

Qt 5 开发实战

【韩】金大砾 著 张红艳 译



Qt 5 Programming Guide



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Qt 5 开发实战

【韩】金大勋 著 张红艳 译



人民邮电出版社

北京

图书在版编目 (C I P) 数据

Qt 5开发实战 / (韩) 金大勋著 ; 张红艳译. — 北京 :
人民邮电出版社, 2015.9
ISBN 978-7-115-40325-4

I. ①Q… II. ①金… ②张… III. ①软件工具—程序
设计 IV. ①TP311.56

中国版本图书馆CIP数据核字(2015)第210004号

内 容 提 要

Qt 支持 C++、Python、Ruby、Perl 等多种编程语言, 常规用户应用程序、多媒体、后端系统管理、汽车、国防、家电、移动设备等多个领域都在广泛应用 Qt。本书在全面阐述 Qt 基本功能的基础上, 对新增的功能和服务进行了重点介绍。同时运用大量示例, 集中讲解了应用程序的开发方法、技巧和必需的 Qt API。读者通过实际动手操作能够迅速获得业务中必备的相关知识与经验, 本书非常适合利用高级 GUI 和 OpenGL ES 开发 3D 图像应用程序的人阅读。

-
- ◆ 著 [韩] 金大勋
 - 译 张红艳
 - 责任编辑 傅志红
 - 执行编辑 陈 曦
 - 责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 33
 - 字数: 780千字 2015年9月第1版
 - 印数: 1-3 000册 2015年9月北京第1次印刷
 - 著作权合同登记号 图字: 01-2014-6029号
-

定价: 99.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

前 言

Qt是一个跨平台的应用程序开发框架，支持嵌入式Linux、Mac OS X、Windows、Linux、Android、QNX、MeeGo、WebOS、Symbian等多个平台。例如，在嵌入式Linux平台中开发的应用程序源代码——除平台附属部分外——可完全适用于Windows、Android、QNX等多种平台。

Qt支持C++、Python、Ruby、Perl等多种编程语言，全球有近40万Qt开发人员在多种计算环境中开发各类软件。因此，常规用户应用程序、多媒体、后端系统管理、汽车、国防、家电、移动设备等多个领域都在广泛应用Qt。

近年来，开发环境中充斥着各种平台，在这种条件下开发可支持多平台的应用程序并非易事。例如，如果要开发一种可同时适用于Android和iPhone这两种不同移动设备平台的应用程序，就需要投入大量时间。

但若使用Qt则可以开发出一种不改变源代码亦可适用于多种平台的应用程序，这样可大大缩短应用程序的开发时间。因此，我认为Qt是一个可适应突变的市场环境，并具有强大竞争力的应用程序开发框架。

父母的爱是培养优秀子女的基础，各位读者就是我的衣食父母。因此，我要向一直喜爱我的读者致以最诚挚的感谢。希望本书可以给各位读者带来些许帮助。同时，向促成本书出版的出版社负责人员致以衷心的感谢。

