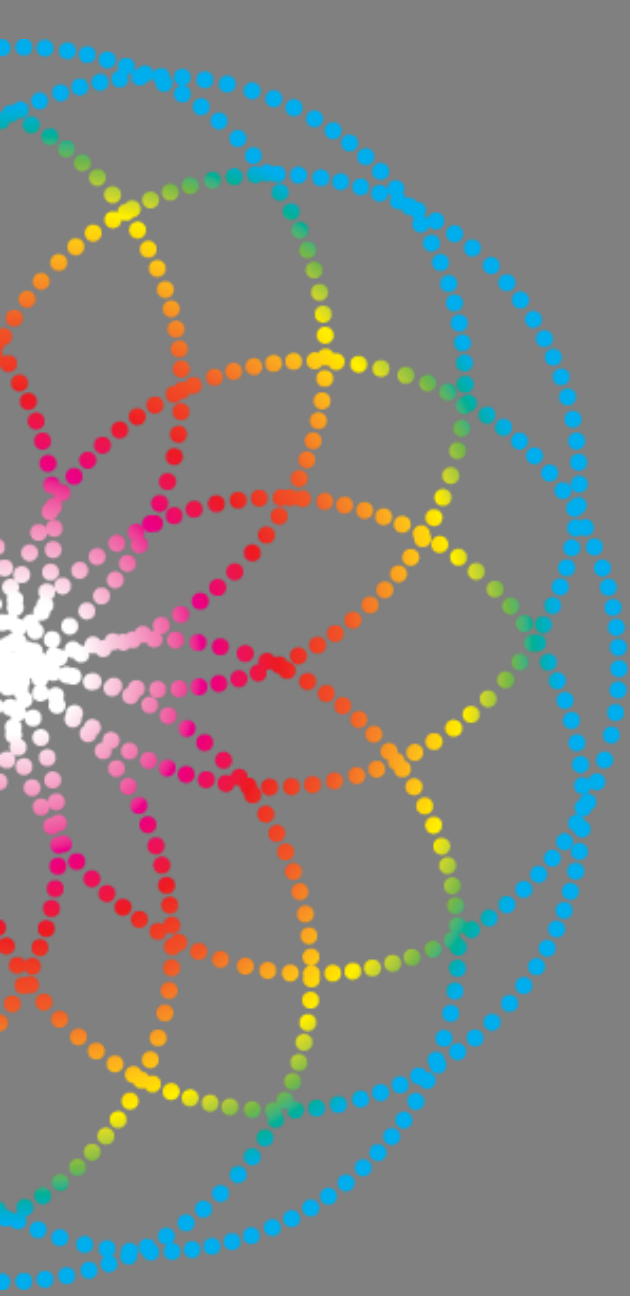


A decorative graphic on the left side of the slide, consisting of several overlapping circles made of small dots in various colors (blue, yellow, orange, red, pink, green).

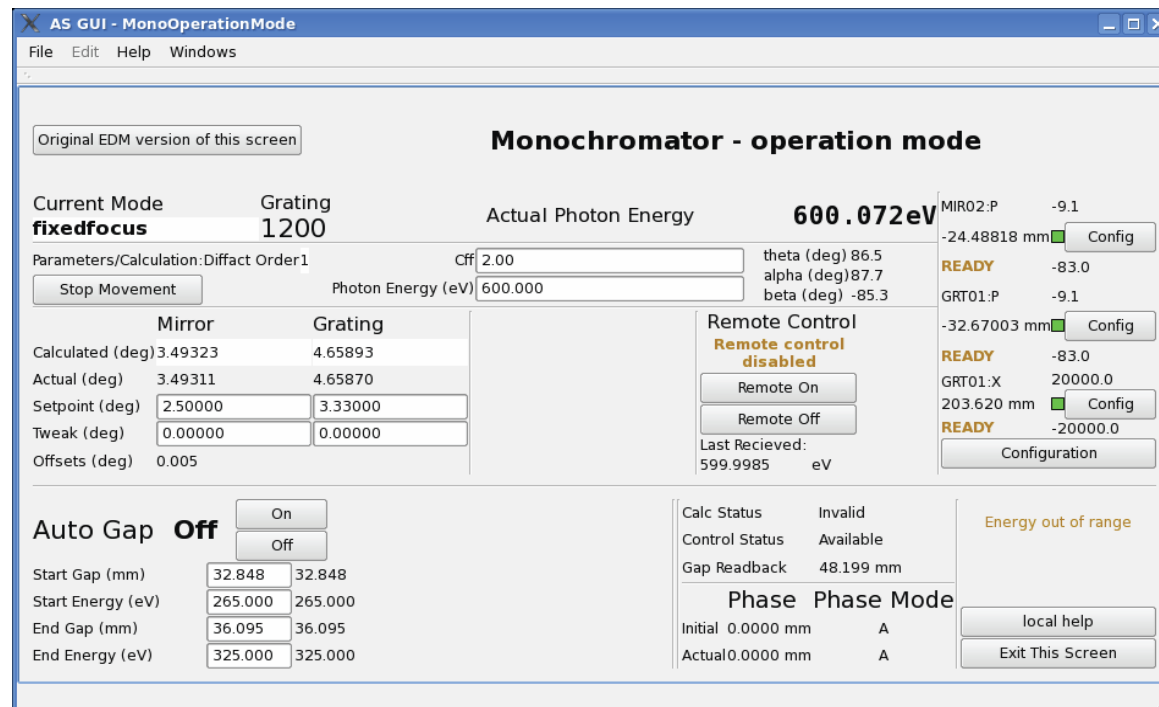
QT BASED GUI SYSTEM FOR EPICS CONTROL SYSTEMS

