

书籍

Jasmin Blanchette, Mark Summerfield ¹ | 万泽²

版本： 1.0

¹翻译：于丁

²编者：万泽邮箱： a358003542@gmail.com。

前言

前言 C++ GUI Programming with Qt 4

By Jasmin Blanchette, Mark Summerfield

.....

Publisher: Prentice Hall

Pub Date: June 21, 2006

作者于丁

学习, 使用 Qt 已经将近两年了, 最开始的时候用 Qt3, 后来升级到 Qt4.1, 自认为对 Qt 的 GUI 编程部分还是很熟悉的。遗憾的是 Qt 的参考书少之又少, 一般就是看文档和 C++ GUI Programming with Qt 3, 这本书是免费提供的, 让我对 Qt, Trolltech 公司有了更多的了解。我这个人就是这样, 学习什么都喜欢了解这个东东发展的历史, 虽然对学习并没有太多帮助。

升级到 Qt4 以后, API 有了很多变化, 苦盼一本系统的参考书, 很久, C++ GUI Programming with Qt 4 才得以面世, 终于盼到后却遗憾的发现这本没有提供免费的电子版, 今天, 一个偶然的机会有幸得到 Yimin 网友的帮助, 得到一份电子版, 真是很谢谢他, 他的 blog<http://liyimin.net/blog>。

C++ GUI Programming with Qt 4 就是这样一本参考书, 从易

到难，从最经典的 `hello Qt` 开始，到构建复杂的程序。我从现在开始阅读学习，同时也把部分心得写出来，发到这里，也希望对 Qt 的学习者们有所帮助。这对我来说有点难度，懒人那，不过一定尽最大努力坚持写完我熟悉的部分。加油！

目 录

前言	i
目录	iii
1 qt 发展小史	1
1.1 从 Hello Qt 开始	2
1.2 连接信号和响应函数	4
1.3 控件的几何排列	5

qt 发展小史

Qt 的创建者 Haarard Nord (Trolltech 公司的 CEO) 和 Eirik Chambe-Eng (Trolltech 公司的总裁) 是一家瑞典公司的同事。那时 (1990) 他们在做一个项目, 这个项目需要在 Unix, Macintosh, Windows 上运行同一个 GUI, 象我们现在的开发人员一样, 工作的很累, 当时可是没有如今这么多的开发工具。一天他们工作之余去公园散步, 晒太阳, 喝咖啡。Haarard 说: “We need an object-oriented display system。”这成为了后来 Qt 最重要的思想: 提供面向对象的跨平台的 GUI 框架。看到这里小女我不仅感慨: 什么时候我们的程序员们可以在工作的时候出来走走, 只有在轻松愉快的环境中才会生产出出色的成果。在沉闷的办公室里, 只是机械的堆砌代码而已。

所做就做, Haarard 开始写代码, Eirik 负责设计, Qt 在襁褓中逐渐成长, 在开始蹒跚学步的时候 (1993 年), 他们开始让 Qt 闯荡江湖, 两个人开始了创业的艰辛历程。

对这两个年轻人, 1994 年是非常艰难的一年, 他们没有客户, 没有钱, 只有还没有完全实现的产品。关键时刻, 他们的妻子帮他们渡过了难关。

字母 Q 作为所有类的前缀, 是因为 Haarard 手写这个字母看起来特别的漂亮, 字母 t 代表 “toolkit”, 在 Xt, X toolkit 等中得

到灵感。

1995 年开始出现转机，他们得到了一个合同。这一年，他们雇佣了 Arnt Gulbrandsen，他在 Trolltech 工作了六年，他为 Qt 实现了优秀的文档系统。

1995 年 5 月，Qt 0.9 发布，有商业和开源两个版本。96 年 9 月，Qt1.0 发布。

1997 年，Matthias Ettrich 开始用 Qt 开发 KDE，使 Qt 成为 Linux 上 GUI 开发的事实上的标准。

1999 年，Qt 2 发布。

2000 年，Qt4 发布。支持 linux 嵌入式开发。

2001 年，Qt 3 发布。

2005 年，Qt 4 发布。

十年来，Qt 就是这样从不知名的一个产品，发展到现在拥有全世界范围内成千上万的客户。

从 Hello Qt 开始

差不多所有的程序教材都从 Hello 开始，下面就是这个程序的 qt 版本。

```
1 #include <QApplication>
2 #include <QLabel>
3 int main(int argc, char *argv[])
```

```
4 {  
5     QApplication app(argc, argv);  
6     QLabel *label = new QLabel("Hello Qt!");  
7     label->show();  
8     return app.exec();  
9 }
```

按行解析以上 9 行代码第一，二行：是代码中需要使用的类的头文件。在 Qt4 中，可以写成 `<QApplication>` 的格式，当然也可写成“`QApplication.h`”。

第三行：是 `main` 函数的标准写法

第五行：创建一个 `QApplication` 对象，管理应用程序的资源。

第六行：`QLabel` 对象，`QLabel` 是一个 Qt 提供的小控件，显示一行文本。

第七行：显示 `QLabel`。

第八行：`QApplication.exec()`，让程序进入消息循环。等待可能的菜单，工具条，鼠标等的输入，进行响应。

将以上代码放到名为 `hello.cpp` 中，保存，编译过程如下：

`qmake -project`，`qmake` 命令创建 `hello.pro`，是平台无关的工程文件。在 `hello.pro` 所在目录下，运行 `make (unix)` 或者 `nmake (windows)`。

用 `geany` 打开目标 `cpp` 文件，新建一个生成命令 `qmake-project`，内容如下：

```
qmake -project -o "%e.pro"
```

然后新建一个命令 `qmake`，内容如下：

```
qmake %e.pro
```

然后依次点击顺序如下：1.qmake-project; 2.qmake; 3.make; 4.执行。

第 6 行代码还可以如下替换：`QLabel *label = new QLabel("<h2><i>Hello</i>
Qt!</h2>");`

这里面包含了 html 文本，显示的字体，颜色会改变。

实际程序中，下面两行是比不可少的。`QApplication app(argc, argv); return app.exec();`

连接信号和响应函数

这个例子用来说明怎么响应信号，和 `hello` 程序的源代码相似，原来的 `Label` 用一个按钮代替，点击时退出程序。

```
1 #include <QApplication>
2 #include <QPushButton>
3
4 int main(int argc, char *argv[])
5 {
6     QApplication app(argc, argv);
7     QPushButton *button = new QPushButton("Quit");
8     QObject::connect(button, SIGNAL(clicked()),
9                      &app, SLOT(quit()));
10    button->show();
```



```
11     return app.exec();  
12 }
```

当有所动作或者状态改变，qt 的控件会发出消息（signal），例如，当点击按钮时，按钮会发送 `clicked()` 消息，这个消息可以连接到一个函数上（这个函数在这里成为 slot）。这样，当一个消息发送时，slot 函数可以自动执行。在这个例子中，我们连接了按钮的 `clicked` 信号和 `QApplication` 的 `quit` 函数，语法如第七，八行所示。

编译以上程序，将以上代码放在 `quit.cpp` 文件中，保存。

依次运行

`qmake -project qmake quit.pro make(unix or linux) or nmake(windows)`
然后运行程序，点击 `Quit` 按钮，程序将会中止。

控件的几何排列