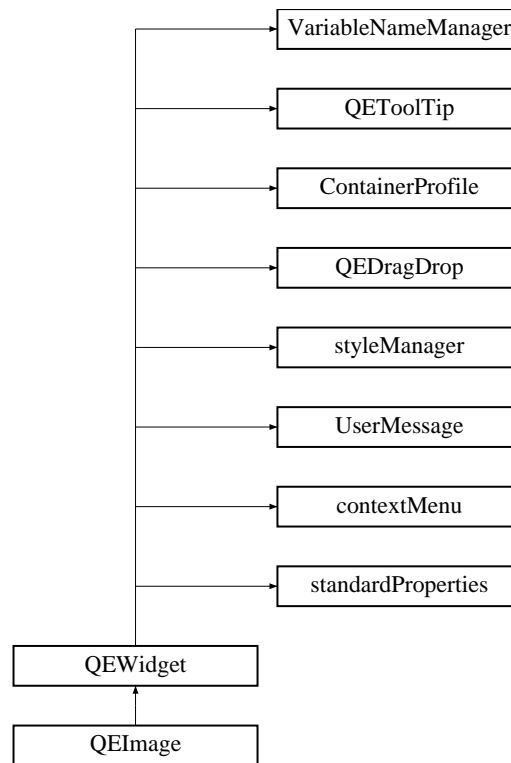
- **QEGroupBoxManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEGroupBox/QEGroupBoxManager.h
- /home/rhydera/epicsqt/framework/widgets/QEGroupBox/QEGroupBoxManager.cpp

10.73 QEImage Class Reference

Inheritance diagram for QEImage:



Public Types

- enum **selectOptions** {
SO_NONE, **SO_PANNING**, **SO_VSLICE**, **SO_HSLICE**,
SO_AREA, **SO_PROFILE**, **SO_TARGET**, **SO_BEAM** }
- enum **formatOptions** { **GREY8**, **GREY12**, **GREY16**, **RGB_888** }
- enum **resizeOptions** { **RESIZE_OPTION_ZOOM**, **RESIZE_OPTION_FIT** }
- enum **rotationOptions** { **ROTATION_0**, **ROTATION_90_RIGHT**, **ROTATION_90_LEFT**, **ROTATION_180** }
- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }
User friendly enumerations for userLevelVisibility and userLevelEnabled properties - refer to userLevelVisibility and userLevelEnabled properties and userLevel enumeration for details.
- enum **FormatOptions** { **Grey_8** = QEImage::GREY8, **Grey_12** = QEImage::GREY12, **Grey_16** = QEImage::GREY16, **RGB** = QEImage::RGB_888 }
- enum **ResizeOptions** { **Zoom** = QEImage::RESIZE_OPTION_ZOOM, **Fit** = QEImage::RESIZE_OPTION_FIT }
- enum **RotationOptions** { **NoRotation** = QEImage::ROTATION_0, **Rotate90Right** = QEImage::ROTATION_90_RIGHT, **Rotate90Left** = QEImage::ROTATION_90_LEFT, **Rotate180** = QEImage::ROTATION_180 }

Public Slots

- void **setSelectPanMode** ()
- void **setSelectVSliceMode** ()
- void **setSelectHSliceMode** ()
- void **setSelectAreaMode** ()
- void **setSelectProfileMode** ()
- void **setSelectTargetMode** ()
- void **setSelectBeamMode** ()
- void **pauseClicked** ()
- void **saveClicked** ()
- void **roiClicked** ()
- void **resetRoiClicked** ()
- void **targetClicked** ()
- void **requestEnabled** (const bool &state)

Signals

- void **dbValueChanged** (const QString &out)
- void **requestResend** ()

Public Member Functions

- **QEImage** (QWidget *parent=0)
- **QEImage** (const QString &variableName, QWidget *parent=0)
- selectOptions **getSelectionOption** ()
- void **setFormatOption** (formatOptions formatOption)
- formatOptions **getFormatOption** ()
- void **setResizeOption** (resizeOptions resizeOptionIn)
- resizeOptions **getResizeOption** ()
- void **setZoom** (int zoomIn)
- int **getZoom** ()
- void **setRotation** (rotationOptions rotationIn)
- rotationOptions **getRotation** ()
- void **setHorizontalFlip** (bool flipHozIn)
- bool **getHorizontalFlip** ()
- void **setVerticalFlip** (bool flipVertIn)
- bool **getVerticalFlip** ()
- void **setInitialHozScrollPos** (int initialHosScrollPosIn)
- int **getInitialHozScrollPos** ()
- void **setInitialVertScrollPos** (int initialVertScrollPosIn)
- int **getInitialVertScrollPos** ()
- void **setDisplayAcquirePeriod** (bool displayAcquirePeriodIn)
- bool **getDisplayAcquirePeriod** ()
- void **setDisplayExposureTime** (bool displayExposureTimeIn)
- bool **getDisplayExposureTime** ()

- void **setDisplayButtonBar** (bool displayButtonBarIn)
- bool **getDisplayButtonBar** ()
- void **setShowTime** (bool pValue)
- bool **getShowTime** ()
- void **setVertSliceMarkupColor** (QColor pValue)
- QColor **getVertSliceMarkupColor** ()
- void **setHozSliceMarkupColor** (QColor pValue)
- QColor **getHozSliceMarkupColor** ()
- void **setProfileMarkupColor** (QColor pValue)
- QColor **getProfileMarkupColor** ()
- void **setAreaMarkupColor** (QColor pValue)
- QColor **getAreaMarkupColor** ()
- void **setTargetMarkupColor** (QColor pValue)
- QColor **getTargetMarkupColor** ()
- void **setBeamMarkupColor** (QColor pValue)
- QColor **getBeamMarkupColor** ()
- void **setTimeMarkupColor** (QColor pValue)
- QColor **getTimeMarkupColor** ()
- void **setDisplayCursorPixelInfo** (bool displayCursorPixelInfoIn)
- bool **getDisplayCursorPixelInfo** ()
- void **setContrastReversal** (bool contrastReversalIn)
- bool **getContrastReversal** ()
- void **setEnabledPan** (bool enablePanIn)
- bool **getEnabledPan** ()
- void **setEnabledVertSliceSelection** (bool enableVSliceSelectionIn)
- bool **getEnabledVertSliceSelection** ()
- void **setEnabledHozSliceSelection** (bool enableHSliceSelectionIn)
- bool **getEnabledHozSliceSelection** ()
- void **setEnabledAreaSelection** (bool enableAreaSelectionIn)
- bool **getEnabledAreaSelection** ()
- void **setEnabledProfileSelection** (bool enableProfileSelectionIn)
- bool **getEnabledProfileSelection** ()
- void **setEnabledTargetSelection** (bool enableTargetSelectionIn)
- bool **getEnabledTargetSelection** ()
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)
- QString **getVariableNameSubstitutionsProperty** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.

- void **setFormatOptionProperty** (FormatOptions formatOption)
- FormatOptions **getFormatOptionProperty** ()
- void **setResizeOptionProperty** (ResizeOptions resizeOption)
- ResizeOptions **getResizeOptionProperty** ()
- void **setRotationProperty** (RotationOptions rotation)
- RotationOptions **getRotationProperty** ()

Protected Types

- enum **variableIndexes** {
IMAGE_VARIABLE, **WIDTH_VARIABLE**, **HEIGHT_VARIABLE**, **ROI_X_VARIABLE**,
ROI_Y_VARIABLE, **ROI_W_VARIABLE**, **ROI_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, **QEIMAGE_NUM_VARIABLES** }

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 v)
- void **resizeEvent** (QResizeEvent *)

Protected Attributes

- [QEIntegerFormatting](#) **integerFormatting**
- resizeOptions **resizeOption**
- int **zoom**
- rotationOptions **rotation**
- bool **flipVert**

- bool **flipHoz**
- int **initialHozScrollPos**
- int **initialVertScrollPos**
- bool **displayButtonBar**

Properties

- QString [imageVariable](#)
- QString [widthVariable](#)
- QString [heightVariable](#)
- QString [regionOfInterestXVariable](#)
- QString [regionOfInterestYVariable](#)
- QString [regionOfInterestWVariable](#)
- QString [regionOfInterestHVariable](#)
- QString [targetXVariable](#)
- QString [targetYVariable](#)
- QString [beamXVariable](#)
- QString [beamYVariable](#)
- QString [targetTriggerVariable](#)
- QString [clippingOnOffVariable](#)
- QString [clippingLowVariable](#)
- QString [clippingHighVariable](#)
- QString [variableSubstitutions](#)
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned int
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- FormatOptions **formatOption**
- bool **enableVertSliceSelection**
- bool **enableHozSliceSelection**
- bool **showTime**
- QColor **vertSliceColor**
- QColor **hozSliceColor**
- QColor **profileColor**
- QColor **areaColor**
- QColor **beamColor**
- QColor **targetColor**
- QColor **timeColor**
- ResizeOptions **resizeOption**
- RotationOptions **rotation**
- bool **verticalFlip**
- bool **horizontalFlip**
- int **initialHosScrollPos**

10.73.1 Member Function Documentation

10.73.1.1 void QEImage::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.73.2 Property Documentation

10.73.2.1 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.

Reimplemented from [QEDragDrop](#).

10.73.2.2 QString QEImage::beamXVariable [read, write]

EPICS variable name (CA PV). This variable is used to write the selected beam X position.

10.73.2.3 QString QEImage::beamYVariable [read, write]

EPICS variable name (CA PV). This variable is used to write the selected beam Y position.

10.73.2.4 QString QEImage::clippingHighVariable [read, write]

EPICS variable name (CA PV). This variable is used to write the areadetector clipping high level.

10.73.2.5 QString QEImage::clippingLowVariable [read, write]

EPICS variable name (CA PV). This variable is used to write the areadetector clipping low level.

10.73.2.6 QString QEImage::clippingOnOffVariable [read, write]

EPICS variable name (CA PV). This variable is used to write the areadetector clipping on/off command.

10.73.2.7 `bool QEImage::enabled` [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.73.2.8 `QString QEImage::heightVariable` [read, write]

EPICS variable name (CA PV). This variable is used to read the height of the image.

10.73.2.9 `QString QEImage::imageVariable` [read, write]

EPICS variable name (CA PV). This variable is used as the source the image waveform.

10.73.2.10 `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.

10.73.2.11 `QString QEImage::regionOfInterestHVariable` [read, write]

EPICS variable name (CA PV). This variable is used to write the region of interest height.

10.73.2.12 `QString QEImage::regionOfInterestWVariable` [read, write]

EPICS variable name (CA PV). This variable is used to write the region of interest width.

10.73.2.13 `QString QEImage::regionOfInterestXVariable` [read, write]

EPICS variable name (CA PV). This variable is used to write the region of interest X position.

10.73.2.14 `QString QEImage::regionOfInterestYVariable` [read, write]

EPICS variable name (CA PV). This variable is used to write the region of interest Y position.

10.73.2.15 QString QElImage::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.

10.73.2.16 QString QElImage::targetXVariable [read, write]

EPICS variable name (CA PV). This variable is used to write the selected target X position.

10.73.2.17 QString QElImage::targetYVariable [read, write]

EPICS variable name (CA PV). This variable is used to write the selected target Y position.

10.73.2.18 UserLevels QElImage::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 through the [QELogin](#) widget, or programatically 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'.

10.73.2.19 QString QElImage::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.

10.73.2.20 QString QElImage::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.

10.73.2.21 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.

10.73.2.22 UserLevels 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 through the [QELogin](#) widget, or programatically 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'.

10.73.2.23 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.

Reimplemented from [QEWidget](#).

10.73.2.24 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.

10.73.2.25 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.

10.73.2.26 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/rhydera/epicsqt/framework/widgets/QEImage/QEImage.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/QEImage.cpp

10.74 QEImageManager Class Reference

Public Member Functions

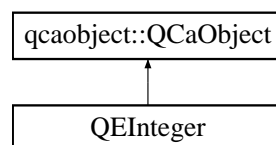
- **QEImageManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/QEImageManager.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/QEImageManager.cpp

10.75 QEInteger Class Reference

Inheritance diagram for QEInteger:



Public Slots

- void **writeInteger** (const 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/rhydera/epicsqt/framework/data/include/QEInteger.h
- /home/rhydera/epicsqt/framework/data/src/QEInteger.cpp

10.76 QEIntegerFormatting Class Reference

Public Member Functions

- long **formatInteger** (const QVariant &value)
- QVector< long > **formatIntegerArray** (const QVariant &value)
- QVariant **formatValue** (const long &integerValue, generic::generic_types value-Type)
- void **setRadix** (unsigned int radix)
- unsigned int **getPrecision** ()
- unsigned int **getRadix** ()

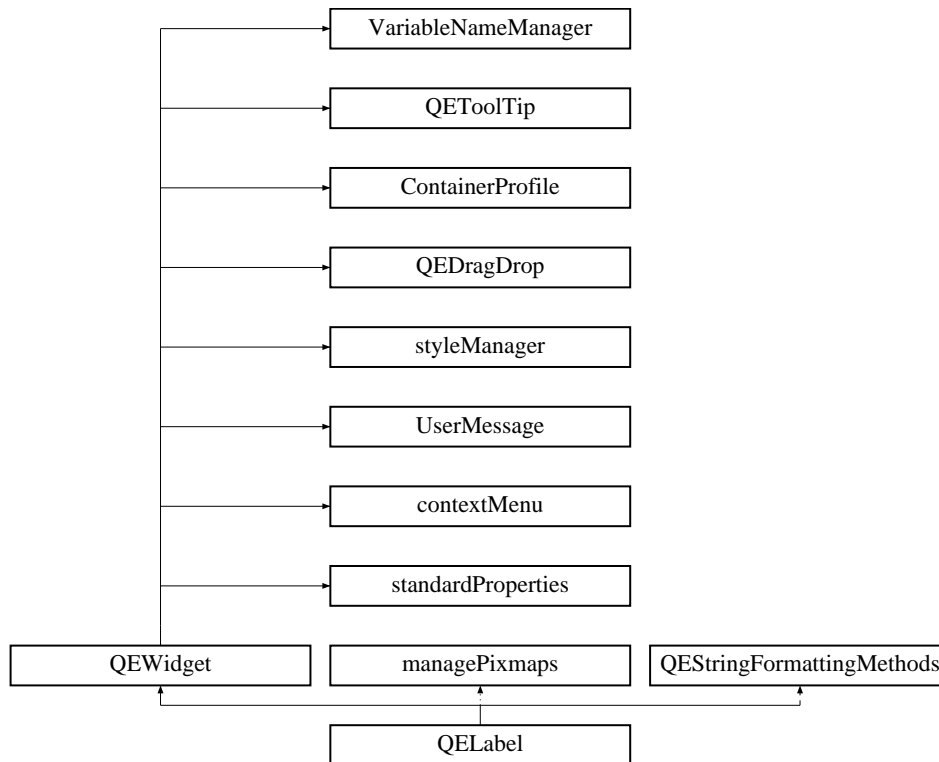
The documentation for this class was generated from the following files:

- /home/rhydera/epicsqt/framework/data/include/QEIntegerFormatting.h
- /home/rhydera/epicsqt/framework/data/src/QEIntegerFormatting.cpp

10.77 QELabel Class Reference

```
#include <QELabel.h>
```

Inheritance diagram for QELabel:



Public Types

- enum `updateOptions` { `UPDATE_TEXT`, `UPDATE_PIXMAP` }
- enum `UserLevels` { `User` = `USERLEVEL_USER`, `Scientist` = `USERLEVEL_SCIENTIST`, `Engineer` = `USERLEVEL_ENGINEER` }
User friendly enumerations for userLevelVisibility and userLevelEnabled properties - refer to userLevelVisibility and userLevelEnabled properties and userLevel enumeration for details.
- 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` }
User friendly enumerations for format property - refer to QStringFormatting::formats for details.
- enum `Notations` { **Fixed** = `QStringFormatting::NOTATION_FIXED`, **Scientific** = `QStringFormatting::NOTATION_SCIENTIFIC`, **Automatic** = `QStringFormatting::NOTATION_AUTOMATIC` }
User friendly enumerations for notation property - refer to QStringFormatting::notations for details.

- enum [ArrayActions](#) { **Append** = QQStringFormatting::APPEND, **Ascii** = QQStringFormatting::ASCII, **Index** = QQStringFormatting::INDEX }
User friendly enumerations for arrayAction property - refer to QQStringFormatting::arrayActions for details.
- enum [UpdateOptions](#) { **Text** = QELabel::UPDATE_TEXT, **Picture** = QELabel::UPDATE_PICTURE }
User friendly enumerations for updateOption property - refer to [QELabel::updateOptions](#) for details.

Public Slots

- void [requestEnabled](#) (const bool &state)

Signals

- void [dbValueChanged](#) (const QString &out)
- void [requestResend](#) ()
Internal use only - Used when a property changes to force a redisplay of the current data.

Public Member Functions

- [QELabel](#) (QWidget *parent=0)
- [QELabel](#) (const QString &variableName, QWidget *parent=0)
- bool [isEnabled](#) () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void [setEnabled](#) (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- 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.
- 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.
- void [setPixmap0Property](#) (QPixmap pixmap)
Access function for 'pixmap0' property - refer to 'pixmap0' property for details.
- void [setPixmap1Property](#) (QPixmap pixmap)
Access function for 'pixmap1' property - refer to 'pixmap1' property for details.
- void [setPixmap2Property](#) (QPixmap pixmap)
Access function for 'pixmap2' property - refer to 'pixmap2' property for details.
- void [setPixmap3Property](#) (QPixmap pixmap)
Access function for 'pixmap3' property - refer to 'pixmap3' property for details.
- void [setPixmap4Property](#) (QPixmap pixmap)
Access function for 'pixmap4' property - refer to 'pixmap4' property for details.
- void [setPixmap5Property](#) (QPixmap pixmap)
Access function for 'pixmap5' property - refer to 'pixmap5' property for details.
- void [setPixmap6Property](#) (QPixmap pixmap)
Access function for 'pixmap6' property - refer to 'pixmap6' property for details.
- void [setPixmap7Property](#) (QPixmap pixmap)
Access function for 'pixmap7' property - refer to 'pixmap7' property for details.
- QPixmap [getPixmap0Property](#) ()
Access function for 'pixmap0' property - refer to 'pixmap0' property for details.
- QPixmap [getPixmap1Property](#) ()
Access function for 'pixmap1' property - refer to 'pixmap1' property for details.
- QPixmap [getPixmap2Property](#) ()
Access function for 'pixmap2' property - refer to 'pixmap2' property for details.
- QPixmap [getPixmap3Property](#) ()
Access function for 'pixmap3' property - refer to 'pixmap3' property for details.
- QPixmap [getPixmap4Property](#) ()
Access function for 'pixmap4' property - refer to 'pixmap4' property for details.
- QPixmap [getPixmap5Property](#) ()
Access function for 'pixmap5' property - refer to 'pixmap5' property for details.
- QPixmap [getPixmap6Property](#) ()
Access function for 'pixmap6' property - refer to 'pixmap6' property for details.
- QPixmap [getPixmap7Property](#) ()
Access function for 'pixmap7' property - refer to 'pixmap7' property for details.

Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned int
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- int [precision](#)
- bool [useDbPrecision](#)
- bool [leadingZero](#)
- bool [trailingZeros](#)
- bool [addUnits](#)
- QString [localEnumeration](#)
- [Formats](#) [format](#)
- [Notations](#) [notation](#)
- [ArrayActions](#) [arrayAction](#)
- [UpdateOptions](#) [updateOption](#)
- QPixmap [pixmap0](#)
- QPixmap [pixmap1](#)
- QPixmap [pixmap2](#)
- QPixmap [pixmap3](#)
- QPixmap [pixmap4](#)
- QPixmap [pixmap5](#)
- QPixmap [pixmap6](#)
- QPixmap [pixmap7](#)

10.77.1 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.

10.77.2 Member Enumeration Documentation

10.77.2.1 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.

10.77.3 Constructor & Destructor Documentation

10.77.3.1 QELabel::QELabel (QWidget * *parent* = 0)

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

10.77.3.2 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.

10.77.4 Member Function Documentation

10.77.4.1 void QELabel::dbValueChanged (const QString & *out*) [signal]

Signal a data update has occurred and supply the new formatted text. Note, the formatted text is still signaled even when it is being used to select a pixmap for the label.

10.77.4.2 void QELabel::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard `setEnabled` slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.77.5 Property Documentation

10.77.5.1 bool QELabel::addUnits [read, write]

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

10.77.5.2 `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.

Reimplemented from [QEDragDrop](#).

10.77.5.3 `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'.

10.77.5.4 `bool QELabel::enabled` [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.77.5.5 `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).

10.77.5.6 `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.

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

Index used to select a single item of data for formatting from an array of data. Default is 0. Only used when the `arrayAction` property is INDEX. Refer to the `arrayAction` property for more details.

10.77.5.7 bool QELabel::leadingZero [read, write]

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

10.77.5.8 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.

10.77.5.9 Notations QELabel::notation [read, write]

Notation used for numerical formatting. Default is fixed.

10.77.5.10 QPixmap QELabel::pixmap0 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 0.

10.77.5.11 QPixmap QELabel::pixmap1 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 1.

10.77.5.12 QPixmap QELabel::pixmap2 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 2.

10.77.5.13 QPixmap QELabel::pixmap3 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 3.

10.77.5.14 QPixmap QELabel::pixmap4 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 4.

10.77.5.15 QPixmap QELabel::pixmap5 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 5.

10.77.5.16 QPixmap QELabel::pixmap6 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 6.

10.77.5.17 QPixmap QELabel::pixmap7 [read, write]

Pixmap displayed when updateOption property is 'Picture' and data is interpreted as 7.

10.77.5.18 int QELabel::precision [read, write]

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

10.77.5.19 bool QELabel::trailingZeros [read, write]

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

10.77.5.20 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.

10.77.5.21 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.

10.77.5.22 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 through the [QELogin](#) widget, or programatically 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'.

10.77.5.23 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.

10.77.5.24 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.

10.77.5.25 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.

10.77.5.26 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 through the [QELogin](#) widget, or programatically 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'.

10.77.5.27 QString QELabel::variable [read, write]

EPICS variable name (CA PV)

10.77.5.28 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.

Reimplemented from [QEWidget](#).

10.77.5.29 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 are also used for other purposes.

10.77.5.30 bool QELabel::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/rhydera/epicsqt/framework/widgets/QELabel/QELabel.h
- /home/rhydera/epicsqt/framework/widgets/QELabel/QELabel.cpp

10.78 QELabelManager Class Reference

Public Member Functions

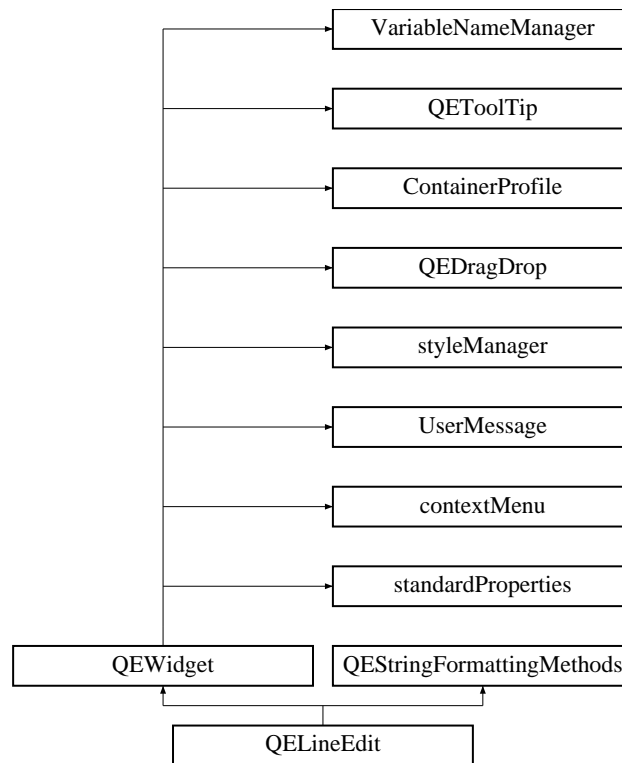
- **QELabelManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QELabel/QELabelManager.h
- /home/rhydera/epicsqt/framework/widgets/QELabel/QELabelManager.cpp

10.79 QELineEdit Class Reference

Inheritance diagram for QELineEdit:



Public Types

- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }
User friendly enumerations for userLevelVisibility and userLevelEnabled properties - refer to userLevelVisibility and userLevelEnabled properties and userLevel enumeration for details.
- 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 }
User friendly enumerations for format property - refer to QQStringFormatting::formats for details.
- enum **Notations** { **Fixed** = QQStringFormatting::NOTATION_FIXED, **Scientific** = QQStringFormatting::NOTATION_SCIENTIFIC, **Automatic** = QQStringFormatting::NOTATION_AUTOMATIC }
User friendly enumerations for notation property - refer to QQStringFormatting::notations for details.
- enum **ArrayActions** { **Append** = QQStringFormatting::APPEND, **Ascii** = QQStringFormatting::ASCII, **Index** = QQStringFormatting::INDEX }

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

Public Slots

- void [requestEnabled](#) (const bool &state)
- void **writeNow** ()

Signals

- void **dbValueChanged** (const QString &out)
- void **userChange** (const QString &oldValue, const QString &newValue, const QString &lastValue)
- void **requestResend** ()

Public Member Functions

- bool [isEnabled](#) () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void [setEnabled](#) (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- 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.

- **QLElineEdit** (QWidget *parent=0)
- **QLElineEdit** (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 **setSubscribe** (bool subscribe)
- bool **getSubscribe** ()
- void **setConfirmWrite** (bool confirmWrite)
- bool **getConfirmWrite** ()

Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Protected Attributes

- bool **localEnabled**

Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- bool **subscribe**
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- int [precision](#)
- bool [useDbPrecision](#)
- bool [leadingZero](#)
- bool [trailingZeros](#)

- bool [addUnits](#)
- QString [localEnumeration](#)
- [Formats](#) format
- [Notations](#) notation
- [ArrayActions](#) arrayAction
- bool **writeOnLoseFocus**
- bool **writeOnEnter**
- bool **writeOnFinish**
- bool **confirmWrite**

10.79.1 Member Function Documentation

10.79.1.1 void QLEdit::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.79.2 Property Documentation

10.79.2.1 bool QLEdit::addUnits [read, write]

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

10.79.2.2 bool QLEdit::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 [QEDragDrop](#).

10.79.2.3 [ArrayActions](#) QLEdit::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'.

10.79.2.4 `bool QLElineEdit::enabled` [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.79.2.5 `Formats QLElineEdit::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).

10.79.2.6 `unsigned QLElineEdit::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!)

Index used to select a single item of data for formatting from an array of data. Default is 0. Only used when the `arrayAction` property is `INDEX`. Refer to the `arrayAction` property for more details.

10.79.2.7 `bool QLElineEdit::leadingZero` [read, write]

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

10.79.2.8 `QString QLElineEdit::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.

10.79.2.9 `Notations QLElineEdit::notation` [read, write]

Notation used for numerical formatting. Default is fixed.

10.79.2.10 `int QLElineEdit::precision` [read, write]

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

10.79.2.11 bool QLEdit::trailingZeros [read, write]

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

10.79.2.12 bool QLEdit::useDbPrecision [read, write]

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

10.79.2.13 UserLevels QLEdit::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 through the [QELogin](#) widget, or programatically 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'.

10.79.2.14 QString QLEdit::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.

10.79.2.15 QString QLEdit::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.

10.79.2.16 QString QLEdit::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.

10.79.2.17 UserLevels QELineEdit::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 through the [QELLogin](#) widget, or programatically 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'.

10.79.2.18 QString QELineEdit::variable [read, write]

EPICS variable name (CA PV)

10.79.2.19 bool QELineEdit::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 [QEWidget](#).

10.79.2.20 QString QELineEdit::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 are also used for other purposes.

10.79.2.21 bool QELineEdit::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/rhydera/epicsqt/framework/widgets/QELineEdit/QELineEdit.h
- /home/rhydera/epicsqt/framework/widgets/QELineEdit/QELineEdit.cpp

10.80 QELineEditManager Class Reference

Public Member Functions

- **QELineEditManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const

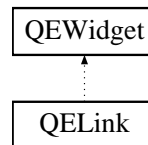
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QLineEdit/QLineEditManager.h
- /home/rhydera/epicsqt/framework/widgets/QLineEdit/QLineEditManager.cpp

10.81 QELink Class Reference

Inheritance diagram for QELink:



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 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 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 **setRunVisible** (bool visibleIn)
- bool **getRunVisible** ()
- void **setConditionProperty** (ConditionNames condition)
- ConditionNames **getConditionProperty** ()

Protected Attributes

- conditions **condition**
- QVariant **comparisonValue**
- bool **signalTrue**
- bool **signalFalse**
- QVariant **outTrueValue**
- QVariant **outFalseValue**
- bool **visible**

Properties

- ConditionNames **condition**
- QString **comparisonValue**
- QString **outTrueValue**
- QString **outFalseValue**
- bool **runVisible**

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

- /home/rhydera/epicsqt/framework/widgets/QELink/QELink.h
- /home/rhydera/epicsqt/framework/widgets/QELink/QELink.cpp

10.82 QELinkManager Class Reference

Public Member Functions

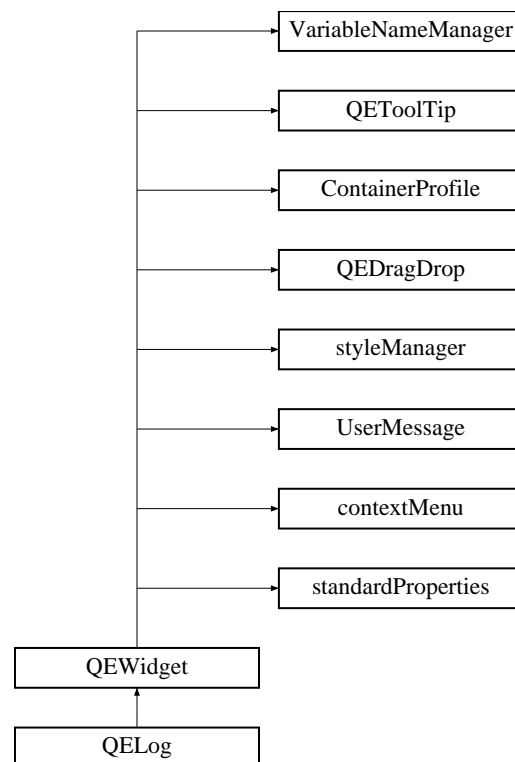
- **QELinkManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QELink/QELinkManager.h
- /home/rhydera/epicsqt/framework/widgets/QELink/QELinkManager.cpp

10.83 QELog Class Reference

Inheritance diagram for QELog:



Public Types

- enum **detailsLayoutProperty** { **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 }

Public Member Functions

- **QELog** (QWidget *pParent=0)
- void **setShowColumnTime** (bool pValue)
- bool **getShowColumnTime** ()
- 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 **setDetailsLayout** (int pValue)
- int **getDetailsLayout** ()
- 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 **setDetailsLayoutProperty** (detailsLayoutProperty pDetailsLayout)
- detailsLayoutProperty **getDetailsLayoutProperty** ()
- MessageFilterOptions **getMessageFormFilter** ()
- void **setMessageFormFilter** (MessageFilterOptions messageFormFilter)
- MessageFilterOptions **getMessageSourceFilter** ()
- void **setMessageSourceFilter** (MessageFilterOptions messageSourceFilter)

Protected Attributes

- [_QTableWidgetLog](#) * **qTableWidgetLog**
- QCheckBox * **qCheckBoxInfoMessage**
- QCheckBox * **qCheckBoxWarningMessage**
- QCheckBox * **qCheckBoxErrorMessage**
- QPushButton * **qPushButtonClear**
- QPushButton * **qPushButtonSave**
- QColor **qColorInfo**
- QColor **qColorWarning**
- QColor **qColorError**
- bool **scrollToBottom**
- int **detailsLayout**

Properties

- bool **showColumnTime**
- bool **showColumnType**
- bool **showColumnMessage**
- bool **showMessageFilter**
- bool **showClear**
- bool **showSave**
- detailsLayoutProperty **detailsLayout**

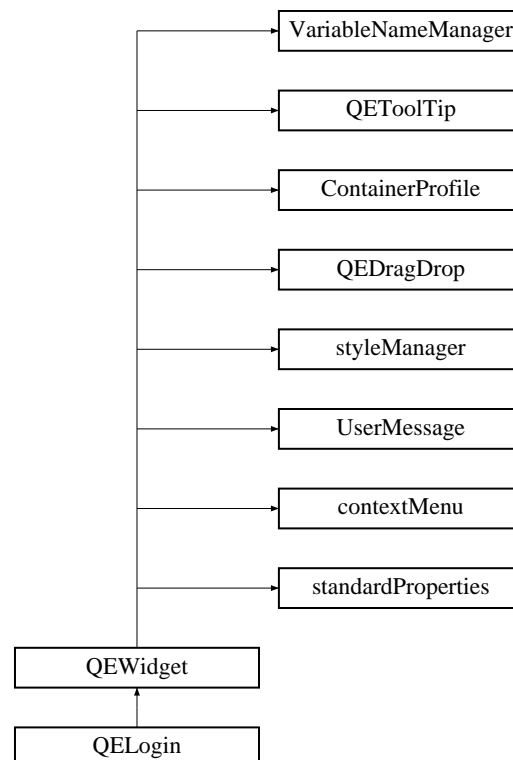
- QColor **infoColor**
- QColor **warningColor**
- QColor **errorColor**
- MessageFilterOptions **messageFormFilter**
- MessageFilterOptions **messageSourceFilter**

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

- /home/rhydera/epicsqt/framework/widgets/QELog/QELog.h
- /home/rhydera/epicsqt/framework/widgets/QELog/QELog.cpp

10.84 QELogin Class Reference

Inheritance diagram for QELogin:



Public Types

- enum **userTypesProperty** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }
- enum **detailsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }

Public Member Functions

- **QELogin** (QWidget *pParent=0)
- void **setShowUserType** (bool pValue)
- bool **getShowUserType** ()
- void **setShowLogin** (bool pValue)
- bool **getShowButtonLogin** ()
- void **setShowLogout** (bool pValue)
- bool **getShowButtonLogout** ()
- void **setUserPassword** (QString pValue)
- QString **getUserPassword** ()
- void **setScientistPassword** (QString pValue)
- QString **getScientistPassword** ()
- void **setEngineerPassword** (QString pValue)
- QString **getEngineerPassword** ()
- void **setCurrentUserType** (int pValue)
- int **getCurrentUserType** ()
- void **setDetailsLayout** (int pValue)
- int **getDetailsLayout** ()
- QString **getUserTypeName** (userLevels type)
- void **logoutCurrentUserType** ()
- void **setCurrentUserTypeProperty** (userTypesProperty pUserType)
- userTypesProperty **getCurrentUserTypeProperty** ()
- void **setDetailsLayoutProperty** (detailsLayoutProperty pDetailsLayout)
- detailsLayoutProperty **getDetailsLayoutProperty** ()

Protected Attributes

- QStack< int > **loginHistory**
- QPushButton * **qPushButtonLogin**
- QPushButton * **qPushButtonLogout**
- QLabel * **qLabelUserType**
- QString **userPassword**
- QString **scientistPassword**
- QString **engineerPassword**
- int **currentUserType**
- int **detailsLayout**

Properties

- bool **showUserType**
- bool **showLogin**
- bool **showLogout**
- userTypesProperty **currentUserType**
- detailsLayoutProperty **detailsLayout**

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

- /home/rhydera/epicsqt/framework/widgets/QELogin/QELogin.h
- /home/rhydera/epicsqt/framework/widgets/QELogin/QELogin.cpp

10.85 QELoginManager Class Reference

Public Member Functions

- **QELoginManager** (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/rhydera/epicsqt/framework/widgets/QELogin/QELoginManager.h
- /home/rhydera/epicsqt/framework/widgets/QELogin/QELoginManager.cpp

10.86 QELogManager Class Reference

Public Member Functions

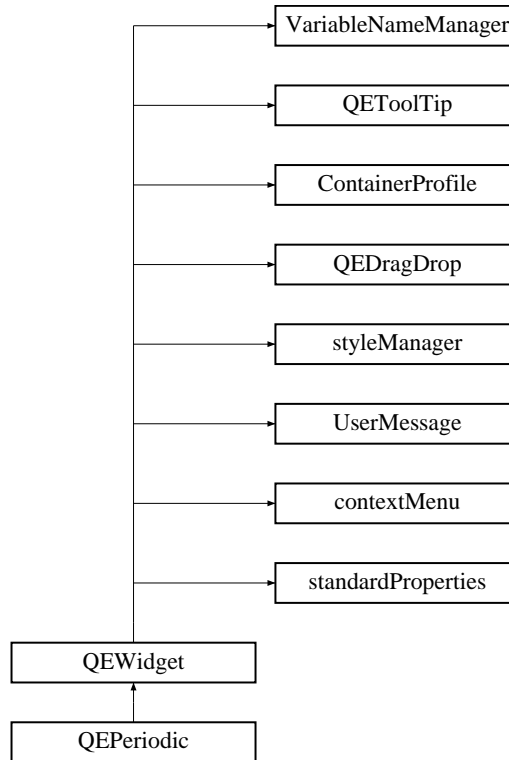
- **QELogManager** (QObject *pParent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *pParent)
- void **initialize** (QDesignerFormEditorInterface *pCore)

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

- /home/rhydera/epicsqt/framework/widgets/QELog/QELogManager.h
- /home/rhydera/epicsqt/framework/widgets/QELog/QELogManager.cpp

10.87 QEPeiodic Class Reference

Inheritance diagram for QEPeiodic:



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 [UserLevels](#) { **User** = **USERLEVEL_USER**, **Scientist** = **USERLEVEL_SCIENTIST**, **Engineer** = **USERLEVEL_ENGINEER** }

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

- 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 }

Public Slots

- void **requestEnabled** (const bool &state)

Signals

- void **dbValueChanged** (const double &out)
- void **dbElementChanged** (const QString &out)
- void **requestResend** ()

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 **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)

- QString **getVariableNameSubstitutionsProperty** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- void **setPresentationOptionProperty** (PresentationOptions presentationOption)
- PresentationOptions **getPresentationOptionProperty** ()
- void **setVariableType1Property** (VariableTypes variableType)
- void **setVariableType2Property** (VariableTypes variableType)
- VariableTypes **getVariableType1Property** ()
- VariableTypes **getVariableType2Property** ()

Public Attributes

- **userInfoStruct** **userInfo** [NUM_ELEMENTS]

Static Public Attributes

- static **elementInfoStruct** **elementInfo** [NUM_ELEMENTS]

Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

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 [enabled](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- PresentationOptions **presentationOption**
- VariableTypes **variableType1**
- VariableTypes **variableType2**
- QString **userInfo**

10.87.1 Member Function Documentation

10.87.1.1 void QEPeiodic::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.87.2 Member Data Documentation

10.87.2.1 bool QEPeiodic::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.