金大砾 敬上

目 录

第 1 章 概述	1
第 2 章 有用的工具和 qmake	6
2.1 Qt Creator IDE	6
2.2 Qt Assistant	14
2.3 Qt Designer	15
2.4 qmake	16
第 3 章 GUI 设计与制作	19
3.1 基础控件	19
3.1.1 QCheckBox 和 QButtonGroup	22
3.1.2 QComboBox	25
3.1.3 QCommandLinkButton	26
3.1.4 QDate 类和 QDateEdit 控件	27
3.1.5 QTime 类和 QTimeEdit 控件	30
3.1.6 QDateTime 类和 QDateTime Edit 控件	33
3.1.7 QDial	35
3.1.8 QSpinBox 和 QDoubleSpinBox	36
3.1.9 QPushButton 和 QFocusFrame	39
3.1.10 QFontComboBox	41
3.1.11 QLabel 和 QLCDNumber	43
3.1.12 QLineEdit	45
3.1.13 QMenu 和 QMenuBar	47
3.1.14 QProgressBar	50
3.1.15 QRadioButton	51
3.1.16 QScrollArea	52
3.1.17 QScrollBar	53
3.1.18 QSizeGrip	55
3.1.19 QSlider	57
3.1.20 QTabWidget	58
3.1.21 QToolBar 和 QAction	60
3.1.22 QWidget	62
3.1.23 QTabBar	67
3.1.24 QToolBox	68
3.1.25 QToolButton	69
3.2 布局管理器	70
3.3 高级控件	73
3.3.1 QColumnView	74
3.3.2 QDataWidgetMapper	75
3.3.3 QDesktopWidget	77
3.3.4 QCalendarWidget	79
3.4 使用 Designer 构建 GUI	81
3.4.1 控件和信号槽使用示例	84
3.4.2 Calculator Form 示例	86
3.5 对话框	88
3.5.1 普通对话框	88
3.5.2 用户自定义对话框	97
3.6 主窗口	98
3.7 模型/视图结构和数据绑定控件	101
3.7.1 模型/视图结构	102
3.7.2 使用模型类和视图控件	103
3.8 用户自定义控件和插件	107
3.8.1 实现用户自定义控件	109
3.8.2 插件制作及布局	118
第 4 章 信号与槽	128
第 5 章 数据类型和有用的数据操作类	131
5.1 比较和判断变量的函数	133
5.2 数据类	136
5.3 容器类	154
5.4 字符串类	159

第 6 章 文件输入输出	163	11.2 文档对象模型 (DOM)	266
第 7 章 图形	171	11.3 SAX 模式	269
7.1 使用 QPainter 类的图形	171	第 12 章 数据传输和网络编程	273
7.1.1 QPainter 的基本绘图	172	12.1 网络	273
7.1.2 渐变	179	12.1.1 使用 QTcpSocket 和	
7.1.3 转换	182	QTcpServer 进行 TCP 编程	277
7.1.4 组合	184	12.1.2 使用 QUdpSocket 的 UDP	285
7.2 图形视图框架	185	12.1.3 主机名和 DNS 查询	291
7.2.1 图形视图架构	187	12.1.4 HTTP 协议	297
7.2.2 图形视图的坐标	190	12.1.5 承载管理	306
7.2.3 图形视图的主要特点	191	12.1.6 基于服务器端/客户端的聊	
7.2.4 事件处理	193	天示例	307
7.2.5 用户自定义 QGraphicsItem	194	12.2 QtWebKit	314
7.2.6 QGraphicsEffect	197	12.2.1 QtWebKit 模块提供的类	316
7.3 使用 Qt OpenGL 模块的 3D 图形	202	12.2.2 DOM 查询	323
7.3.1 用于渲染 3D 图形的 QGL		12.2.3 QtWebKit 模块环境下的	
Widget 类	203	jQuery 应用	325
7.3.2 Hello OpenGL 示例	206	12.2.4 QtWebKit 和 QObject 模型	
7.3.3 使用硬盘加速器的 2D 图形		间的网桥	330
渲染	212	12.2.5 混合客户端应用程序示例	336
7.3.4 QPainter 类的复绘	218	12.3 进程间通信	339
7.4 可缩放向量图形	224	12.3.1 TCP/IP	340
第 8 章 动画框架和状态机	227	12.3.2 共享内存	341
8.1 动画框架和类	228	12.3.3 D-Bus 协议	346
8.2 动画和状态机	235	12.3.4 QProcess	353
第 9 章 线程	238	第 13 章 多媒体	355
9.1 QThread 类	242	13.1 音频组件	356
9.2 多线程编程	244	13.2 视频组件	359
9.3 线程等待条件	247	13.3 camera 组件	362
9.4 使用信号量进行同步	249	13.4 收音机组件	364
9.5 使用 QtConcurrent 的多线程	252	第 14 章 数据库	365
第 10 章 容器类和迭代器类	254	14.1 数据库和 SQL 简介	367
10.1 Java 风格的迭代器模式	255	14.2 关联和支持数据库的驱动	369
10.2 STL 风格的迭代器模式	257	14.3 使用 SQL 语句的数据库查询	377
10.3 使用 foreach 关键字访问迭代器	259	14.4 Model 类	383
第 11 章 XML	261	14.5 列表视图应用	386
11.1 使用数据流的 XML 数据处理	263	14.6 SQL 记录文件字段和 GUI 控件	
		映射	388

