

Socrateos

A newbie to Linux/Ubuntu and Ruby programming

PAGES

- [Home](#)

SEARCH THIS BLOG

LABELS

- [adobe flash](#) (3)
- [Android](#) (1)
- [anjuta](#) (2)
- [Apache](#) (3)
- [Aptana](#) (1)
- [backup](#) (1)
- [blogger](#) (1)
- [Bowline](#) (2)
- [Bridged Network](#) (1)
- [Chrome](#) (1)
- [clipgrab](#) (1)
- [Code::Blocks](#) (3)

MONDAY, MAY 30, 2011

Using QtDesigner for Ruby Programming

QtDesigner is a GUI designer for Qt.
I'd like to try using it for ruby apps.

1. Preparation

- [Qt Creator](#) - All-in-one Qt IDE, which includes Designer. Installed through Ubuntu Software Center
- [qtbindings](#) - Ruby bindings for Qt lib. See [this](#) for installation
- rbui4 - A tool to convert Designer outputs (ui) to ruby files, installed with qtbindings gem

2. Create a Project Folder

```
1 | ~/work/ruby/gt/ui_test01      # project folder
2 | ~/work/ruby/qt/ui_test01/ui   # subfolder to place
   | QtDesigner outputs (ui files)
```

3. Create Forms using Designer

I followed an example shown in the second chapter of "[Foundations of Qt Development](#)" (by Johan Thelin from Apress). And I created two forms that look like:

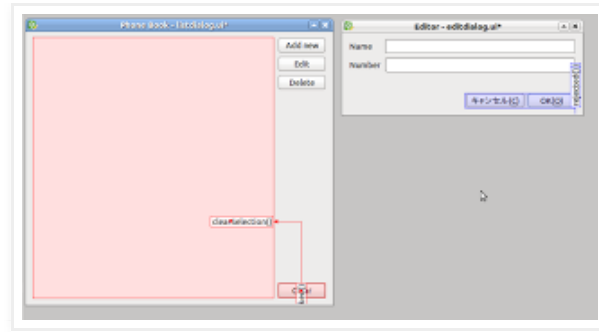
ABOUT ME

Socrateos

Learning Ubuntu and
Ruby

[View my complete
profile](#)

- [database](#) (1)
- [Debian](#) (1)
- [DevKit](#) (2)
- [dicom](#) (3)
- [editor](#) (6)
- [ExtJS](#) (2)
- [ffmpeg](#) (2)
- [filesystem](#) (1)
- [firefox](#) (1)
- [gconf-editor](#) (1)
- [Geany](#) (2)
- [gem](#) (2)
- [gem documentation](#) (1)
- [gem server](#) (1)
- [git](#) (2)
- [gLabels](#) (1)
- [Glade](#) (1)
- [glade3](#) (2)
- [GTK](#) (13)
- [gtk-ruby](#) (9)
- [gtk3](#) (3)
- [gui](#) (2)
- [java](#) (1)
- [javascript](#) (1)
- [jre](#) (1)
- [LAMP Server](#) (1)
- [libyaml](#) (1)
- [Markaby](#) (1)
- [Media Server](#) (6)
- [members](#) (1)
- [Mingw32](#) (1)
- [mp3](#) (2)
- [MySQL](#) (1)
- [nautilus](#) (1)



This is a simple Phone Book application. The idea is that when you click the "Add New" or "Edit" button in the List Dialog (shown on the left), it opens the Edit Dialog (shown on the right) to add or edit item. The "Delete" button will delete a selected item from the list, and the "Clear" button will delete selected items in the list.

Note that there is a red line that links the "Clear" button and the List box. The line signifies Qt's event handling mechanism: **Signal and Slot**. In the Designer, you draw the line by dragging from one object (Clear button) to another (Listbox) in Edit Connections mode, selecting **clicked()** as it's signal and the **clearSelection()** as its signal-receiving slot. When you make an event connection like this within the Designer, you will not have to write codes to make the connection between signal and slot as you will see below. (I did not create any link for the other 3 buttons. I will have to write codes manually for these connections.)

The Edit Dialog form is created using an existing template which already contains a pair of the CANCEL/OK buttons. And these buttons already have connections so that I will not have to write codes for their connections.

One important thing to remember when using the Designer is to add a grid layout to the dialog itself (not any widget group within); otherwise, widgets in the dialog will not line up nicely.

Save these files (listdialog.ui and editdialog.ui) in **/ui** folder.

