

Les auto-connexions de QMetaObject

Sous Qt, les connexions entre les signaux et slots peuvent être mis en place soit manuellement, soit automatiquement, en utilisant la capacité qu'a QMetaObject à établir des liens entre ces derniers.

Cette partie, ne concerne pas les connexions manuelles (traitées à cette [adresse](#)), en revanche elle traite de la méthode automatique.

Voici la boîte de dialogue que nous souhaitons réaliser:



Cette boîte de dialogue, contient un seul élément, à savoir un QPushButton, que l'on a nommé *myButton* et que nous connecterons à un slot, en se gardant de faire appel à la fonction **connect()** de QMetaObject. Pour cela, nous disposons des 5 fichiers suivants:

- autoconnexionform.ui (fichier de Qt Designer)
- autoconnexionform.pro (fichier de projet Qt)
- autoconnexionform.h
- autoconnexionform.cpp
- main.cpp

Bien qu'il soit plutôt aisé d'implémenter un slot dans un QDialog et de le connecter dans le constructeur, nous pouvons tout aussi bien utiliser l'outil d'auto-connexion de QMetaObject pour connecter le signal **clicked()** de *myButton* à un slot dans notre classe.

Lors de la génération du code de la fonction **setupUi()** de notre QDialog par *uic*, une ligne qui nous intéresse particulièrement est générée:

```
QMetaObject::connectSlotsByName(autoconnexionform);
```

Cette méthode, du QMetaObject de Qt, connecte automatiquement tout slot qui respecte la convention **on_Nomobjet_Nomsignal()** au signal **Nomsignal()** correspondant de *Nomobjet*.

Il nous reste alors, seulement, à déclarer le slot, toujours en suivant la convention standard suivante:

```
void on_Nomobjet_Nomsignal(Parametresignal);
```

Et de l'implémenter.

Voici le code de notre boîte de dialogue *AutoConnexionForm*, mettant en oeuvre une connexion automatique établie par le QMetaObject, dont le rôle est d'afficher « myButton has been clicked... » sur la console lorsque l'utilisateur clique sur le bouton *myButton*.

```
#ifndef AUTOCONNEXIONFORM_H  
#define AUTOCONNEXIONFORM_H
```

```

#include "ui_autoconnexionform.h"

class AutoConnexionForm : public QDialog, public
Ui::autoconnexionform
{
    Q_OBJECT

public:
    AutoConnexionForm(QWidget *parent = 0);

public slots:
    void on_myButton_clicked();
};

#endif

```

Dans le code ci-dessus, on déclare le slot `on_myButton_clicked()`, hérité de `Ui::autoconnexionform`.

```

#include <iostream>
#include "autoconnexionform.h"

AutoConnexionForm::AutoConnexionForm(QWidget *parent)
    : QDialog(parent)
{
    setupUi(this);
}

void AutoConnexionForm::on_myButton_clicked()
{
    std::cout << "myButton has been clicked..." << std::endl;
}

```

Dans cette portion de code, nous implémentons le slot, pour qu'il fasse ce que l'on souhaite, dans notre cas, il se contente d'afficher « myButton has been clicked... » sur la sortie standard.

Note: On souhaite voir le message s'afficher dans notre console, pour cela, nous rajoutons dans `autoconnexionform`, la ligne suivante: `CONFIG += console`

On termine par l'écriture de notre *main*:

```

#include "autoconnexionform.h"

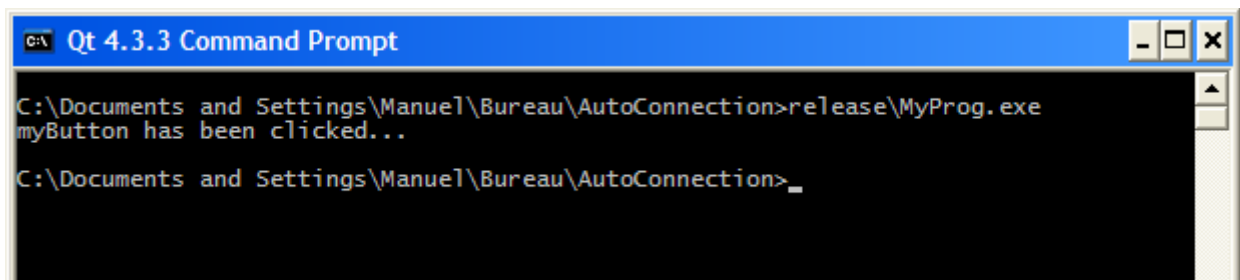
int main(int argc, char *argv[]) {
    QApplication app(argc, argv);
    AutoConnexionForm myForm;

    myForm.show();

    return app.exec();
}

```

Une fois notre programme compilé et lancé, je clique sur le bouton « myButton » au centre de ma boîte de dialogue et j'observe le résultat...



A screenshot of a Qt 4.3.3 Command Prompt window. The title bar is blue and contains the text "Qt 4.3.3 Command Prompt" and standard window control buttons. The main area is black with white text. The prompt shows the current directory as "C:\Documents and Settings\Manuel\Bureau\AutoConnection". The command "release\MyProg.exe" has been entered and executed, resulting in the output "myButton has been clicked...". The prompt is now waiting for the next command, indicated by a white cursor.

```
C:\Documents and Settings\Manuel\Bureau\AutoConnection>release\MyProg.exe
myButton has been clicked...
C:\Documents and Settings\Manuel\Bureau\AutoConnection>_
```

Mac & Cheese