



QEMenuButton Widget

Andrew Starritt

18th March 2022

Copyright (c) 2022 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
Description	3
Substitutions	5
Restrictions	5
Customisation Menus	5
Properties.....	5
defaultSubstitutions : QString.....	5
labelText: QString.....	5
icon : QIcon	5
iconSize : QSize	6
menuEntries: QString.....	6

Introduction

This document describes in detail the QEMenuButton 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 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>

Description

The QEMenuButton widget is a QPushButton with an associated menu. Each menu entry provides a sub-set of the functionality provided by an individual QEPushButton, i.e. each menu item provides the following non-exclusive functions:

- a) Write a value to a process variable;
- b) Issue a command to the operating system; and/or
- c) Open a new GUI form.

The functionality provided is a sub-set of QEPushButton, and as such this widget does:

- a) **not** read and present process variable values;
- b) **not** emit dbChanged like signals; and
- c) **only** provides a 'clickText' value only (as opposed to pressed, released and checked values).

The QEMenuButton may be configured from within designer by right-clicking on the widget and selecting the "Edit Menu Info..." option which launches the Menu Button Setup dialog (see example in Figure 1 Menu Button Setup dialog below).

The left-hand side of the set up dialog provides a menu tree, while the right-hand side the set of "properties" associated with the selected menu item. The context menu over the tree provides three options:

- a) Add Menu Item – creates a menu action item
- b) Add Sub menu – create a sub menu item holder (like the shutter node in Figure 1 below); and
- c) Delete menu Item – delete the menu item and any associated sub menu items.

Menu items may also be dragged and dropped *within* the menu tree to allow the menu tree to be arranged. The default allocated menu names are of the form X followed by a 5-digit number, e.g. X00011, and should be renamed to something much more meaningful.

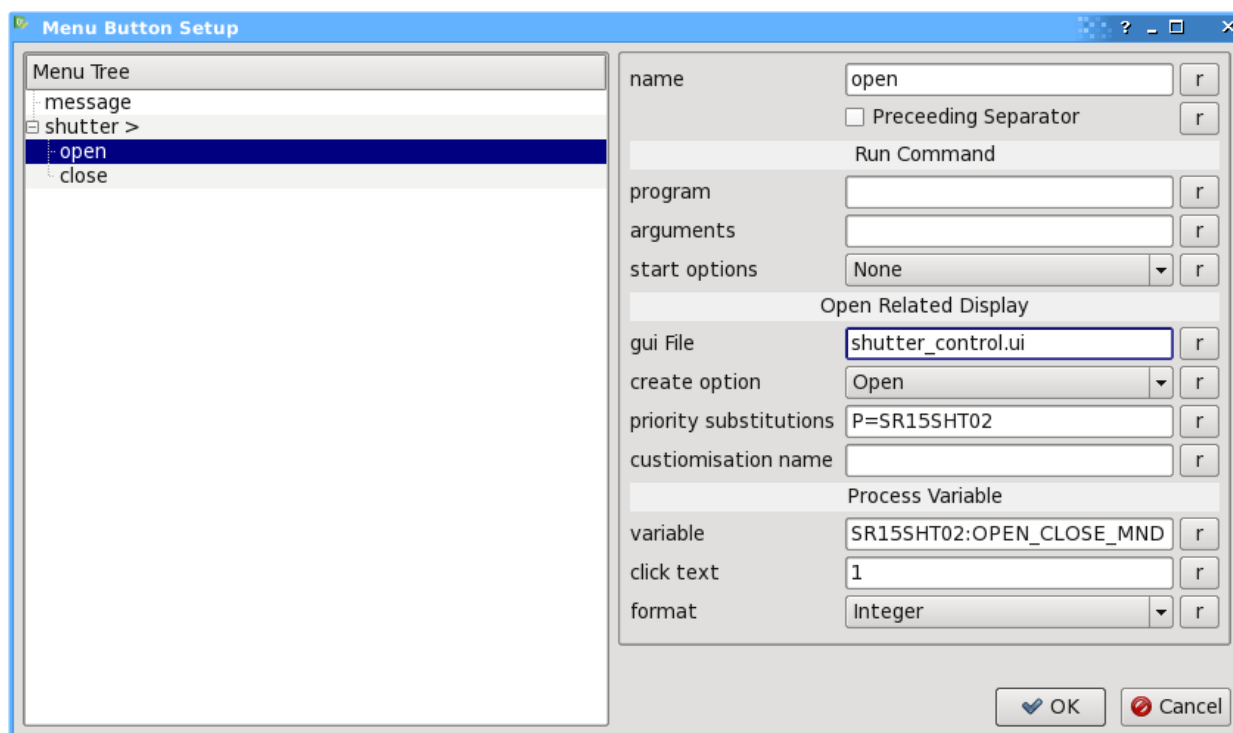


Figure 1 Menu Button Setup dialog

The right hand side allows the program to run, gui file to open and/or the process variable to be written to and value to be written to be specified. These are essentially as described in the QEPushButton, QERadioButton and QECheckBox section of the main widget specification document.

The only additional "property" is the preceding separator checkbox which adds a menu separator.

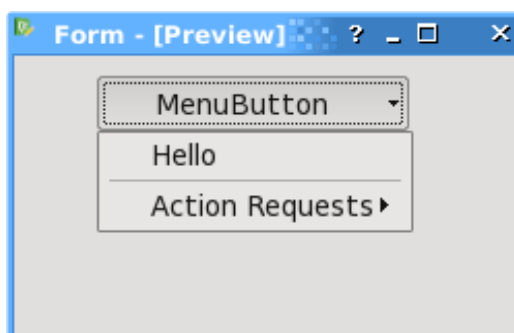


Figure 2 Menu Button example

Substitutions

The standard macro substitution mechanism can be used for the following items within the QEMenuButton widget. These are:

- a) The button text – see labelText property below;
- b) The menu and submenu names;
- c) Program names and arguments;
- d) Gui file names and priority substitutions; and
- e) Process variable name and click text.

Restrictions

The following are not (currently) implement for QEMenuButton:

- a) Copy/paste within the menu hierarchy tree;
- b) Checkable menu items; and
- c) User Level visibility control of individual menu items (although the QEMenuButton as a whole is subject to the regular user level style/visibility controls).

Customisation Menus

An alternative to the QEMenuButton is the definition of a customisation file which is described in the QE_QEGuiAndUserInterfaceDesign document.

Properties

The QEMenuButton inherits directly from QEAbstractWidget and as such inherits all its properties. The widget has the following class specific properties.

defaultSubstitutions : QString

This defines the default substitutions that are applied to the QEMenuButton widget as described in the Substitutions paragraph above.

labelText: QString

default value: "MenuButton"

The text that appears on the button.

icon : QIcon

default value: none

The icon that appears on the menu button.

iconSize : QSize

default value: 16 x 16

Defines the size of icon that appears on the menu button.

menuEntries: QString

default value: '<MenuButton Version="1"/>'

This property holds the xml string that defines the menu/sub-menu structure together with the associated program names, arguments, gui file names, process variables etc. This property is **not** intended to be directly edited by the user. The "Edit Menu Info..." option described above should be used.