Ricardo Fernandes

*Supported
by*

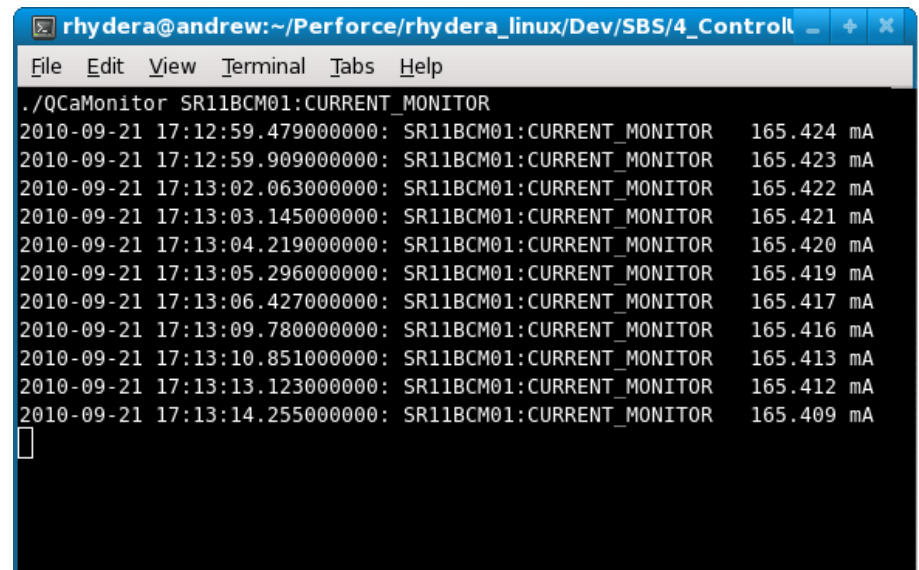
What is the QE Framework?

- Developed and maintained by the Australian Synchrotron
- Implemented in C++ and using Qt
- Display EPICS data (PVs) using appropriate Qt widgets



Benefits

- Reduce usage of different GUI systems (EDM, MEDM, ...)
- Rapid development of control system GUIs
- Development of console-based applications (e.g. servers)
- Able to run in different platforms