Reimplemented from [QEDragDrop](#).

10.87.3 Property Documentation

10.87.3.1 `bool QEPeiodic::enabled` [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.87.3.2 `unsigned QEPeiodic::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.

10.87.3.3 `QString QEPeiodic::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.

10.87.3.4 `QString QEPeiodic::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.

10.87.3.5 `UserLevels QEPeiodic::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 through the [QELogin](#) widget, or programatically 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'.

10.87.3.6 `QString QEPeiodic::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.

10.87.3.7 QString QEPeiodic::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.

10.87.3.8 QString QEPeiodic::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.

10.87.3.9 UserLevels QEPeiodic::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 through the [QELogin](#) widget, or programatically 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'.

10.87.3.10 bool QEPeiodic::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 [QEWidget](#).

10.87.3.11 QString QEPeiodic::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.

10.87.3.12 bool QEPeiodic::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.

10.87.3.13 QString QEPeiodic::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.

10.87.3.14 QString QEPeiodic::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/rhydera/epicsqt/framework/widgets/QEPeiodic/QEPeiodic.h
- /home/rhydera/epicsqt/framework/widgets/QEPeiodic/QEPeiodic.cpp

10.88 QEPeiodicComponentData 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/rhydera/epicsqt/framework/widgets/QEPeiodic/QEPeiodic.h

10.89 QEPeiodicManager Class Reference

Public Member Functions

- **QEPeiodicManager** (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/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodicManager.h
- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodicManager.cpp

10.90 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/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskMenu.h
- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskMenuExtension.cpp

10.91 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/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskMenu.h
- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodicTaskMenuExtension.cpp

10.92 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 **valueInitd** (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

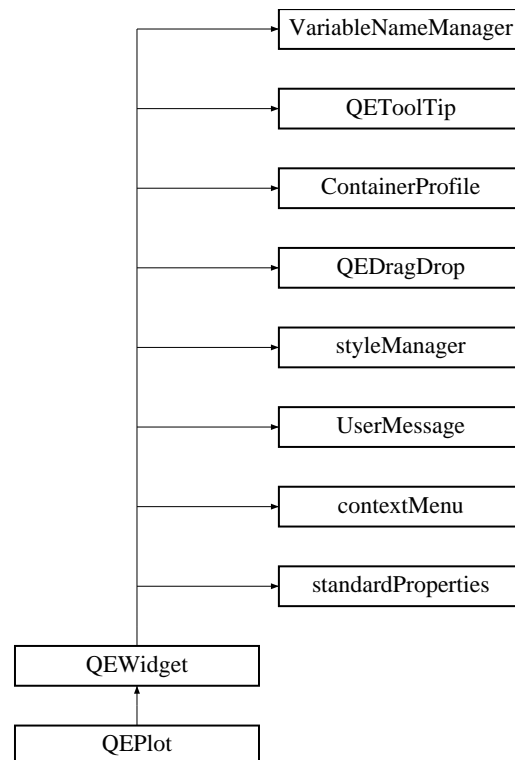
- static void **setDebugLevel** (unsigned level=0)
- static QVariant **get** (const QString &_pvName, int delay=defaultDelay)
- static QVariant **set** (QString &_pvName, const QVariant &value, int delay=-1)

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

- /home/rhydera/epicsqt/framework/data/include/qepicspv.h
- /home/rhydera/epicsqt/framework/data/src/qepicspv.cpp

10.93 QEPlot Class Reference

Inheritance diagram for QEPlot:



Public Types

- enum [UserLevels](#) { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

- enum **TraceStyles** { **Lines** = QwtPlotCurve::Lines, **Sticks** = QwtPlotCurve::Sticks, **Steps** = QwtPlotCurve::Steps, **Dots** = QwtPlotCurve::Dots }

Public Slots

- void [requestEnabled](#) (const bool &state)

Signals

- 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)
- void **setYMin** (double yMin)
- double **getYMin** ()
- void **setYMax** (double yMax)
- double **getYMax** ()
- void **setAutoScale** (bool autoScale)
- bool **getAutoScale** ()
- void **setAxisEnableX** (bool axisEnableXIn)
- bool **getAxisEnableX** ()
- void **setAxisEnableY** (bool axisEnableYIn)
- bool **getAxisEnableY** ()
- QString **getTitle** ()
- void **setBackgroundColor** (QColor backgroundColor)
- QColor **getBackgroundColor** ()
- void **setTraceStyle** (QwtPlotCurve::CurveStyle traceStyle, const unsigned int variableIndex)
- QwtPlotCurve::CurveStyle **getTraceStyle** (const unsigned int variableIndex)
- 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)
- QColor **getTraceColor1** ()
- QColor **getTraceColor2** ()
- QColor **getTraceColor3** ()
- QColor **getTraceColor4** ()
- void **setTraceLegend1** (QString traceLegend)
- void **setTraceLegend2** (QString traceLegend)
- void **setTraceLegend3** (QString traceLegend)
- void **setTraceLegend4** (QString traceLegend)
- QString **getTraceLegend1** ()
- QString **getTraceLegend2** ()
- QString **getTraceLegend3** ()
- QString **getTraceLegend4** ()
- void **setXUnit** (QString xUnit)
- QString **getXUnit** ()
- void **setYUnit** (QString yUnit)
- QString **getYUnit** ()
- void **setGridEnableMajorX** (bool gridEnableMajorXIn)
- void **setGridEnableMajorY** (bool gridEnableMajorYIn)
- void **setGridEnableMinorX** (bool gridEnableMinorXIn)
- void **setGridEnableMinorY** (bool gridEnableMinorYIn)
- bool **getGridEnableMajorX** ()

- bool **getGridEnableMajorY** ()
- bool **getGridEnableMinorX** ()
- bool **getGridEnableMinorY** ()
- void **setGridMajorColor** (QColor gridMajorColorIn)
- void **setGridMinorColor** (QColor gridMinorColorIn)
- QColor **getGridMajorColor** ()
- QColor **getGridMinorColor** ()
- void **setXStart** (double xStart)
- double **getXStart** ()
- void **setXIncrement** (double xIncrement)
- double **getXIncrement** ()
- void **setTimeSpan** (unsigned int timeSpan)
- unsigned int **getTimeSpan** ()
- void **setTickRate** (unsigned int tickRate)
- unsigned int **getTickRate** ()
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)
- QString **getVariableNameSubstitutionsProperty** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- 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 **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **mousePressEvent** (QMouseEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Protected Attributes

- [QEFloatingFormatting](#) **floatingFormatting**
- bool **localEnabled**
- bool [allowDrop](#)

Properties

- QString [variable1](#)
- QString [variable2](#)
- QString [variable3](#)
- QString [variable4](#)
- QString [variableSubstitutions](#)
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [visible](#)
- unsigned int
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- 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**

10.93.1 Member Function Documentation

10.93.1.1 `void QEPlot::requestEnabled (const bool & state) [inline, slot]`

Similar to standard `setEnabled` slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.93.2 Member Data Documentation

10.93.2.1 `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.

Reimplemented from [QEDragDrop](#).

10.93.3 Property Documentation

10.93.3.1 `bool QEPlot::enabled [read, write]`

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.93.3.2 `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.

10.93.3.3 `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 through the [QELogin](#) widget, or programatically 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'.

10.93.3.4 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.

10.93.3.5 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.

10.93.3.6 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.

10.93.3.7 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 through the [QELogin](#) widget, or programatically 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'.

10.93.3.8 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.

10.93.3.9 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.

10.93.3.10 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.

10.93.3.11 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.

10.93.3.12 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.

Reimplemented from [QEWWidget](#).

10.93.3.13 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.

10.93.3.14 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/rhydera/epicsqt/framework/widgets/QEPlot/QEPlot.h
- /home/rhydera/epicsqt/framework/widgets/QEPlot/QEPlot.cpp

10.94 QEPlotManager Class Reference

Public Member Functions

- **QEPlotManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const

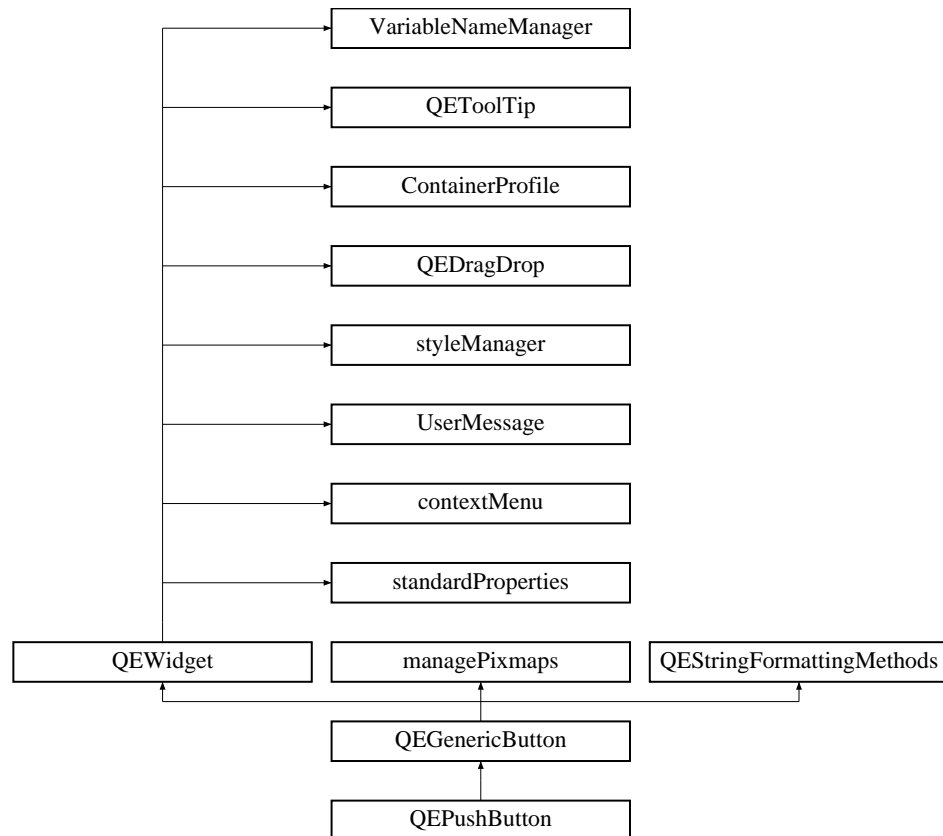
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEPlot/QEPlotManager.h
- /home/rhydera/epicsqt/framework/widgets/QEPlot/QEPlotManager.cpp

10.95 QEPushButton Class Reference

Inheritance diagram for QEPushButton:



Public Types

- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

- enum **UpdateOptions** { **Text** = QEPushButton::UPDATE_TEXT, **Icon** = QEPushButton::UPDATE_ICON, **TextAndIcon** = QEPushButton::UPDATE_TEXT_AND_ICON, **State** = QEPushButton::UPDATE_STATE }
- 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 **Notations** { **Fixed** = QStringFormatting::NOTATION_FIXED, **Scientific** = QStringFormatting::NOTATION_SCIENTIFIC, **Automatic** = QStringFormatting::NOTATION_AUTOMATIC }
- enum **CreationOptionNames** { **Open** = QForm::CREATION_OPTION_OPEN, **NewTab** = QForm::CREATION_OPTION_NEW_TAB, **NewWindow** = QForm::CREATION_OPTION_NEW_WINDOW }

Public Slots

- void **launchGui** (QString guiName, QForm::creationOptions creationOption)
- void **onGeneralMessage** (QString message)
- void **requestEnabled** (const bool &state)

Signals

- void **dbValueChanged** (const QString &out)
- void **newGui** (QString guiName, QForm::creationOptions creationOption)
- void **requestResend** ()

Public Member Functions

- **QEPushButton** (QWidget *parent=0)
- **QEPushButton** (const QString &variableName, QWidget *parent=0)
- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)
- QString **getVariableNameSubstitutionsProperty** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- void **setUpdateOptionProperty** (UpdateOptions updateOption)
- UpdateOptions **getUpdateOptionProperty** ()
- void **setPixmap0Property** (QPixmap pixmap)
- void **setPixmap1Property** (QPixmap pixmap)
- void **setPixmap2Property** (QPixmap pixmap)
- void **setPixmap3Property** (QPixmap pixmap)
- void **setPixmap4Property** (QPixmap pixmap)
- void **setPixmap5Property** (QPixmap pixmap)
- void **setPixmap6Property** (QPixmap pixmap)
- void **setPixmap7Property** (QPixmap pixmap)
- QPixmap **getPixmap0Property** ()
- QPixmap **getPixmap1Property** ()
- QPixmap **getPixmap2Property** ()
- QPixmap **getPixmap3Property** ()
- QPixmap **getPixmap4Property** ()
- QPixmap **getPixmap5Property** ()
- QPixmap **getPixmap6Property** ()
- QPixmap **getPixmap7Property** ()
- void **setFormatProperty** (Formats format)
- Formats **getFormatProperty** ()
- void **setNotationProperty** (Notations notation)
- Notations **getNotationProperty** ()
- void **setCreationOptionProperty** (CreationOptionNames creationOptionIn)
- CreationOptionNames **getCreationOptionProperty** ()

Protected Member Functions

- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Properties

- QString [variable](#)
- QString [altReadbackVariable](#)
- QString **variableSubstitutions**
- bool **subscribe**
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- UpdateOptions **updateOption**
- QPixmap **pixmap0**
- QPixmap **pixmap1**
- QPixmap **pixmap2**
- QPixmap **pixmap3**
- QPixmap **pixmap4**
- QPixmap **pixmap5**
- QPixmap **pixmap6**
- QPixmap **pixmap7**
- bool **useDbPrecision**
- bool **leadingZero**
- bool **trailingZeros**
- bool **addUnits**
- QString **localEnumeration**
- Qt::Alignment **alignment**
- Formats **format**
- Notations **notation**
- QString **password**
- bool **writeOnPress**
- bool **writeOnRelease**
- bool **writeOnClick**
- QString **pressText**
- QString **releaseText**
- QString **clickText**
- QString **clickCheckedText**
- QString **labelText**
- QString **program**
- QStringList **arguments**
- QString **guiFile**
- CreationOptionNames **creationOption**

10.95.1 Member Function Documentation

10.95.1.1 void QEPushButton::requestEnabled (const bool & state) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.95.2 Property Documentation

10.95.2.1 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.

Reimplemented from [QEDragDrop](#).

10.95.2.2 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.

10.95.2.3 bool QEPushButton::enabled [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.95.2.4 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.

10.95.2.5 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 through the [QELogin](#) widget, or programatically 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'.

10.95.2.6 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.

10.95.2.7 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.

10.95.2.8 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.

10.95.2.9 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 through the [QELogin](#) widget, or programatically 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'.

10.95.2.10 QString QEPushButton::variable [read, write]

EPICS variable name (CA PV). This variable is used for both writing (on button press), and reading if subscribed and no alternate readback variable is provided.

10.95.2.11 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.

Reimplemented from [QEWWidget](#).

10.95.2.12 `bool QEPushButton::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/rhydera/epicsqt/framework/widgets/QEButton/QEPushButton.h
- /home/rhydera/epicsqt/framework/widgets/QEButton/QEPushButton.cpp

10.96 QEPushButtonManager Class Reference

Public Member Functions

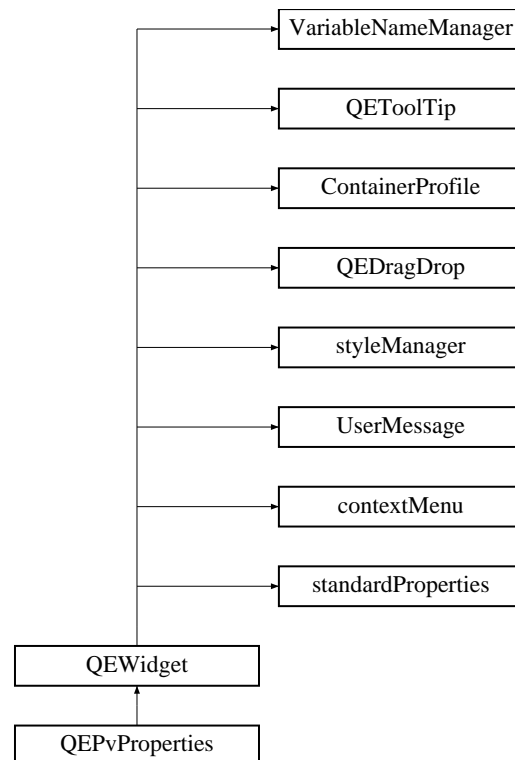
- **QEPushButtonManager** (QObject *parent=0)
- `bool isContainer` () const
- `bool isInitialized` () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEButton/QEPushButtonManager.h
- /home/rhydera/epicsqt/framework/widgets/QEButton/QEPushButtonManager.cpp

10.97 QEPvProperties Class Reference

Inheritance diagram for QEPvProperties:



Classes

- struct **WidgetHolder**

Public Types

- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

Public Slots

- void **requestEnabled** (const bool &state)

Signals

- void **setCurrentBoxIndex** (int index)

Public Member Functions

- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- **QEPvProperties** (QWidget *parent=0)
- **QEPvProperties** (const QString &variableName, QWidget *parent=0)
- QSize **sizeHint** () const
- void **establishConnection** (unsigned int variableIndex)
- void **updateToolTip** (const QString &tip)

Protected Member Functions

- void **setup** ()
- **qcaobject::QCaObject * createQcaltem** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **mousePressEvent** (QMouseEvent *event)
- QString **copyVariable** ()
- QVariant **copyData** ()
- void **paste** (QVariant s)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Properties

- QString **variable**
- QString **variableSubstitutions**
- bool **variableAsToolTip**
- bool **enabled**
- bool **allowDrop**
- bool **visible**

- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)

10.97.1 Member Function Documentation

10.97.1.1 void QEPvProperties::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.97.2 Property Documentation

10.97.2.1 bool QEPvProperties::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 [QEDragDrop](#).