14.7	Cached 列表示例	392	16.5	项目定位	444
14.8	Drill Down 示例	395	16.6	State 与 Transition	446
第 15 章	Qt 测试框架	405	16.7	动画	450
15.1	单元测试	409	16.8	数据的显示	456
15.2	数据驱动测试	410	16.9	模块化编程	461
15.3	模拟 GUI 事件	412	16.10	图形效果	466
15.4	GUI 事件重复测试	414	16.11	有用的 QML 类型	489
15.5	基准测试	415	16.12	Canvas	492
第 16 章	Qt Quick 与 QML	418	16.13	集成 QML 和 C++	495
16.1	Qt Quick 项目创建和有用的工具	419	16.14	多媒体	500
16.2	基本类型和扩展数据类型	421	第 17 章	Qt 应用程序的发布	504
16.3	对象	427	17.1	多种平台下 Qt 应用程序的发布	504
16.3.1	类型	428	17.1.1	Unix 和 Linux	504
16.3.2	定义项目属性	433	17.1.2	Windows	505
16.3.3	Repeater 和 Loader	436	17.1.3	Mac OS X	506
16.3.4	转换	437	17.1.4	Android 平台 Qt 应用程序的 发布	507
16.4	用户输入和事件处理	438			

第1章 概述

本章将介绍Qt应用程序开发框架。

Qt向开发人员提供了可以轻松设计现代GUI和开发应用程序所必需的多种API。Qt可用于开发C++应用程序,也可使用Qt Quick进行开发。Qt Quick使用QML语言。QML是解释型语言,开发人员即使不熟悉C++知识,也可以用QML设计GUI。

QML是一种描述性脚本语言(过程化语言),具有扩展JavaScript类型的语法结构。QML具有的Qt Declarative模块可以解释JavaScript之类编程语言。使用C++编写源代码后,需要编译程序生成可执行的二进制文件。与之类似,Qt Declarative模块可转换为可执行QML代码的应用程序,而且QML中可使用C++ API。

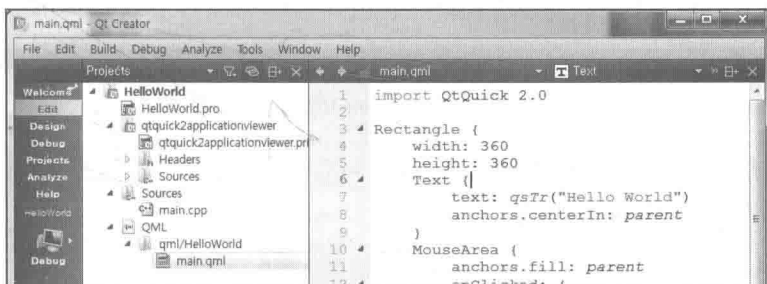


图1-1 QML项目

使用Qt Quick和QML不能创建完美的应用程序。例如,要创建一个可以查询网络流量的应用程序时,需要使用本地库以控制系统层级的网络流量。因此,提供给用户的GUI使用Qt Quick,以图形的形式显示网络流量,而测定网络流量的API最好是传统的Qt API或系统层级的API。

用Qt Quick实现GUI时,实际运行的模块是传统的Qt API。这种方式与使用HTML5创建应用程序的操作类似。换言之,用HTML5创建GUI时,实际使用本地API模块实现应用程序。

Qt Quick的优势在于,设计逻辑和程序逻辑是可分离的。也就是说,不需要C++知识,只用QML就可以设计GUI。QML类似于HTML5,既直观又容易理解,而且支持传统C++中提供的所有功能。例如,Qt WebKit、动画、多媒体等都可使用QML。

Qt支持多种平台,如MS Windows、Linux/Unix、Mac OS及Android、iOS、Windows 8(WinRT)、BlackBerry 10、QNX、VxWorks、嵌入式Linux等移动设备和嵌入式平台。不必重新编写源代码即可将Qt应用于多种平台,平台附属功能除外。