- [nautilus-open-terminal](#) (1)
- [Netbeans](#) (3)
- [Network](#) (1)
- [Netzke](#) (2)
- [nginx](#) (4)
- [Octopress](#) (1)
- [Passenger](#) (1)
- [pg](#) (6)
- [pgAdmin](#) (1)
- [pik](#) (1)
- [Pony](#) (1)
- [PostgreSQL](#) (21)
- [PostgreSQL 9.1](#) (1)
- [Primergy TX100 S2](#) (1)
- [PS3](#) (2)
- [PS3 Media Server](#) (1)
- [ps3mediaserver](#) (1)
- [Qt](#) (1)
- [Qt Designer](#) (1)
- [Qt-Ruby](#) (5)
- [Qt4](#) (1)
- [qtbindings](#) (4)
- [Rails](#) (1)
- [ramaze](#) (9)
- [rhino](#) (1)
- [RMagic](#) (1)
- [Ruboto](#) (1)
- [ruby](#) (24)
- [Ruby 1.9.2](#) (2)
- [Ruby 1.9.3](#) (1)
- [Ruby in Windows](#) (1)
- [ruby mail](#) (2)

```
1 | $ cd ~/work/ruby/qt/ui_test01/ui
2 | $ ls
3 | editdialog.ui  listdialog.ui
```

4. Convert to Ruby files using rbuic4

```
1 | $ rbuic4 listdialog.ui -x -o listdialog_ui.rb
2 | $ rbuic4 editdialog.ui -x -o editdialog_ui.rb
```

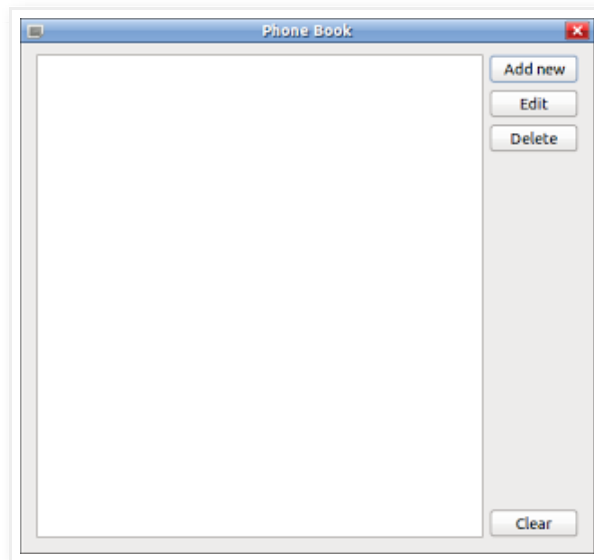
-x = generate extra code to test the class

-o = output file

5. A Quick Test for Generated ruby files

```
1 | $ ruby listdialog_ui.rb
```

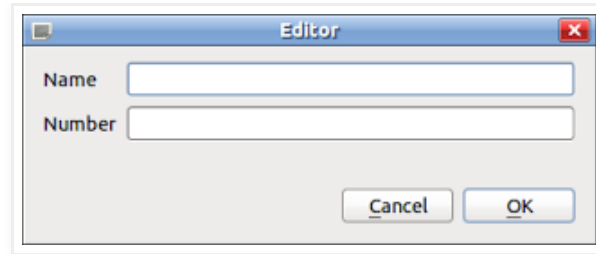
The result is:



```
1 | $ ruby editdialog_ui.rb
```

The result is:

- [ruby shoes](#) (3)
- [ruby-dicom](#) (3)
- [rubygems](#) (1)
- [RubyMine](#) (1)
- [rush](#) (2)
- [RVM](#) (1)
- [samba](#) (1)
- [sequel](#) (3)
- [Sequel](#) (1)
- [shotgun](#) (2)
- [Shutter](#) (1)
- [Sinatra](#) (3)
- [Slim](#) (1)
- [Sony Tablet](#) (1)
- [SQLite](#) (3)
- [sqlite3-ruby](#) (3)
- [terminal](#) (1)
- [ubuntu](#) (19)
- [Ubuntu 12.04](#) (1)
- [Ubuntu 13.04](#) (1)
- [Ubuntu Server 12.04](#) (3)
- [unicorn](#) (3)
- [virtual hosts](#) (1)
- [VirtualBox](#) (2)
- [vsftpd](#) (1)
- [Web design](#) (1)
- [Windows](#) (1)
- [Windows 8.1](#) (1)
- [Windows Server 2003](#) (1)
- [WinFF](#) (1)
- [WordPress](#) (2)
- [wxRuby](#) (7)
- [wxSmith](#) (1)



