QESelector Widget

Andrew Starritt

13th January 2022

Contents

QESelector Widget Specification

Introduction

This document describes in detail the QESelector 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

The QESelector widget extends the functionality of the QComboBox widget and provides EPICS-awareness via a single control Process Variable (PV), typically associated with a stringin/stringout record or in general a string field of any record. The use case for its development was to enable writing to the positioner and detector name fields of the sscan record.

Description

The QESelector inherits directly from Qt's own QEAbstractWidget which provides many standard properties. It holds/contains an embedded QCombBox widget that provides the graphical presentation functionality for the widgets.

The QESelector provides the user with one or more string options, which upon selection, is written to the nominated PV. The list of strings may either be defined using the stringList QStringList property, or may be defined in a text based configuration file. An example configuration file is show below.

```
# Comments allowed

MEX1MOT01.VAL  - first positioner
MEX1MOT02.VAL  - zoom positioner
MEX1MOT03.VAL  - x-axis positioner
MEX1MOT04.VAL  - y-axis positioner
MEX1MOT05.VAL  - z-axis positioner

# end
```

Comments are allowed (starting with #) and blank lines are ignored. The widget may also define a delimiter that allow extra support/descriptive test to be shown in the combo box, however only text up-to and before delimiter is written to the PV. In this example, the delimited is set to space, so only the PV name part of the selected entry is written to the PV.

In Figure 1 below, we see a QESelector widget on the left and a QELabel widget on the right. Both are looking at the same PV. The PV is currently empty and the severity/status is INVALID/UDF (undefined).

There is an implicit option provided by the QESelector widget which is an empty string. This is shown in the first position of the combo box as None.

Figure 1 Initial state

Figure 2 below shows the drop down menu provided by the combo box. It shows None together with the five options extracted from the configuration file.

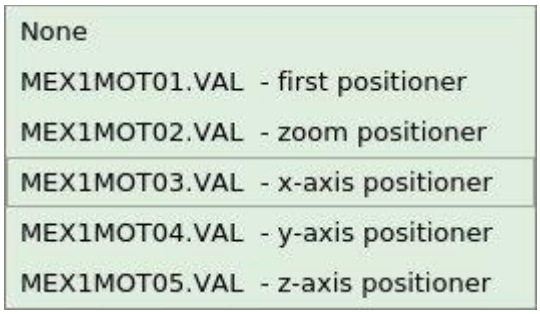

Figure 2 ComboBox drop down options

After making the selection, everything before the first space is written to the PV. See Figure 3 below.

Figure 3 Post item selection

Note: the same configuration file could be used for multiple instances of the QESelctor widget.

Properties

variable : QString
default value: empty string

This property specifies the PV variable name associated with this widget. Standard macro substitutions are allowed.

variableSubstitutions : QString
default value: empty string

This property specifies the default substitutions to be used. This may be particularly useful when creating/modifying a form within *designer*.

elementsRequired : int
default value: 0

The maximum number of elements to be read, 0 means all elements.

arrayIndex : int
default value: 0

The element of an array PV to be updated by the widget.

source : enumeration
allowed values: stringListSource, textFileSource
default value: stringListSource

This property selects weather the presented options are defines by the stringList propery or extracted from the file specified in the sourceFilename property.

stringList : QStringList
default value: empty list

This property can be used to define the selectable options presented to the user.

sourceFilename : QString
default value: empty string

This property defines the file name to read to extract the selectable options presented to the user.

This may be a absolute path file name, a relative path name (relative to the ui file holding the widget), or a resource file name.

maxVisibleItems : int

allowed values: 1 – lots
default value: 40

This property controls the maximum number items allowed on screen. All item are accessible using scrolling.

delimiter : enumeration

allowed values: NoDelimiter, SpaceDelimiter, CommaDelimiter
default value: SpaceDelimiter

This property defines the delimiter to be used, if any, to isolate the text written to the PV.

subscribe : bool

default value: true

writeOnChange : bool

default value: true

allowFocusUpdate : bool

default value: false