Qt可以跨平台使用网络、图形图像、数据库、OpenGL/ES、WebKit、HTML5、多媒体等多种API创建应用程序,并将之应用于多种平台,且可在各平台使用统一的IDE工具。Qt Creator是Qt的官方IDE,它提供了集Qt Designer、Help、Assistant功能于一体的综合环境。因此,开发程序时,如果想从Qt API中获得更详细的信息,则将光标定位到相应类或函数上并按F1键即可。

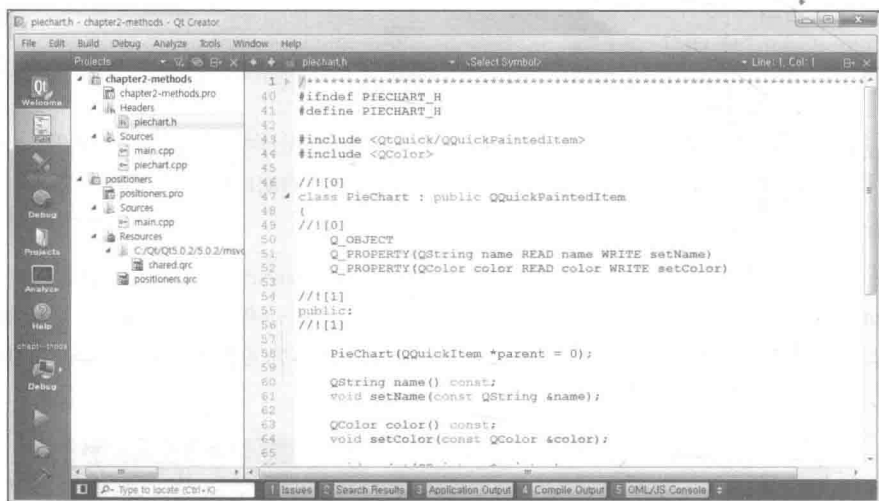


图1-2 Qt Creator IDE工具

Qt API可应用于多个领域,Qt拥有丰富的API和易操作且结构直观的库,不仅可轻松创建源代码,而且能够大大缩减应用程序的开发时间。

1. Qt中的GUI控件

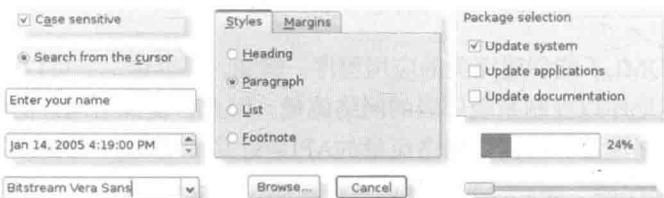


图1-3 Qt中的GUI控件

用Qt进行UI设计时,用户可利用默认控件、组件或用户自定义控件和QML直接设计控件。

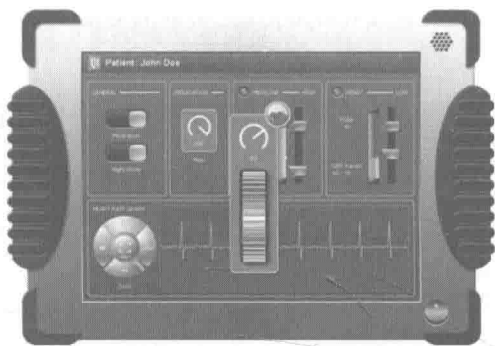


图1-6 自定义控件

4. Qt 5新增功能

Qt 5是最新版本。Qt 5不仅可以使⽤QML和JavaScript,还可以开发HTML5等网络应用程序。Qt 5增加了利⽤OpenGL创建的多种动画效果。

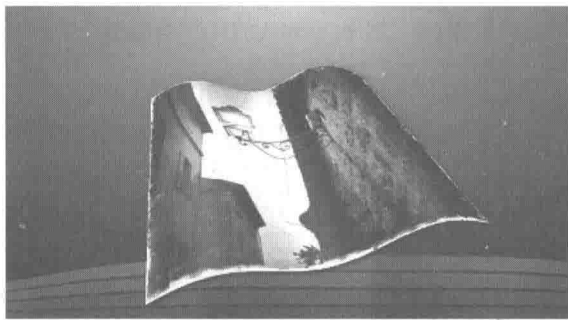


图1-7 适用于图像的着色效果 (Shader Effect)

