



QENumericEdit Widget

Andrew Starritt

29th July 2025

Copyright (c) 2020-2025 Australian Synchrotron

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License" within the QE_QEGuiAndUserInterfaceDesign document.

Contents

Introduction.....	3
QNumericEdit.....	3
Description.....	3
Properties.....	5
QENumericEdit.....	7
Description.....	7
Properties.....	7

Introduction

This document describes in detail the QNumericEdit and QENumericEdit widgets 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 unwieldy 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>

The QNumericEdit is a non-EPICS aware widget that allows the editing of numerical values. QENumericEdit extends the functionality of the QNumericEdit widget and provides EPICS-awareness via a single control Process Variable.

QNumericEdit

Description

QNumericEdit extends the functionality of the QLineEdit widget, and is somewhat like a spin box, save that the spin or increment/decrement value depends upon which character of the numerical field is selected/highlighted. This widget also supports the following functionality:

- a) Radix selection: 10 (default), 16, 8, 2;
- b) Optional “thousands” : comma (','), underscore ('_') or space (' ');
- c) Notation: Fixed point (default) or scientific.
Note: only the decimal radix is allowed for scientific notation;
- d) Value wrapping.

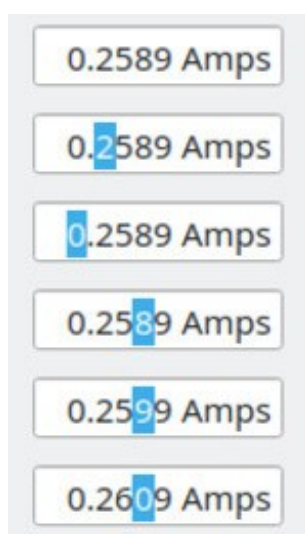
The QNumericEdit widget provides the ability to modify the value of a single numerical value, either integer or floating point. Figure 1 shows examples of the widget in several configurations, and in each case the widgets' suffix values have been set to “sec”.



Figure 1 QNumericEdit examples

The first example shows a QNumericEdit in its default configuration, and in appearance at least, looks very much like its QLineEdit counterpart. The second example shows the appearance with the separator property set to "comma". The 3rd, 4th and 5th show the same with the radix property set to Hexadecimal, Octal and Binary respectively. The widgets tool tip will be annotated accordingly.

Unlike QLineEdit, the user may only enter valid radix digits and if a sign is present enter a plus/minus ("+", "-"). A sign is displayed if and only if the allowed range of values encompasses negative values or the forceSign property is set True. The user may also use the left and right keys to navigate sideways to select a digit and use the up and down keys to increment or decrement the overall value by an amount corresponding to the unit value of the selected digit. An example sequence is shown and then described below :



1. widget not in focus
2. widget gets focus – the currently selected digit is after the first decimal point
3. left key – first digit selected – note: the decimal point skipped
4. right key three times – third digit after point selected
5. up key – increment value by 0.001
6. up key – increment value by 0.001, second digit has changed from 5 to 6.

Properties

The QNumericEdit widget specific properties are shown In Figure 2.

Property	Value
QObject	
QWidget	
QNumericEdit	
frame	<input checked="" type="checkbox"/>
▶ suffix	
▶ prefix	
▶ alignment	AlignRight, AlignVCenter
notation	Fixed
radix	Decimal
separator	NoSeparator
leadingZeros	3
precision	4
forceSign	<input type="checkbox"/>
wrapValue	<input type="checkbox"/>
minimum	-999.999900
maximum	999.999900
value	0.000000

Figure 2 QNumericEdit properties

These are described below:

- a) `frame` (default: `true`): when true the widget is displayed with a boarder;
- b) `suffix` (default: `""`): fixed text appended to the end of the numerical text;
- c) `prefix` (default: `""`): fixed text prepended to the start of the numerical text;
- d) `alignment` (default: `right`, vertical centre): alignment applied to embedded `QLineEdit`;
- e) `notation` (default `Fixed`): selects the notation used: fixed point or scientific;
- f) `radix` (default `Decimal`): allows the selection of display/editing radix. Unlike other widgets, this is restricted to just four options: `Decimal`, `Hexadecimal`, `Octal` or `Binary`.

