

昵称： NQian
园龄： 2年6个月
粉丝： 407
关注： 17
+加关注

积分与排名

积分 - 397116

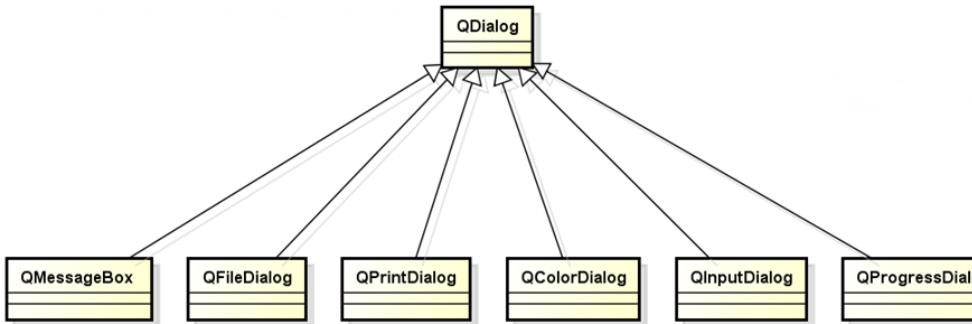
排名 - 684

随笔分类

- 1.C与算法(16)
- 2.STM32(8)
- 3.2440应用项目(7)
- 3.ARM与汇编 (11)
- 3.linux_shell(4)
- 3.linux底层驱动(54)
- 3.linux电源管理(1)
- 3.linux命令学习(23)
- 3.Makefile(6)
- 3.u-boot(22)
- 3.裸机驱动(4)
- 3.移植transplant(10)
- 4.1 C++(31)
- 4.2 C++数据结构
- 4.3 QT学习之路(51)
- 5.android入门(4)
- 5.JAVA(19)
- github(1)
- LeetCode(2)

9.QT-标准对话框

Qt提供的可复用的标准对话框,全部继承自QDialog类,如下图所示:



- **QMessageBox**: 信息对话框, 用于显示信息、询问问题等;
- **QFileDialog**: 文件对话框
- **QColorDialog**: 颜色对话框
- **QInputDialog**: 输入对话框(允许用户输入一次数据)
- **QFontDialog**: 字体对话框
- **QProgressDialog**: 进度对话框
- **QPrintDialog**: 打印对话框
- **QPrintPreviewDialog**: 打印预览对话框
- **QPageSetupDialog**: 打印设置对话框,为打印机提供纸张相关的选项

标准对话框使用方式

```
QDialogType dialog(this); //对话框对象的定义

dialog.setPropertyxxxx(value); //对话框属性设置

if(dialog.exec() == QDialogType::vaule)
{
    Type v = dialog.getDialogValue(); //获取对话框数据
    //... .. 处理数据
}
```

QMessageBox消息对话框
代码如下:

其它(3)
心情(1)

随笔档案
2019年12月(8)
2019年11月(1)
2019年8月(5)
2019年6月(3)
2019年5月(14)
2019年4月(11)
2019年3月(2)
2019年2月(5)
2019年1月(4)
2018年12月(7)
2018年11月(2)
2018年10月(3)
2018年9月(3)
2018年8月(4)
2018年7月(3)
2018年6月(10)
2018年5月(10)
2018年4月(23)
2018年3月(25)
2018年2月(11)
2018年1月(8)
2017年12月(12)
2017年11月(10)
2017年10月(13)
2017年9月(24)
2017年8月(30)
2017年7月(16)
2017年6月(10)

```
#include <QtGui/QApplication>
#include <QDebug>
#include <QMessageBox>
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QMessageBox msg;
    msg.setWindowTitle("Message");
    msg.setText("This is Message content!");
    msg.setIcon(QMessageBox::Information); //选择图标为:信息类型
    msg.setStandardButtons(QMessageBox::Ok|QMessageBox::Cancel);
    msg.setButtonText(QMessageBox::Ok,QString("确定"));
    msg.setButtonText(QMessageBox::Cancel,QString("取消"));
    if(msg.exec()==QMessageBox::Ok)
    {
        qDebug()<<"QMessageBox::Ok"; //当点击确定按钮,则会打印信息
    }
    return 0;
}
```

效果:



