1 菜单栏

```
#include "mainwindow.h"
#include <QMenuBar>
#include <QMenu>
#include <QAction>
MainWindow::MainWindow(QWidget *parent)
  : QMainWindow(parent)
{
 //取出菜单栏
 QMenuBar *menubar1 = this->menuBar();
  //向菜单栏上去添加 菜单
  QMenu * filemenu = menubar1->addMenu("文件");
  QMenu * fileedit = menubar1->addMenu("编辑");
  //向菜单添加菜单项
  QAction *openaction = filemenu->addAction("打开");
  filemenu->addSeparator();//添加分割线
  QAction *saveaction = filemenu->addAction("保存");

}
```

2 工具栏

```
#include "mainwindow.h"
#include <QMenuBar>
#include <QMenu>
#include <QAction>
#include <QToolBar>
```

```
MainWindow::MainWindow(QWidget *parent)
  : QMainWindow(parent)
{
 //取出菜单栏
  QMenuBar *menubar1 = this->menuBar();
  //向菜单栏上去添加 菜单
  QMenu * filemenu = menubar1->addMenu("文件");
  QMenu * fileedit = menubar1->addMenu("编辑");
  //向菜单添加菜单项
  QAction *openaction = filemenu->addAction("打开");
  filemenu->addSeparator();//添加分割线
  QAction *saveaction = filemenu->addAction("保存");

  //获取工具栏
  QToolBar *toolbar = this->addToolBar("");
  //向工具栏中添加工具(添加菜单项)
  toolbar->addAction(openaction);
  toolbar->addAction(saveaction);


 }
```

3状态栏

```
#include "mainwindow.h"
#include <QMenuBar>
#include <QMenu>
#include <QAction>
#include <QToolBar>
#include<QStatusBar>
#include<QLabel>
MainWindow::MainWindow(QWidget *parent)
  : QMainWindow(parent)
 {
  this->resize(600,400);
  //取出菜单栏
  QMenuBar *menubar1 = this->menuBar();
  //向菜单栏上去添加 菜单
  QMenu * filemenu = menubar1->addMenu("文件");
  QMenu * fileedit = menubar1->addMenu("编辑");
  //向菜单添加菜单项
```

```
18  QAction *openaction = filemenu->addAction("打开");
19  filemenu->addSeparator();//添加分割线
20  QAction *saveaction = filemenu->addAction("保存");
21
22  //获取工具栏
23  QToolBar *toolbar = this->addToolBar("");
24  //向工具栏中添加工具(添加菜单项)
25  toolbar->addAction(openaction);
26  toolbar->addAction(saveaction);
27
28  //取出状态栏
29  QStatusBar *status = this->statusBar();
30  status->addWidget(new QLabel("状态"));//向状态添加控件
31
32
33 }
```

4 铆接控件

```
1  #include "mainwindow.h"
2  #include <QMenuBar>
3  #include <QMenu>
4  #include <QAction>
5  #include <QToolBar>
6  #include<QStatusBar>
7  #include<QLabel>
8  #include <QDockWidget>
9  MainWindow::MainWindow(QWidget *parent)
10   : QMainWindow(parent)
11 {
12  this->resize(600,400);
13  //取出菜单栏
14  QMenuBar *menubar1 = this->menuBar();
15  //向菜单栏上去添加 菜单
16  QMenu * filemenu = menubar1->addMenu("文件");
17  QMenu * fileedit = menubar1->addMenu("编辑");
18  //向菜单添加菜单项
19  QAction *openaction = filemenu->addAction("打开");
20  filemenu->addSeparator();//添加分割线
```

```
21   QAction *saveaction = filemenu->addAction("保存");
22
23   //获取工具栏
24   QToolBar *toolbar = this->addToolBar("");
25   //向工具栏中添加工具(添加菜单项)
26   toolbar->addAction(openaction);
27   toolbar->addAction(saveaction);
28
29   //取出状态栏
30   QStatusBar *status = this->statusBar();
31   status->addWidget(new QLabel("状态"));//向状态添加控件
32   //创建铆接部件
33   QDockWidget *dockwidget = new QDockWidget("这是一个铆接部件", this);
34   this->addDockWidget(Qt::TopDockWidgetArea, dockwidget);//将浮动窗口添加到
mainwindow
35
36  }
```