Qt 4开始支持Qt Quick。Qt 5使⽤Qt Quick 2.0版本,新增了类似HTML5的Canvas功能和多种2D图形效果功能。

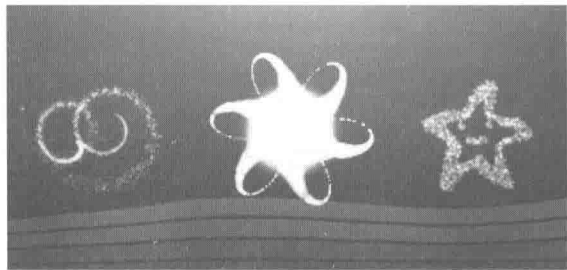


图1-8 2D图形效果

Qt Quick可使用图形效果（Graphical Effect）模块，制作软阴影（Soft drop shadow）、模糊（Blur）、发光（Glow）、着色（Colorize）等多种效果。

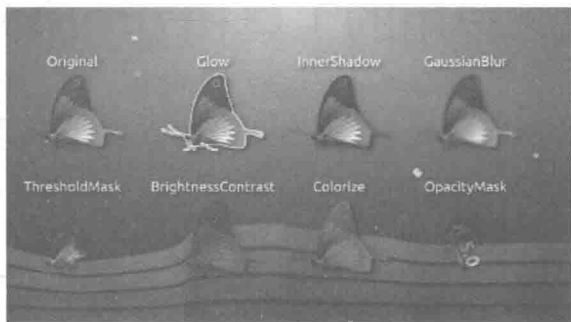


图1-9 图形效果

Qt 5使用Qt Quick 2.0版本，支持基于OpenGL/ES 2.0的Qt Quick Graphic Scene图像，而且在动画中增加了像素图序列（SpriteSequence）风格。

Qt 5中，不仅仅是C++，还有Qt Quick等QML中均可使用多媒体模块。例如，Qt Quick可通过组件输出多媒体影像，通过API访问照相机（Camera）和收音机（Radio）功能。

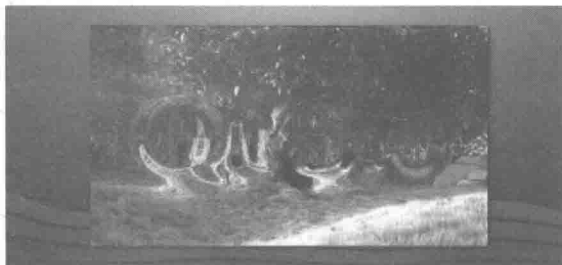


图1-10 使用Qt Quick输出多媒体影像

第2章 有用的工具和 qmake

本章将介绍Qt提供的一些有用的应用程序开发工具。

Qt主要包含开发所需的IDE（Integrated Development Environment，集成开发环境）工具Qt Creator、帮助文件Qt Assistant和GUI设计工具Qt Designer。

Qt还提供了qmake工具，可以将用Qt创建的应用程序源代码编译为二进制可执行文件。

表2-1 Qt提供的有用工具

工 具	说 明
Qt Creator	IDE工具，专为Qt开发人员量身定做
Qt Assistant	查看Qt API的帮助文件工具
Qt Designer	无需生成源代码也可轻松设计GUI或拖放控件
qmake	可轻松编译Qt生成的示例代码

本章将主要介绍以上4种Qt工具。

2.1 Qt Creator IDE

Qt Creator是使用Qt框架支持应用程序开发的IDE。Qt支持Unix/Linux、Windows、Mac OS X、Android、BlackBerry、QNX等多种平台。Qt Creator为不同平台提供了统一的开发环境。使用Qt Creator IDE工具创建的应用程序可部署到开发目标平台。

