

Cours Qt

Création d'une application en 10 étapes

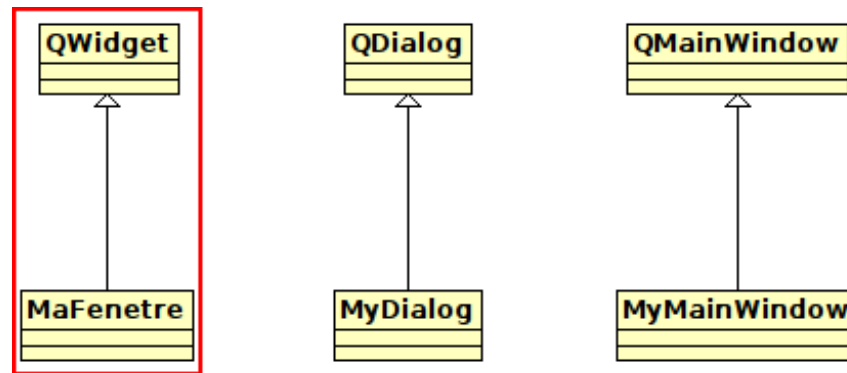
Thierry Vaira

IUT Arles

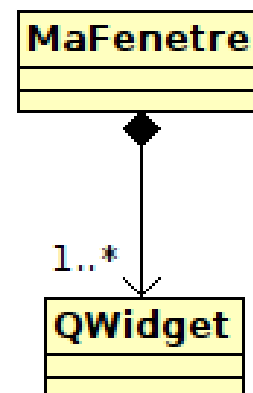
tvaira@free.fr © v1

Création d'une application "fenêtre"

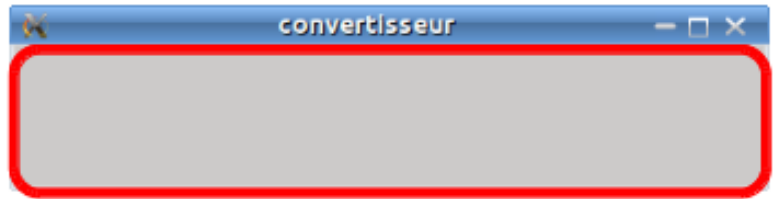
La création de **fenêtres personnalisées** est réalisée à partir d'une nouvelle classe qui **hérite** de QWidget, QDialog ou QMainWindow :



Ensuite, on **compose** sa fenêtre personnalisée en y intégrant des *widgets* :



Principes I



Un widget « fenêtre »
(QWidget)

```
class MaFenetre : public QWidget
{
    Q_OBJECT

private:

public:
    MaFenetre(QWidget *parent=0);
    ~MaFenetre();
};
```

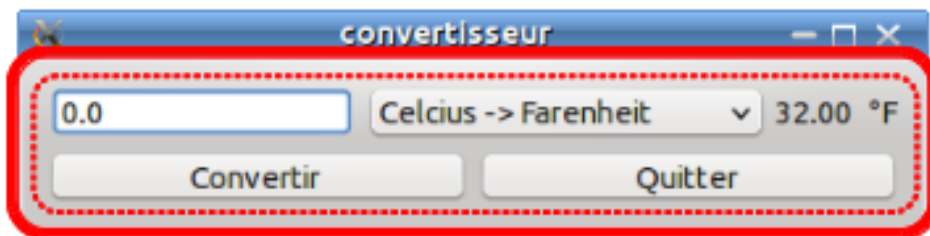
```
int main( int argc, char **argv )
{
    QApplication a( argc, argv );

    MaFenetre w; // rappel : pas de parent = fenêtre !
    w.show();

    return a.exec();
}
```

Un *widget* qui n'est pas intégré dans un *widget* parent est appelé une **fenêtre**.





**Un widget « fenêtre »
(QWidget) composé
d'autres widgets**

```
class MaFenetre : public QWidget
{
    Q_OBJECT

private:
    QLineEdit    *valeur;
    QLabel       *resultat;
    QLabel       *unite;
    QComboBox    *choixConversion;
    QPushButton  *bConvertir;
    QPushButton  *bQuitter;

public:
    MaFenetre(QWidget *parent=0);
    ~MaFenetre();
};
```

On va créer cette application simple en **10 étapes** !