5 中心控件

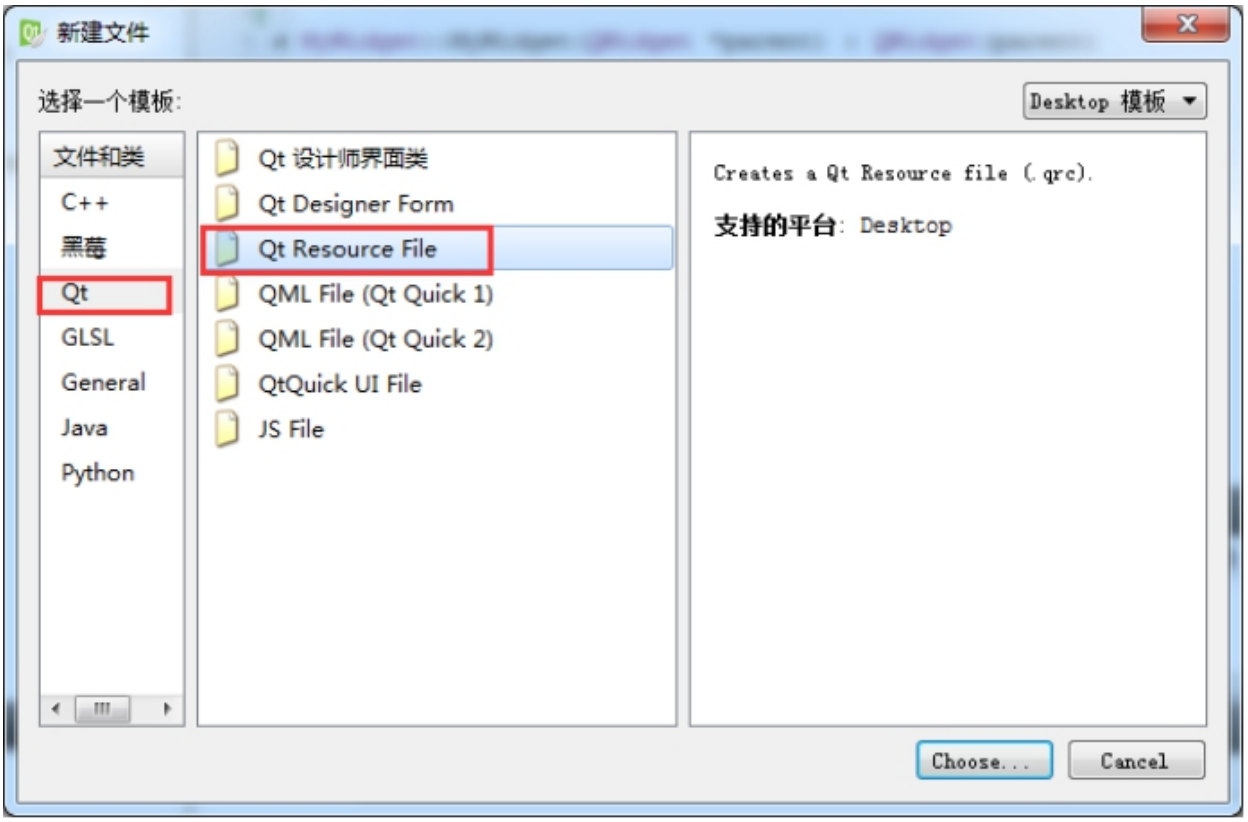
```
1  #include "mainwindow.h"
2  #include <QMenuBar>
3  #include <QMenu>
4  #include <QAction>
5  #include <QToolBar>
6  #include<QStatusBar>
7  #include<QLabel>
8  #include <QDockWidget>
9  #include <QTextEdit>
10  MainWindow::MainWindow(QWidget *parent)
11    : QMainWindow(parent)
12  {
13   this->resize(600,400);
14   //取出菜单栏
15   QMenuBar *menubar1 = this->menuBar();
16   //向菜单栏上去添加 菜单
17   QMenu * filemenu = menubar1->addMenu("文件");
18   QMenu * fileedit = menubar1->addMenu("编辑");
```

```
19   //向菜单添加菜单项
20   QAction *openaction = filemenu->addAction("打开");
21   filemenu->addSeparator();//添加分割线
22   QAction *saveaction = filemenu->addAction("保存");
23
24   //获取工具栏
25   QToolBar *toolbar = this->addToolBar("");
26   //向工具栏中添加工具(添加菜单项)
27   toolbar->addAction(openaction);
28   toolbar->addAction(saveaction);
29
30   //取出状态栏
31   QStatusBar *status = this->statusBar();
32   status->addWidget(new QLabel("状态"));//向状态添加控件
33   //创建铆接部件
34   //QDockWidget *dockwidget = new QDockWidget("这是一个铆接部件", this);
35   // this->addDockWidget(Qt::TopDockWidgetArea, dockwidget);//将浮动窗口添
加到mainwindow
36   QTextEdit *edit = new QTextEdit("文本编辑器", this);
37   this->setCentralWidget(edit);
38
39 }
40
41 MainWindow::~MainWindow()
42 {
43
44 }
45
```

