Qt Surprises


CÔTES DU


PAIR
FIRST
SECOND
POINT
SIZE

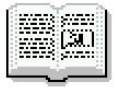

GUI Tool
Linux GUI

TROLL TECH

Software That Makes Sense

Welcome to Troll Tech's web site. Most of the material on this server is about our multi-platform C++ GUI application framework, Qt.


Qt Product Information


On-Line Qt Documentation

Learn to Program With Qt


Example Programs and Screen Shots


Download Qt


What Users Say About Qt


The Free Software License


Company Profile

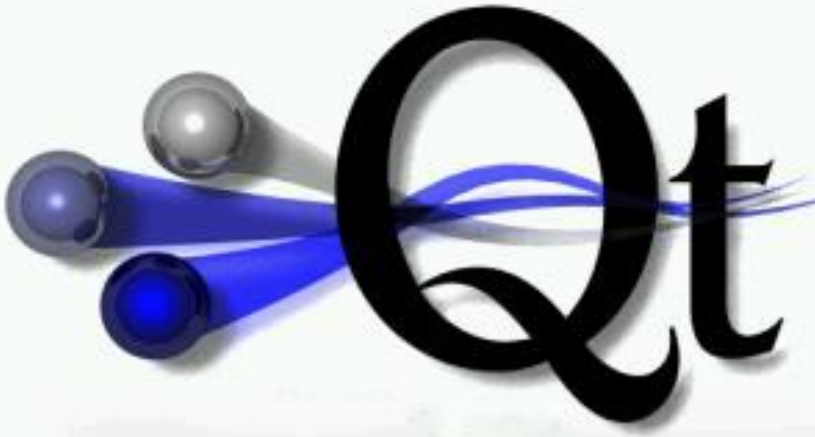

Availability and Pricing


Contact Us

Win $2000

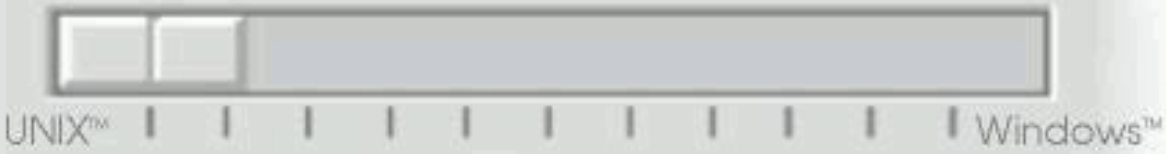
The Qt GUI programming contest has free entry, US$2000+ first prize, US$500+ second and third prizes. Read all about it!