10.97.2.2 bool QEPvProperties::enabled [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.97.2.3 unsigned QEPvProperties::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.

10.97.2.4 [UserLevels](#) QEPvProperties::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 through the [QELogin](#) widget, or programatically 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'.

10.97.2.5 QString QEPvProperties::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.

10.97.2.6 QString QEPvProperties::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.

10.97.2.7 QString QEPvProperties::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.

10.97.2.8 UserLevels QEPvProperties::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 through the [QELogin](#) widget, or programatically 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'.

10.97.2.9 QString QEPvProperties::variable [read, write]

EPICS variable name (CA PV)

10.97.2.10 bool QEPvProperties::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 [QEWidget](#).

10.97.2.11 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.

10.97.2.12 bool QEPvProperties::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/rhydera/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.h
- /home/rhydera/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.cpp

10.98 QEPvPropertiesManager Class Reference

Public Member Functions

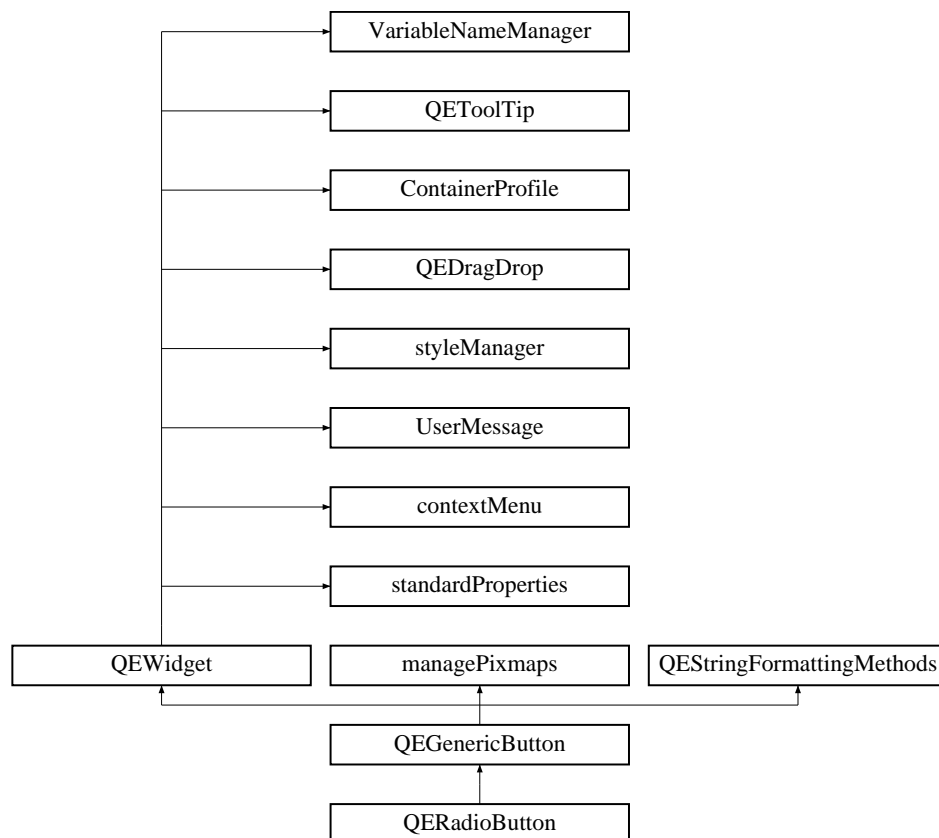
- **QEPvPropertiesManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEPvProperties/QEPvPropertiesManager.h
- /home/rhydera/epicsqt/framework/widgets/QEPvProperties/QEPvPropertiesManager.cpp

10.99 QERadioButton Class Reference

Inheritance diagram for QERadioButton:



Public Types

- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }
User friendly enumerations for userLevelVisibility and userLevelEnabled properties - refer to userLevelVisibility and userLevelEnabled properties and userLevel enumeration for details.
- enum **UpdateOptions** { **Text** = QERadioButton::UPDATE_TEXT, **Icon** = QERadioButton::UPDATE_ICON, **TextAndIcon** = QERadioButton::UPDATE_TEXT_AND_ICON, **State** = QERadioButton::UPDATE_STATE }
- enum **Formats** {
Default = QStringFormatting::FORMAT_DEFAULT, **Floating** = QStringFormatting::FORMAT_FLOATING, **Integer** = QStringFormatting::FORMAT_INTEGER, **UnsignedInteger** = QStringFormatting::FORMAT_UNSIGNEDINTEGER,
Time = QStringFormatting::FORMAT_TIME }

- enum **Notations** { **Fixed** = QStringFormatting::NOTATION_FIXED, **Scientific** = QStringFormatting::NOTATION_SCIENTIFIC, **Automatic** = QStringFormatting::NOTATION_AUTOMATIC }
- enum **CreationOptionNames** { **Open** = QForm::CREATION_OPTION_OPEN, **NewTab** = QForm::CREATION_OPTION_NEW_TAB, **NewWindow** = QForm::CREATION_OPTION_NEW_WINDOW }

Public Slots

- void **launchGui** (QString guiName, QForm::creationOptions creationOption)
- void **onGeneralMessage** (QString message)
- void **requestEnabled** (const bool &state)

Signals

- void **dbValueChanged** (const QString &out)
- void **newGui** (QString guiName, QForm::creationOptions creationOption)
- void **requestResend** ()

Public Member Functions

- **QERadioButton** (QWidget *parent=0)
- **QERadioButton** (const QString &variableName, QWidget *parent=0)
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- void **setUpdateOptionProperty** (UpdateOptions updateOption)
- UpdateOptions **getUpdateOptionProperty** ()
- void **setPixmap0Property** (QPixmap pixmap)
- void **setPixmap1Property** (QPixmap pixmap)
- void **setPixmap2Property** (QPixmap pixmap)
- void **setPixmap3Property** (QPixmap pixmap)

- void **setPixmap4Property** (QPixmap pixmap)
- void **setPixmap5Property** (QPixmap pixmap)
- void **setPixmap6Property** (QPixmap pixmap)
- void **setPixmap7Property** (QPixmap pixmap)
- QPixmap **getPixmap0Property** ()
- QPixmap **getPixmap1Property** ()
- QPixmap **getPixmap2Property** ()
- QPixmap **getPixmap3Property** ()
- QPixmap **getPixmap4Property** ()
- QPixmap **getPixmap5Property** ()
- QPixmap **getPixmap6Property** ()
- QPixmap **getPixmap7Property** ()
- void **setFormatProperty** (Formats format)
- Formats **getFormatProperty** ()
- void **setNotationProperty** (Notations notation)
- Notations **getNotationProperty** ()
- void **setCreationOptionProperty** (CreationOptionNames creationOptionIn)
- CreationOptionNames **getCreationOptionProperty** ()

Protected Member Functions

- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- bool **subscribe**
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- UpdateOptions **updateOption**
- QPixmap **pixmap0**
- QPixmap **pixmap1**
- QPixmap **pixmap2**

- QPixmap **pixmap3**
- QPixmap **pixmap4**
- QPixmap **pixmap5**
- QPixmap **pixmap6**
- QPixmap **pixmap7**
- bool **useDbPrecision**
- bool **leadingZero**
- bool **trailingZeros**
- bool **addUnits**
- Qt::Alignment **alignment**
- Formats **format**
- Notations **notation**
- QString **password**
- bool **writeOnPress**
- bool **writeOnRelease**
- bool **writeOnClick**
- QString **pressText**
- QString **releaseText**
- QString **clickText**
- QString **clickCheckedText**
- QString **labelText**
- QString **program**
- QStringList **arguments**
- QString **guiFile**
- CreationOptionNames **creationOption**

10.99.1 Member Function Documentation

10.99.1.1 void QERadioButton::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.99.2 Property Documentation

10.99.2.1 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.

Reimplemented from [QEDragDrop](#).

10.99.2.2 `bool QERadioButton::enabled` [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.99.2.3 `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.

10.99.2.4 `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 through the [QELogin](#) widget, or programatically 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'.

10.99.2.5 `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.

10.99.2.6 `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.

10.99.2.7 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.

10.99.2.8 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 through the [QELogin](#) widget, or programatically 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'.

10.99.2.9 QString QERadioButton::variable [read, write]

EPICS variable name (CA PV)

10.99.2.10 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.

Reimplemented from [QEWidget](#).

10.99.2.11 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 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.

10.99.2.12 bool QERadioButton::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/rhydera/epicsqt/framework/widgets/QEButton/QERadioButton.h
- /home/rhydera/epicsqt/framework/widgets/QEButton/QERadioButton.cpp

10.100 QERadioButtonManager Class Reference

Public Member Functions

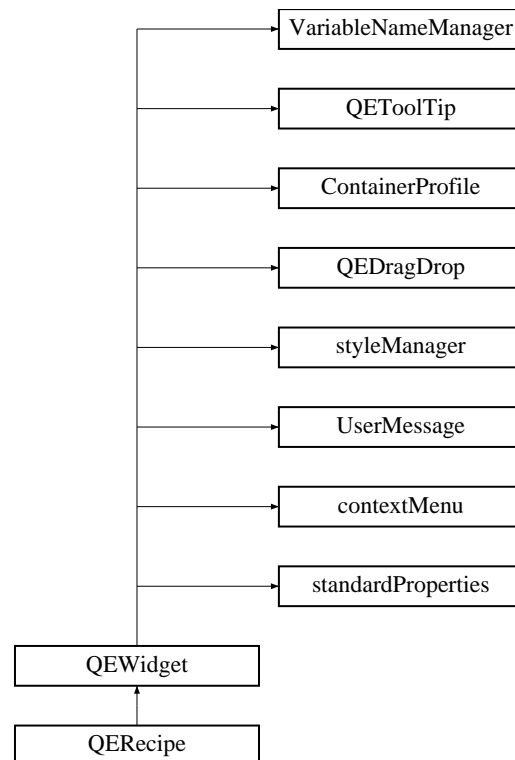
- **QERadioButtonManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEButton/QERadioButtonManager.h
- /home/rhydera/epicsqt/framework/widgets/QEButton/QERadioButtonManager.cpp

10.101 QERecipe Class Reference

Inheritance diagram for QERecipe:



Public Types

- enum **configurationTypesProperty** { **File** = FROM_FILE, **Text** = FROM_TEXT }
- enum **detailsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }
- enum **userTypesProperty** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

Public Member Functions

- **QERecipe** (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 **setDetailsLayout** (int pValue)
- int **getDetailsLayout** ()
- void **setCurrentUserType** (int pValue)
- int **getCurrentUserType** ()
- bool **saveRecipeList** ()
- void **refreshRecipeList** ()
- void **refreshButton** ()
- void **userLevelChanged** (userLevels pValue)
- void **setConfigurationTypeProperty** (configurationTypesProperty pConfigurationType)
- configurationTypesProperty **getConfigurationTypeProperty** ()
- void **setDetailsLayoutProperty** (detailsLayoutProperty pDetailsLayout)
- detailsLayoutProperty **getDetailsLayoutProperty** ()
- 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 **detailsLayout**
- 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**
- detailsLayoutProperty **detailsLayout**
- userTypesProperty **currentUserType**

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

- /home/rhydera/epicsqt/framework/widgets/QERecipe/QERecipe.h
- /home/rhydera/epicsqt/framework/widgets/QERecipe/QERecipe.cpp

10.102 QERecipeManager Class Reference

Public Member Functions

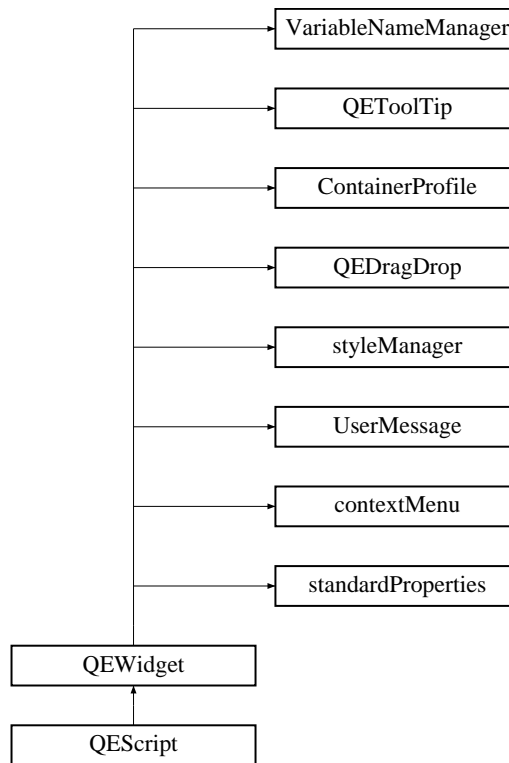
- **QERecipeManager** (QObject *pParent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *pParent)
- void **initialize** (QDesignerFormEditorInterface *pCore)

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

- /home/rhydera/epicsqt/framework/widgets/QERecipe/QERecipeManager.h
- /home/rhydera/epicsqt/framework/widgets/QERecipe/QERecipeManager.cpp

10.103 QEScript Class Reference

Inheritance diagram for QEScript:



Public Types

- enum **detailsLayoutProperty** { **Top** = TOP, **Bottom** = BOTTOM, **Left** = LEFT, **Right** = RIGHT }

Signals

- void **selected** (QString pFilename)

Public Member Functions

- **QEScript** (QWidget *pParent=0)
- 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 **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 **setDetailsLayout** (int pValue)
- int **getDetailsLayout** ()
- void **updateTable** ()
- void **setDetailsLayoutProperty** (detailsLayoutProperty pDetailsLayout)
- detailsLayoutProperty **getDetailsLayoutProperty** ()

Protected Attributes

- QLineEdit * **qlineEditDirectoryPath**
- QPushButton * **qPushButtonDirectoryBrowser**
- QPushButton * **qPushButtonRefresh**
- [_QTableWidgetScript](#) * **qTableWidgetScript**
- QString **fileFilter**
- bool **showFileExtension**
- int **detailsLayout**

Properties

- QString **directoryPath**
- bool **showDirectoryPath**
- bool **showDirectoryBrowser**
- bool **showRefresh**
- bool **showColumnTime**
- bool **showColumnSize**
- bool **showColumnFilename**
- detailsLayoutProperty **detailsLayout**

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

- /home/rhydera/epicsqt/framework/widgets/QEScript/QEScript.h
- /home/rhydera/epicsqt/framework/widgets/QEScript/QEScript.cpp

10.104 QEScriptManager Class Reference

Public Member Functions

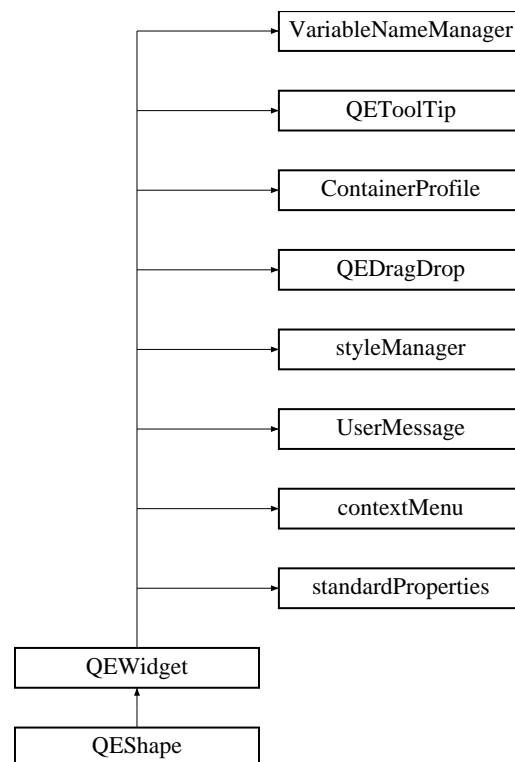
- **QEScriptManager** (QObject *pParent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *pParent)
- void **initialize** (QDesignerFormEditorInterface *pCore)

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

- /home/rhydera/epicsqt/framework/widgets/QEScript/QEScriptManager.h
- /home/rhydera/epicsqt/framework/widgets/QEScript/QEScriptManager.cpp

10.105 QEShape Class Reference

Inheritance diagram for QEShape:



Public Types

- enum **shapeOptions** {
Line, Points, Polyline, Polygon,
Rect, RoundedRect, Ellipse, Arc,
Chord, Pie, Path, Text,
Pixmap }
- enum **animationOptions** {
Width, Height, X, Y,
Transperency, Rotation, ColourHue, ColourSaturation,
ColourValue, ColourIndex, Penwidth }
- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST,
Engineer = USERLEVEL_ENGINEER }

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

Public Slots

- void **requestEnabled** (const bool &state)

Signals

- void **dbValueChanged1** (const qlonglong &out)
- void **dbValueChanged2** (const qlonglong &out)
- void **dbValueChanged3** (const qlonglong &out)
- void **dbValueChanged4** (const qlonglong &out)
- void **dbValueChanged5** (const qlonglong &out)
- void **dbValueChanged6** (const qlonglong &out)