例如，应用程序可在嵌入式设备之类的手机、平板电脑、嵌入式Linux等远程开发目标平台上直接运行，还可使Qt应用程序在实际设备中运行，以方便调试。

Qt Creator集成了Qt Designer、Help、Qt Assistant。因此，开发程序时，如果想从Qt API中获

得更详细的信息，只要将光标定位到相应类或函数并按F1键即可。

开发程序时，会有多人合作开发的情况。此时通常使用Git、CVS等版本控制工具。Qt Creator集成了此版本控制系统，可管理Qt项目。Qt Creator支持GNU Symbolic Debugger (GDB)、Microsoft Console Debugger (CDB) 和QML/JavaScript Debugger，以调试Qt应用程序，也支持源代码中检测分析内存泄漏等问题的工具。下面详细介绍Qt Creator的功能。

2

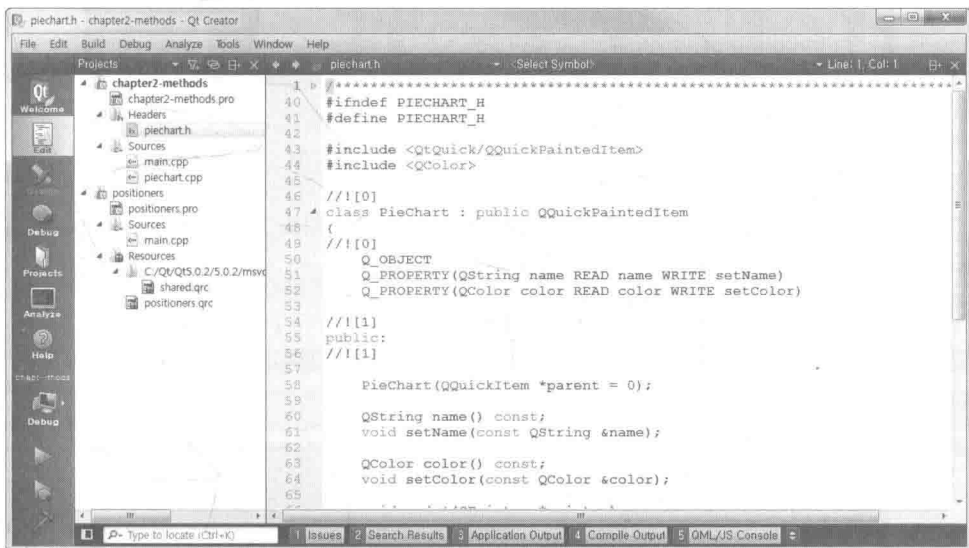


图2-1 Qt Creator IDE运行画面

1. 创建Qt项目

在Qt Creator中按照项目创建过程，使用项目向导分阶段创建新项目。依据项目向导生成新项目时，使用Qt构建工具qmake。后面会详细介绍qmake。

可以用Qt项目向导创建项目，也可以只使用qmake在记事本之类的编辑器上创建。后面会详细介绍这种方法，本章主要介绍如何使用项目向导创建新项目。

Qt可跨平台使用统一的开发环境（IDE）Qt Creator界面，即使平台不同，也可用统一的方法创建项目。

如图2-2所示，运行项目向导创建新项目，只需选择Qt Creator的File（文件）菜单，或者按Ctrl+N快捷键即可运行。

向导的第一个画面中，可根据想要创建的Qt应用程序的种类选择项目，如基于控件的GUI应用程序、基于Mobile的程序、基于系统的程序等。此处准备创建基于GUI应用程序的项目，所以在Projects（项目）栏选择Applications（应用程序），在中间的菜单栏选择Qt Gui Application，点击下拉菜单中的Choose（选择）键，然后选择出现的向导画面。