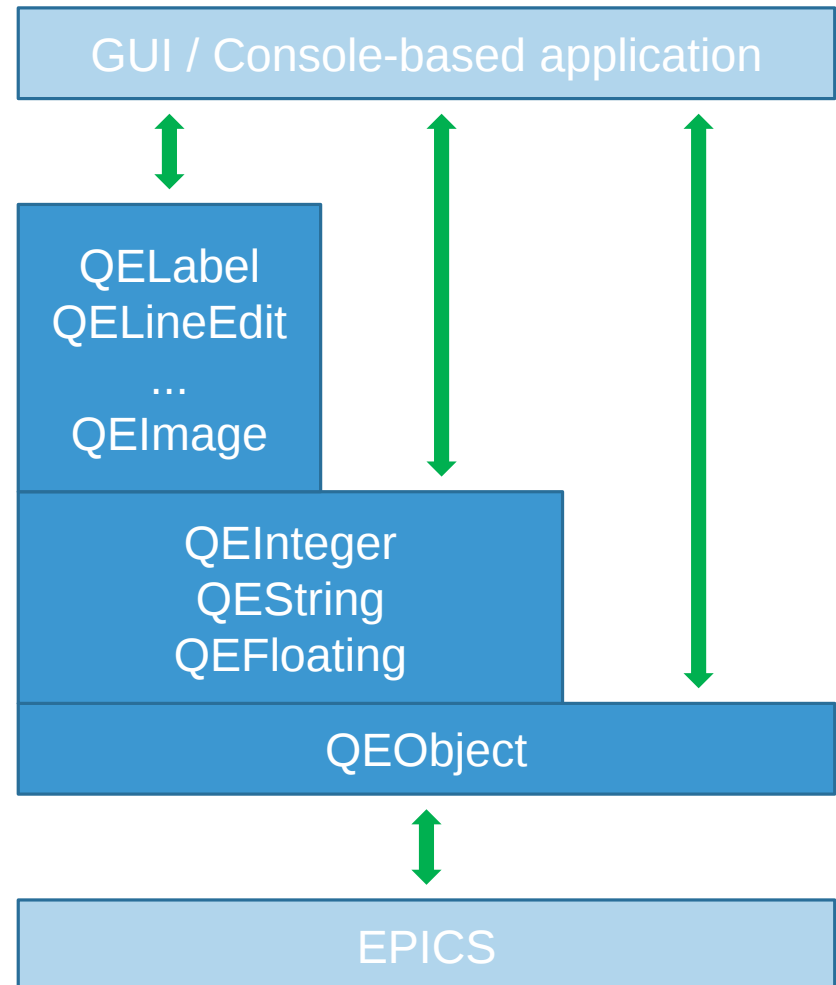


The screenshot shows a terminal window titled "rhydera@andrew:~/Perforce/rhydera_linux/Dev/SBS/4_Control". The window contains a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal output shows a command prompt followed by a series of timestamped log entries for a current monitor.

```
./QCaMonitor SR11BCM01:CURRENT_MONITOR
2010-09-21 17:12:59.479000000: SR11BCM01:CURRENT_MONITOR 165.424 mA
2010-09-21 17:12:59.909000000: SR11BCM01:CURRENT_MONITOR 165.423 mA
2010-09-21 17:13:02.063000000: SR11BCM01:CURRENT_MONITOR 165.422 mA
2010-09-21 17:13:03.145000000: SR11BCM01:CURRENT_MONITOR 165.421 mA
2010-09-21 17:13:04.219000000: SR11BCM01:CURRENT_MONITOR 165.420 mA
2010-09-21 17:13:05.296000000: SR11BCM01:CURRENT_MONITOR 165.419 mA
2010-09-21 17:13:06.427000000: SR11BCM01:CURRENT_MONITOR 165.417 mA
2010-09-21 17:13:09.780000000: SR11BCM01:CURRENT_MONITOR 165.416 mA
2010-09-21 17:13:10.851000000: SR11BCM01:CURRENT_MONITOR 165.413 mA
2010-09-21 17:13:13.123000000: SR11BCM01:CURRENT_MONITOR 165.412 mA
2010-09-21 17:13:14.255000000: SR11BCM01:CURRENT_MONITOR 165.409 mA
```

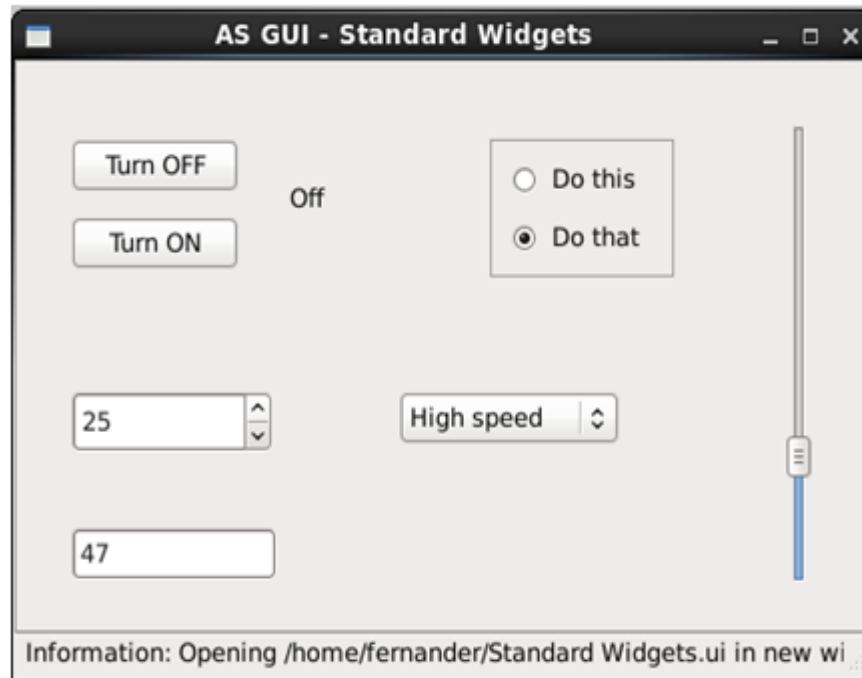
QE Framework Overview

- Composed by 28 C++ classes
- Organized in 3 logical groups
 - Data objects (4 classes)
 - Standard widgets (10 classes)
 - Extended widgets (14 classes)