Public Member Functions

- **QEShape** (QWidget *parent=0)
- **QEShape** (const QString &variableName, QWidget *parent=0)
- void **setAnimation** (animationOptions animation, const int index)
- animationOptions **getAnimation** (const int index)
- void **setScale** (const double scale, const int index)
- double **getScale** (const int index)
- void **setOffset** (const double offset, const int index)
- double **getOffset** (const int index)
- void **setBorder** (const bool border)
- bool **getBorder** ()
- void **setFill** (const bool fill)
- bool **getFill** ()
- void **setShape** (shapeOptions shape)
- shapeOptions **getShape** ()
- void **setNumPoints** (const unsigned int numPoints)
- unsigned int **getNumPoints** ()
- void **setOriginTranslation** (const QPoint originTranslation)
- QPoint **getOriginTranslation** ()
- void **setPoint** (const QPoint point, const int index)
- QPoint **getPoint** (const int index)
- void **setColor** (const QColor color, const int index)
- QColor **getColor** (const int index)
- void **setDrawBorder** (const bool drawBorder)
- bool **getDrawBorder** ()
- void **setLineWidth** (const unsigned int lineWidth)
- unsigned int **getLineWidth** ()
- void **setStartAngle** (const double startAngle)
- double **getStartAngle** ()
- void **setRotation** (const double rotation)
- double **getRotation** ()
- void **setArcLength** (const double arcLength)
- double **getArcLength** ()
- void **setText** (const QString text)
- QString **getText** ()

- void **setVariableNameSubstitutionsProperty** (QString variableNameSubstitutions)
- QString **getVariableNameSubstitutionsProperty** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.
- void **setAnimation1Property** (animationOptions animation)
- animationOptions **getAnimation1Property** ()
- void **setAnimation2Property** (animationOptions animation)
- animationOptions **getAnimation2Property** ()
- void **setAnimation3Property** (animationOptions animation)
- animationOptions **getAnimation3Property** ()
- void **setAnimation4Property** (animationOptions animation)
- animationOptions **getAnimation4Property** ()
- void **setAnimation5Property** (animationOptions animation)
- animationOptions **getAnimation5Property** ()
- void **setAnimation6Property** (animationOptions animation)
- animationOptions **getAnimation6Property** ()
- void **setScale1Property** (double scale)
- double **getScale1Property** ()
- void **setScale2Property** (double scale)
- double **getScale2Property** ()
- void **setScale3Property** (double scale)
- double **getScale3Property** ()
- void **setScale4Property** (double scale)
- double **getScale4Property** ()
- void **setScale5Property** (double scale)
- double **getScale5Property** ()
- void **setScale6Property** (double scale)
- double **getScale6Property** ()
- void **setOffset1Property** (double offset)
- double **getOffset1Property** ()
- void **setOffset2Property** (double offset)

- double **getOffset2Property** ()
- void **setOffset3Property** (double offset)
- double **getOffset3Property** ()
- void **setOffset4Property** (double offset)
- double **getOffset4Property** ()
- void **setOffset5Property** (double offset)
- double **getOffset5Property** ()
- void **setOffset6Property** (double offset)
- double **getOffset6Property** ()
- void **setShapeProperty** (shapeOptions shape)
- shapeOptions **getShapeProperty** ()
- void **setPoint1Property** (QPoint point)
- QPoint **getPoint1Property** ()
- void **setPoint2Property** (QPoint point)
- QPoint **getPoint2Property** ()
- void **setPoint3Property** (QPoint point)
- QPoint **getPoint3Property** ()
- void **setPoint4Property** (QPoint point)
- QPoint **getPoint4Property** ()
- void **setPoint5Property** (QPoint point)
- QPoint **getPoint5Property** ()
- void **setPoint6Property** (QPoint point)
- QPoint **getPoint6Property** ()
- void **setPoint7Property** (QPoint point)
- QPoint **getPoint7Property** ()
- void **setPoint8Property** (QPoint point)
- QPoint **getPoint8Property** ()
- void **setPoint9Property** (QPoint point)
- QPoint **getPoint9Property** ()
- void **setPoint10Property** (QPoint point)
- QPoint **getPoint10Property** ()
- void **setColor1Property** (QColor color)
- QColor **getColor1Property** ()
- void **setColor2Property** (QColor color)
- QColor **getColor2Property** ()
- void **setColor3Property** (QColor color)
- QColor **getColor3Property** ()
- void **setColor4Property** (QColor color)
- QColor **getColor4Property** ()
- void **setColor5Property** (QColor color)
- QColor **getColor5Property** ()
- void **setColor6Property** (QColor color)
- QColor **getColor6Property** ()
- void **setColor7Property** (QColor color)
- QColor **getColor7Property** ()
- void **setColor8Property** (QColor color)

- QColor **getColor8Property** ()
- void **setColor9Property** (QColor color)
- QColor **getColor9Property** ()
- void **setColor10Property** (QColor color)
- QColor **getColor10Property** ()

Protected Member Functions

- void **colorChange** (unsigned int index)
- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **mousePressEvent** (QMouseEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Protected Attributes

- [QEIntegerFormatting](#) **integerFormatting**
- double **offsets** [OFFSETS_SIZE]
- double **scales** [SCALES_SIZE]
- shapeOptions **shape**
- QPoint **originTranslation**
- QPoint **points** [POINTS_SIZE]
- unsigned int **numPoints**
- QColor **colors** [COLORS_SIZE]
- animationOptions **animations** [6]
- double **startAngle**
- double **arcLength**
- QString **text**
- double **rotation**
- unsigned int **lineWidth**
- bool **fill**
- bool **drawBorder**
- unsigned int **currentColor**

Properties

- QString [variable1](#)
- QString [variable2](#)
- QString [variable3](#)
- QString [variable4](#)
- QString [variable5](#)
- QString [variable6](#)
- QString [variableSubstitutions](#)

- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- animationOptions **animation1**
- animationOptions **animation2**
- animationOptions **animation3**
- animationOptions **animation4**
- animationOptions **animation5**
- animationOptions **animation6**
- double **scale1**
- 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**

10.105.1 Member Function Documentation

10.105.1.1 void QEShape::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.105.2 Property Documentation

10.105.2.1 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.

Reimplemented from [QEDragDrop](#).

10.105.2.2 bool QEShape::enabled [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.105.2.3 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.

10.105.2.4 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 through the [QELogin](#) widget, or programatically 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'.

10.105.2.5 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.

10.105.2.6 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.

10.105.2.7 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.

10.105.2.8 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 through the [QELogin](#) widget, or programatically 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'.

10.105.2.9 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.

10.105.2.10 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.

10.105.2.11 `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.

10.105.2.12 `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.

10.105.2.13 `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.

10.105.2.14 `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.

10.105.2.15 `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.

Reimplemented from [QEObject](#).

10.105.2.16 `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 example, 'SAMPLE=SAM1, NAME = "Ref foil"' These substitutions are applied to all the variable names.

10.105.2.17 `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/rhydera/epicsqt/framework/widgets/QEShape/QEShape.h`

- /home/rhydera/epicsqt/framework/widgets/QEShape/QEShape.cpp

10.106 QEShapeManager Class Reference

Public Member Functions

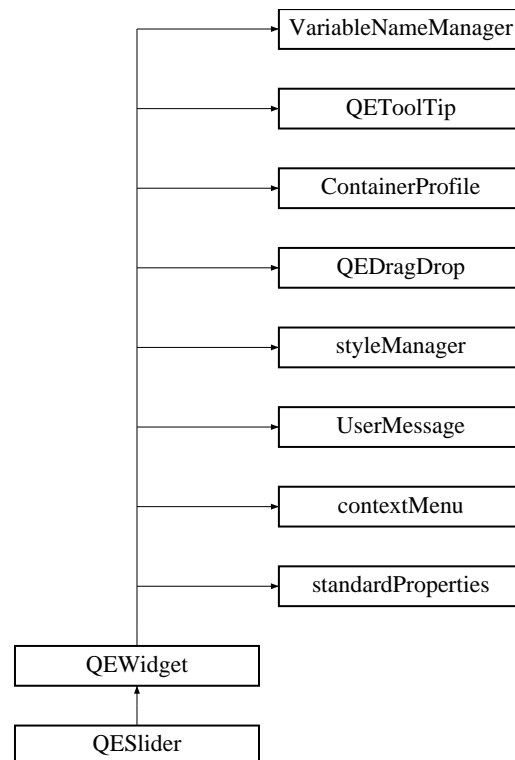
- **QEShapeManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QEShape/QEShapeManager.h
- /home/rhydera/epicsqt/framework/widgets/QEShape/QEShapeManager.cpp

10.107 QESlider Class Reference

Inheritance diagram for QESlider:



Public Types

- enum **UserLevels** { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

Public Slots

- void **writeNow** ()
- void **requestEnabled** (const bool &state)

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** ()
- void **setSubscribe** (bool subscribe)
- bool **getSubscribe** ()
- void **setScale** (double scaleIn)
- double **getScale** ()
- void **setOffset** (double offsetIn)
- double **getOffset** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.

Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Protected Attributes

- **QEFloatingFormatting** **floatingFormatting**
- bool **writeOnChange**

Properties

- QString **variable**
- QString **variableSubstitutions**
- bool **subscribe**

- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)

10.107.1 Member Function Documentation

10.107.1.1 void QESlider::requestEnabled (const bool & *state*) [inline, slot]

Similar to standard setEnabled slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.107.2 Property Documentation

10.107.2.1 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.

Reimplemented from [QEDragDrop](#).

10.107.2.2 bool QESlider::enabled [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.107.2.3 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.

10.107.2.4 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 through the [QELogin](#) widget, or programatically 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'.

10.107.2.5 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.

10.107.2.6 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.

10.107.2.7 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.

10.107.2.8 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 through the [QELogin](#) widget, or programatically 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'.

10.107.2.9 QString QESlider::variable [read, write]

EPICS variable name (CA PV)

10.107.2.10 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.

Reimplemented from [QESlider](#).

10.107.2.11 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.

10.107.2.12 bool QESlider::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 [QESlider](#) 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/rhydera/epicsqt/framework/widgets/QESlider/QESlider.h
- /home/rhydera/epicsqt/framework/widgets/QESlider/QESlider.cpp

10.108 QESliderManager Class Reference

Public Member Functions

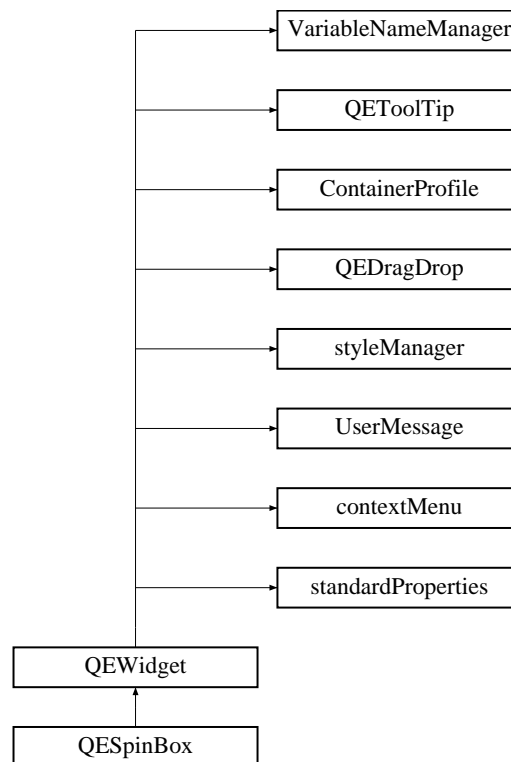
- **QESliderManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QESlider/QESliderManager.h
- /home/rhydera/epicsqt/framework/widgets/QESlider/QESliderManager.cpp

10.109 QESpinBox Class Reference

Inheritance diagram for QESpinBox:



Public Types

- enum `UserLevels` { **User** = USERLEVEL_USER, **Scientist** = USERLEVEL_SCIENTIST, **Engineer** = USERLEVEL_ENGINEER }

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

Public Slots

- void **writeNow** ()
- void `requestEnabled` (const bool &state)

Signals

- void **dbValueChanged** (const double &out)
- void **userChange** (const QString &oldValue, const QString &newValue, const QString &lastValue)

Public Member Functions

- **QESpinBox** (QWidget *parent=0)
- **QESpinBox** (const QString &variableName, QWidget *parent=0)
- void **setWriteOnChange** (bool writeOnChangeIn)
- bool **getWriteOnChange** ()
- void **setSubscribe** (bool subscribe)
- bool **getSubscribe** ()
- void **setAddUnitsAsSuffix** (bool addUnitsAsSuffixIn)
- bool **getAddUnitsAsSuffix** ()
- void **setUseDbPrecisionForDecimals** (bool useDbPrecisionForDecimalsIn)
- bool **getUseDbPrecisionForDecimals** ()
- bool **isEnabled** () const
Access function for 'enabled' property - refer to 'enabled' property for details.
- void **setEnabled** (bool state)
Access function for 'enabled' property - refer to 'enabled' 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.

Protected Member Functions

- void **establishConnection** (unsigned int variableIndex)
- void **dragEnterEvent** (QDragEnterEvent *event)
- void **dropEvent** (QDropEvent *event)
- void **setDrop** (QVariant drop)
- QVariant **getDrop** ()

Protected Attributes

- [QEFloatingFormatting](#) **floatingFormatting**
- bool **writeOnChange**
- bool **addUnitsAsSuffix**
- bool **useDbPrecisionForDecimal**

Properties

- QString [variable](#)
- QString [variableSubstitutions](#)
- bool [variableAsToolTip](#)
- bool [enabled](#)
- bool [allowDrop](#)
- bool [visible](#)
- unsigned [int](#)
- QString [userLevelUserStyle](#)
- QString [userLevelScientistStyle](#)
- QString [userLevelEngineerStyle](#)
- [UserLevels](#) [userLevelVisibility](#)
- [UserLevels](#) [userLevelEnabled](#)
- bool **subscribe**
- bool **useDbPrecision**
- bool **addUnits**

10.109.1 Member Function Documentation

10.109.1.1 void [QESpinBox::requestEnabled](#) (const bool & *state*) [[inline](#), [slot](#)]

Similar to standard `setEnabled` slot, but allows QE widget to determine if the widget remains disabled due to invalid data. If disabled due to invalid data, a request to enable the widget will be honoured when the data is no longer invalid.

10.109.2 Property Documentation

10.109.2.1 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.

Reimplemented from [QEDragDrop](#).

10.109.2.2 `bool QESpinBox::enabled` [read, write]

Set the preferred 'enabled' state. Default is true. This property is copied to the standard Qt 'enabled' property if the data being displayed is valid. If the data being displayed is invalid the standard Qt 'enabled' property will always be set to false to indicate invalid data. The value of this property will only be copied to the standard Qt 'enabled' property once data is valid.

10.109.2.3 `unsigned QESpinBox::id` [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.

10.109.2.4 `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 through the [QELogin](#) widget, or programatically 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'.

10.109.2.5 `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.

10.109.2.6 `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.

10.109.2.7 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.

10.109.2.8 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 through the [QELogin](#) widget, or programatically 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'.

10.109.2.9 QString QESpinBox::variable [read, write]

EPICS variable name (CA PV)

10.109.2.10 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.

Reimplemented from [QEWidget](#).

10.109.2.11 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.

10.109.2.12 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/rhydera/epicsqt/framework/widgets/QESpinBox/QESpinBox.h
- /home/rhydera/epicsqt/framework/widgets/QESpinBox/QESpinBox.cpp

10.110 QESpinBoxManager Class Reference

Public Member Functions

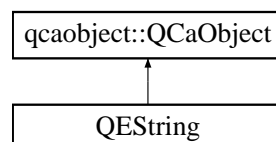
- **QESpinBoxManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QESpinBox/QESpinBoxManager.h
- /home/rhydera/epicsqt/framework/widgets/QESpinBox/QESpinBoxManager.cpp

10.111 QEStrString Class Reference

Inheritance diagram for QEStrString:



Public Slots

- void **writeString** (const QString &data)

Signals

- void **stringConnectionChanged** (QCaConnectionInfo &connectionInfo, const unsigned int &variableIndex)
- void **stringChanged** (const QString &value, QCaAlarmInfo &alarmInfo, QCaDateTime &timeStamp, const unsigned int &variableIndex)

Public Member Functions

- **QString** (QString recordName, QObject *eventObject, [QStringFormatting](#) *stringFormattingIn, unsigned int variableIndexIn)
- **QString** (QString recordName, QObject *eventObject, [QStringFormatting](#) *stringFormattingIn, unsigned int variableIndexIn, [UserMessage](#) *userMessageIn)

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

- /home/rhydera/epicsqt/framework/data/include/QString.h
- /home/rhydera/epicsqt/framework/data/src/QString.cpp

10.112 QStringFormatting Class Reference

Public Types

- enum **formats** {
FORMAT_DEFAULT, **FORMAT_FLOATING**, **FORMAT_INTEGER**, **FORMAT_UNSIGNEDINTEGER**,
FORMAT_TIME, **FORMAT_LOCAL_ENUMERATE** }
- enum **notations** { **NOTATION_FIXED** = QTextStream::FixedNotation, **NOTATION_SCIENTIFIC** = QTextStream::ScientificNotation, **NOTATION_AUTOMATIC** = QTextStream::SmartNotation }
- enum **arrayActions** { **APPEND**, **ASCII**, **INDEX** }

Public Member Functions

- QString **formatString** (const QVariant &value)
- QVariant **formatValue** (const QString &text)
- void **setDbEgu** (QString egu)
- void **setDbEnumerations** (QStringList enumerations)
- void **setDbPrecision** (unsigned int dbPrecisionIn)
- void **setDbVariablesStatField** (bool isStatField)
- void **setPrecision** (int precision)
- void **setUseDbPrecision** (bool useDbPrecision)
- void **setLeadingZero** (bool leadingZero)
- void **setTrailingZeros** (bool trailingZeros)
- void **setFormat** (formats format)
- void **setRadix** (unsigned int radix)
- void **setNotation** (notations notation)
- void **setArrayAction** (arrayActions arrayActionIn)
- void **setArrayIndex** (unsigned int arrayIndexIn)
- void **setAddUnits** (bool addUnits)
- void **setLocalEnumeration** (QString localEnumerationIn)
- int **getPrecision** ()
- bool **getUseDbPrecision** ()

