EPICS Qt Environment Variables

# Environment Variables

Environment variables influence the behavior both at build time and run time.

## Build Time Environment Variables

### EPICS\_BASE (Required)

This defines the location of EPICS base. For headless building, this is defined in the configure/RELEASE file, however when using *qtcreator* it must be explicitly defined.

### EPICS\_HOST\_ARCH (Required)

This defines the host architecture, .e.g. linux-x86\_64, windows-x64, win32-x86-mingw.

### QE\_FRAMEWORK (Required)

This defines the location of the qeframework *installation* directory , i.e. the qeframework's *top* directory. For headless building, this is defined in the configure/RELEASE file, however when using *qtcreator* it must be explicitly defined. When using *qtcreator*, this must be defined for qeframework clients such as the qeplugin.pro and QEGuiApp.pro projects, but not for the framework itself. See note on QE\_TARGET\_DIR below.

### QWT\_INCLUDE\_PATH (Required)

This defines the location of the QWT header files. Typically this is /usr/include/qwt on Linux.

### QWT\_ROOT (Optional)

If not defined, the default location of QWT library is used. If defined then this is used to locate the QWT library.

### QE\_FFMPEG (Optional)

If you want MPEG support install FFmpeg and define this environment variable. This can be defined as anything on Linux, but must point to the FFmpeg directory on windows.

### QE\_CAQTDM (Optional)

If integration with PSI's caQtDM is required, this variable specified the location of the caQtDM\_Project directory.

### QE\_CAQTDM\_LIB (Optional)

Location of caQtDM\_Lib if not within location specified by QE\_CAQTDM.

### QTINC (Optional)

Applicable to Qt5 only, and defines include files needed for QtPrintSupport.

### QE\_ARCHAPPL\_SUPPORT (Optional)

When not defined, QE Framework will only be built to support EPICS Channel Archiver. If set to YES, the framework will be built for both EPICS Channel Access Archive and EPICS Archiver Appliance.

### PROTOBUF\_LIBS\_DIR (Optional)

If you want to build QE Framework with AA support (QE\_ARCHAPPL\_SUPPORT=YES), Google Protocol Buffers have to be installed on the system. If the directory containing the libraries is not on the library path, it can be defined using this variable.

### QE\_TARGET\_DIR (Optional/Deprecated)

When not defined, the executables, the libraries and the include files are installed the <*top>/bin*, <*top>/lib* and <*top>/include* directories as one would expect for an EPICS module. However, if this environment variable is defined it specifies the installation location out side of the top directory. If used such that the QEFramework files are installed elsewhere, then QE\_FRAMEWORK must be defined accordingly. This is a legacy option and we suggest it is not used.

## Run Time Environment Variables

### QT\_PLUGIN\_PATH (Required)

This must include **<where-epicsqt-is-located>**/qeframework/lib/*epics\_host\_arch*, so that qegui (and designer) may load and create QEFramework widgets. Using an environment variable is the easiest way to do this, there are other ways – please refer to the Qt documentation.

### PATH / LD\_LIBRARY\_PATH (Required)

This is OS dependent. The qegui executable and the EPICS and QEFramework libraries must be must be on the appropriate paths.

For Linux, the **-Wl,-rpath** link flags are used, so LD\_LIBRARY\_PATH need only be specified if the libraries are relocated.

On Windows builds, PATH must include %EPIC\_BASE%\bin\%EPICS\_HOST\_ARCH% which is where the ca.dll and Com.dll files are built; and  
**<where-epicsqt-is-located>\**qeframework\lib\%EPICS\_HOST\_ARCH% which is where the QEFramework.dll is located.

### QE\_UI\_PATH (Optional)

This defines alternative/additional paths used when searching for a ui file. This augments qegui's -u command line option.

### QE\_ARCHIVE\_TYPE (Optional)

This specified the type of archiver the framework will try to connect to. Currently supported options are CA for EPICS Channel Access Archive and ARCHAPPL for EPICS Archiver Appliance. If not defined, the default is CA. In order to connect to EPICS Archiver Appliance, the framework needs to be built with AA support.

### QE\_ARCHIVE\_LIST (Optional)

This specifies a space separated list of Channel Access archive servers. In turn each server is specified by a slash ('/') separated host name, port number and cgi program.

Example: "cr01arc01v:80/cgi-bin/ArchiveDataServer.cgi cr01arc02:80/cgi-bin/ArchiveDataServer.cgi"

### QE\_ARCHIVE\_PATTERN (Optional)

pattern match applied when extracting PV list from the archives.

### QE\_STRIPCHART\_PREDEFINED\_PVS (Optional)

This defines up to ten space separated PV names that are added to the Strip Chart context menu. If you don't know which PVs to define here, speak to your operators.

Example: "SR11BCM01:CURRENT\_MONITOR SR11BCM01:LIFETIME\_MONITOR"

### QE\_GLOBAL\_STYLE\_SHEET (Optional)

This defines a global style that is applied to the application. This also works within designer provided at least one QE widget is included on the form being designed. Running *qegui -h* provides a nice example of this, suitable for a Qt 5 to “fix” the way QGroupBox widgets are presented.

### QE\_RECORD\_FIELD\_LIST (Optional)

specifies a file that defines or replaces the set of field names associated with each record type. The framework already knows about all the records from EPICS base, most of the record types from the synApps distribution, together with the AS developed concat record type.  
Refer to the QEPvProperties widget in the QE\_QEGuiAndUserInterfaceDesign documentation for details.

### General

Running *qegui -h* provides further details about each of these environment variables.

As the EPICS Qt framework is a Channel Access client, the values assigned to:

EPICS\_CA\_AUTO\_ADDR\_LIST,

EPICS\_CA\_ADDR\_LIST,

EPICS\_CA\_MAX\_ARRAY\_BYTES,

EPICS\_CA\_SERVER\_PORT etc.

can affect the operation of this program. Please refer to EPICS R3.14 Channel Access Reference Manual for details.

*Last updated: Wed Apr 18 18:31:04 AEST 2018*