Last modified 1996/12/07 by the Webmaster at Troll Tech



Multi-platform GUI Toolkit

for
Windows 95/NT
and UNIX/X11



TROLL
T·E·C·H

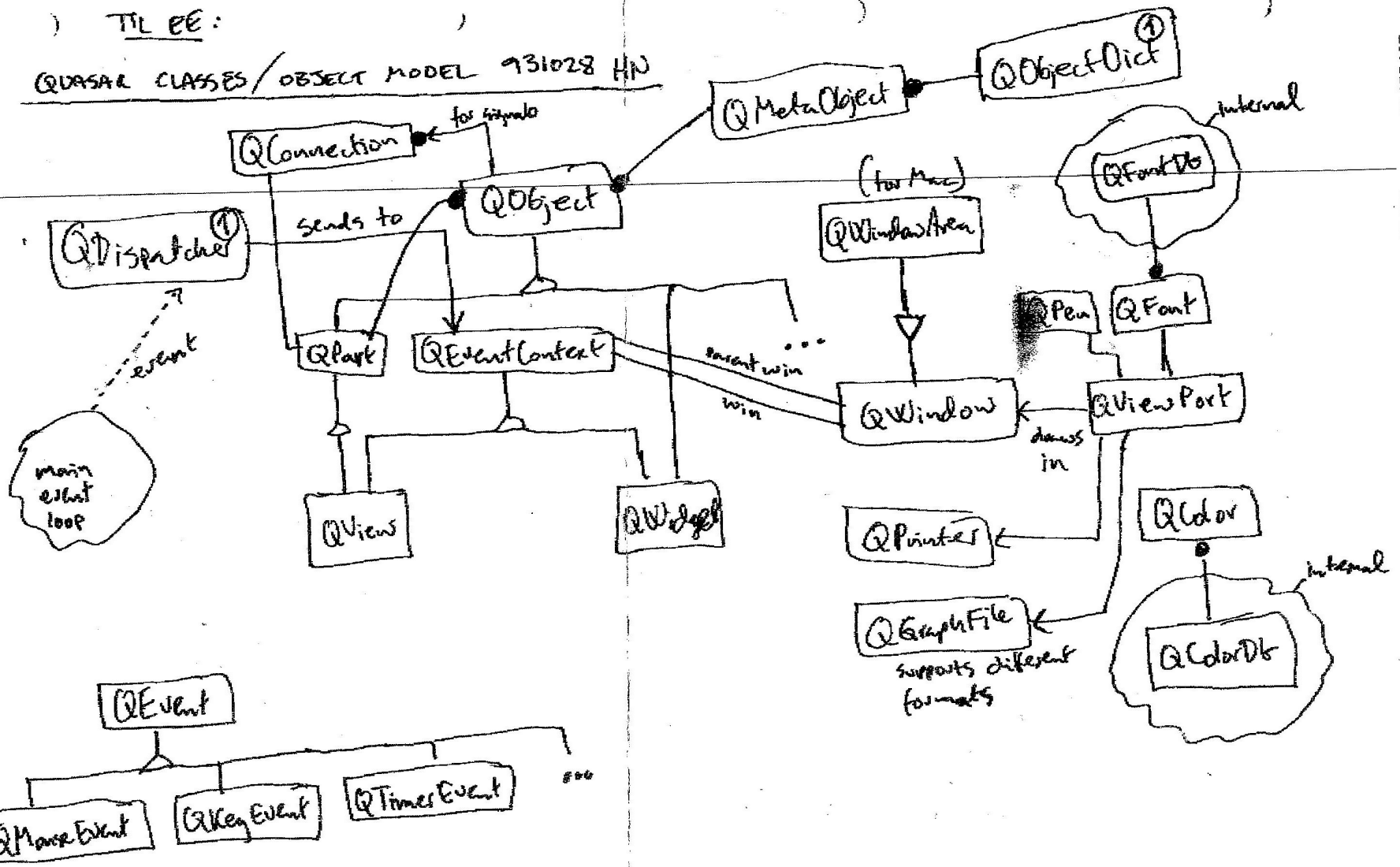

Troll Tech AS
957
Qt
Toolkit


The Multi-pla
C++ GUI To


Qt
[cooler]
www.troll-tech.com

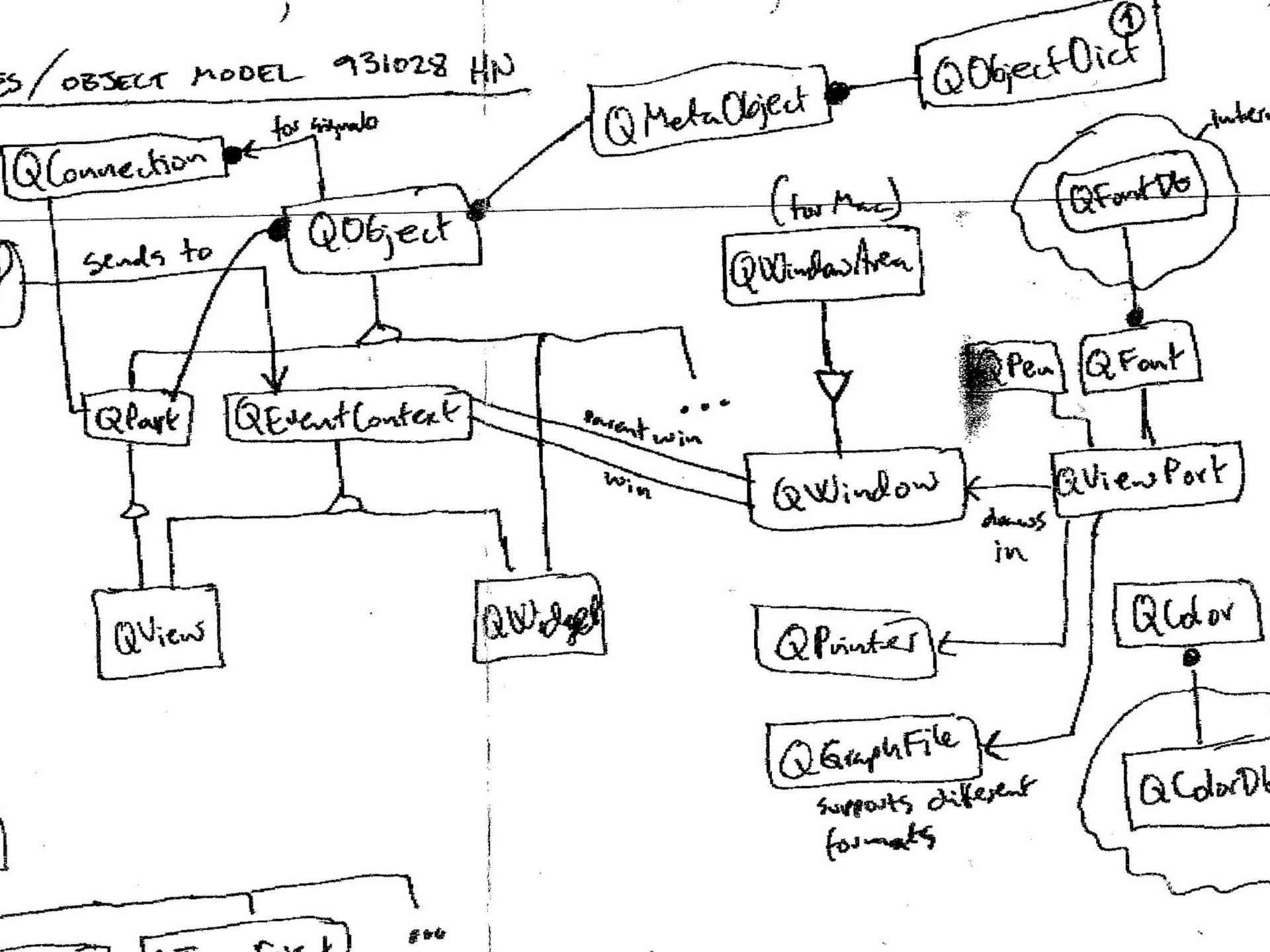
TIL EE:

QUASAR CLASSES / OBJECT MODEL 931028 HN



29.10.96 16:26 43 1 8777421 13 DIASONICS/SONOTRON WIEN S.01

ES / OBJECT MODEL 931028 HN



signal/slot access specifiers

Slot access check

```
bool QObject::connect(
    const QObject *sender,
    const char *signal,
       const char *member ) const;

static bool QObject::connect(
   const QObject *sender,
   const char *signal,
      const QObject *receiver,
   const char *member );
```

Signal access specifiers

Qt 0.60 – 0.90 (almost):

```
private signals:
   void something_secret_happened();

protected signals:
   void something_confidential_happened();

public signals:
   void something_cool_happened();
```

Qt 0.90 lsm-file

```
Begin3
Title:          Qt toolkit
Version:        0.90
Entered-date:   20may95
Description:    C++ GUI class library with Motif look and feel
Keywords:       gui library motif
Author:         qt-bugs@troll.no
Maintained-by:  qt-buts@troll.no