5. Contents of ui files

QtDesigner's forms are just xml files. Here is the content of **listdialog.ui**:

```

01 <ui version="4.0">
02   <class>ListDialog</class>
03   <widget class="QDialog" name="ListDialog">
04     <property name="geometry">
05       <rect>
06         <x>0</x>
07         <y>0</y>
08         <width>539</width>
09         <height>481</height>
10       </rect>
11     </property>
12     <property name="windowTitle">
13       <string>Phone Book</string>
14     </property>
15     <layout class="QGridLayout" name="gridLayout">
16       <item column="1" row="0">
17         <layout class="QVBoxLayout" name="verticalLayout">
18           <item>
19             <widget class="QPushButton" name="addButton">
20               <property name="text">
21                 <string>Add new</string>
22               </property>
23             </widget>
24           </item>
25           <item>
26             <widget class="QPushButton" name="editButton">
27               <property name="text">
28                 <string>Edit</string>
29               </property>

```

- [wxSugar](#) (1)
- [wxWidgets](#) (5)
- [Xbuntu](#) (1)
- [Youtube](#) (2)
- [Zen-CMS](#) (1)

BLOG ARCHIVE

- [2016](#) (3)
- [2015](#) (8)
- [2014](#) (8)
- [2013](#) (6)
- [2012](#) (23)
- ▼ [2011](#) (86)
 - [December](#) (2)
 - [November](#) (3)
 - [October](#) (5)
 - [September](#) (4)
 - [August](#) (10)
 - [July](#) (6)
 - [June](#) (1)
 - ▼ [May](#) (11)

Using
QtDesigner
for Ruby
Programming

Ruby Sites

Installing
qtbindings (alt
Qt-Ruby) on
Ubuntu

Installing json

Displaying List
of Tables in
PostgreSQL

```

30     </widget>
31   </item>
32 </item>
33   <widget class="QPushButton" name="deleteButton">
34     <property name="text">
35       <string>Delete</string>
36     </property>
37   </widget>
38 </item>
39 </item>
40   <spacer name="verticalSpacer">
41     <property name="orientation">
42       <enum>Qt::Vertical</enum>
43     </property>
44     <property name="sizeHint" stdset="0">
45       <size>
46         <width>20</width>
47         <height>40</height>
48       </size>
49     </property>
50   </spacer>
51 </item>
52 </item>
53   <widget class="QPushButton" name="clearButton">
54     <property name="text">
55       <string>Clear</string>
56     </property>
57   </widget>
58 </item>
59 </layout>
60 </item>
61 <item column="0" row="0">
62   <widget class="QListWidget" name="list">
63   </widget></item>
64 </layout>
65 </widget>
66 <resources>
67 <connections>
68   <connection>
69     <sender>clearButton</sender>
70     <signal>clicked()</signal>
71     <receiver>list</receiver>
72     <slot>clearSelection()</slot>
73   </hints>
74   <hint type="sourcelabel">
75     <x>495</x>
76     <y>455</y>
77   </hint>
78   <hint type="destinationlabel">

```

Switching to
PostgreSQL
in Netzk
Demo

Installing a
Netzk demo
app on
Ubuntu

Installing pg on
Ubuntu with
rvm

Ruby Cheat
Sheets

First Bowline
App Failed to
Run on
Ubuntu

Installing
Bowline

► April (15)

► March (18)

► January (11)

► 2010 (93)

```
79      <x>337</x>
80      <y>391</y>
81    </hint>
82  </hints>
83  </connection>
84  </connections>
85 </resources></ui>
```