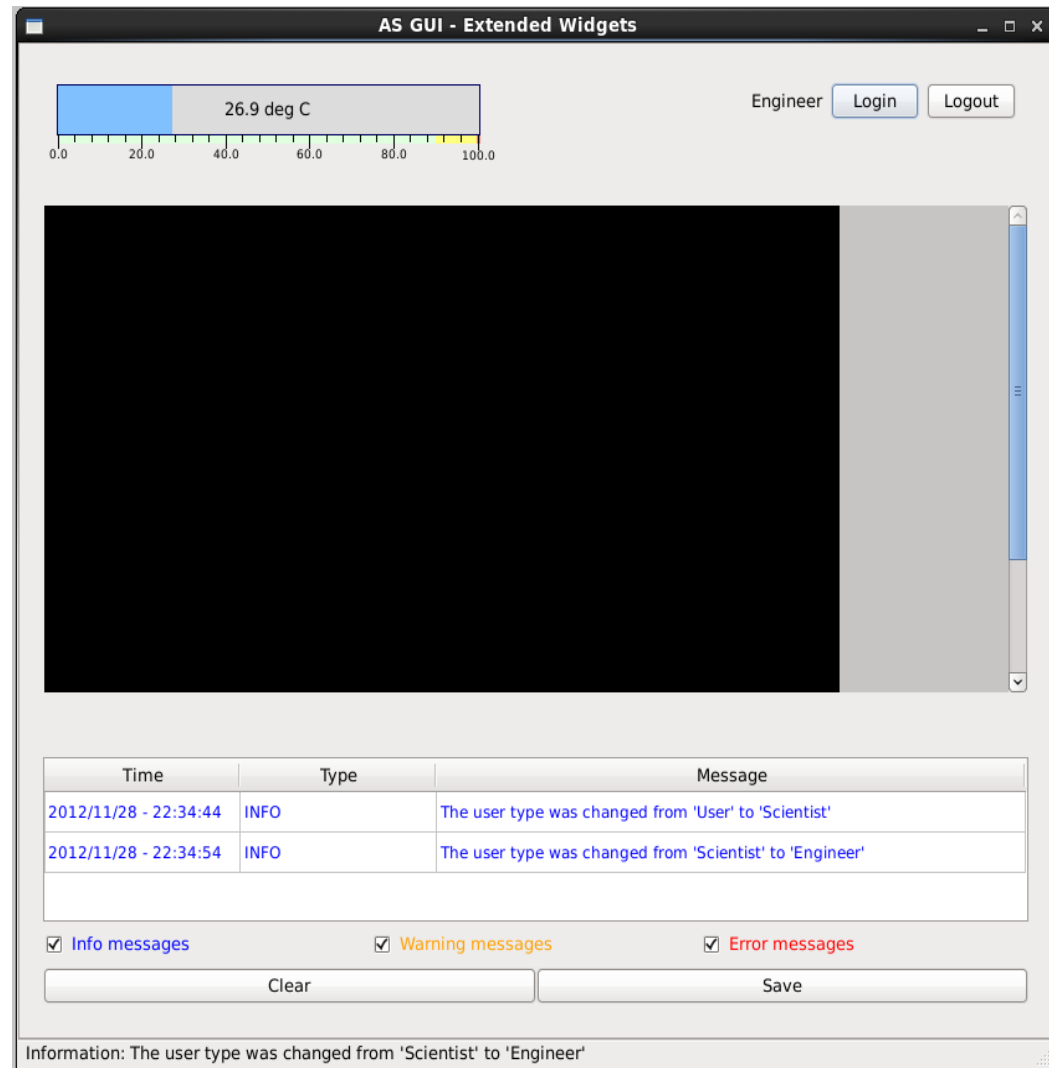
QE Framework Overview (Standard Widgets)

- QEComboBox
- QEForm
- QEFrame
- QEGroupBox
- QELabel
- QELineEdit
- QEPushButton
- QERadioButton
- QESlider
- QESpinBox



QE Framework Overview (Extended Widgets)

- QEAnalogProgressBar
- QEBitStatus
- QEFileBrowser
- QEImage
- QELink
- QELog
- QELogin
- QEPeriodic
- QEPlot
- QEPvProperties
- QERecipe
- QEScript
- QEShape
- QEStripChart