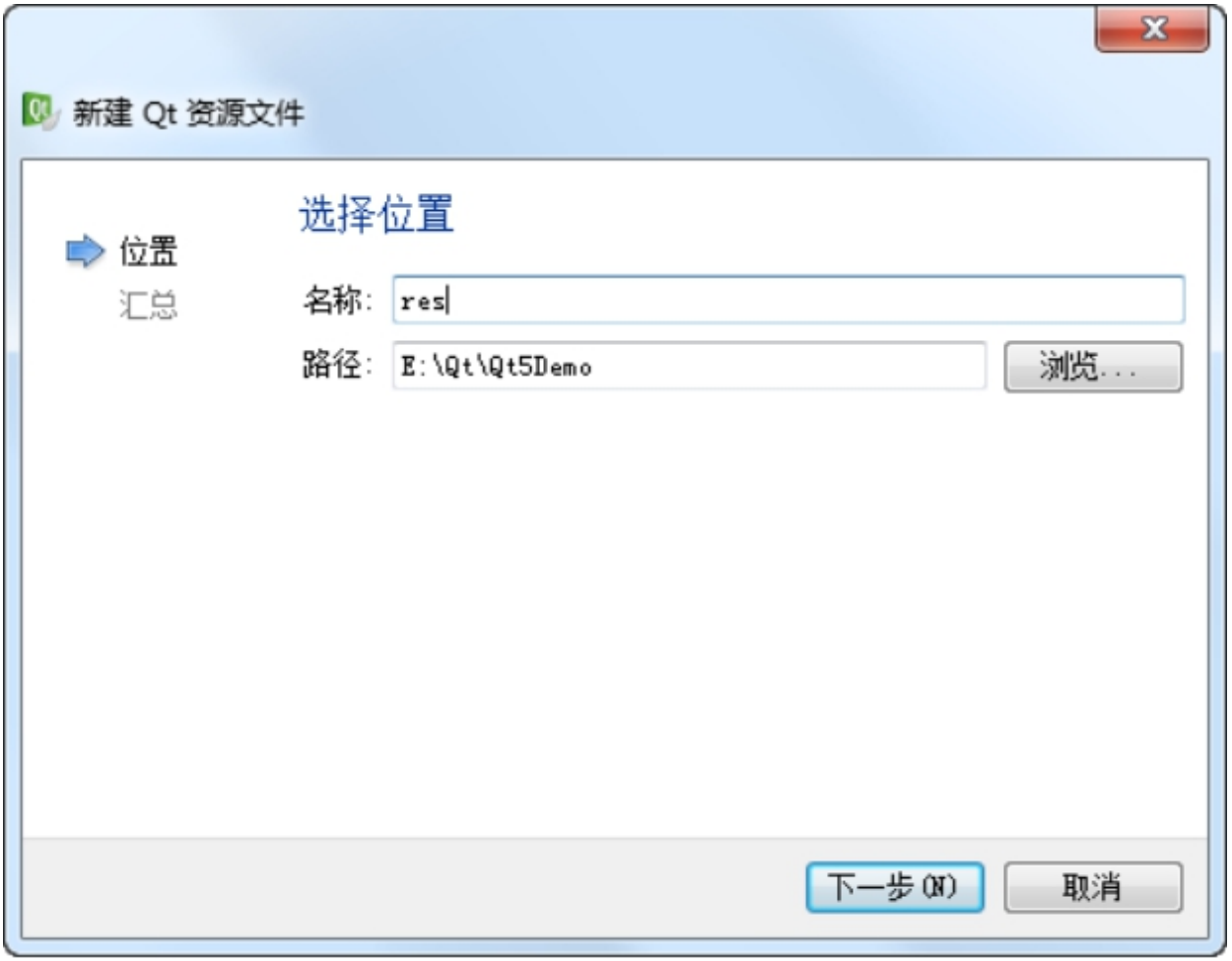
6 资源文件

Qt 资源系统是一个跨平台的资源机制，用于将程序运行时所需要的资源以二进制的形式存储于可执行文件内部。如果你的程序需要加载特定的资源（图标、文本翻译等），那么，将其放置在资源文件中，就再也不需要担心这些文件的丢失。也就是说，如果你将资源以资源文件形式存储，它是会编译到可执行文件内部。