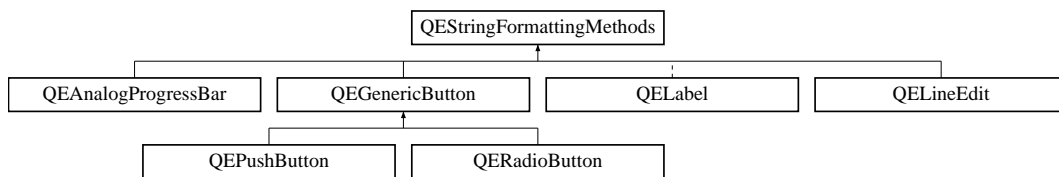
- bool **getLeadingZero** ()
- bool **getTrailingZeros** ()
- formats **getFormat** ()
- unsigned int **getRadix** ()
- notations **getNotation** ()
- arrayActions **getArrayAction** ()
- unsigned int **getArrayIndex** ()
- bool **getAddUnits** ()
- QString **getLocalEnumeration** ()

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

- /home/rhydera/epicsqt/framework/data/include/QQStringFormatting.h
- /home/rhydera/epicsqt/framework/data/src/QQStringFormatting.cpp

10.113 QQStringFormattingMethods Class Reference

Inheritance diagram for QQStringFormattingMethods:



Public Member Functions

- virtual void **stringFormattingChange** ()=0
- void **setPrecision** (int precision)
- int **getPrecision** ()
- void **setUseDbPrecision** (bool useDbPrecision)
- bool **getUseDbPrecision** ()
- void **setLeadingZero** (bool leadingZero)
- bool **getLeadingZero** ()
- void **setTrailingZeros** (bool trailingZeros)
- bool **getTrailingZeros** ()
- void **setAddUnits** (bool addUnits)
- bool **getAddUnits** ()
- void **setLocalEnumeration** (QString localEnumeration)
- QString **getLocalEnumeration** ()
- void **setFormat** (QQStringFormatting::formats format)
- QQStringFormatting::formats **getFormat** ()
- void **setRadix** (unsigned int radix)

- unsigned int **getRadix** ()
- void **setNotation** (QCStringFormatting::notations notation)
- QCStringFormatting::notations **getNotation** ()
- void **setArrayAction** (QCStringFormatting::arrayActions arrayAction)
- QCStringFormatting::arrayActions **getArrayAction** ()
- void **setArrayIndex** (unsigned int arrayIndex)
- unsigned int **getArrayIndex** ()

Protected Attributes

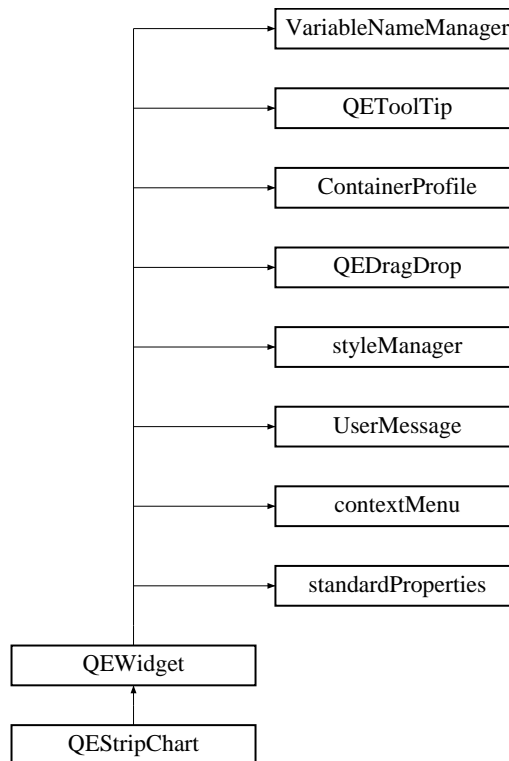
- [QCStringFormatting](#) **stringFormatting**

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

- /home/rhydera/epicsqt/framework/widgets/include/QCStringFormattingMethods.h
- /home/rhydera/epicsqt/framework/widgets/src/QCStringFormattingMethods.cpp

10.114 QEStripChart Class Reference

Inheritance diagram for QEStripChart:



Classes

- class [PrivateData](#)

Public Types

- enum **Constants** { **NUMBER_OF_PVS** = 12 }

Public Member Functions

- **QEStripChart** (QWidget *parent=0)
- QSize **sizeHint** () const
- QDateTime **getStartDateTime** ()
- QDateTime **getEndDateTime** ()
- void **setEndDateTime** (QDateTime endTimeIn)
- int **getDuration** ()
- void **setDuration** (int durationIn)
- double **getYMinimum** ()
- void **setYMinimum** (double yMinimumIn)
- double **getYMaximum** ()
- void **setYMaximum** (double yMaximumIn)
- void **plotData** ()

Protected Member Functions

- 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)
- void **setup** ()
- [qcaobject::QCaObject](#) * **createQcaltem** (unsigned int variableIndex)
- void **establishConnection** (unsigned int variableIndex)
- void **updateToolTip** (const QString &tip)

Properties

- int **duration**
- double **yMinimum**
- double **yMaximum**
- QString **variable1**
- QString **variable2**

- QString **variable3**
- QString **variable4**
- QString **variable5**
- QString **variable6**
- QString **variable7**
- QString **variable8**
- QString **variable9**
- QString **variable10**
- QString **variable11**
- QString **variable12**
- QColor **colour1**
- QColor **colour2**
- QColor **colour3**
- QColor **colour4**
- QColor **colour5**
- QColor **colour6**
- QColor **colour7**
- QColor **colour8**
- QColor **colour9**
- QColor **colour10**
- QColor **colour11**
- QColor **colour12**

Friends

- class **PrivateData**
- class [QEStripChartItem](#)

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

- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChart.h
- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChart.cpp

10.115 QEStripChartItem Class Reference

Classes

- class [PrivateData](#)

Friends

- class [QEStripChart](#)

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

- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartItem.h
- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartItem.cpp

10.116 QEStripChartItemDialog Class Reference

Public Member Functions

- **QEStripChartItemDialog** (QWidget *parent=0)
- void **setPvName** (QString pvNameIn)
- QString **getPvName** ()
- void **setColour** (QColor colourIn)
- QColor **getColour** ()
- bool **isClear** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartItemDialog.h
- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartItemDialog.cpp

10.117 QEStripChartManager Class Reference

Public Member Functions

- **QEStripChartManager** (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/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartManager.h
- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartManager.cpp

10.118 QEStripChartTimeDialog Class Reference

Public Member Functions

- **QEStripChartTimeDialog** (QWidget *parent=0)
- void **setMaximumDateTime** (QDateTime datetime)
- void **setStartDateTime** (QDateTime datetime)

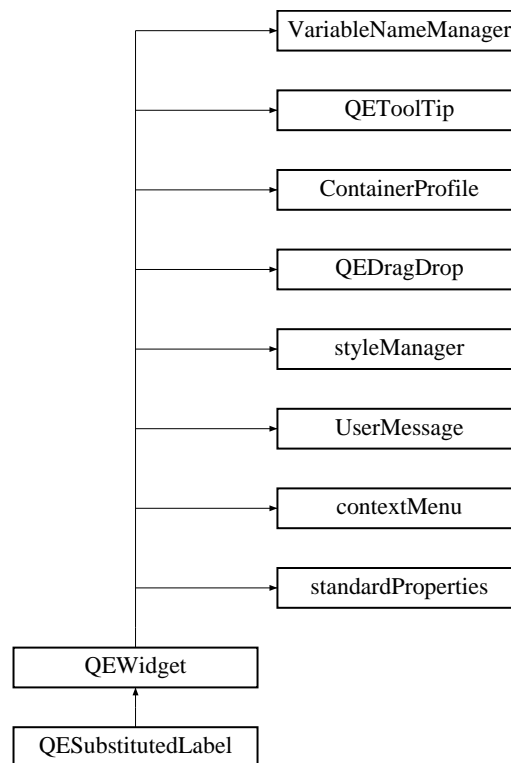
- QDateTime **getStartDateTime** ()
- void **setEndDateTime** (QDateTime datetime)
- QDateTime **getEndDateTime** ()

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

- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartTimeDialog.h
- /home/rhydera/epicsqt/framework/widgets/QEStripChart/QEStripChartTimeDialog.cpp

10.119 QESubstitutedLabel Class Reference

Inheritance diagram for QESubstitutedLabel:



Public Member Functions

- **QESubstitutedLabel** (QWidget *parent=0)
- void **establishConnection** (unsigned int variableIndex)
- void **setLabelTextProperty** (QString labelTextIn)
- QString **getLabelTextProperty** ()
- QString **getLabelTextPropertyFormat** ()
- void **setLabelTextPropertyFormat** (QString labelTextIn)

Protected Attributes

- QString [labelText](#)

Properties

- QString [textSubstitutions](#)

10.119.1 Member Data Documentation

10.119.1.1 **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

10.119.2 Property Documentation

10.119.2.1 **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/rhydera/epicsqt/framework/widgets/QESubstitutedLabel/QESubstitutedLabel.h
- /home/rhydera/epicsqt/framework/widgets/QESubstitutedLabel/QESubstitutedLabel.cpp

10.120 QESubstitutedLabelManager Class Reference

Public Member Functions

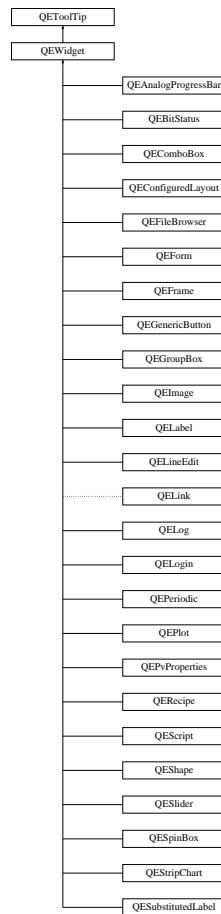
- **QESubstitutedLabelManager** (QObject *parent=0)
- bool **isContainer** () const
- bool **isInitialized** () const
- QIcon **icon** () const
- QString **group** () const
- QString **includeFile** () const
- QString **name** () const
- QString **toolTip** () const
- QString **whatsThis** () const
- QWidget * **createWidget** (QWidget *parent)
- void **initialize** (QDesignerFormEditorInterface *core)

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

- /home/rhydera/epicsqt/framework/widgets/QSubstitutedLabel/QSubstitutedLabelManager.h
- /home/rhydera/epicsqt/framework/widgets/QSubstitutedLabel/QSubstitutedLabelManager.cpp

10.121 QEToolTip Class Reference

Inheritance diagram for QEToolTip:



Public Member Functions

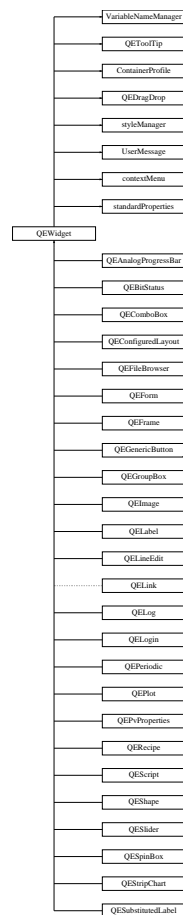
- void **updateToolTipVariable** (const QString &variable)
- void **updateToolTipAlarm** (const QString &alarm)
- void **updateToolTipConnection** (bool connection)
- virtual void **updateToolTip** (const QString &)

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

- `/home/rhydera/epicsqt/framework/widgets/include/QEToolTip.h`
- `/home/rhydera/epicsqt/framework/widgets/src/QEToolTip.cpp`

10.122 QEWidget Class Reference

Inheritance diagram for QEWidget:



Public Member Functions

- **QEWidget** (QWidget *ownerIn)
- void **activate** ()
- unsigned int **getMessageSourceId** ()
- void **setMessageSourceId** (unsigned int messageSourceId)
- [qcaobject::QCaObject](#) * **getQcaltem** (unsigned int variableIndex)
- void **setupContextMenu** (QWidget *w)
- QColor **getColor** ([QCaAlarmInfo](#) &alarmInfo, const int saturation)

- void **readNow** ()
- virtual void **writeNow** ()
- virtual void **setVariableNameAndSubstitutions** (QString variableNameIn, QString variableNameSubstitutionsIn, unsigned int variableIndex)
- QFile * **openQEFile** (QString name, QFile::OpenModeFlag mode)
- QString **defaultFileLocation** ()

Static Public Member Functions

- static bool **inDesigner** ()

Protected Member Functions

- void **setNumVariables** (unsigned int numVariablesIn)
- [qcaobject::QCaObject](#) * **createConnection** (unsigned int variableIndex)
- virtual [qcaobject::QCaObject](#) * **createQcaltem** (unsigned int variableIndex)
- virtual void **establishConnection** (unsigned int variableIndex)
- void **setVariableAsToolTip** (bool variableAsToolTip)
- bool **getVariableAsToolTip** ()

Protected Attributes

- bool **subscribe**
- bool **variableAsToolTip**

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

- /home/rhydera/epicsqt/framework/widgets/include/QEWidget.h
- /home/rhydera/epicsqt/framework/widgets/src/QEWidget.cpp

10.123 QEWidgets Class Reference

Public Member Functions

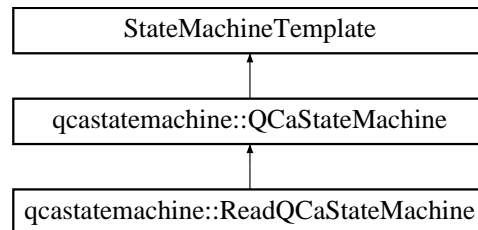
- **QEWidgets** (QObject *parent=0)
- virtual QList< QDesignerCustomWidgetInterface * > **customWidgets** () const

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

- /home/rhydera/epicsqt/framework/widgets/include/QEDesignerPlugin.h
- /home/rhydera/epicsqt/framework/widgets/src/QEDesignerPlugin.cpp

10.124 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/rhydera/epicsqt/framework/data/include/QCaStateMachine.h
- /home/rhydera/epicsqt/framework/data/src/QCaStateMachine.cpp

10.125 RecordSpec Class Reference

Public Member Functions

- **RecordSpec** (const QString theRecordType)
- QString **getRecordType** ()
- QString **getFieldName** (const int index)

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

- /home/rhydera/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.cpp

10.126 RecordSpecList Class Reference

Public Member Functions

- [RecordSpec](#) * **find** (const QString recordType)

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

- /home/rhydera/epicsqt/framework/widgets/QEPvProperties/QEPvProperties.cpp

10.127 selectMenu Class Reference

Public Member Functions

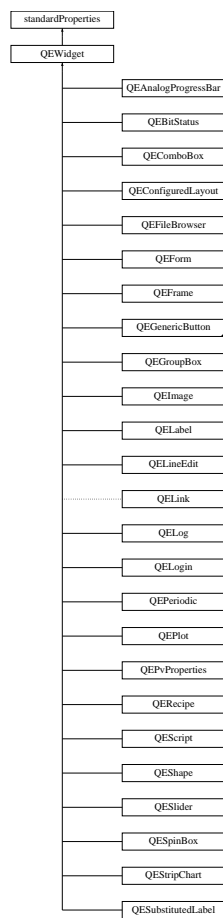
- **selectMenu** (QWidget *parent=0)
- imageContextMenu::imageContextMenuOptions **getSelectOption** (const QPoint &pos)
- void **setChecked** (const int mode)
- void **setPanEnabled** (bool enablePan)
- void **setVSliceEnabled** (bool enableVSliceSelection)
- void **setHSliceEnabled** (bool enableHSliceSelection)
- void **setAreaEnabled** (bool enableAreaSelection)
- void **setProfileEnabled** (bool enableProfileSelection)
- void **setTargetEnabled** (bool enableTargetSelection)

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

- /home/rhydera/epicsqt/framework/widgets/QEImage/selectMenu.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/selectMenu.cpp

10.128 standardProperties Class Reference

Inheritance diagram for standardProperties:



Public Member Functions

- **standardProperties** (QWidget *ownerIn)

Protected Member Functions

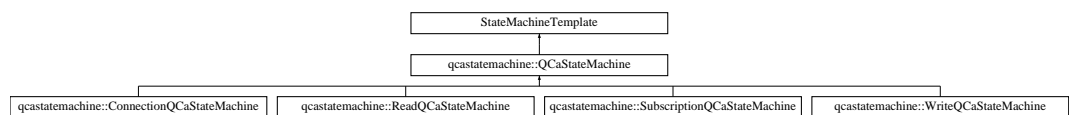
- userLevels **getUserLevelVisibility** ()
- void **setUserLevelVisibility** (userLevels level)
- userLevels **getUserLevelEnabled** ()
- void **setUserLevelEnabled** (userLevels level)
- bool **getApplicationEnabled** () const
- void **setApplicationEnabled** (bool state)
- void **setDataDisabled** (bool disable)
- void **setRunVisible** (bool visibleIn)
- bool **getRunVisible** ()
- void **checkVisibilityEnabledLevel** (userLevels level)

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

- /home/rhydera/epicsqt/framework/widgets/include/standardProperties.h
- /home/rhydera/epicsqt/framework/widgets/src/standardProperties.cpp

10.129 StateMachineTemplate Class Reference

Inheritance diagram for StateMachineTemplate:



Public Member Functions

- virtual bool **process** (int requestedState)=0

Public Attributes

- int **currentState**
- int **requestState**

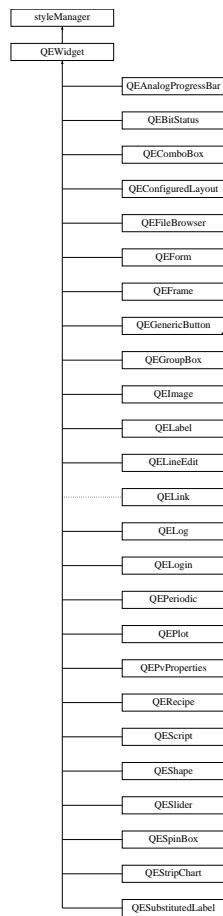
The documentation for this class was generated from the following file:

- /home/rhydera/epicsqt/framework/data/include/QCaStateMachine.h

10.130 styleManager Class Reference

```
#include <styleManager.h>
```

Inheritance diagram for styleManager:



Public Member Functions

- **styleManager** (QWidget *ownerIn)

Protected Member Functions

- void [setStyleUser](#) (QString style)
- QString [getStyleUser](#) ()
- void [setStyleScientist](#) (QString style)
- QString [getStyleScientist](#) ()
- void [setStyleEngineer](#) (QString style)
- QString [getStyleEngineer](#) ()
- void [updateDataStyle](#) (QString style)
- void [updateStatusStyle](#) (QString style)
- void [styleUserLevelChanged](#) (userLevels levelIn)

10.130.1 Detailed Description

This class adds common style support to all QE widgets if required.

Standard properties for all QE widgets specify a style to be applied for user, scientist, and engineer mode. Also QE widgets can specify data or status related style changes.