And here is the contents of **editdialog.ui** file:

```
001 <ui version="4.0">
002   <class>EditDialog</class>
003   <widget class="QDialog" name="EditDialog">
004     <property name="geometry">
005       <rect>
006         <x>0</x>
007         <y>0</y>
008         <width>421</width>
009         <height>143</height>
010       </rect>
011     </property>
012     <property name="windowTitle">
013       <string>Editor</string>
014     </property>
015     <layout class="QGridLayout" name="gridLayout_2">
016       <item column="0" row="0">
017         <layout class="QGridLayout" name="gridLayout">
018           <item column="0" row="0">
019             <widget class="QLabel" name="nameLabel">
020               <property name="text">
021                 <string>Name</string>
022               </property>
023             </widget>
024           </item>
025           <item column="1" row="0">
026             <widget class="QLineEdit" name="nameEdit">
027             </widget></item>
028           <item column="0" row="1">
029             <widget class="QLabel" name="numberLabel">
030               <property name="text">
031                 <string>Number</string>
032               </property>
033             </widget>
034           </item>
035           <item column="1" row="1">
036             <widget class="QLineEdit" name="numberEdit">
037             </widget></item>
```

```
038     </layout>
039   </item>
040   <item column="0" row="1">
041     <spacer name="verticalSpacer">
042       <property name="orientation">
043         <enum>Qt::Vertical</enum>
044       </property>
045       <property name="sizeHint" stdset="0">
046         <size>
047           <width>20</width>
048           <height>15</height>
049         </size>
050       </property>
051     </spacer>
052   </item>
053   <item column="0" row="2">
054     <layout class="QHBoxLayout"
name="horizontalLayout_3">
055       <item>
056         <spacer name="horizontalSpacer">
057           <property name="orientation">
058             <enum>Qt::Horizontal</enum>
059           </property>
060           <property name="sizeHint" stdset="0">
061             <size>
062               <width>108</width>
063               <height>20</height>
064             </size>
065           </property>
066         </spacer>
067       </item>
068       <item>
069         <widget class="QDialogButtonBox"
name="buttonBox">
070           <property name="orientation">
071             <enum>Qt::Horizontal</enum>
072           </property>
073           <property name="standardButtons">
074             <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
075           </property>
076         </widget>
077       </item>
078     </layout>
079   </item>
080 </layout>
081 </widget>
082 <resources>
083 <connections>
084   <connection>
```

```

085     <sender>buttonBox</sender>
086     <signal>accepted()</signal>
087     <receiver>EditDialog</receiver>
088     <slot>accept()</slot>
089     <hints>
090         <hint type="sourcelabel">
091             <x>410</x>
092             <y>132</y>
093         </hint>
094         <hint type="destinationlabel">
095             <x>157</x>
096             <y>274</y>
097         </hint>
098     </hints>
099 </connection>
100 <connection>
101     <sender>buttonBox</sender>
102     <signal>rejected()</signal>
103     <receiver>EditDialog</receiver>
104     <slot>reject()</slot>
105     <hints>
106         <hint type="sourcelabel">
107             <x>410</x>
108             <y>132</y>
109         </hint>
110         <hint type="destinationlabel">
111             <x>286</x>
112             <y>274</y>
113         </hint>
114     </hints>
115 </connection>
116 </connections>
117 </resources></ui>

```

6. Contents of generated rb files

The tool **rbuic4** (Ruby UI Compiler) generates ruby files (rb) from QtDesigner's output files (ui). So **listdialog_ui.rb** looks like this:

```

001 =begin
002 ** Form generated from reading ui file 'listdialog.ui'
003 **
004 ** Created: 日 6月 19 11:49:30 2011
005 **      by: Qt User Interface Compiler version 4.7.0
006 **
007 ** WARNING! All changes made in this file will be lost
008 when recompiling ui file!
009 =end