使用 Qt Creator 可以很方便地创建资源文件。我们可以在工程上点右键，选择“添加新文件…”，可以在 Qt 分类下找到“Qt 资源文件”：



点击“选择…”按钮，打开“新建 Qt 资源文件”对话框。在这里我们输入资源文件的名字和路径：

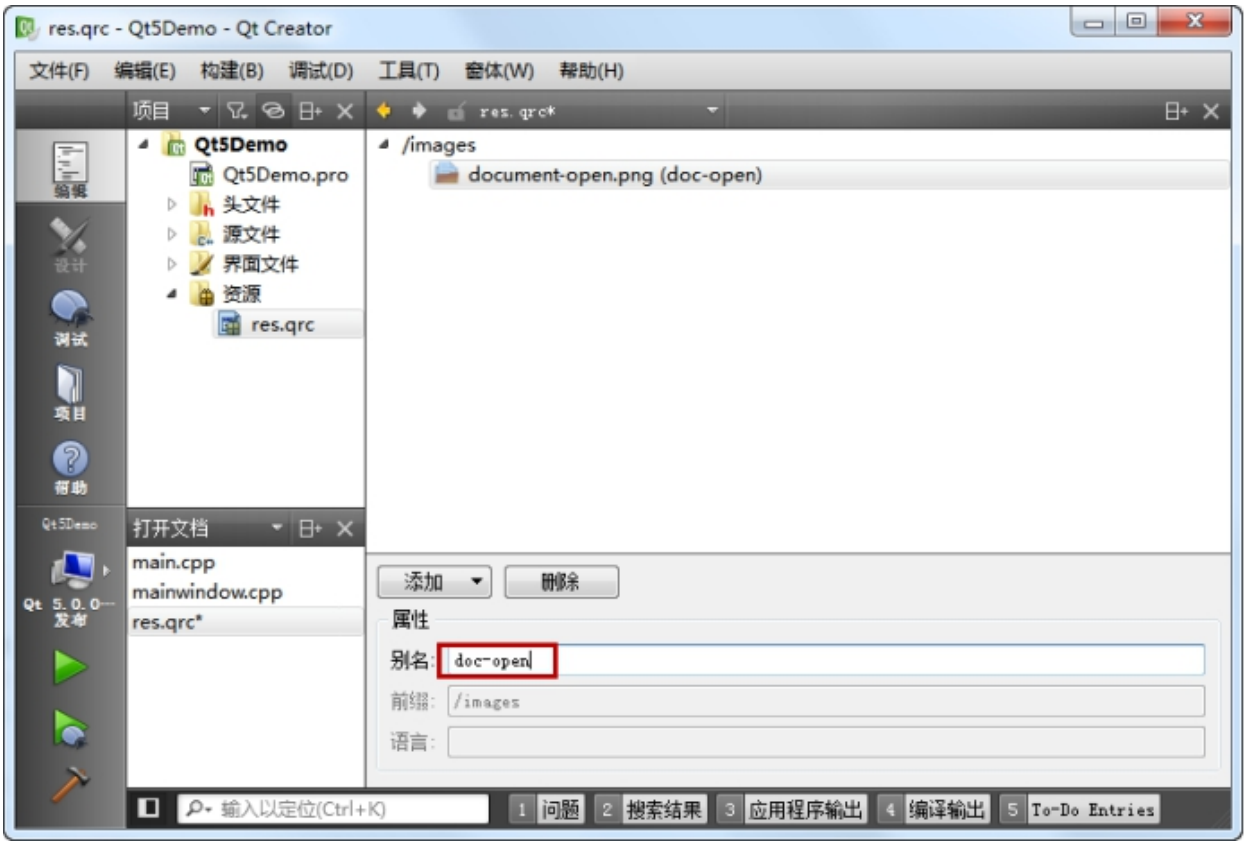


点击下一步，选择所需要的版本控制系统，然后直接选择完成。我们可以在 Qt Creator 的左侧文件列表中看到“资源文件”一项，也就是我们新创建的资源文件：

右侧的编辑区有个“添加”，我们首先需要添加前缀，比如我们将前缀取名为 images。然后选中这个前缀，继续点击添加文件，可以找到我们所需添加的文件。这里，我们选择 document-open.png 文件。当我们完成操作之后，Qt Creator 应该是这样子的：

接下来，我们还可以添加另外的前缀或者另外的文件。这取决于你的需要。当我们添加完成之后，我们可以像前面一章讲解的那样，通过使用 ：开头的路径来找到这个文件。比如，我们的前缀是 /images，文件是 document-open.png，那么就可以使用:/images/document-open.png找到这个文件。

这么做带来的一个问题是，如果以后我们要更改文件名，比如将 docuemnt-open.png 改成 docopen.png，那么，所有使用了这个名字的路径都需要修改。所以，更好的办法是，我们给这个文件去一个“别名”，以后就以这个别名来引用这个文件。具体做法是，选中这个文件，添加别名信息：