也可以使用一个静态函数来创建消息对话框:

```
StandardButton QMessageBox::information(QWidget * parent, const QString & title, const
QString & text, StandardButtons buttons = Ok, StandardButtondefaultButton = NoButton);
//显示信息对话框, 将显示一个反叹号图标

StandardButton QMessageBox::question(QWidget * parent, const QString & title, const
QString & text, StandardButtons buttons = Ok, StandardButtondefaultButton = NoButton);
//显示问题对话框, 将显示一个问号图标

StandardButton QMessageBox::warning(QWidget * parent, const QString & title, const
QString & text, StandardButtons buttons = Ok, StandardButtondefaultButton = NoButton);
//显示警告对话框, 将显示一个黄色叹号图标

StandardButton QMessageBox::critical(QWidget * parent, const QString & title, const
QString & text, StandardButtons buttons = Ok, StandardButton defaultButton= NoButton);
//显示严重错误对话框, 将显示一个红色的错误符号

void QMessageBox::about(QWidget * parent, const QString & title, const QString &
text);
//显示关于对话框,该对话框只有一个OK按钮
```

比如:

```
#include <QtGui/QApplication>
#include <QDebug>
#include <QMessageBox>
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QMessageBox msg;
    int ret = msg.question(NULL,"Question","This is Message
Question",QMessageBox::Ok|QMessageBox::Cancel);
    if(ret==QMessageBox::Ok)
    {
        qDebug()<<"QMessageBox::Ok"; //当点击Ok按钮,则会打印QDebug
    }
}
```

最新评论

1. Re:42.QT-QSqlQuery类
操作SQLite数据库(创建、查
询、删除、修改)详解

@ 走在花开左边db表示
数据库连接类,"scooters.d
at"是一个.dat文件,QSqlQ
uery query是定义一个执
行和操作SQL语句的方法类,
最好是先学C++,不然学不
精的...

--NQian

2. Re:42.QT-QSqlQuery类
操作SQLite数据库(创建、查
询、删除、修改)详解

大神你好,我是一个超级
想入门的小白,有几个超级
简单但对于我来说又是超级
障碍的问题, 1、在创建数
据库的时候用到了"db", 又
用到了"scooters.dat",这两
个名称都是代表什么? scoo
ters....

--走在花开左边

3. Re:STM32-对芯片启动读
保护,实现加密(详解)

1, 不要在程序中直接出
现id地址, 例如1FFFF7E8
1FFFF7EC 1FFFF7F0 2,
利用校验和或是crc对程序
区进行校验, 防止改程序

--银胖胖

4. Re:14.QT-QFile文件,QB
uffer缓冲区,QDir目录,QFileSy
stemWatcher文件系统监视

很全面, 很多细节都讲到
了.能说下平时是怎么学习
的么? 很多细节都要看文档
才知道

--duohappy

5. Re:44.Linux君正X1000-
添加st7789v显示

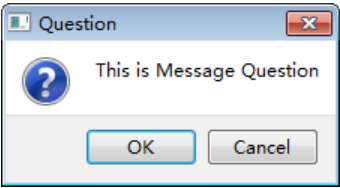
@ 阳光普照你也很帅哟,
这段时间忙完,开年加强更一
波~...

--NQian

阅读排行榜

```
return 0;  
}
```

效果:



QFileDialog文件对话框

QFileDialog的exec()返回值为**QFileDialog::Accepted**或者**QFileDialog::Rejected**

需要以下成员函数:



```
setAcceptMode (QFileDialog::AcceptOpen);  
                //设置文本对话框是用来打开文件(AcceptOpen),还是用来保存文件(AcceptSave)  
  
setDirectory (const QString & directory );           //设置文本对话框路径  
  
setFileMode ( FileMode mode );  
                //设置文本对话框打开的文件模式,是单个文件?还是多个文件? 还是包括目录?  
void selectFile(const QString & filename);  
                //预置文件对话框默认打开的文件名  
QStringList QFileDialog::selectedFiles();  
                //用来获取用户选择的多个 文件信息 位置(绝对路径),比  
如:"C:/Users/Administrator/Desktop/新建文本文档.txt"  
QString selectedFilter();  
                //获取用户选择的文件过滤器
```



文件类型过滤器

通过**setFilter()**成员函数实现,文件过滤器定义规则如下:

- 显示名(*.后缀名1 *.后缀名2 ...*.后缀名n)

如果,有多个规则,则可以通过;;来隔开,例如:

```
setFilter("Image (*.jpg *.png *.bmp);;Text (*.txt);;All (*.*)");
```