How to Use the QE Framework

- Programmatically via C++

```
void show(void)
{
    QFrame *myFrame;
    QLabel *myLabel;

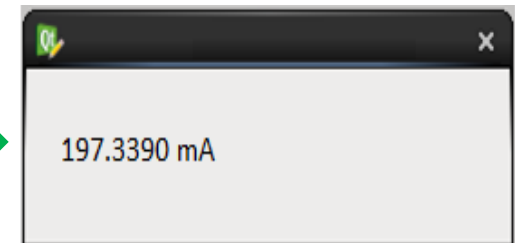
    myFrame = new QFrame();

    //creates object called 'myLabel' of type QLabel (belonging to QE framework)
    myLabel = new QLabel();

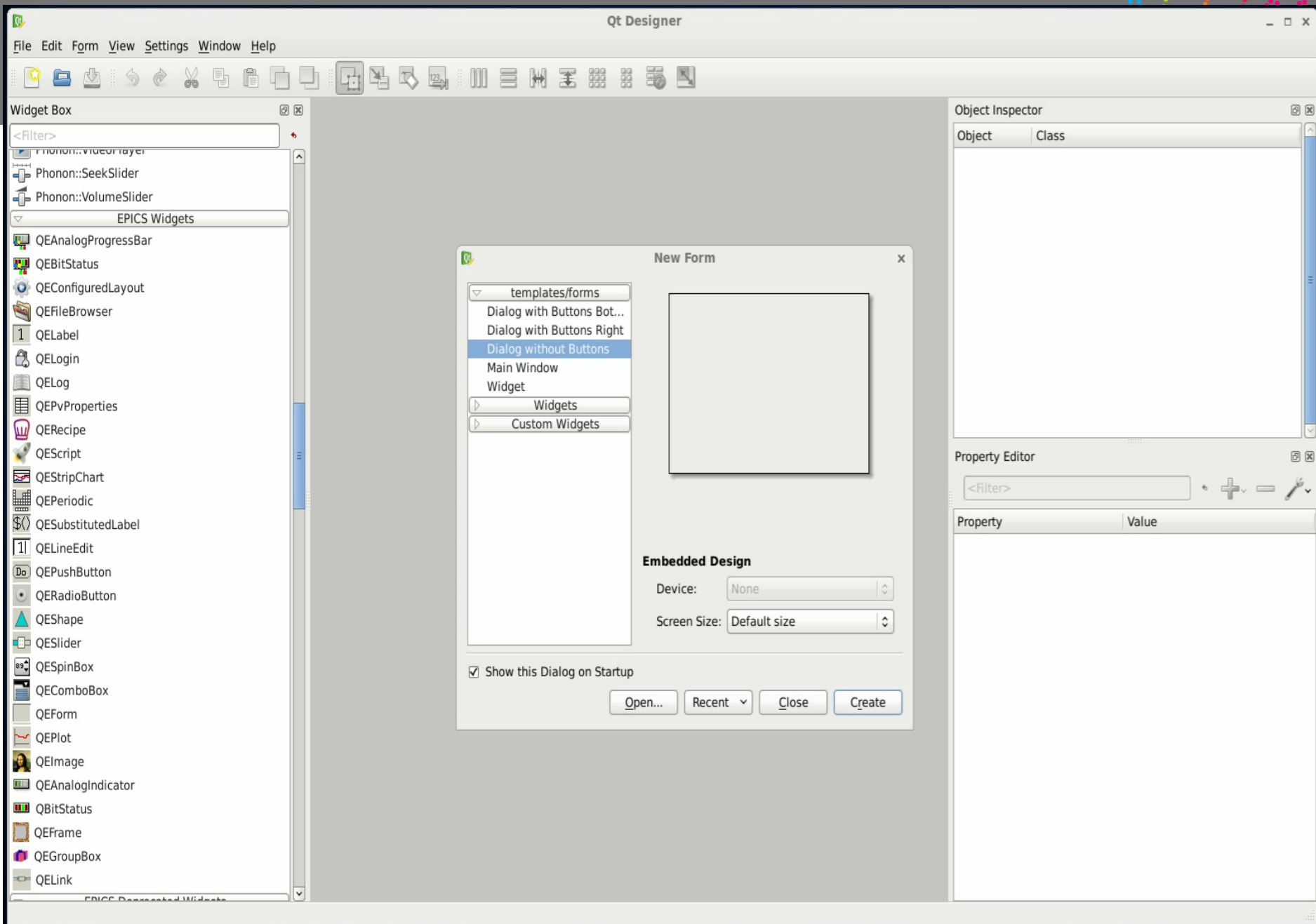
    //connects 'myLabel' to the PV called 'MACHINE_CURRENT'
    myLabel->setVariableName("MACHINE_CURRENT");

    //adds 'myLabel' to the frame layout and display machine current
    myFrame->layout()->addWidget(myLabel);

    myFrame->show();
}
```