The syntax for all Style Sheet strings used by this class is the standard Qt Style Sheet syntax. For example, 'background-color: red'. Refer to Qt Style Sheets Reference for full details.

Note, as well as the large number of defined Style Sheet properties, the Style Sheet syntax allows setting any widget property using the 'qproperty' property. For example 'qproperty-geometry:rect(10 10 100 100);' Caution, any amount of weird behaviour can be effected using Style Sheet strings. Without carefull consideration they should only be used for simple visualisation effects such as altering the background color.

All QE widgets (eg, [QELabel](#), [QELineEdit](#)) have an instance of this class as they based on [QEWidget](#) which itself uses this class as a base class.

To use the functionality provided by this class, QE widgets must include the following:

- `Q_PROPERTY(QString userLevelUserStyle READ getStyleUser WRITE setStyleUser)`
- `Q_PROPERTY(QString userLevelScientistStyle READ getStyleScientist WRITE setStyleScientist)`
- `Q_PROPERTY(QString userLevelEngineerStyle READ getStyleEngineer WRITE setStyleEngineer)`

To use this class to manage style changes related to data and status the QE widget must include the following:

- On presentation of new data call [updateDataStyle\(\)](#) with the Style Sheet string related to the new data (if any).
- On change of data status, call [updateStatusStyle\(\)](#) with the Style Sheet string related to the new status (if any).

Note, this class notes the initial style when instantiated and uses that style as the base style for all style changes. This means any style changes not performed through this class will be lost the next time this class changes the style.

10.130.2 Member Function Documentation

10.130.2.1 `QString styleManager::getStyleEngineer ()` [protected]

Get the Style Sheet string to be applied when the widget is displayed in 'Engineer' mode.

10.130.2.2 QString styleManager::getStyleScientist () [protected]

Get the Style Sheet string to be applied when the widget is displayed in 'Scientist' mode.

10.130.2.3 QString styleManager::getStyleUser () [protected]

Get the Style Sheet string to be applied when the widget is displayed in 'User' mode.

10.130.2.4 void styleManager::setStyleEngineer (QString style) [protected]

Set the Style Sheet string to be applied when the widget is displayed in 'Engineer' mode. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red'

10.130.2.5 void styleManager::setStyleScientist (QString style) [protected]

Set the Style Sheet string to be applied when the widget is displayed in 'Scientist' mode. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red'

10.130.2.6 void styleManager::setStyleUser (QString style) [protected]

Set the Style Sheet string to be applied when the widget is displayed in 'User' mode. The syntax is the standard Qt Style Sheet syntax. For example, 'background-color: red'.

10.130.2.7 void styleManager::styleUserLevelChanged (userLevels levelIn) [protected]

Set the current user level.

10.130.2.8 void styleManager::updateDataStyle (QString style) [protected]

Set the Style Sheet string to be applied to reflect an aspect of the current data. For example, a value over a high limit may be displayed in red.

10.130.2.9 void styleManager::updateStatusStyle (QString style) [protected]

Set the Style Sheet string to be applied to reflect an aspect of the current status. For example, invalid data may be displayed with a white background.

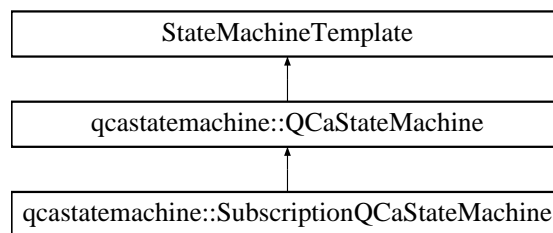
The documentation for this class was generated from the following files:

- /home/rhydera/epicsqt/framework/widgets/include/styleManager.h

- /home/rhydera/epicsqt/framework/widgets/src/styleManager.cpp

10.131 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/rhydera/epicsqt/framework/data/include/QCaStateMachine.h
- /home/rhydera/epicsqt/framework/data/src/QCaStateMachine.cpp

10.132 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**

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

- /home/rhydera/epicsqt/framework/widgets/QEPlot/QEPlot.h

10.133 TrackRange Class Reference

Public Member Functions

- void **clear** ()
- void **merge** (const double d)
- void **merge** (const [TrackRange](#) that)
- bool **getMinMax** (double &min, double &max)

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

- /home/rhydera/epicsqt/framework/widgets/QESTripChart/QESTripChartItem.h
- /home/rhydera/epicsqt/framework/widgets/QESTripChart/QESTripChartItem.cpp

10.134 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/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h

10.135 QEPeriodic::userInfoStructArray Struct Reference

Public Attributes

- [userInfoStruct](#) **array** [NUM_ELEMENTS]

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

- /home/rhydera/epicsqt/framework/widgets/QEPeriodic/QEPeriodic.h

10.136 userLevelSignal Class Reference

Signals

- void **userChanged** (userLevels level)

Public Member Functions

- void **setLevel** (userLevels levelIn)
- userLevels **getLevel** ()

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

- /home/rhydera/epicsqt/framework/widgets/include/ContainerProfile.h
- /home/rhydera/epicsqt/framework/widgets/src/ContainerProfile.cpp

10.137 userLevelSlot Class Reference

Public Slots

- void **userChanged** (userLevels level)

Public Member Functions

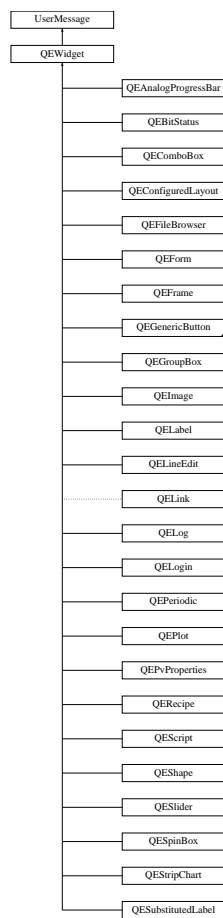
- void **setOwner** ([ContainerProfile](#) *ownerIn)

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

- /home/rhydera/epicsqt/framework/widgets/include/ContainerProfile.h
- /home/rhydera/epicsqt/framework/widgets/src/ContainerProfile.cpp

10.138 UserMessage Class Reference

Inheritance diagram for UserMessage:



Public Types

- enum **message_filter_options** { MESSAGE_FILTER_ANY, MESSAGE_FILTER_MATCH, MESSAGE_FILTER_NONE }

Public Member Functions

- void **setSourceId** (unsigned int sourceId)
- void **setFormId** (unsigned int formId)
- void **setFormFilter** (message_filter_options formFilterIn)
- void **setSourceFilter** (message_filter_options sourceFilterIn)
- unsigned int **getSourceId** ()
- unsigned int **getFormId** ()
- message_filter_options **getFormFilter** ()
- message_filter_options **getSourceFilter** ()
- void **setChildFormId** (unsigned int)
- unsigned int **getChildFormId** ()

- unsigned int **getNextMessageFormId** ()
- void **sendMessage** (QString message, message_types type=MESSAGE_TYPE_INFO)
- void **sendMessage** (QString message, QString source, message_types type=MESSAGE_TYPE_INFO)
- QString **getMessageTypeName** (message_types type)
- virtual void **newMessage** (QString, message_types)

Friends

- class [UserMessageSlot](#)
- class [UserMessageSignal](#)

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

- /home/rhydera/epicsqt/framework/widgets/include/UserMessage.h
- /home/rhydera/epicsqt/framework/widgets/src/UserMessage.cpp

10.139 UserMessageSignal Class Reference

Signals

- void **message** (QString msg, message_types type, unsigned int formId, unsigned int sourceId, [UserMessage](#) *originator)

Public Member Functions

- void **sendMessage** (QString msg, message_types type, unsigned int formId, unsigned int sourceId, [UserMessage](#) *originator)

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

- /home/rhydera/epicsqt/framework/widgets/include/UserMessage.h
- /home/rhydera/epicsqt/framework/widgets/src/UserMessage.cpp

10.140 UserMessageSlot Class Reference

Public Slots

- void **message** (QString msg, message_types type, unsigned int formId, unsigned int sourceId, [UserMessage](#) *originator)

Public Member Functions

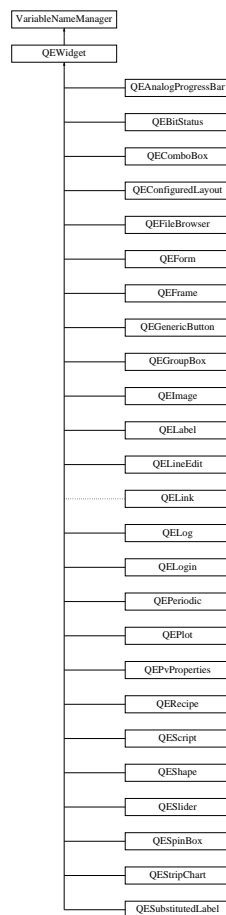
- void **setOwner** (UserMessage *ownerIn)

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

- /home/rhydera/epicsqt/framework/widgets/include/UserMessage.h
- /home/rhydera/epicsqt/framework/widgets/src/UserMessage.cpp

10.141 VariableNameManager Class Reference

Inheritance diagram for VariableNameManager:



Public Member Functions

- void **variableNameManagerInitialise** (unsigned int numVariables)

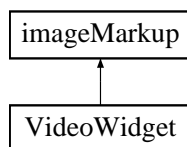
- QString **getOriginalVariableName** (unsigned int variableIndex)
- QString **getVariableNameSubstitutions** ()
- QString **getSubstitutedVariableName** (unsigned int variableIndex)
- void **initialiseVariableNameSubstitutions** (const QString &substitutions)
- void **setVariableNameSubstitutionsOverride** (const QString &substitutions)
- void **setVariableName** (const QString &variableName, unsigned int variableIndex)
- void **setVariableNameSubstitutions** (const QString &substitutions)
- QString **substituteThis** (const QString string)

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

- /home/rhydera/epicsqt/framework/widgets/include/VariableNameManager.h
- /home/rhydera/epicsqt/framework/widgets/src/VariableNameManager.cpp

10.142 VideoWidget Class Reference

Inheritance diagram for VideoWidget:



Signals

- void **userSelection** (imageMarkup::markupIds mode, QPoint point1, QPoint point2)
- void **zoomInOut** (int zoomAmount)
- void **currentPixelInfo** (QPoint pos)
- void **pan** (QPoint pos)

Public Member Functions

- **VideoWidget** (QWidget *parent=0)
- void **setNewImage** (const QImage image, [QCaDateTime](#) &time)
- void **setPanning** (bool panningIn)
- bool **getPanning** ()
- QPoint **scalePoint** (QPoint pnt)
- int **scaleOrdinate** (int ord)
- QImage **getImage** ()

Protected Member Functions

- void **paintEvent** (QPaintEvent *)
- void **mousePressEvent** (QMouseEvent *event)
- void **mouseReleaseEvent** (QMouseEvent *event)
- void **mouseMoveEvent** (QMouseEvent *event)
- void **wheelEvent** (QWheelEvent *event)
- void **markupChange** (QImage &markups, QVector< QRect > &changedAreas)
- void **resizeEvent** (QResizeEvent *event)
- void **markupSetCursor** (QCursor cursor)
- void **markupAction** (markupIds mode, QPoint point1, QPoint point2)

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

- /home/rhydera/epicsqt/framework/widgets/QImage/videowidget.h
- /home/rhydera/epicsqt/framework/widgets/QImage/videowidget.cpp

10.143 WidgetRef Class Reference

Public Member Functions

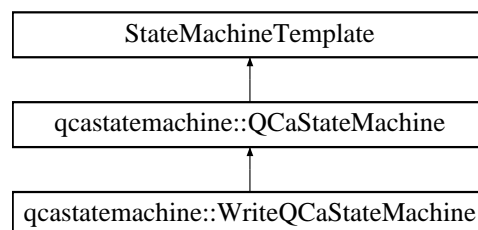
- **WidgetRef** ([QWidget](#) *refIn)
- [QWidget](#) * **getRef** ()

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

- /home/rhydera/epicsqt/framework/widgets/include/ContainerProfile.h

10.144 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/rhydera/epicsqt/framework/data/include/QCaStateMachine.h
- /home/rhydera/epicsqt/framework/data/src/QCaStateMachine.cpp

10.145 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/rhydera/epicsqt/framework/widgets/QEImage/zoomMenu.h
- /home/rhydera/epicsqt/framework/widgets/QEImage/zoomMenu.cpp

Index

- [_Field, 29](#)
- [_Item, 30](#)
- [_QDialogItem, 30](#)
- [_QDialogLogin, 30](#)
- [_QPushButtonGroup, 31](#)
- [_QTableWidgetFileBrowser, 31](#)
- [_QTableWidgetLog, 32](#)
- [_QTableWidgetScript, 32](#)
- addUnits
 - [QEAnalogProgressBar, 64](#)
 - [QELabel, 117](#)
 - [QELineEdit, 126](#)
- allowDrop
 - [QEAnalogProgressBar, 64](#)
 - [QEBitStatus, 71](#)
 - [QEComboBox, 77](#)
 - [QEFrame, 92](#)
 - [QEGroupBox, 99](#)
 - [QEImage, 107](#)
 - [QELabel, 117](#)
 - [QELineEdit, 126](#)
 - [QEPeriodic, 141](#)
 - [QEPlot, 151](#)
 - [QEPushButton, 158](#)
 - [QEPvProperties, 163](#)
 - [QERadioButton, 169](#)
 - [QEShape, 185](#)
 - [QESlider, 191](#)
 - [QESpinBox, 196](#)
- altReadbackVariable
 - [QEPushButton, 158](#)
- arrayAction
 - [QEAnalogProgressBar, 64](#)
 - [QELabel, 118](#)
 - [QELineEdit, 126](#)
- beamXVariable
 - [QEImage, 107](#)
- beamYVariable
 - [QEImage, 107](#)
- clippingHighVariable
 - [QEImage, 107](#)
- clippingLowVariable
 - [QEImage, 107](#)
- clippingOnOffVariable
 - [QEImage, 107](#)
- ContainerProfile, [33](#)
- contextMenu, [35](#)
- contextMenuObject, [37](#)
- dbValueChanged
 - [QELabel, 117](#)
- enabled
 - [QEAnalogProgressBar, 65](#)
 - [QEBitStatus, 71](#)
 - [QEComboBox, 77](#)
 - [QEFrame, 92](#)
 - [QEGroupBox, 99](#)
 - [QEImage, 107](#)
 - [QELabel, 118](#)
 - [QELineEdit, 126](#)
 - [QEPeriodic, 142](#)
 - [QEPlot, 151](#)
 - [QEPushButton, 158](#)
 - [QEPvProperties, 163](#)
 - [QERadioButton, 169](#)
 - [QEShape, 185](#)
 - [QESlider, 191](#)
 - [QESpinBox, 196](#)
- flipRotateMenu, [38](#)
- format
 - [QEAnalogProgressBar, 65](#)
 - [QELabel, 118](#)
 - [QELineEdit, 127](#)
- getStyleEngineer
 - [styleManager, 216](#)
- getStyleScientist
 - [styleManager, 216](#)
- getStyleUser

- styleManager, [217](#)
- heightVariable
 - QEImage, [108](#)
- imageContextMenu, [38](#)
- imageMarkup, [39](#)
- imageVariable
 - QEImage, [108](#)
- int
 - QEAAnalogProgressBar, [65](#)
 - QEBitStatus, [71](#)
 - QEComboBox, [77](#)
 - QEFrame, [92](#)
 - QEGroupBox, [99](#)
 - QEImage, [108](#)
 - QELabel, [118](#)
 - QELineEdit, [127](#)
 - QEPeriodic, [142](#)
 - QEPlot, [151](#)
 - QEPushButton, [158](#)
 - QEPvProperties, [163](#)
 - QERadioButton, [170](#)
 - QEShape, [185](#)
 - QESlider, [191](#)
 - QESpinBox, [197](#)
- labelText
 - QESubstitutedLabel, [207](#)
- leadingZero
 - QEAAnalogProgressBar, [65](#)
 - QELabel, [118](#)
 - QELineEdit, [127](#)
- localEnumeration
 - QEAAnalogProgressBar, [65](#)
 - QELabel, [119](#)
 - QELineEdit, [127](#)
- localEnumerationItem, [40](#)
- managePixmap, [41](#)
- markupBeam, [41](#)
- markupHLine, [42](#)
- markupItem, [43](#)
- markupLine, [44](#)
- markupRegion, [45](#)
- markupTarget, [45](#)
- markupText, [46](#)
- markupVLine, [47](#)
- notation
 - QEAAnalogProgressBar, [65](#)
 - QELabel, [119](#)
 - QELineEdit, [127](#)
- PeriodicDialog, [47](#)
- PeriodicElementSetupForm, [48](#)
- PeriodicSetupDialog, [48](#)
- pixmap0
 - QELabel, [119](#)
- pixmap1
 - QELabel, [119](#)
- pixmap2
 - QELabel, [119](#)
- pixmap3
 - QELabel, [119](#)
- pixmap4
 - QELabel, [119](#)
- pixmap5
 - QELabel, [119](#)
- pixmap6
 - QELabel, [119](#)
- pixmap7
 - QELabel, [120](#)
- precision
 - QEAAnalogProgressBar, [65](#)
 - QELabel, [120](#)
 - QELineEdit, [127](#)
- profilePlot, [49](#)
- PushButtonSpecifications, [49](#)
- QBitStatus, [50](#)
- QBitStatusManager, [51](#)
- QCaAlarmInfo, [52](#)
- QCaConnectionInfo, [52](#)
- QCaDataPoint, [53](#)
- QCaDataPointList, [53](#)
- QCaDateTime, [53](#)
- QCaEventFilter, [54](#)
- QCaEventItem, [54](#)
- QCaEventUpdate, [54](#)
- QCaInstalledFiltersListItem, [55](#)
- qcaobject::QCaObject, [55](#)
- qcastatemachine::ConnectionQCaStateMachine, [33](#)
- qcastatemachine::QCaStateMachine, [57](#)
- qcastatemachine::ReadQCaStateMachine, [211](#)
- qcastatemachine::SubscriptionQCaStateMachine, [218](#)
- qcastatemachine::WriteQCaStateMachine, [225](#)