```



```
009
010 require 'Qt4'
011
012 class Ui_ListDialog
013     attr_reader :gridLayout
014     attr_reader :verticalLayout
015     attr_reader :addButton
016     attr_reader :editButton
017     attr_reader :deleteButton
018     attr_reader :verticalSpacer
019     attr_reader :clearButton
020     attr_reader :list
021
022     def setupUi(listDialog)
023         if listDialog.objectName.nil?
024             listDialog.objectName = "listDialog"
025         end
026         listDialog.resize(539, 481)
027         @gridLayout = Qt::GridLayout.new(listDialog)
028         @gridLayout.setObjectName = "gridLayout"
029         @verticalLayout = Qt::VBoxLayout.new()
030         @verticalLayout.setObjectName = "verticalLayout"
031         @addButton = Qt::PushButton.new(listDialog)
032         @addButton.setObjectName = "addButton"
033
034         @verticalLayout.addWidget(@addButton)
035
036         @editButton = Qt::PushButton.new(listDialog)
037         @editButton.setObjectName = "editButton"
038
039         @verticalLayout.addWidget(@editButton)
040
041         @deleteButton = Qt::PushButton.new(listDialog)
042         @deleteButton.setObjectName = "deleteButton"
043
044         @verticalLayout.addWidget(@deleteButton)
045
046         @verticalSpacer = Qt::SpacerItem.new(20, 40,
Qt::SizePolicy::Minimum, Qt::SizePolicy::Expanding)
047
048         @verticalLayout.addItem(@verticalSpacer)
049
050         @clearButton = Qt::PushButton.new(listDialog)
051         @clearButton.setObjectName = "clearButton"
052
053         @verticalLayout.addWidget(@clearButton)
054
055
056         @gridLayout.addLayout(@verticalLayout, 0, 1, 1, 1)
```

```
057
058     @list = Qt::ListWidget.new(listDialog)
059     @list.setObjectName = "list"
060
061     @gridLayout.addWidget(@list, 0, 0, 1, 1)
062
063
064     retranslateUi(listDialog)
065     Qt::Object.connect(@clearButton,
066     SIGNAL('clicked()'), @list, SLOT('clearSelection()'))
067
068     Qt::MetaObject.connectSlotsByName(listDialog)
069     end # setupUi
070
071     def setup_ui(listDialog)
072         setupUi(listDialog)
073     end
074
075     def retranslateUi(listDialog)
076         listDialog.windowTitle =
077         Qt::Application.translate("ListDialog", "Phone Book",
078         nil, Qt::Application::UnicodeUTF8)
079         @addButton.text =
080         Qt::Application.translate("ListDialog", "Add new",
081         nil, Qt::Application::UnicodeUTF8)
082         @editButton.text =
083         Qt::Application.translate("ListDialog", "Edit", nil,
084         Qt::Application::UnicodeUTF8)
085         @deleteButton.text =
086         Qt::Application.translate("ListDialog", "Delete", nil,
087         Qt::Application::UnicodeUTF8)
088         @clearButton.text =
089         Qt::Application.translate("ListDialog", "Clear", nil,
090         Qt::Application::UnicodeUTF8)
091     end # retranslateUi
092
093     def retranslate_ui(listDialog)
094         retranslateUi(listDialog)
095     end
096
097 end
098
099 module Ui
100     class ListDialog < Ui_ListDialog
101     end
102 end # module Ui
103
104 if $0 == __FILE__
105     a = Qt::Application.new(ARGV)
```

```

095     u = Ui_ListDialog.new
096     w = Qt::Dialog.new
097     u.setupUi(w)
098     w.show
099     a.exec
100 end

```

And editdialog_ui.rb looks like this:

```

001 =begin
002 ** Form generated from reading ui file 'editdialog.ui'
003 **
004 ** Created: 金 6月 3 20:45:49 2011
005 **      by: Qt User Interface Compiler version 4.7.0
006 **
007 ** WARNING! All changes made in this file will be lost
when recompiling ui file!
008 =end
009
010 require 'Qt4'
011
012 class Ui_EditDialog
013     attr_reader :gridLayout_2
014     attr_reader :gridLayout
015     attr_reader :nameLabel
016     attr_reader :nameEdit
017     attr_reader :numberLabel
018     attr_reader :numberEdit
019     attr_reader :verticalSpacer
020     attr_reader :horizontalLayout_3
021     attr_reader :horizontalSpacer
022     attr_reader :buttonBox
023
024     def setupUi(editDialog)"
025     if editDialog.objectName.nil?
026         editDialog.objectName = "editDialog"
027     end
028     editDialog.resize(421, 143)
029     @gridLayout_2 = Qt::GridLayout.new(editDialog)
030     @gridLayout_2.setObjectName = "gridLayout_2"
031     @gridLayout = Qt::GridLayout.new()
032     @gridLayout.setObjectName = "gridLayout"
033     @nameLabel = Qt::Label.new(editDialog)
034     @nameLabel.setObjectName = "nameLabel"
035
036     @gridLayout.addWidget(@nameLabel, 0, 0, 1, 1)
037
038     @nameEdit = Qt::LineEdit.new(editDialog)
039     @nameEdit.setObjectName = "nameEdit"