以打开文件为例:



```
QFileDialog dlg;  
  
dlg.setAcceptMode(QFileDialog::AcceptOpen);  
dlg.setFileMode(QFileDialog::ExistingFile); //模式为:已存在的文件,如果为保存文  
件,则不需要指定模式  
dlg.setFilter("Image (*.jpg *.png *.bmp);;Text (*.txt);;All (*.*)");  
dlg.setDirectory("D:");  
dlg.selectFile("新建文件.txt");  
  
if(dlg.exec() == QFileDialog::Accepted)  
{  
    QString Str=dlg.selectedFiles()[0];           //获取用户选择的文件  
    QString filter = dlg.selectedFilter();         //获取用户选择的过滤器  
  
    qDebug() << Str;  
    qDebug() << filter;  
}
```

0



1. STM32—无需中断来实现使用DMA接收串口数据(26159)
2. 1.移植uboot-分析uboot启动流程(详解)(17773)
3. 28.QT-QPainter介绍(16243)
4. Linux-chown命令(1)(14782)
5. 6.QT-简易计算器实现(详解)(12685)

评论排行榜
1. 2.制作第一个驱动程序(14)
2. STM32—无需中断来实现使用DMA接收串口数据(12)
3. 重新改变,以此共勉(12)
4. 46.Linux-创建rc红外遥控平台设备,实现重复功能(2)(9)
5. 12.Linux之输入子系统分析(详解)(8)

推荐排行榜
1. 32.QT-制作最强电压电阻表盘,可以自定义阴影效果,渐变颜色,图标,文字标签等-附带demo程序(13)
2. 33.Linux-实现U盘自动挂载(详解)(8)
3. 28.Linux-IIC驱动(详解)(8)
4. 12.Linux之输入子系统分析(详解)(8)
5. github网站介绍、并使用git命令管理github(详细描述)(7)

也可以使用一个静态函数直接打开文件,或者保存文件:

```
QString QFileDialog::getOpenFileName (
QWidget * parent = 0,      //父组件
const QString & caption = QString(),    //标题
const QString & dir = QString(),    //设置路径, .表示当前路径, /表示更目录
const QString & filter = QString(), //过滤器
QString * selectedFilter = 0,    //用户选择文件后的过滤器信息
Options options = 0 );    //参数,它的取值为enum QFileDialog::Option ,比如:
QFileDialog::ShowDirsOnly (只显示目录)

QString getSaveFileName(
QWidget *parent = 0,
const QString &caption = QString(),
const QString &dir = QString(),
const QString &filter = QString(),
QString *selectedFilter = 0,
Options options = 0);
```

QInputDialog输入对话框

QInputDialog的exec()返回值为**QFileDialog::Accepted**或者**QFileDialog::Rejected**
常用函数:

```
setLabelText ( const QString & text );    //设置对话框标签

setOkButtonText ( const QString & text );    //设置OK按钮文本

setCancelButtonText ( const QString & text )    //设置Cancel按钮文本

setInputMode ( InputMode mode);    //设置输入模式, 是文本, 还是整数, 还是浮点数

setIntRange( int min, int max );    //如果是整数模式, 则该函数可以设置允许整数范围

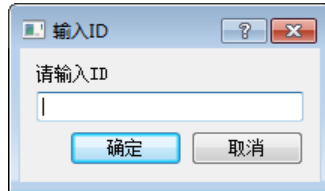
setDoubleRange ( double min, double max );    //如果是浮点数模式, 则该函数可以设置允许浮点数范围

QString textValue ();    //返回文本模式的输入对话框返回值
int intValue ();    //返回整数模式的输入对话框返回值
double doubleValue ();    //返回浮点数模式的输入对话框返回值
```

示例:

```
QInputDialog log;
log.setInputMode(QInputDialog::TextInput);
log.setCancelButtonText("取消");
log.setOkButtonText("确定");
log.setWindowTitle("输入ID");
log.setLabelText("请输入ID");
if(log.exec() == QInputDialog::Accepted)
{
qDebug() << log.textValue();
}
}
```

效果:



也可以使用一个静态函数实现输入对话框:

设置文本输入对话框:



```
QString QInputDialog::getText( QWidget * parent, //父组件
const QString & title, //窗口标题
const QString & label, //标签提示
QLineEdit::EchoMode mode = QLineEdit::Normal, //输入框模式
const QString & text = QString(), //预定义的文本
bool * ok = 0, //如果用户点击ok,则ok =true
Qt::WindowFlags flags = 0 );
```