这样，我们可以直接使用:/images/doc-open引用到这个资源，无需关心图片的真实文件名。

如果我们使用文本编辑器打开 res.qrc 文件，就会看到一下内容：

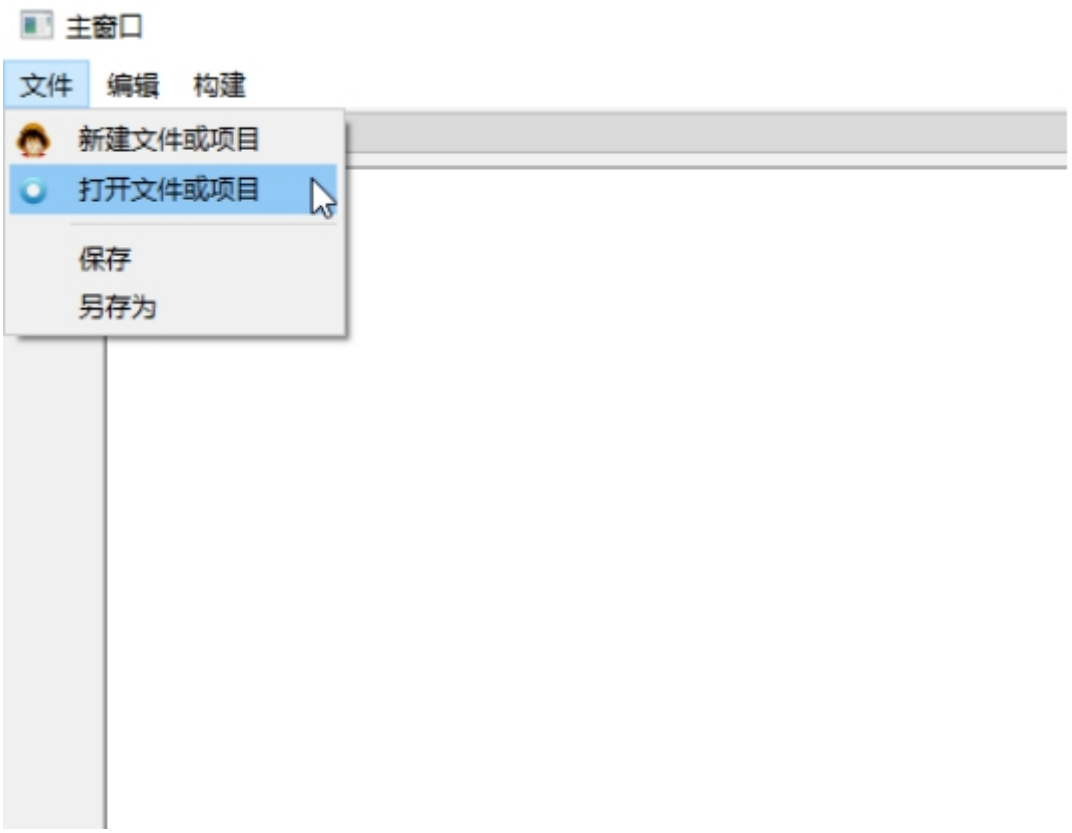
```
<RCC>
    <qresource prefix="/images">
        <file alias="doc-open">document-open.png</file>
    </qresource>
    <qresource prefix="/images/fr" lang="fr">
        <file alias="doc-open">document-open-fr.png</file>
    </qresource>
</RCC>
```

我们可以对比一下，看看 Qt Creator 帮我们生成的是怎样的 qrc 文件。当我们编译工程之后，我们可以在构建目录中找到 qrc_res.cpp 文件，这就是 Qt 将我们的资源编译成了 C++ 代码。

```
//在指定的菜单项上设置图片
//创建图片控件
QPixmap pix;
//选择图片
pix.load(":/image/Luffy.png");
//给菜单项设置图片
act1->setIcon(QIcon(pix));
```

```
pix.load(":/image/Start.png");
act2->setIcon(QIcon(pix));
```

执行结果




```
#include "mainwindow.h"
#include <QMenuBar>
#include <QMenu>
#include <QAction>
#include <QToolBar>
#include<QStatusBar>
#include<QLabel>
#include <QDockWidget>
#include <QTextEdit>
 #include <QPixmap>
 MainWindow::MainWindow(QWidget *parent)
   : QMainWindow(parent)
 {
  this->resize(600,400);
  //取出菜单栏