Note: the widget assumes that the precision/leading zeros are appropriate for the selected radix;

Note: Scientific notation and non-decimal radix selections are mutually exclusive.

- g) `separator` (default `None`): allows the use of a character to break up the textual representation of the numerical value. This may be one of `None`, `Comma`, `Underscore` or `Space`. For `Decimal` and `Octal`, this is between every third digits, whereas for `Hexadecimal` and `Binary`, this is every 4th digit;
- h) `leadingZeros` (default `3`): specified the number of digits before the decimal point;
- i) `precision` (default `4`): specifies the number digits after the decimal point for display and editing;
- j) `forceSign` (default: `false`): when true a '+' sign is always display. When false a '+' sign only displayed when the value could be negative;
- k) `wrapValue` (default: `false`): When false, value editing is constrained to the lie between the minimum and maximum values. When true, the value will wrap from the maximum value to the minimum value when then maximum value is reached and vice-versa. Example use cases are wrapping phase angles and storage ring bucket numbers.
Note: the minimum is allowed, attempting to select the maximum value causes a wrap. For example if the minimum value is `-180` and the maximum values is `+180` then the sequeunce of values are : `177 → 178 → 179 → -180 → -179 → -178`.
- l) `minimum`: specifies the minimum value allowed to be entered;
- m) `maximum`: specifies the maximum value allowed to be entered; and
- n) `value`: specifies the current value.

QENumericEdit

Description

QENumericEdit is derived directly from QEAbstractWidget and thus inherits many standard properties used by QEWidgets, and includes an embedded QNumericEdit widget in order to provide the numerical editing capability (recall Qt only allows direct inheritance from one QObject/QWidget only). This allows the QENumericEdit to set the value of a numeric PV.

Properties

Like the QELineEdit widget, the subscribe, writeOnLoseFocus, writeOnEnter, writeOnFinish, confirmWrite and allowFocusUpdate properties modify the behaviour in exactly the same manner. The widget specific properties are shown in Figure 3. The additional properties, in addition to those provided by QNumericEdit, are described below:

- a) autoScale (default: true): when true the number of leading zeros, precision, minimum and maximum values will be determined from the PV's associated meta values. When false (or when not connected), the precision, leadingZeros, minimum and maximum property values are used; and
- b) addUnits (default: false): the widget displays includes any engineering units.

Note: the widget ensures consistency. For example: the maximum value is always greater than or equal to the minimum value. When in decimal mode, the sum of (b) and (c) is never greater than 15 which is approximately the maximum significance of an IEEE 64 bit float which is used to hold the underlying widget value (and indeed is the "best" significance supported by Channel Access).

Property Editor	
qnumericedit : QENumericEdit	
Property	Value
QObject	
QWidget	
QAbstractWidget	
variableAsToolTip	<input checked="" type="checkbox"/>
allowDrop	<input type="checkbox"/>
visible	<input checked="" type="checkbox"/>
messageSourceId	0
defaultStyle	
userLevelUserStyle	
userLevelScientistStyle	
userLevelEngineerStyle	
userLevelVisibility	User
userLevelEnabled	User
displayAlarmStateOption	Always
QENumericEdit	
variable	
variableSubstitutions	
arrayIndex	0
frame	<input checked="" type="checkbox"/>
alignment	AlignRight, AlignVCenter
autoScale	<input checked="" type="checkbox"/>
notation	Fixed
radix	Decimal
separator	None
leadingZeros	3
precision	2
minimum	-999.990000
maximum	999.990000
addUnits	<input checked="" type="checkbox"/>
writeOnLoseFocus	<input type="checkbox"/>
writeOnEnter	<input checked="" type="checkbox"/>
writeOnFinish	<input checked="" type="checkbox"/>
writeOnChange	<input type="checkbox"/>
confirmWrite	<input type="checkbox"/>
allowFocusUpdate	<input type="checkbox"/>

Figure 3 QENumericEdit properties