设置列表输入对话框:



```
QString QInputDialog::getItem ( QWidget * parent, //父组件
const QString & title, //窗口标题
const QString & label, //标签提示
const QStringList & items, //字符串链表,用来显示列表
int current = 0, //预定义,表示列表中第哪组
bool editable = true, //true表示列表可编译
bool * ok = 0, //如果用户点击ok,则ok =true
Qt::WindowFlags flags = 0 )
```



除此之外还有:**getDouble(), getInt();**

以**getItem()**为例:



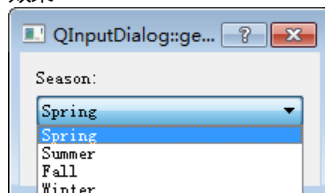
```
QStringList items;
items << "Spring" << "Summer" << "Fall" << "Winter";
bool ok;

QString item = QInputDialog::getItem(0, "QInputDialog::getItem()", "Season:", items, 0,
false, &ok);

if (ok && !item.isEmpty())
qDebug() << item;
```



效果:



QColorDialog颜色对话框

QColorDialog的exec()返回值为**QFileDialog::Accepted**或者**QFileDialog::Rejected**

需要用到以下函数:



```

void setCurrentColor(const QColor & color )    //设置打开后的颜色对话框预定颜色
                                              //它的取值可以为enum Qt::GlobalColor, 比如白色

Qt::white

RGB(100,100,100)                                //也可以为QColor类, 比如QColor(100,100,100),表示

QColor selectedColor();                        //返回用户选择的哪个颜色,如果用户点击了退出,则返回的
QColor::isValid()为false

bool QColor::isValid();                        //颜色有效返回true,无效则返回false

```



示例:



```

QColorDialog dlg;

dlg.setCurrentColor(Qt::white); //预定义白色
dlg.exec();
QColor color=dlg.selectedColor();
if(color.isValid()) //如果用户选择了颜色
{
    qDebug()<<color.red();
    qDebug()<<color.green();
    qDebug()<<color.blue();
}

```



也可以使用一个静态函数直接打开颜色对话框:

```

static QColor getColor(
    const QColor &initial = Qt::white, //预定义颜色
    QWidget *parent = 0);              //父组件

```

示例:



```

QColor color= QColorDialog::getColor(Qt::white);
if(color.isValid()) //如果用户选择了颜色
{
    qDebug()<<color.red();
    qDebug()<<color.green();
    qDebug()<<color.blue();
}

```



QFontDialog字体对话框

QColorDialog的exec()返回值为**QFileDialog::Accepted**或者**QFileDialog::Rejected**

需要用到以下函数:

```

void setCurrentFont ( const QFont & font )    //设置打开后的字体对话框预定字体
                                              //以"Courier New"字体为例,则参数填为font("Courier
New",10,QFont::Bold,true);
                                              //表示字体大小为10,加粗,斜体

QFont QFontDialog::selectedFont ();          //返回用户选择的字体

```

示例:



```

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget w;

```

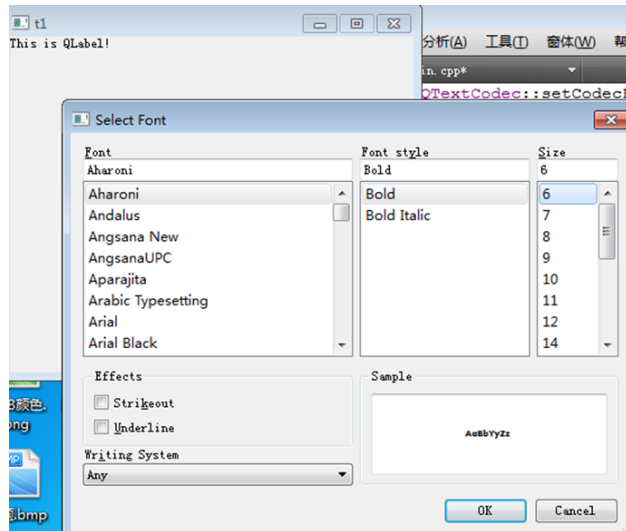
```
w.setFixedSize(360,300);
QLabel label("This is QLabel!",&w);
w.show();

QFontDialog dlg(&w);
dlg.setFont(label.font());

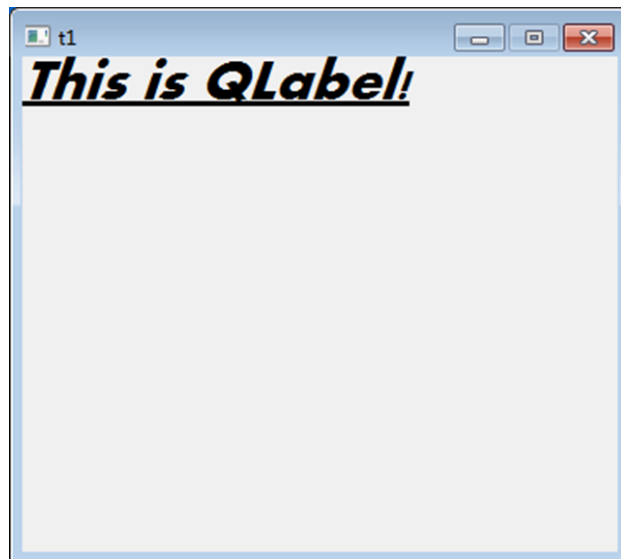
if(dlg.exec()==QFontDialog::Accepted)
{
    label.setFont(dlg.selectedFont());
    label.adjustSize();
}

return a.exec();
}
```