```

```
040
041     @gridLayout.addWidget(@nameEdit, 0, 1, 1, 1)
042
043     @numberLabel = Qt::Label.new(editDialog)
044     @numberLabel.setObjectName = "numberLabel"
045
046     @gridLayout.addWidget(@numberLabel, 1, 0, 1, 1)
047
048     @numberEdit = Qt::LineEdit.new(editDialog)
049     @numberEdit.setObjectName = "numberEdit"
050
051     @gridLayout.addWidget(@numberEdit, 1, 1, 1, 1)
052
053
054     @gridLayout_2.addLayout(@gridLayout, 0, 0, 1, 1)
055
056     @verticalSpacer = Qt::SpacerItem.new(20, 15,
Qt::SizePolicy::Minimum, Qt::SizePolicy::Expanding)
057
058     @gridLayout_2.addItem(@verticalSpacer, 1, 0, 1, 1)
059
060     @horizontalLayout_3 = Qt::HBoxLayout.new()
061     @horizontalLayout_3.setObjectName =
"horizontalLayout_3"
062     @horizontalSpacer = Qt::SpacerItem.new(108, 20,
Qt::SizePolicy::Expanding, Qt::SizePolicy::Minimum)
063
064     @horizontalLayout_3.addItem(@horizontalSpacer)
065
066     @buttonBox = Qt::DialogButtonBox.new(editDialog)
067     @buttonBox.setObjectName = "buttonBox"
068     @buttonBox.orientation = Qt::Horizontal
069     @buttonBox.standardButtons =
Qt::DialogButtonBox::Cancel|Qt::DialogButtonBox::Ok
070
071     @horizontalLayout_3.addWidget(@buttonBox)
072
073
074     @gridLayout_2.addLayout(@horizontalLayout_3, 2, 0,
1, 1)
075
076
077     retranslateUi(editDialog)
078     Qt::Object.connect(@buttonBox,
SIGNAL('accepted()'), editDialog, SLOT('accept()'))
079     Qt::Object.connect(@buttonBox,
SIGNAL('rejected()'), editDialog, SLOT('reject()'))
080
081     Qt::MetaObject.connectSlotsByName(editDialog)
```

```

082     end # setupUi
083
084     def setup_ui(editDialog)
085         setupUi(editDialog)
086     end
087
088     def retranslateUi(editDialog)
089         editDialog.windowTitle =
Qt::Application.translate("EditDialog", "Editor", nil,
Qt::Application::UnicodeUTF8)
090         @nameLabel.text =
Qt::Application.translate("EditDialog", "Name", nil,
Qt::Application::UnicodeUTF8)
091         @numberLabel.text =
Qt::Application.translate("EditDialog", "Number", nil,
Qt::Application::UnicodeUTF8)
092     end # retranslateUi
093
094     def retranslate_ui(editDialog)
095         retranslateUi(editDialog)
096     end
097
098 end
099
100 module Ui
101     class EditDialog < Ui_EditDialog
102     end
103 end # module Ui
104
105 if $0 == __FILE__
106     a = Qt::Application.new(ARGV)
107     u = Ui_EditDialog.new
108     w = Qt::Dialog.new
109     u.setupUi(w)
110     w.show
111     a.exec
112 end

```

7. Some notes on generated ruby codes (1) You should not modify these files manually because they will be lost when re-generated. (2) The block "if \$0 == __FILE__ ... end" was added by -x option.

```

1 | $ rbuic4 listdialog.ui -x -o listdialog_ui.rb      #
   | when -x option is used....
1 | if $0 == __FILE__                                #
   | this block is added due to -x option above
2 |     a = Qt::Application.new(ARGV)

```

```

3      u = Ui_ListDialog.new
4      w = Qt::Dialog.new
5      u.setupUi(w)
6      w.show
7      a.exec
8  end

```

This is convenient in that you can run it immediately (before writing any codes) to see how its form looks like. (3) Another convenience for this block is that it can be used as a template for your main ruby program as you will see below (main.rb). (4) The connection between signal and slot is made using string literals of method names (including parenthesis):

```

1  Qt::Object.connect(@clearButton, SIGNAL('clicked()'),
   @listView, SLOT('clearSelection()'))

1  Qt::Object.connect(@buttonBox, SIGNAL('accepted()'),
   editDialog, SLOT('accept()'))
2  Qt::Object.connect(@buttonBox, SIGNAL('rejected()'),
   editDialog, SLOT('reject()'))

```

(5) A dialog within Qt application is an instance of Qt::Dialog or Qt::Widget class. However, you see from the examples above that dialog classes in generated ruby ui files (Ui_ListDialog and Ui_EditDialog) do not inherit from Qt::Dialog or from Qt::Widget.

```

1  class Ui_ListDialog
2  ...
3  end
4  class Ui_EditDialog
5  ...
6  end

```

This means that somewhere in my application, I have to creates a dialog object that inherits from Qt::Dialog and that somehow I have to link it to the dialog class that defines its ui elements created by the Designer. That is done by the 3 lines in above example:

```

1  u = Ui_ListDialog.new # insatance of ui class
2  w = Qt::Dialog.new    # instance of Qt::Dialog
3  u.setupUi(w)          # link the two to make the Qt
   dialog object to use UI elements

