



QCaMonitor

- QCaMonitor.pro
- Headers
- Sources
  - src
    - main.cpp
    - monitor.cpp

```
21  * You should have received a copy of the GNU General Public License
22  * along with the EPICS QT Framework.  If not, see <http://www.gnu.org/licenses/>.
23  *
24  * Copyright (c) 2009, 2010
25  *
26  * Author:
27  *   Andrew Rhyder
28  * Contact details:
29  *   andrew.rhyder@synchrotron.org.au
30  */
31
32  #include <QTime>
33  #include <QString>
34  #include <monitor.h>
35
36  monitor::monitor( QString pvIn )
37  {
38      stream = new QTextStream( stdout );
39
40      // Save the PV for logging updates
41      pv = pvIn;
42
43      // Create the data source, connect to data update and message signals, then subscribe to updates.
44      source = new QCaString( pv, this, &formatting, 1, &messages );
45      QObject::connect( source, SIGNAL( stringChanged( const QString&, QCaAlarmInfo&, QCaDateTime&, const unsigned int & ) ),
46                      this, SLOT( log( const QString&, QCaAlarmInfo&, QCaDateTime&, const unsigned int & ) ) );
47
48      QObject::connect( source, SIGNAL( connectionChanged( QCaConnectionInfo& ) ),
49                      this, SLOT( connectionChanged( QCaConnectionInfo& ) ) );
50
51      QObject::connect( &messages, SIGNAL( generalMessage( const QString& ) ), this, SLOT( message( const QString & ) ) );
52      source->subscribe();
53  }
54
55  // Log connection issues
56  void monitor::connectionChanged( QCaConnectionInfo& connectionInfo )
57  {
58      if( !connectionInfo.isChannelConnected() )
59          *stream << QString( "%1: %2  Channel not connected\n").arg( QTime::currentTime().toString() ).arg( pv );
60      if( !connectionInfo.isLinkUp() )
61          *stream << QString( "%1: %2  Link not up\n").arg( QTime::currentTime().toString() ).arg( pv );
62
63      stream->flush();
64  }
65
66  // Log data updates and messages
67  void monitor::log( const QString& data, QCaAlarmInfo&, QCaDateTime& timeStamp, const unsigned int & )
68  {
69      *stream << QString( "%1: %2  %3\n").arg( timeStamp.text() ).arg( pv ).arg( data );
70      stream->flush();
71  }
72
73  // Log messages
74  void monitor::message( const QString& message )
75  {
76      *stream << QString( "%1 %2  %3\n").arg( QTime::currentTime().toString() ).arg( pv ).arg( message );
77      stream->flush();
78  }
79
```

Open Documents

monitor.cpp

Scanning