Release Notes

Release Index

[r3.6.1 1](#_Toc511654276)

[r3.5.3 1](#_Toc511654277)

[r3.5.2 3](#_Toc511654278)

[r3.5.1 3](#_Toc511654279)

[r3.4.3 5](#_Toc511654280)

[r3.4.2 6](#_Toc511654281)

[Earlier Releases 6](#_Toc511654282)

This document provides a summary of the changes between releases. Use the **git log** command to obtain details of any changes, The section headings in this document correspond to the tags used within the repositories. While not guaranteed, we will endeavor to keep the latest master set of files compile-able and functionally correct.

# r3.6.1

Release date: 20th April 2018. The main change for this release is the inclusion of Archiver Appliance support. Please see <https://qtepics.github.io/archiver_appliance.html>

# r3.5.3

Release date: 16th April 2018. The main changes in this release are:

### qegui

In the standard File menu, there are two new menu items. These are:

1. List PV Names …  
   This will offer the user a Save File dialog and if a file nominated, then it will write all the PV names used on the current window to the nominated file; and
2. Screen Capture…  
   This will offer the user a Save File dialog and if a file nominated, then it will perform a screen scrape of the the window and write the image to the nominated file (png format).

### qeframework

The QEPlotter now checks for and does not attempt to plot NaN and +/- Inf numbers. The widgits minimum and maximum X and Y range values are now exposed as properties.

The QEStripChart widget modified to provide an expression evaluation capability. This is similar to that already provided by QEPlotter.

Reorganized the how ffmpeg functionality is/is not included into QEImage. No functional change per se but does now avoids a seg-fault in designer.

QEUtilities include the functionality for List PV names and Screen Capture, so available to application specific Qt applications.

For single variable widgets such as QELabel, QESimpleShape etc., added an elementsRequired integer property in order to limit the number of requested elements from an array PV. The default value is 0 which is read as subscribe for all elements. While restricting the number of elements subscribed for is sensible for monitors such as QESimpleShape and QEBitStatus (the reason behind this change) care should be taken when using it with control widgets.

Make disconnected graphical widgets unambiguously greyer. Modify QBitStatus to use the same washed-out colours as QESimpleShape.

Issue the disconnected signals immediately on subscribe - then await connection. This ensures widgets look disconnected asap if the associated PV does not exist.

Updated/added font change event in the event filter (as opposed to overriding the fontChange function which isn't virtual in Qt5) to these widgets: QEHistogram, QEMenuButton, QNumericEdit, QRadioGroup, QERadioGroup and QETable. This widgets now respond property to font changes within designer and at run time.

For QLabel based widgets (QELabel, QESubstitutedLabel) widgets now have a light gray colour style, both within designer and at run time (by setting the defaultStyle property). This colour is very close to, but sufficiently different from the standard form background colour, to be unobtrusive and yet help visualisation when designing forms and at run time.

For alarm sensitive labels, this will be superseded by the standard alarm colour when displaying the form at run time.

Note: For non alarm sensitive QELabels on forms with a non standard background colour, you will have to clear the defaultStyle property.

QEDescriptionLabel widget - new. This is directly inherited from QELabel. There is no extra functionality; however the widget has different default property values (smaller font, not alarm sensitive, clear style-sheet, no indentation). It is suitable for / intended for displaying descriptive text, typically from the .DESC field of a record.

QECalcout widget – new. This widget provides a calcout-like widget. So much so that where applicable the property names have been chosen to match the calcout record. This widget can be used instead of and/or to complement QELink. This widget has no PV variables of its own. The inputs, A to L, must be provided either by signals from other (QE) widgets or preset as 'constants' using the a to l properties. Under the covers, this widget uses the same calculation engine as the calc/calcout record.

The QEPVLoadSave widget has been updated to show not only the snap-shot/load/save value, but also show the live value and difference value.

The built in list of records and associated record fields used by the QEPvProperties widget has been updated to reflect base 3.15.5.

Updated support documentation.

# r3.5.2

Release date: 11th November 2017. The main changes for this release are:

### qegui

Above and beyond any application scaling (see option -a in documentation and/or qegui -h), qegui will now scale an individual window in response to a number of control+key combinations in a similar fashion to many browsers and other programs. The control+key actions are:

* control+plus-key or (control+equal-key): increase scaling by 2%;
* control+minus-key: decrease scaling by 2%; and
* control+zero-key: reset scaling.

### qeframework

The framework project file now automatically defines the \_MINGW macro if the EPICS host architecture is "win32-x86-mingw" or "windows-x64-mingw".

The QEPvProperties' context menu now provides a process record option (this writes 1 to the record's PROC field) and now provides consistent context menu even when clicking on the value field within this widget.