选择前-效果:



选择后-效果:



也可以使用一个静态函数直接打开字体对话框:

```
QFont QFontDialog::getFont( bool * ok, const QFont & initial, QWidget * parent,
const QString & title,FontDialogOptions options );
```

0

QProgressDialog进度条对话框

常用函数:

```
setMinimum(int minimum);           //设置进度条最小值

setMaximum(int maximum);           //设置进度条最大值

setValue(int value);               //设置当前进度值

setAutoClose(bool close);          //设置自动关闭,如果进度值=最大值时,则会自动关闭对话框

setLabelText(const QString &text); //设置标签提示

setCancelButtonText(const QString &text); //设置Cancel按钮提示

setCancelButton(QPushButton *button); //从新自定义按钮框架,如果添NULL,则表示隐藏按钮

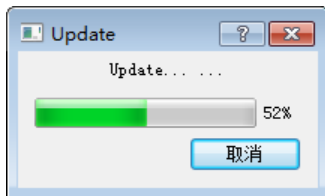
setBar( QProgressBar * bar );       //从新自定义进度条框架

setLabel ( QLabel * label );        //从新自定义标签
```

示例:

```
QProgressDialog dlg;
dlg.setMinimum(0);
dlg.setMaximum(1000);
dlg.setValue(520);
dlg.setWindowTitle("Update");
dlg.setLabelText("Update... ..");
dlg.setCancelButtonText("取消");
dlg.exec();
```

效果:



也可以直接通过构造函数设置:

```
QProgressDialog ( const QString & labelText,
const QString & cancelButtonText,
int minimum,
int maximum,
QWidget * parent = 0,
Qt::WindowFlags f = 0 );
```

QPrintDialog打印机对话框

QPrinter类介绍:

- **QPrinter**是打印设备及其参数的封装
- **QPrinter**封装了系统中打印设备的驱动接口

试验:

```
QPrintDialog printDialog;
if (printDialog.exec() == QDialog::Accepted)
{
    QPrinter *printer=printDialog.printer(); //获取Qprinter
    QTextDocument td;
```

0


```
td.setPlainText("TEST TEST TEST"); //设置文本
//printer->setOutputFileName("D:\\1.pdf"); //设置文件路径
td.print(printer); //打印td
}
```

效果:



版权声明：本文为博主原创文章，转载请标注文章来源,觉得文章不错,不妨点个赞呗~
标注出处于:珍惜他人的码字成果,并且文章有更新或者修改出错的地方,也能方便他人查找到

分类: 4.3 QT学习之路

好文要顶 关注我 收藏该文





NQian
关注 - 17
粉丝 - 407
[+加关注](#)

« 上一篇: 8.QT-对话框(模态与非模态)
» 下一篇: 10.QT-定时器

posted @ 2018-04-25 11:01 NQian 阅读(1009) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。

- 【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】腾讯云热门云产品限时秒杀，爆款1核2G云服务器99元/年！
- 【推荐】阿里云双11返场来袭，热门产品低至一折等你来抢！
- 【活动】京东云服务器_云主机低于1折，低价高性能产品备战双11
- 【活动】ECUG For Future 技术者的年度盛会（杭州，1月4-5日）

相关博文:

- QT+使用标准对话框+关于对话框+问题对话框+文件对话框
- Qt标准对话框
- 8.QT-对话框(模态与非模态)
- Qt中的标准对话框之QMessageBox
- 第十九课、Qt中的标准对话框（上）-----狄泰软件学院
- » 更多推荐...
- 12知识点+20干货案例+110面试题，助你拿offer！ | Python工程师面试宝典