Archived:Implementando uma TreeView em um QComboBox - Parte 2

Aquivado: Este artigo foi arquivado, pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template {{ForArchiveReview|*escreva a sua justificativa*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (based on QWidget) is deprecated.

Este código mostra o uso da QTreeView implementada em um QComboBox. Aqui você pode expandir a arvóre de diretórios e ainda selecionar o arquivo que você deseja, sem a restrição do QComboBox fechar com o clique, você pode selecionar alguns itens até sua confirmação

Introdução

Nome de classes usadas

- QTreeView-Provêr uma implementação de modelo/visão, padrão de uma árvore de diretórios.
- QDirModel-Provêr uma modelo de dados para o sistema local de arquivos.

Código fonte

```
#include <QtGui/QApplication>
#include "widget.h"
#include<QComboBox>
#include<QDirModel>
#include<QTreeView>
#include<QEvent>
#include<QMouseEvent>
#include<QModelIndex>
#include<QDir>
class TreeBox : public QComboBox
{
public:
        TreeBox(QWidget* parent = 0) : QComboBox(parent), skipNextHide(false) //Widget
creation
        {
                setView(new QTreeView(this));
                view()->viewport()->installEventFilter(this);
                QComboBox::resize(200,30);
        }

        bool eventFilter(QObject* object, QEvent* event)
        {
                if (event->type() == QEvent::MouseButtonPress && object == view()-
>viewport())
                {
                        QMouseEvent* mouseEvent = static_cast<QMouseEvent*>(event);
                        QModelIndex index = view()->indexAt(mouseEvent->pos());
                        if (!view()->visualRect(index).contains(mouseEvent->pos()))
                                skipNextHide = true;
                }
                return false;
        }

        virtual void showPopup()
```


(C) Copyright Nokia 2014. All rights reserved.

```
        {
                setRootModelIndex(static_cast<QDirModel*>(model())-
>index(QDir::rootPath()));
                QComboBox::showPopup();
        }

        virtual void hidePopup()
        {
                setRootModelIndex(view()->currentIndex().parent());
                setCurrentIndex(view()->currentIndex().row());
                if (skipNextHide)
                        skipNextHide = false;
                else
                        QComboBox::hidePopup();
        }

private:
        bool skipNextHide;
};
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
        TreeBox combo; //Class object declaration
        QDirModel model;
        combo.setModel(&model); //implemention QDir model
        combo.show();
    return a.exec();
}
```

Captura de tela

 (C) Copyright Nokia 2014. All rights reserved.

Links relacionados

Archived:Implementando uma TreeView em um QComboBox - Parte 1

(C) Copyright Nokia 2014. All rights reserved.