  QMenuBar *menubar1 = this->menuBar();
  //向菜单栏上去添加 菜单
  QMenu * filemenu = menubar1->addMenu("文件");
```

```
19  QMenu * fileedit = menubar1->addMenu("编辑");
20  //向菜单添加菜单项
21  QAction *openaction = filemenu->addAction("打开");
22  filemenu->addSeparator();//添加分割线
23  QAction *saveaction = filemenu->addAction("保存");
24
25  //获取工具栏
26  QToolBar *toolbar = this->addToolBar("");
27  //向工具栏中添加工具(添加菜单项)
28  toolbar->addAction(openaction);
29  toolbar->addAction(saveaction);
30
31  //取出状态栏
32  QStatusBar *status = this->statusBar();
33  status->addWidget(new QLabel("状态"));//向状态添加控件
34  //创建铆接部件
35  //QDockWidget *dockwidget = new QDockWidget("这是一个铆接部件", this);
36  // this->addDockWidget(Qt::TopDockWidgetArea, dockwidget);//将浮动窗口添加到mainwindow
37  QTextEdit *edit = new QTextEdit("文本编辑器", this);
38  this->setCentralWidget(edit);
39
40  //:/new/prefix1/a.bmp
41
42  QPixmap pic;//定义一个图片对象
43  pic.load(":/new/prefix1/a.bmp");//给图片对象加载图片
44  openaction->setIcon(QIcon(pic));
45
46 }
```