Primary-site:   sunsite.unc.edu /pub/Linux/devel/c++
Platform:       linux/X11R6
Copying-policy: freely distributable with certain restrictions
End
```

Case study:
Loading X11 font names

Loading X11 font names

```
char **XListFonts(display, pattern, maxnames, actual_names)
      Display *display;
      char *pattern;
      int maxnames;
      int *actual_names;
```

“-misc-fixed-medium-r-normal--16-160-72-72-m-80-tis620.2529-1”

From Qt 1.x (qfont_x11.cpp)

```
static char **getXFontNames( const char *pattern,
                             int *count )
{
    static int maxFonts = 256;
    char **list;
    while( 1 ) {
        list = XListFonts( QPaintDevice::x__Display(),
                   (char*)pattern, maxFonts, count );
        // I know precisely why 32768 is 32768.
        if ( *count != maxFonts || maxFonts >= 32768 )
            return list;
        XFreeFontNames( list );
        maxFonts *= 2;
    }
}
```

From Qt 2.0 (qfontdatabase.cpp)

```
void QFontDatabase::createDatabase()
{
    ...

    xFontList = XListFonts( qt_xdisplay(), "*”,
                            32767, &xFontCount );

    if ( xFontCount >= 32767 )
          qWarning( "More than 32k fonts, please notify
qt-
                    bugs@troll.no" );
```

