QELineEdit Widget

Zai Wang

10th July 2020

Contents

QELineEdit Widget Specification

Introduction

This document describes in detail the QELineEdit widget which is an EPICS aware widget provided by the EPICS Qt, aka QE, Framework.

This document was created as a separate widget specification document. The main reason for this is ease of maintenance and avoiding editing large and unwieldly word documents.

The QE Framework is distributed under the GNU Lesser General Public License version 3, distributed with the framework in the file LICENSE. It may also be obtained from here: http://www.gnu.org/licenses/lgpl-3.0-standalone.html

Description

The QELineEdit widget provides the ability to textually modify the value of a single PV. This widget is (indirectly) derived from QLineEdit. The example in Figure shows a QELineEdit widget connected to an ao record. While this widget is primarily intended for writing to string PVs, it can also be used with numerical PVs as in this example. However, in this case, a **Error! Reference source not found.** or a **Error! Reference source not found.** widget may be may be more appropriate.

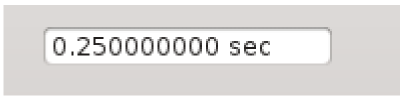
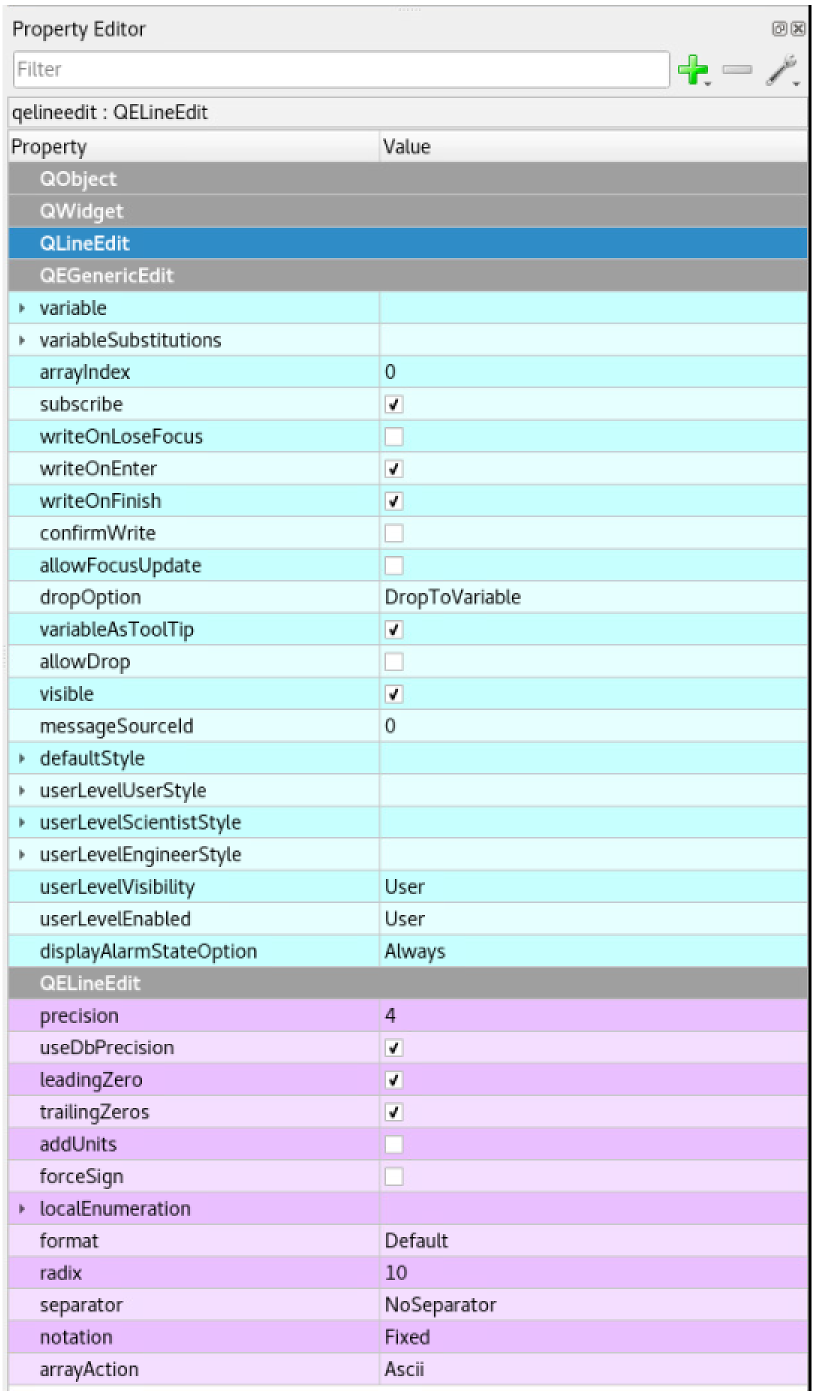


Figure 1 QELineEdit example

Properties

The behaviour of the widget is defined by the widget specific properties as shown in Figure .



Property Editor

Filter

qelineedit : QELineEdit

Property	Value
QObject	
QWidget	
QLineEdit	
QEGenericEdit	
variable	
variableSubstitutions	
arrayIndex	0
subscribe	✓
writeOnLoseFocus	☐
writeOnEnter	✓
writeOnFinish	✓
confirmWrite	☐
allowFocusUpdate	☐
dropOption	DropToVariable
variableAsToolTip	✓
allowDrop	☐
visible	✓
messageSourceId	0
defaultStyle	
userLevelUserStyle	
userLevelScientistStyle	
userLevelEngineerStyle	
userLevelVisibility	User
userLevelEnabled	User
displayAlarmStateOption	Always
QELineEdit	
precision	4
useDbPrecision	✓
leadingZero	✓
trailingZeros	✓
addUnits	☐
forceSign	☐
localEnumeration	
format	Default
radix	10
separator	NoSeparator
notation	Fixed
arrayAction	Ascii

Figure 2 QELineEdit properties

As well as the usual PV variable name, substitutions, display format, user level etc. properties, the widget has additional properties to control it mode of operation:

subscribe: bool

This determines if the widget subscribes for data updates and displays current data.

default value: true

writeOnLoseFocus: bool

This defines when true this widget automatically writes any changes when it loses focus.

default value: false

writeOnEnter: bool

This defines when true writes when the user presses 'enter'. Note, the current value will be written even if the user has not changed it.

default value: true

writeOnFinish: bool

This defines when true writes any changes when the user finished editing (the QLineEdit 'editingFinished' signal is emitted). No writing occurs if no changes were made.

default value: true

writeOnFinish: bool

This defines when true this widget will ask for confirmation (using a dialog box) prior to writing data focus.

default value: false

allowFocusUpdate: bool

This defines when true this widget update even if the widget currently has focus.

default value: false

dropOption: DropOption

Date type definition: enum DropOption { DropToVariable, DropToText, DropToTextAndWrite }

This defines the behaviour of drag and drop text to the QELineEdit widget.

DropToVariable: The dropped text will be used as a new PV name.

DropToText: The dropped text will be written on the widget.

DropToTextAndWrite: The dropped text will be written on the widget and to the defined PV as a value.

default value: DropToVariable