- “Code-free” (drag & drop operations)
 - Creation of user interfaces through Qt Designer
 - Presentation of user interfaces through QEGui



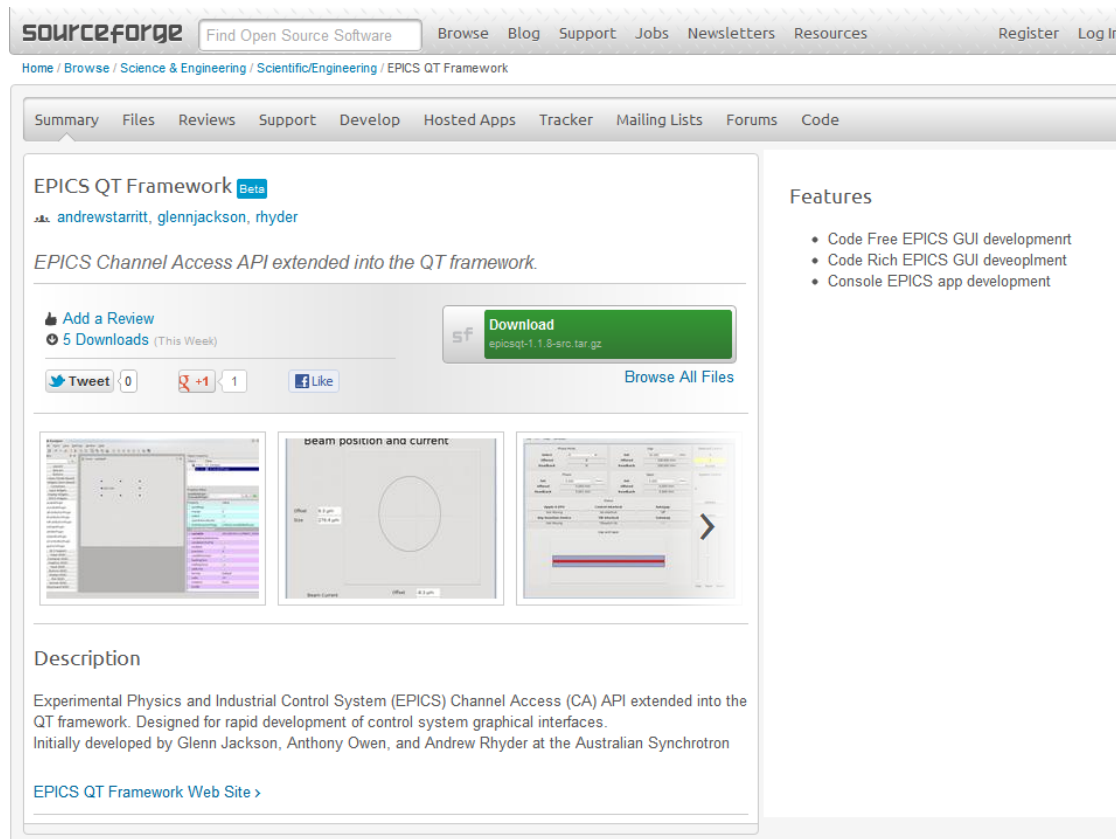
Future Developments



- Testing
- Export the framework into Python
- Further separation between data and graphical classes
- Support of EPICS v4

Location

- Official website: <http://sourceforge.net/projects/epicsqt/>
- Subversion repository: <https://epicsqt.svn.sourceforge.net/svnroot/>



The screenshot shows the SourceForge project page for the EPICS QT Framework. The page has a navigation bar at the top with links for Home, Browse, Science & Engineering, Scientific/Engineering, EPICS QT Framework, Register, and Log In. Below the navigation bar is a search bar and a list of tabs: Summary, Files, Reviews, Support, Develop, Hosted Apps, Tracker, Mailing Lists, Forums, and Code. The main content area is divided into two columns. The left column contains the project title "EPICS QT Framework" with a "Beta" badge, the authors "andrewstarritt, glennjackson, rhyder", and a description "EPICS Channel Access API extended into the QT framework." Below this is a section for reviews and downloads, showing "Add a Review", "5 Downloads (This Week)", and a "Download" button for "epicsqt-1.1.8-src.tar.gz". There are also social media links for Tweet, +1, and Like. The right column contains a "Features" section with three bullet points: "Code Free EPICS GUI development", "Code Rich EPICS GUI development", and "Console EPICS app development". Below the features section are three screenshots of the EPICS QT Framework GUI, showing a file browser, a beam position and current monitor, and a control panel. The bottom of the page has a "Description" section with a paragraph about the project and a link to the "EPICS QT Framework Web Site".