```

(6) A new module "Ui" is created and empty subclasses (ListDialog, EditDialog) inherited from the generated classes (Ui_ListDialog, ListDialog) is created in the

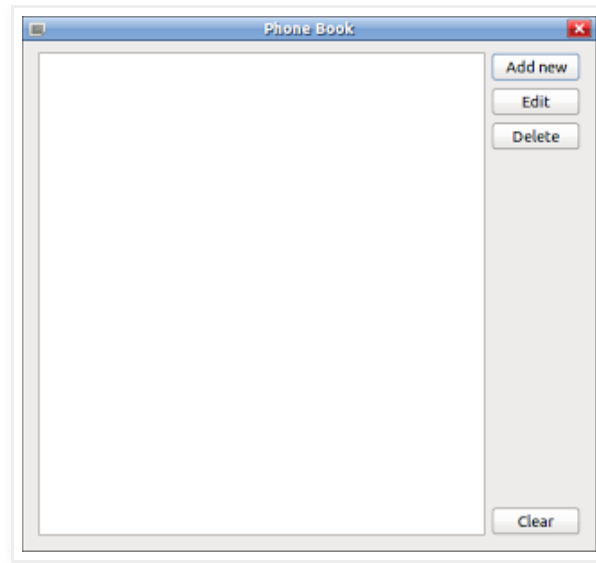
new module Ui. The idea is that your own codes and changes are to be made in subclasses so that re-generating will not erase your changes. (7) The method `setupUi()` makes its form alive in your application by creating a bunch of instance variables for widgets and layouts in the form. A more Ruby-like method **`setup_ui()`** is created to call `setupUi()`. (8) The method `retranslateUi()` is defined for localization/internationalization of UI. **8. Creating main.rb** My first version of main program started with a copy from a generated ui file (`listdialog_ui.rb`) and modified little. First I load those two ui files. I use subclass `Ui::ListDialog` instead of the parent `Ui_ListDialog` class. I can changes the subclass without worrying about being over-written by re-generation of ui codes. Here the goal is just to open the Edit Dialog. There is no functionality: clicking on a button does nothing.

```
01 # main.rb
02 # version 1
03 #
04 require 'Qt4'
05 require './ui/listdialog_ui' # load ui file
    generated by the Designer. I get an error without
    "/" in Ruby 1.9.2
06 require './ui/editdialog_ui' # load ui file
    generated by the Designer
07
08 if $0 == __FILE__
09     a = Qt::Application.new(ARGV)
10     w = Qt::Dialog.new
11     u = Ui::ListDialog.new # use subclass
12     u.setup_ui(w)         # more rubyish,
    replacing setupUi()
13     w.show
14     a.exec
15 end
```

Run it.

```
1 $ ruby main.rb
```

The result.



9. Updating main.rb

Now I would like to add a functionality: clicking "Add New" or "Edit" button opens an Edit Dialog. To do this, I

create two `Qt::Dialog` classes (**MainForm** and **EditForm**) and link them to **ui** objects created with the Designer. Then I declare 3 slot methods that respond to `clicked()` signals. (I do not have to declare 'clear_selection()' slot method - already done within the Designer by drawing a line from Clear button to the List Widget.) Those slot methods create an instance of **EditForm** and calls **exec** method so that the dialog opens in modal mode. Note that I compare the return value of **exec** with fixed number 1. I am supposed to use predefined constant **Qt::Accepted**, but it crashes on me when I use it.

```
01 # main.rb
02 # version 2
03 #
04
05 require 'Qt4'
06 require './ui/listdialog_ui'
07 require './ui/editdialog_ui'
08
09 class MainForm < Qt::Dialog
10
11   slots 'add_item()', 'edit_item()', 'delete_item()'
12   # declaration
13   # slots 'clear_selection' -- in parent class:
14   Ui_ListDialog
```



```

13
14   def initialize
15     super
16     @ui = Ui::ListDialog.new
17     @ui.setup_ui(self)
18     Qt::Object.connect(@ui.addButton,
19       SIGNAL('clicked()'), self, SLOT('add_item()'))
20     Qt::Object.connect(@ui.editButton,
21       SIGNAL('clicked()'), self, SLOT('edit_item()'))
22     Qt::Object.connect(@ui.deleteButton,
23       SIGNAL('clicked()'), self, SLOT('delete_item()'))
24     # -- in parent class: Ui_ListDialog
25     #Qt::Object.connect(@ui.clearButton,
26       SIGNAL('clicked()'), self, SLOT('clear_selection()'))
27
28     self.show
29   end
30
31   def add_item()
32     d = EditForm.new(self)
33     if(d.exec == 1) # I use "1" instead of
34       Qt::Accepted because it crashes on me
35     end
36   end
37
38   def edit_item()
39     d = EditForm.new(self)
40     if(d.exec == 1) # I use "1" instead of
41       Qt::Accepted because it crashes on me
42     end
43   end
44
45   def delete_item()
46   end
47
48   # -- in parent class: Ui_ListDialog
49   # def clear_selection()
50   # end
51
52 end
53
54 class EditForm < Qt::Dialog
55   def initialize(parent=nil)
56     super(parent)
57     @ui = Ui::EditDialog.new
58     @ui.setup_ui(self)
59     self.show
60   end