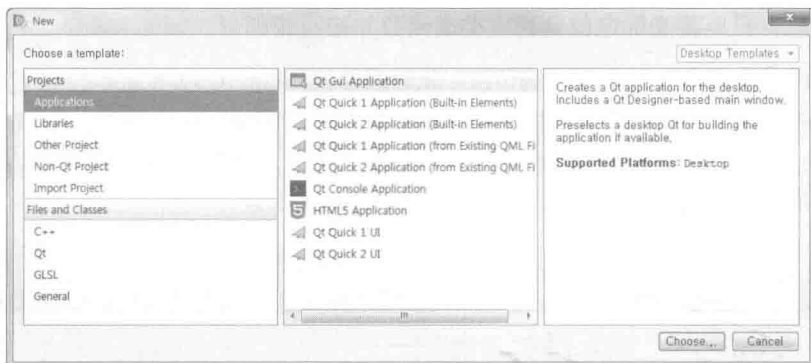


图2-2 Qt Creator的项目向导运行画面

第二个向导画面是类向导。此阶段中，要指定C++源文件和头文件名，以及生成项目类的基类（Base class）。基类是QMainWindow、QDialog、QWidget之一。此处选择了QWidget，类名定义为MyWidget。

勾选图2-3中的Generate form项是指，可以使用项目设计师指定UI文件。设计师可以使用鼠标拖曳GUI组件，以放置UI文件。

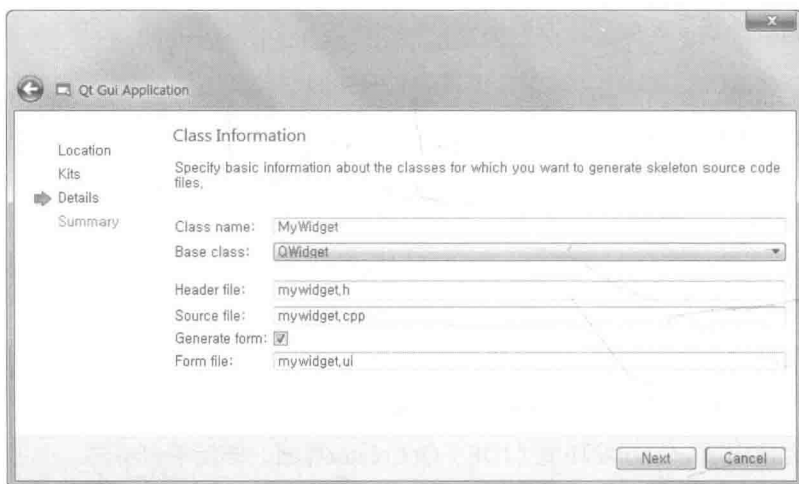


图2-3 类向导画面

点击图2-3中的Next（下一步）键，进入下个界面。下个界面是目前为止用项目向导生成的结果汇总，点击下方的Finish（完成）键完成项目创建。

生成项目后，项目文件中已生成了扩展名为.pro的文件。该文件指定了项目编译所需的库和模块。例如，如果开发的Qt应用程序想使用Network模块，就要如图2-4所示，向.pro文件添加Network模块。

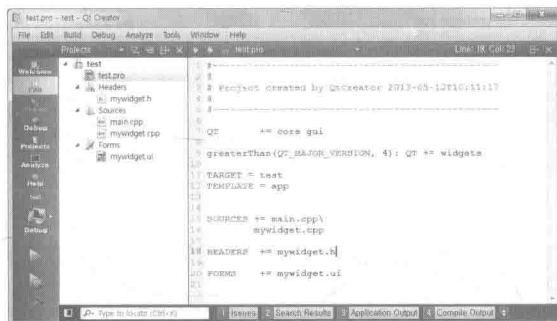


图2-4 Qt Creator项目运行画面

除Qt应用程序外，Qt Creator项目向导还可生成Qt库开发向导等多种项目。

2. 项目版本控制/配置控制

Qt Creator支持多种版本控制系统，以集成项目版本控制/配置控制。

表2-2 Qt Creator支持的版本控制系统

类 型	URL	备 注
Bazaar	http://bazaar.canonical.com/	Qt Creator 2.2以上版本
ClearCase	http://www-01.ibm.com/software/awdtools/clearcase/	Qt Creator 2.6插件
CVS	http://www.cvshome.org	
Git	http://git-scm.com/	
Mercurial	http://mercurial.selenic.com/	Qt Creator 2.0以上版本
Perforce	http://www.perforce.com	Server 2006.1以上版本
Subversion	http://subversion.apache.org/	