SOURCEFORGE Find Open Source Software Browse Blog Support Jobs Newsletters Resources Register Log In

Home / Browse / Science & Engineering / Scientific/Engineering / EPICS QT Framework

Summary Files Reviews Support Develop Hosted Apps Tracker Mailing Lists Forums Code

EPICS QT Framework Beta

by andrewstarritt, glennjackson, rhyder

EPICS Channel Access API extended into the QT framework.

[Add a Review](#)
5 Downloads (This Week)

[Download](#)
epicsqt-1.1.8-src.tar.gz

[Browse All Files](#)

[Tweet](#) 0 [+1](#) 1 [Like](#)

Features

- Code Free EPICS GUI development
- Code Rich EPICS GUI development
- Console EPICS app development

Description

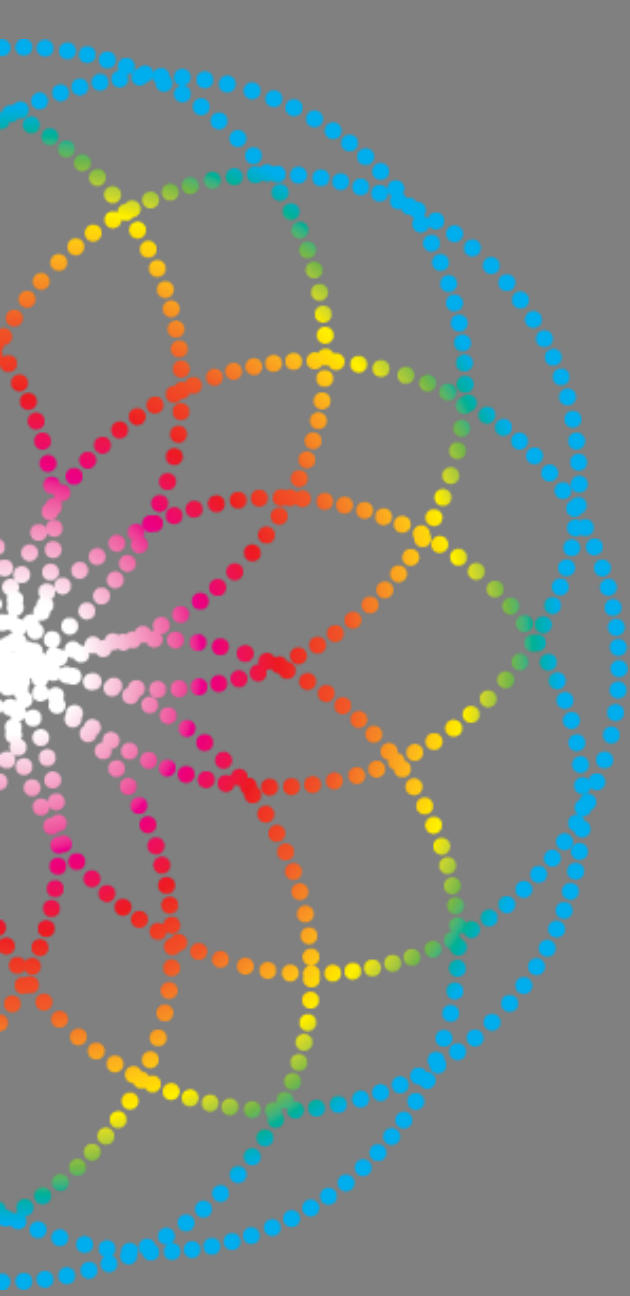
Experimental Physics and Industrial Control System (EPICS) Channel Access (CA) API extended into the QT framework. Designed for rapid development of control system graphical interfaces.
Initially developed by Glenn Jackson, Anthony Owen, and Andrew Rhyder at the Australian Synchrotron

[EPICS QT Framework Web Site](#)