From gtk 1.1.2 (gtkfontsel.c)

```
/* Fontnames  - A maximum of MAX_FONTS (32767) fontnames will be
retrieved from X Windows with XListFonts(). Any more are ignored.

   I think this limit may have been set because of a limit in
GtkList.

*/

#define MAX_FONTS 32767

xfontnames = XListFonts (GDK_DISPLAY(), "-*", MAX_FONTS,
&num_fonts);


if (num_fonts == MAX_FONTS)

  g_warning(_("MAX_FONTS exceeded. Some fonts may be missing."));
```

Q_METHOD

From Qt 0.60 (qobjdefs.h)

```
#define METHOD(a)  "0"#a
#define SLOT(a)    "1"#a
#define SIGNAL(a)  "2"#a

#define METHOD_CODE  0
#define SLOT_CODE    1
#define SIGNAL_CODE  2
```

Answer to Jesper Pedersen, KDAB, Jan 9, 2002

”He, this is funny. METHOD is a leftover from the old days and it was never used. I didn't know it still existed in the header file. We will probably remove it in an upcoming Qt release.” –Haavard Nord

From Qt 5.0 beta 1 (qobjdefs.h)

```
# ifndef QT_NO_KEYWORDS
#  define METHOD(a)   qFlagLocation("0" QTOSTRING(a) QLOCATION)
# endif
# define SLOT(a)     qFlagLocation("1" QTOSTRING(a) QLOCATION)
# define SIGNAL(a)   qFlagLocation("2" QTOSTRING(a) QLOCATION)
#else
# ifndef QT_NO_KEYWORDS
#  define METHOD(a)   "0" QTOSTRING(a)
# endif
# define SLOT(a)     "1" QTOSTRING(a)
# define SIGNAL(a)   "2" QTOSTRING(a)
#endif
#define QMETHOD_CODE  0                 // member type codes
#define QSLOT_CODE    1
#define QSIGNAL_CODE  2
#endif // QT_NO_META_MACROS
```

Easter eggs in Qt

xpm writing

From Qt 1.2

```
static char* xpm_color_name( int cpp, int index )
{
...
        if ( index == 1 )
            index = 64*44+21+1;
        else if ( index == 64*44+21+1 )
            index = 1;
        returnable[0] = ".#abcdefghijklmnopqrstuvwxyz"
           "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"[(index-1)/64];
        returnable[1] = ".#abcdefghijklmnopqrstuvwxyz"
           "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"[(index-1)%64];
        returnable[2] = '\0';
}
```

From Qt 5.0 beta

```
static const char* xpm_color_name(int cpp, int index)
{
...
// the following 4 lines are a joke!
        if (index == 0)
            index = 64*44+21;
        else if (index == 64*44+21)
            index = 0;
```

DEMO !

xpm
OS/2 (sort of)
(Qt 1.20)


SONY
RESET
PlayStation
1
2

A true easter egg

From Qt 2.0 (qmessagebox.cpp)

```
".........aa********aaaaa........",
"...........aaaaaaaaaaa..........",
".............aaaaaaa............"};
#define QT_END_TEXT qt_dialog_default_key

...

static const char *textAboutQt =
"<h3>This program is developed with Qt, the multi-platform C++ GUI toolkit.</h3>"
"<p>Qt version running with this application: <tt>%1</tt></p>"
"<p>Qt is a product of <b>Troll Tech AS </b>(http://www.troll.no).</p>"
"<p>Qt is available under two different licenses:</p>"
"<ul><li>The Free Edition, which may be used free of charge to develop"
"  Free Software on the X Window System.</li>"
"<li>The Professional Edition, which may be used to develop commercial"
"  software on both X and Microsoft Windows.</li></ul>";
void QT_END_TEXT() { QMessageBox::aboutQt(0,"E" "g" "g"); }
```

From Qt 2.0 (qapplication_x11.cpp)

```
bool QETWidget::translateKeyEventInternal( const XEvent *event, int& count,
        QString& text, int& state,
        char& ascii, int &code )
{

#ifndef Q_EE
    static int c  = 0;
    extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
    if ( tlw && state == '0' ) {
     switch ( code ) {
     case 0x4f: Q_EE(Key_Backtab); break;
     case 0x52: Q_EE(Key_Tab); break;
     case 0x54: Q_EE(Key_Escape); break;
     case 0x4c:
          if (c == Key_Return )
           qt_dialog_default_key();
          else
           Q_EE(Key_Backspace);
          break;
     }
}
#undef Q_EE
#endif
```