Qt Creator使用版本控制系统时，必须从版本控制系统中为PC客户端下载指定的目录。确定指定目录PATH前，要先在Qt Creator菜单中依次选择Tools→Options→Version Control（工具→选项→版本控制），然后设置要使用的版本控制系统。

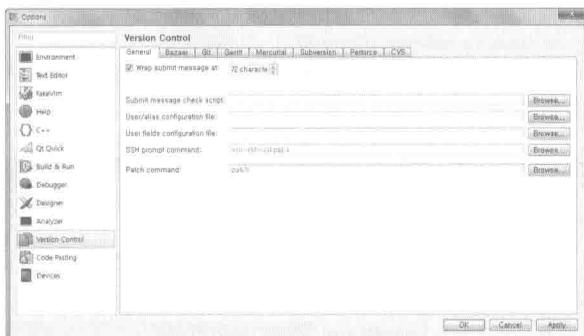


图2-5 Qt Creator版本控制系统设置画面

3. 设置项目的构建和运行配置项

Qt Creator可根据要开发的平台设置Build & Run（构建和运行）。例如，在Android、QNX或嵌入式Linux等被物理隔离的开发平台中，创建适宜的开发环境以设置构建和运行。现在，为了修改设置，按CTRL+5快捷键即可设置构建和运行。在菜单中依次选择Tools→Options→Build & Run→Kits（工具→选项→构建和运行→构建套件）即可。

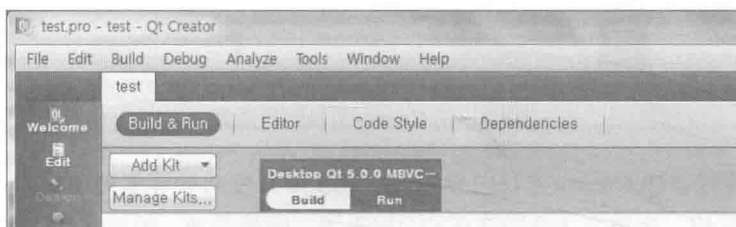


图2-6 构建和运行画面

4. 调试Qt应用程序

Qt Creator支持调试功能。可使用GDB和CDB调试工具调试当前已编译的可执行二进制文件。在菜单中选择Tools→Options→Debugger（工具→选项→调试器）即可设置GDB和CDB。

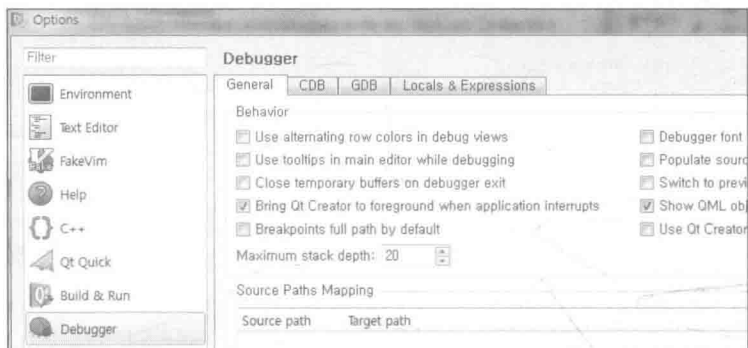


图2-7 调试器设置画面

Qt Creator还可以在GUI环境下调试程序。在Qt Creator的Debug菜单中可对用C++或QML开发的应用程序的调试配置进行设置。

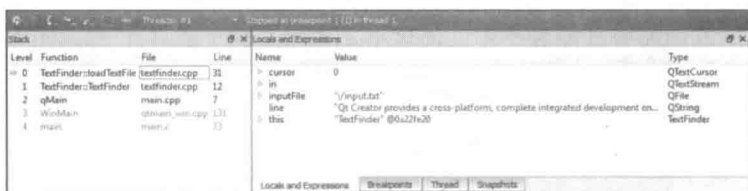


图2-8 GUI环境下的调试