- QCaVariableNamePropertyManager, 57
- QEAAnalogIndicator, 58
- QEAAnalogIndicator::Band, 32
- QEAAnalogIndicator::BandList, 33
- QEAAnalogIndicatorManager, 60
- QEAAnalogProgressBar, 61
 - addUnits, 64
 - allowDrop, 64
 - arrayAction, 64
 - enabled, 65
 - format, 65
 - int, 65
 - leadingZero, 65
 - localEnumeration, 65
 - notation, 65
 - precision, 65
 - requestEnabled, 64
 - trailingZeros, 66
 - useDbPrecision, 66
 - userLevelEnabled, 66
 - userLevelEngineerStyle, 66
 - userLevelScientistStyle, 66
 - userLevelUserStyle, 66
 - userLevelVisibility, 67
 - variable, 67
 - variableAsToolTip, 67
 - variableSubstitutions, 67
 - visible, 67
- QEAAnalogProgressBarManager, 68
- QEBitStatus, 68
 - allowDrop, 71
 - enabled, 71
 - int, 71
 - requestEnabled, 71
 - userLevelEnabled, 71
 - userLevelEngineerStyle, 71
 - userLevelScientistStyle, 72
 - userLevelUserStyle, 72
 - userLevelVisibility, 72
 - variable, 72
 - variableAsToolTip, 72
 - variableSubstitutions, 72
 - visible, 73
- QEBitStatusManager, 73
- QEByteArray, 73
- QEComboBox, 74
 - allowDrop, 77
 - enabled, 77
 - int, 77
 - requestEnabled, 77
 - userLevelEnabled, 77
 - userLevelEngineerStyle, 78
 - userLevelScientistStyle, 78
 - userLevelUserStyle, 78
 - userLevelVisibility, 78
 - variable, 78
 - variableAsToolTip, 79
 - variableSubstitutions, 79
 - visible, 79
- QEComboBoxManager, 79
- QEConfiguredLayout, 80
- QEConfiguredLayoutManager, 82
- QEDragDrop, 82
- QEFileBrowser, 84
- QEFileBrowserManager, 86
- QEFloating, 86
- QEFloatingFormatting, 87
- QEForm, 87
- QEFormManager, 89
- QEFrame, 90
 - allowDrop, 92
 - enabled, 92
 - int, 92
 - requestEnabled, 92
 - userLevelEnabled, 92
 - userLevelEngineerStyle, 92
 - userLevelScientistStyle, 93
 - userLevelUserStyle, 93
 - userLevelVisibility, 93
 - variableAsToolTip, 93
 - visible, 93
- QEFrameManager, 94
- QEGenericButton, 94
- QEGroupBox, 97
 - allowDrop, 99
 - enabled, 99
 - int, 99
 - requestEnabled, 99
 - userLevelEnabled, 99
 - userLevelEngineerStyle, 99
 - userLevelScientistStyle, 100
 - userLevelUserStyle, 100
 - userLevelVisibility, 100
 - variableAsToolTip, 100
 - visible, 100
- QEGroupBoxManager, 101
- QEImage, 101
 - allowDrop, 107
 - beamXVariable, 107
 - beamYVariable, 107

- clippingHighVariable, [107](#)
- clippingLowVariable, [107](#)
- clippingOnOffVariable, [107](#)
- enabled, [107](#)
- heightVariable, [108](#)
- imageVariable, [108](#)
- int, [108](#)
- regionOfInterestHVariable, [108](#)
- regionOfInterestWVariable, [108](#)
- regionOfInterestXVariable, [108](#)
- regionOfInterestYVariable, [108](#)
- requestEnabled, [107](#)
- targetTriggerVariable, [108](#)
- targetXVariable, [109](#)
- targetYVariable, [109](#)
- userLevelEnabled, [109](#)
- userLevelEngineerStyle, [109](#)
- userLevelScientistStyle, [109](#)
- userLevelUserStyle, [109](#)
- userLevelVisibility, [110](#)
- variableAsToolTip, [110](#)
- variableSubstitutions, [110](#)
- visible, [110](#)
- widthVariable, [110](#)
- QEImageManager, [111](#)
- QEInteger, [111](#)
- QEIntegerFormatting, [112](#)
- QELabel, [112](#)
 - addUnits, [117](#)
 - allowDrop, [117](#)
 - arrayAction, [118](#)
 - dbValueChanged, [117](#)
 - enabled, [118](#)
 - format, [118](#)
 - int, [118](#)
 - leadingZero, [118](#)
 - localEnumeration, [119](#)
 - notation, [119](#)
 - pixmap0, [119](#)
 - pixmap1, [119](#)
 - pixmap2, [119](#)
 - pixmap3, [119](#)
 - pixmap4, [119](#)
 - pixmap5, [119](#)
 - pixmap6, [119](#)
 - pixmap7, [120](#)
 - precision, [120](#)
 - QELabel, [117](#)
 - requestEnabled, [117](#)
 - trailingZeros, [120](#)
- UPDATE_PIXMAP, [117](#)
- UPDATE_TEXT, [117](#)
- updateOption, [120](#)
- updateOptions, [117](#)
- useDbPrecision, [120](#)
- userLevelEnabled, [120](#)
- userLevelEngineerStyle, [120](#)
- userLevelScientistStyle, [121](#)
- userLevelUserStyle, [121](#)
- userLevelVisibility, [121](#)
- variable, [121](#)
- variableAsToolTip, [121](#)
- variableSubstitutions, [121](#)
- visible, [122](#)
- QELabelManager, [122](#)
- QELineEdit, [122](#)
 - addUnits, [126](#)
 - allowDrop, [126](#)
 - arrayAction, [126](#)
 - enabled, [126](#)
 - format, [127](#)
 - int, [127](#)
 - leadingZero, [127](#)
 - localEnumeration, [127](#)
 - notation, [127](#)
 - precision, [127](#)
 - requestEnabled, [126](#)
 - trailingZeros, [127](#)
 - useDbPrecision, [128](#)
 - userLevelEnabled, [128](#)
 - userLevelEngineerStyle, [128](#)
 - userLevelScientistStyle, [128](#)
 - userLevelUserStyle, [128](#)
 - userLevelVisibility, [128](#)
 - variable, [129](#)
 - variableAsToolTip, [129](#)
 - variableSubstitutions, [129](#)
 - visible, [129](#)
- QELineEditManager, [129](#)
- QELink, [130](#)
- QELinkManager, [132](#)
- QELog, [132](#)
- QELogin, [135](#)
- QELoginManager, [137](#)
- QELogManager, [137](#)
- QEPeriodic, [138](#)
 - allowDrop, [141](#)
 - enabled, [142](#)
 - int, [142](#)
 - readbackLabelVariable1, [142](#)

- readbackLabelVariable2, 142
- requestEnabled, 141
- userLevelEnabled, 142
- userLevelEngineerStyle, 142
- userLevelScientistStyle, 142
- userLevelUserStyle, 143
- userLevelVisibility, 143
- variableAsToolTip, 143
- variableSubstitutions, 143
- visible, 143
- writeButtonVariable1, 143
- writeButtonVariable2, 144
- QEPeiodic::elementInfoStruct, 37
- QEPeiodic::userInfoStructArray, 219
- QEPeiodicComponentData, 144
- QEPeiodicManager, 144
- QEPeiodicTaskMenu, 145
- QEPeiodicTaskMenuFactory, 145
- QEpicsPV, 145
- QEPlot, 146
 - allowDrop, 151
 - enabled, 151
 - int, 151
 - requestEnabled, 151
 - userLevelEnabled, 151
 - userLevelEngineerStyle, 151
 - userLevelScientistStyle, 152
 - userLevelUserStyle, 152
 - userLevelVisibility, 152
 - variable1, 152
 - variable2, 152
 - variable3, 152
 - variable4, 153
 - variableAsToolTip, 153
 - variableSubstitutions, 153
 - visible, 153
- QEPlotManager, 153
- QEPushButton, 154
 - allowDrop, 158
 - altReadbackVariable, 158
 - enabled, 158
 - int, 158
 - requestEnabled, 158
 - userLevelEnabled, 158
 - userLevelEngineerStyle, 158
 - userLevelScientistStyle, 159
 - userLevelUserStyle, 159
 - userLevelVisibility, 159
 - variable, 159
 - variableAsToolTip, 159
 - visible, 160
- QEPushButtonManager, 160
- QEPvProperties, 160
 - allowDrop, 163
 - enabled, 163
 - int, 163
 - requestEnabled, 163
 - userLevelEnabled, 163
 - userLevelEngineerStyle, 164
 - userLevelScientistStyle, 164
 - userLevelUserStyle, 164
 - userLevelVisibility, 164
 - variable, 164
 - variableAsToolTip, 164
 - variableSubstitutions, 165
 - visible, 165
- QEPvPropertiesManager, 165
- QERadioButton, 166
 - allowDrop, 169
 - enabled, 169
 - int, 170
 - requestEnabled, 169
 - userLevelEnabled, 170
 - userLevelEngineerStyle, 170
 - userLevelScientistStyle, 170
 - userLevelUserStyle, 170
 - userLevelVisibility, 171
 - variable, 171
 - variableAsToolTip, 171
 - variableSubstitutions, 171
 - visible, 171
- QERadioButtonManager, 172
- QERecipe, 172
- QERecipeManager, 175
- QEScript, 176
- QEScriptManager, 178
- QEShape, 178
 - allowDrop, 185
 - enabled, 185
 - int, 185
 - requestEnabled, 185
 - userLevelEnabled, 185
 - userLevelEngineerStyle, 185
 - userLevelScientistStyle, 186
 - userLevelUserStyle, 186
 - userLevelVisibility, 186
 - variable1, 186
 - variable2, 186
 - variable3, 186
 - variable4, 187

- variable5, [187](#)
- variable6, [187](#)
- variableAsToolTip, [187](#)
- variableSubstitutions, [187](#)
- visible, [187](#)
- QEShapeManager, [188](#)
- QESlider, [188](#)
 - allowDrop, [191](#)
 - enabled, [191](#)
 - int, [191](#)
 - requestEnabled, [191](#)
 - userLevelEnabled, [191](#)
 - userLevelEngineerStyle, [192](#)
 - userLevelScientistStyle, [192](#)
 - userLevelUserStyle, [192](#)
 - userLevelVisibility, [192](#)
 - variable, [192](#)
 - variableAsToolTip, [193](#)
 - variableSubstitutions, [193](#)
 - visible, [193](#)
- QESliderManager, [193](#)
- QESpinBox, [194](#)
 - allowDrop, [196](#)
 - enabled, [196](#)
 - int, [197](#)
 - requestEnabled, [196](#)
 - userLevelEnabled, [197](#)
 - userLevelEngineerStyle, [197](#)
 - userLevelScientistStyle, [197](#)
 - userLevelUserStyle, [197](#)
 - userLevelVisibility, [198](#)
 - variable, [198](#)
 - variableAsToolTip, [198](#)
 - variableSubstitutions, [198](#)
 - visible, [198](#)
- QESpinBoxManager, [199](#)
- QEStr, [199](#)
- QEStrFormatting, [200](#)
- QEStrFormattingMethods, [201](#)
- QEStrStripChart, [202](#)
- QEStrStripChart::PrivateData, [48](#)
- QEStrStripChartItem, [204](#)
- QEStrStripChartItem::PrivateData, [49](#)
- QEStrStripChartItemDialog, [205](#)
- QEStrStripChartManager, [205](#)
- QEStrStripChartTimeDialog, [205](#)
- QESubstitutedLabel, [206](#)
 - labelText, [207](#)
 - textSubstitutions, [207](#)
- QESubstitutedLabelManager, [207](#)
- QEToolTip, [208](#)
- QEWid, [209](#)
- QEWid, [210](#)
- readbackLabelVariable1
 - QEPeriodic, [142](#)
- readbackLabelVariable2
 - QEPeriodic, [142](#)
- RecordSpec, [211](#)
- RecordSpecList, [211](#)
- regionOfInterestHVariable
 - QEImage, [108](#)
- regionOfInterestWVariable
 - QEImage, [108](#)
- regionOfInterestXVariable
 - QEImage, [108](#)
- regionOfInterestYVariable
 - QEImage, [108](#)
- requestEnabled
 - QEAnalogProgressBar, [64](#)
 - QEBitStatus, [71](#)
 - QEComboBox, [77](#)
 - QEFrame, [92](#)
 - QEGroupBox, [99](#)
 - QEImage, [107](#)
 - QELabel, [117](#)
 - QELineEdit, [126](#)
 - QEPeriodic, [141](#)
 - QEPlot, [151](#)
 - QEPushButton, [158](#)
 - QEPvProperties, [163](#)
 - QERadioButton, [169](#)
 - QEShape, [185](#)
 - QESlider, [191](#)
 - QESpinBox, [196](#)
- selectMenu, [212](#)
- setStyleEngineer
 - styleManager, [217](#)
- setStyleScientist
 - styleManager, [217](#)
- setStyleUser
 - styleManager, [217](#)
- standardProperties, [212](#)
- StateMachineTemplate, [214](#)
- styleManager, [214](#)
 - getStyleEngineer, [216](#)
 - getStyleScientist, [216](#)
 - getStyleUser, [217](#)
 - setStyleEngineer, [217](#)

- setStyleScientist, 217
- setStyleUser, 217
- styleUserLevelChanged, 217
- updateDataStyle, 217
- updateStatusStyle, 217
- styleUserLevelChanged
 - styleManager, 217
- targetTriggerVariable
 - QEImage, 108
- targetXVariable
 - QEImage, 109
- targetYVariable
 - QEImage, 109
- textSubstitutions
 - QESubstitutedLabel, 207
- trace, 218
- TrackRange, 219
- trailingZeros
 - QEAnalogProgressBar, 66
 - QELabel, 120
 - QELineEdit, 127
- UPDATE_PIXMAP
 - QELabel, 117
- UPDATE_TEXT
 - QELabel, 117
- updateDataStyle
 - styleManager, 217
- updateOption
 - QELabel, 120
- updateOptions
 - QELabel, 117
- updateStatusStyle
 - styleManager, 217
- useDbPrecision
 - QEAnalogProgressBar, 66
 - QELabel, 120
 - QELineEdit, 128
- userInfoStruct, 219
- userLevelEnabled
 - QEAnalogProgressBar, 66
 - QEBitStatus, 71
 - QEComboBox, 77
 - QEFrame, 92
 - QEGroupBox, 99
 - QEImage, 109
 - QELabel, 120
 - QELineEdit, 128
 - QEPeriodic, 142
- QEPlot, 151
- QEPushButton, 158
- QEPvProperties, 163
- QERadioButton, 170
- QEShape, 185
- QESlider, 191
- QESpinBox, 197
- userLevelEngineerStyle
 - QEAnalogProgressBar, 66
 - QEBitStatus, 71
 - QEComboBox, 78
 - QEFrame, 92
 - QEGroupBox, 99
 - QEImage, 109
 - QELabel, 120
 - QELineEdit, 128
 - QEPeriodic, 142
 - QEPlot, 151
 - QEPushButton, 158
 - QEPvProperties, 164
 - QERadioButton, 170
 - QEShape, 185
 - QESlider, 192
 - QESpinBox, 197
- userLevelScientistStyle
 - QEAnalogProgressBar, 66
 - QEBitStatus, 72
 - QEComboBox, 78
 - QEFrame, 93
 - QEGroupBox, 100
 - QEImage, 109
 - QELabel, 121
 - QELineEdit, 128
 - QEPeriodic, 142
 - QEPlot, 152
 - QEPushButton, 159
 - QEPvProperties, 164
 - QERadioButton, 170
 - QEShape, 186
 - QESlider, 192
 - QESpinBox, 197
- userLevelSignal, 219
- userLevelSlot, 220
- userLevelUserStyle
 - QEAnalogProgressBar, 66
 - QEBitStatus, 72
 - QEComboBox, 78
 - QEFrame, 93
 - QEGroupBox, 100
 - QEImage, 109

- QELabel, 121
- QELineEdit, 128
- QEPeiodic, 143
- QEPlot, 152
- QEPushButton, 159
- QEPvProperties, 164
- QERadioButton, 170
- QEShape, 186
- QESlider, 192
- QESpinBox, 197
- userLevelVisibility
 - QEAnalogProgressBar, 67
 - QEBitStatus, 72
 - QEComboBox, 78
 - QEFrame, 93
 - QEGroupBox, 100
 - QEImage, 110
 - QELabel, 121
 - QELineEdit, 128
 - QEPeiodic, 143
 - QEPlot, 152
 - QEPushButton, 159
 - QEPvProperties, 164
 - QERadioButton, 171
 - QEShape, 186
 - QESlider, 192
 - QESpinBox, 198
- UserMessage, 220
- UserMessageSignal, 222
- UserMessageSlot, 222
- variable
 - QEAnalogProgressBar, 67
 - QEBitStatus, 72
 - QEComboBox, 78
 - QELabel, 121
 - QELineEdit, 129
 - QEPushButton, 159
 - QEPvProperties, 164
 - QERadioButton, 171
 - QESlider, 192
 - QESpinBox, 198
- variable1
 - QEPlot, 152
 - QEShape, 186
- variable2
 - QEPlot, 152
 - QEShape, 186
- variable3
 - QEPlot, 152
 - QEShape, 186
- variable4
 - QEPlot, 153
 - QEShape, 187
- variable5
 - QEShape, 187
- variable6
 - QEShape, 187
- variableAsToolTip
 - QEAnalogProgressBar, 67
 - QEBitStatus, 72
 - QEComboBox, 79
 - QEFrame, 93
 - QEGroupBox, 100
 - QEImage, 110
 - QELabel, 121
 - QELineEdit, 129
 - QEPeiodic, 143
 - QEPlot, 153
 - QEPushButton, 159
 - QEPvProperties, 164
 - QERadioButton, 171
 - QEShape, 187
 - QESlider, 193
 - QESpinBox, 198
- VariableNameManager, 223
- variableSubstitutions
 - QEAnalogProgressBar, 67
 - QEBitStatus, 72
 - QEComboBox, 79
 - QEImage, 110
 - QELabel, 121
 - QELineEdit, 129
 - QEPeiodic, 143
 - QEPlot, 153
 - QEPvProperties, 165
 - QERadioButton, 171
 - QEShape, 187
 - QESlider, 193
 - QESpinBox, 198
- VideoWidget, 224
- visible
 - QEAnalogProgressBar, 67
 - QEBitStatus, 73
 - QEComboBox, 79
 - QEFrame, 93
 - QEGroupBox, 100
 - QEImage, 110
 - QELabel, 122
 - QELineEdit, 129

- QEPeiodic, [143](#)
- QEPlot, [153](#)
- QEPushButton, [160](#)
- QEPvProperties, [165](#)
- QERadioButton, [171](#)
- QEShape, [187](#)
- QESlider, [193](#)
- QESpinBox, [198](#)

- WidgetRef, [225](#)
- widthVariable
 - QImage, [110](#)
- writeButtonVariable1
 - QEPeiodic, [143](#)
- writeButtonVariable2
 - QEPeiodic, [144](#)

- zoomMenu, [226](#)