```

```
56
57 end
58
59 if $0 == __FILE__
60   a = Qt::Application.new(ARGV)
61   MainForm.new
62   a.exec
63 end
```

10. Final main.rb

Finally I define all slot methods, giving all required functions to the program.

```
01 # main.rb
02 # version 3
03 #
04 require 'Qt4'
05 require './ui/listdialog_ui'
06 require './ui/editdialog_ui'
07
08 class MainForm < Qt::Dialog
09
10   slots 'add_item()', 'edit_item()', 'delete_item()'
11
12   def initialize
13     super
14
15     @ui = Ui::ListDialog.new
16     @ui.setup_ui(self)
17
18     Qt::Object.connect(@ui.addButton,
19       SIGNAL('clicked()'), self, SLOT('add_item()'))
20     Qt::Object.connect(@ui.editButton,
21       SIGNAL('clicked()'), self, SLOT('edit_item()'))
22     Qt::Object.connect(@ui.deleteButton,
23       SIGNAL('clicked()'), self, SLOT('delete_item()'))
24
25     self.show
26
27   end
28
29   def add_item()
30     d = EditForm.new(self)
```

```
28     if(d.exec == 1) # OK clicked
29         @ui.list.add_item(d.name + ": " + d.number)
30     end
31 end
32
33 def edit_item()
34     if(@ui.list.current_item) # if any item is
35         selected
36         temp = @ui.list.current_item.text
37         a = temp.split(/: /)
38         d = EditForm.new(self)
39         d.name = a[0]
40         d.number = a[1]
41         if(d.exec == 1) # OK clicked
42             @ui.list.current_item.text = d.name + ": " +
43             d.number
44         end
45     end
46 end
47
48 def delete_item()
49     @ui.list.current_item.dispose # delete selected
50     object
51 end
52
53 end
54
55 class EditForm < Qt::Dialog
56
57     def initialize(parent=nil)
58         super(parent)
59         @ui = Ui::EditDialog.new
60         @ui.setup_ui(self)
61         self.show
62     end
63
64     def name
65         @ui.nameEdit.text
66     end
67
68     def name=(s)
69         @ui.nameEdit.set_text(s)
70     end
71
72     def number
73         @ui.numberEdit.text
74     end
75
76     def number=(s)
```

```
74     @ui.numberEdit.setText(s)
75   end
76
77 end
78
79 if $0 == __FILE__
80   a = Qt::Application.new(ARGV)
81   MainForm.new
82   a.exec
83 end
```

Posted by [Socrateos](#) at 10:21 AM

+2 Recommend this on Google

Labels: [Qt Designer](#), [Qt-Ruby](#), [qtbindings](#)

4 comments:

**Anonymous** January 23, 2013 at 7:35 AM

Great tutorial, thanks a lot!

[Reply](#)**zipizap** September 20, 2013 at 7:27 PM

This is the best tutorial I've found explaining Ruby, Qt and QtCreator... really thanks

[Reply](#)**MatrakMatrix** December 15, 2014 at 6:54 AM

Thank you so much kind sir! You saved my life :) Awesome tutorial

[Reply](#)**Anonymous** July 21, 2015 at 10:08 AM

Thanks for the tutorial but I still have a couple of questions to ask. How can I safely read and save binary files? You see, I've been trying to allow my code to store info in such a way both Ruby and QtRuby can read it. I know Marshal.dump and load

methods don't work, their Marshal versions conflict some how. What should I do then?

[Reply](#)

Enter your comment...

Comment as: Ángel (Google)

[Sign out](#)

[Publish](#)

[Preview](#)

☐ [Notify me](#)

Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Powered by [Blogger](#).