How to use QIntValidator in Qt

Introduction

The QIntValidator class provides a validation that ensures a string contains a valid integer within a specified range.

Values consisting of a number of digits equal to or less than the max value are considered intermediate.

The minimum and maximum values are set in one call with setRange(), or individually with setBottom() and setTop().

Various Functions

Note: val is object of QValidator.

- This property holds the validator's lowest acceptable value.

```
val->setbottom(1);
```

- This property holds the validator's highest acceptable value.

```
val->setTop(10);
```

Source File

main.cpp

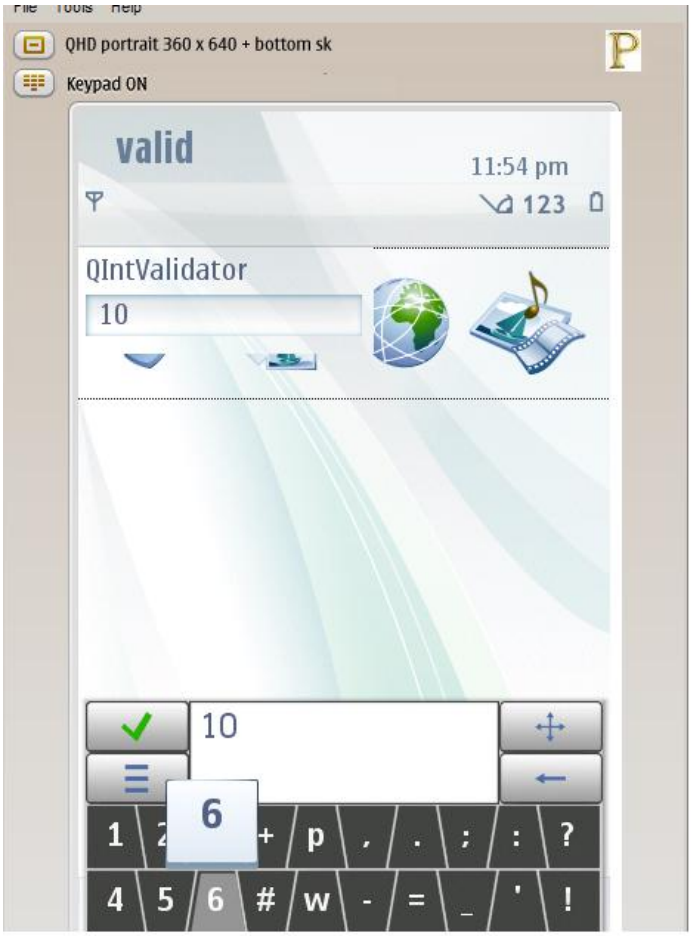
```
#include <QtGui/QApplication>
#include "valid.h"
#include<QWidget>
#include<QLineEdit>
#include<QVBoxLayout>
#include <QIntValidator>
#include<QValidator>
#include<QLabel>
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget *win=new QWidget();
    QLabel *lbl=new QLabel("QIntValidator");
    QLineEdit *edit = new QLineEdit();
    lbl->setBuddy(edit);
    QValidator *val=new QIntValidator(1,10,edit);
    edit->setValidator(val);
    QVBoxLayout *lay=new QVBoxLayout();
    lay->addWidget(lbl);
    lay->addWidget(edit);
    win->setLayout(lay);
    win->show();
    return a.exec();
}
```

Note: From the above code validator we can write the integer value between 1 to 10.You can't write value more than 10.

Screenshot

More AboutQIntValidator..

(C) Copyright Nokia 2014. All rights reserved.



As you can see in the above image that we are unable to enter three digit value in QLineEdit,because the last and maximum possible value is "10" that is 2 digit.

From above image we can say that after pressing the "1" in keypad the Validator does not allowing us enter "11" in line

(C) Copyright Nokia 2014. All rights reserved.

edit as it maximum value is "10"

More AboutQIntValidator..

(C) Copyright Nokia 2014. All rights reserved.