The QEPlotter has been updated so that it does not attempt to plot data if the associated array PV has an invalid severity. In now also clears any old plot data when establishing a new channel connection.

Incorporated Bob Gunion's change to the QEMenuButton which has modified to introduce a labelText property to allow the button's text to be set and also to honor the widget's font setting.

All single PV widgets now emit a parameter-less dbValueChanged signal in addition to various other signals.

The units (typically specified in the EGU field) for a DBF\_CHAR type has now available to the QELabel and other widget type. String formatting has modified to ignore the units when a DBF\_CHAR array PV is interpreted as a long string.

The way in which QEPvLoadSave widget handles array PVs and enumeration PVs has been improved.

# r3.5.1

Release date: 9th July 2017. The main change for this release is that the QEPlugin library has been split the into two distinct libraries which are:

* QEFramework - the functional library which contains the widgets themselves together with support functionality. qegui and any other bespoke display managers programs will need to link against this library.
* QEPlugin - strictly the widget plugin library usable by any application using the ui loader, such as qegui and designer. The QEPlugin library is dependent on the QEFramework library.

Both libraries are part of the the qeframework repository, with QEFramework built from the qeframeworkSup application and the QEPlugin library built from the new

qepluginApp application. Using linux-x86\_64 as an example EPICS host architecture the libraries are built and installed into:

*<top>*/lib/linux-x86\_64/libQEFramework.so

*<top>*/lib/linux-x86\_64/designer/libQEPlugin.so

This means that QT\_PLUGIN\_PATH needed not change as the plugin library is installed into a directory called designer (as was previously "faked" by use of

a symbolic link). Application building using the framework library need to reference QEFramework as opposed to QEPlugin.

Because the qepluginApp build uses the qeframeworkSup, the following has been added to the qeframework configure/RELEASE file:

QE\_FRAMEWORK=...

This need to be configured to point to the qeframeworkSup top directory. Unfortunately one cannot say QE\_FRAMEWORK=$(TOP). Please see comments in configure/RELEASE.

There have also been a number of other changes to qegui and qegramework since release r3.4.3 as outlined below. Apart from referencing the QEFramework library there have been no other changes to qeByteArrayTest, qeMonitor, qeReadArchive etc.

### qegui

* For windows - if the main window is off screen, then re-position windows to position (0,0)
* Introduced independent font scaling capability. Use the -f options. Run qegui -h for details.

### qeframework

* Modified QEConfiguredLayout, QEFileBrowser, QELog, QERecipe, QEScript to scope local enumeration definitions so that they don't clash if two or more headers using in same compilation unit.
* mpeg - more version related conditional compilation.
* Added $$(EPICS\_BASE)/include/compiler/msvc to the include path and for windows defined EPICS\_CALL\_DLL when compiling the QE framework.
* Updated some of the EPICS aware widget designer icons to be distinct from non EPICS aware counter parts.
* QEPvProperties - ensured correct field referenced when fields are sorted. Also added RMOD, ADEL, MDEL, ALST, MLST and SYNC fields to the build in motor record field list.
* QEPlotter and QEStripChart - save, set and restore all line styles when drawing graphic markups.
* Add forceSign boolean property to widgets using string formatting, the default value is false.
* Introduced independent font scaling functionality.
* QEPVLoadSave - ensure all array elements are of the same and a suitable type before writing to the channel.
* windowCustomisation - include a QAction object as opposed to inheriting from QAction. This is to support running on MAC OS.
* QEPlot - remove outward dependency on QWT, so that plugin library build does not need to include QWT header files, and do a general tidy up. Also inherit from QEFrame in order to provides the standard properties.

# r3.4.3

Release date: 5th June 2017. This release basically consolidates the transition from SourceForge to GitHub, and also accommodates the support of headless builds (on Windows). However, it also include a major bug fix and a number of minor functional changes

* qegui - modified the about dialog to include both the EPICS and QWT versions
* QEScratchPad avoid segmentation fault in drag drop handling - check before de-referencing.
* QEScratchPad - use comma separator for the value QELabel widgets.
* QEGraphic - set minimum size - keep Qt5 happy.
* Modify contextMenu to allow Edit PV action selection criteria to be specified as opposed to hard coded to Engineer User Level and enable by default for QEPVProperties and QEScratchPad.
* Truncates strings to 40 chars before attempting to write DBF\_STRING mode to a channel.
* Added .gitignore files to each repository.
* Component App/Sup make files are now more OS independent
* Updated sample .ui files to use current widget names (long over due).

# r3.4.2

Release date: 29th April 2017.

This is initial release at GitHub. This is functionally equivalent to the last SourceForge release despite the overall restructure.

# Earlier Releases

Refer to SourceForge for all history prior to release r.3.4.2 as the SourceForge history has not been transferred to GitHub.