From Qt 2.0 (qapplication_x11.cpp)

```
  static int c  = 0;
  extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
  if ( tlw && state == '0' ) {
    switch ( code ) {
    case 0x4f: Q_EE(Key_Backtab); break;
    case 0x52: Q_EE(Key_Tab); break;
    case 0x54: Q_EE(Key_Escape); break;
    case 0x4c:
        if (c == Key_Return )
    qt_dialog_default_key();
        else
    Q_EE(Key_Backspace);
        break;
    }
  }
```

From Qt 2.0 (qapplication_x11.cpp)

```
  static int c  = 0;
  extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
  if ( tlw && state == '0' ) {
    switch ( code ) {
    case 0x4f: Q_EE(Key_Backtab); break;
    case 0x52: Q_EE(Key_Tab); break;
    case 0x54: Q_EE(Key_Escape); break;
    case 0x4c:
        if (c == Key_Return )
    qt_dialog_default_key();
        else
    Q_EE(Key_Backspace);
        break;
    }
  }
```

From Qt 2.0 (qapplication_x11.cpp)

```
  static int c  = 0;
  extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
  if ( tlw && state == '0' ) { // ‘0’ = 0X30 (Key_Ctrl | Key_Alt)
    switch ( code ) {
    case 0x4f: Q_EE(Key_Backtab); break;
    case 0x52: Q_EE(Key_Tab); break;
    case 0x54: Q_EE(Key_Escape); break;
    case 0x4c:
        if (c == Key_Return )
    qt_dialog_default_key();
        else
    Q_EE(Key_Backspace);
        break;
    }
  }
```

From Qt 2.0 (qapplication_x11.cpp)

```
  static int c  = 0;
  extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
  if ( tlw && state == (Key_Ctrl | Key_Alt) ) {
    switch ( code ) {
    case 0x4f: Q_EE(Key_Backtab); break;  // ASCII for “o”
    case 0x52: Q_EE(Key_Tab); break;      // ASCII for “r”
    case 0x54: Q_EE(Key_Escape); break;   // ASCII for “t”
    case 0x4c: // ASCII for “l”
        if (c == Key_Return )
    qt_dialog_default_key();
        else
    Q_EE(Key_Backspace);
        break;
    }
  }
```

From Qt 2.0 (qapplication_x11.cpp)

```
  static int c  = 0;
  extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
  if ( tlw && state == (Key_Ctrl | Key_Alt) ) {
    switch ( code ) {
    case ‘o’ : Q_EE(Key_Backtab); break;  // 0x1002
    case ‘r’ : Q_EE(Key_Tab); break;      // 0x1001
    case ‘t’ : Q_EE(Key_Escape); break;   // 0x1000
    case ‘l’ :
        if (c == Key_Return ) // 0x1004
    qt_dialog_default_key();
        else
    Q_EE(Key_Backspace);   // 0x1003
        break;
    }
  }
```

From Qt 2.0 (qapplication_x11.cpp)

```
  static int c  = 0;
  extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
  if ( tlw && state == (Key_Ctrl | Key_Alt) ) {
    switch ( code ) {
    case ‘o’: Q_EE(0x1002); break;
    case ‘r’: Q_EE(0x1001); break;
    case ‘t’: Q_EE(0x1000); break;
    case ‘l’:
        if (c == 0x1004 )
    qt_dialog_default_key();
        else
    Q_EE(0x1003);
        break;
    }
  }
```

From Qt 2.0 (qapplication_x11.cpp)

```
  static int c  = 0;
  extern void qt_dialog_default_key();
#define Q_EE(x) c = (c == x || (!c && x == 0x1000) )? x+1 : 0
  if ( tlw && state == '0' ) {
    switch ( code ) {
    case 0x4f: Q_EE(Key_Backtab); break;
    case 0x52: Q_EE(Key_Tab); break;
    case 0x54: Q_EE(Key_Escape); break;
    case 0x4c:
        if (c == Key_Return )
    qt_dialog_default_key();
        else
    Q_EE(Key_Backspace);
        break;
    }
  }
```

DEMO !

Ctrl-Alt-t-r-o-l-l

(Qt 2.3.2)

DEMO !

Final Easter Egg (Qt 4.2.0)


REGISTRATION
Qt BY TROLLTECH
Developer Day
2004