Acknowledgements

- Andrew Rhyder (main author)
- Andrew Starritt
- Glenn Jackson
- Anthony Owen



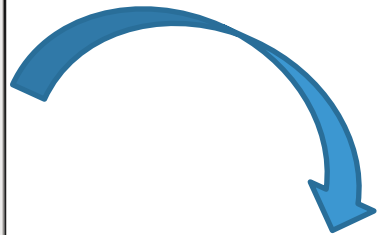
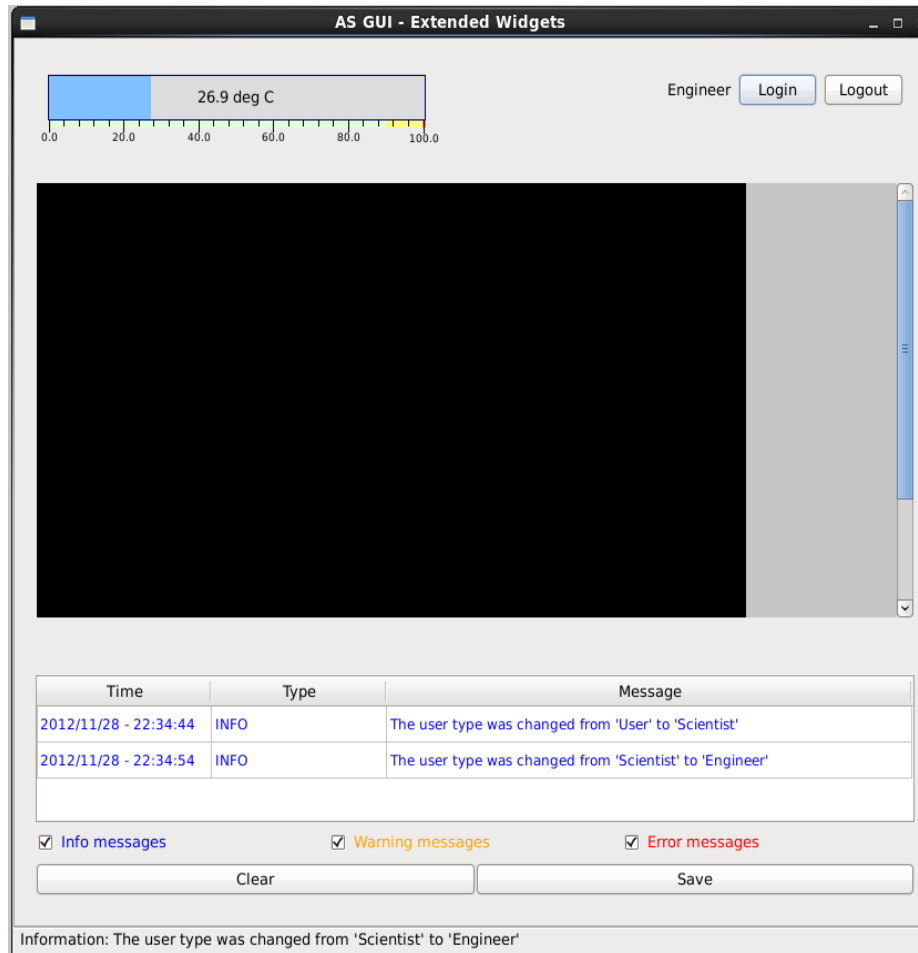
A decorative graphic on the left side of the slide, consisting of several overlapping circles made of small dots. The dots are colored in a gradient from blue to yellow to red to pink to white, creating a vibrant, multi-colored effect.

Thank you!

Supported
by

Framework Overview (Extended Widgets)

- QEAnalogProgressBar
- QEBitStatus
- QEFileBrowser
- QEImage
- QELink
- QELog
- QELogin
- QEPeriodic
- QEPlot
- QEPvProperties
- QERecipe
- QEScript
- QEShape
- QEStripChart



QEAnalogProgressBar	
variable	SR08FE01TES02:TEMPERATURE_MONITOR
variableSubstitutions	
variableAsToolTip	☒
enabled	☒
allowDrop	☐
visible	☒
messageSourceId	0
userLevelUserStyle	
userLevelScientistStyle	
userLevelEngineerStyle	
userLevelVisibility	User
userLevelEnabled	User
precision	4
useDbPrecision	☒
leadingZero	☒
trailingZeros	☒
addUnits	☒
localEnumeration	
format	Default
radix	10
notation	Fixed
arrayAction	Ascii
arrayIndex	0
useDbDisplayLimits	☒
alarmSeverityDisplayMode	none

Framework Overview (Data Objects)



- QObject
- QInteger
- QString
- QEFloating

```
// Stream data to the logger
source = new QString( pv, this, &format );
QObject::connect( source, SIGNAL( stringChanged() ),
                  logger, SLOT( log( const QString& ) ) );
```

Metrics

- Scenario
- Environment
 - 3 Ghz quad-core processor (2 GB of RAM)
 - Linux Fedora 16
 - EPICS 3.14.12
 - Qt 4.8
- Result
 - 7000 updates per second!