Étape n°1 : déclaration de la classe MaFenetre

```
// Par héritage ici d'un QWidget
class MaFenetre : public QWidget
{
    Q_OBJECT

private:

public:
    MaFenetre(QWidget *parent=0);
    ~MaFenetre();
};
```



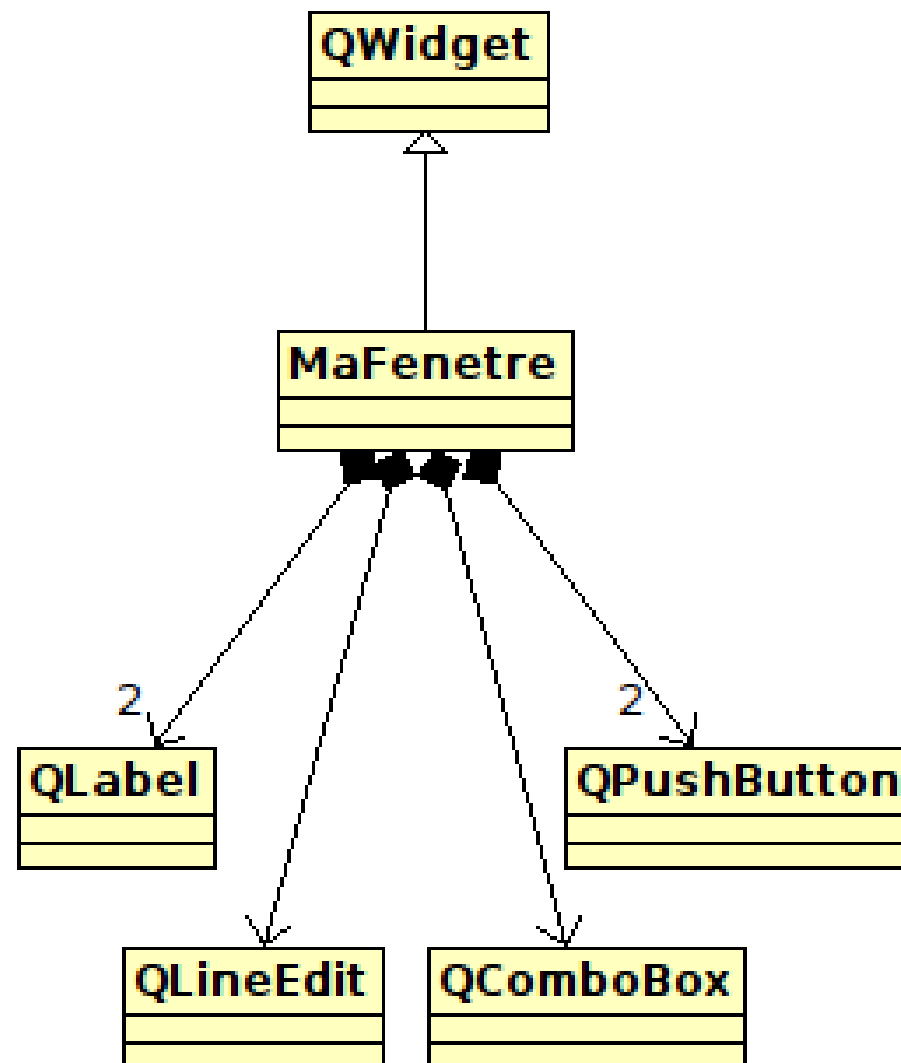
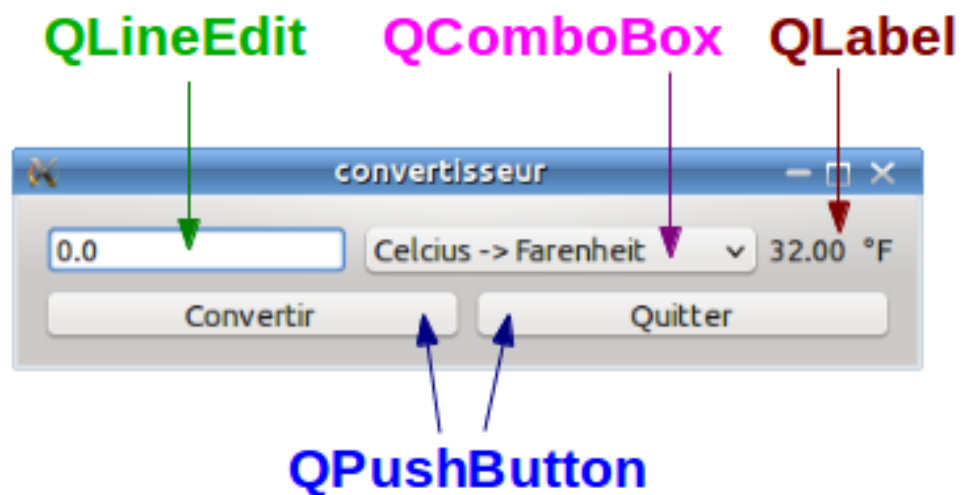
Étape n°2 : définition de la classe MaFenetre

```
// Le constructeur : appel du constructeur de la classe mère
MaFenetre::MaFenetre(QWidget *parent) : QWidget(parent)
{
    setWindowTitle("convertisseur"); // le titre
}

// Le destructeur
MaFenetre::~~MaFenetre()
{
}
```



Exemple : les widgets de la fenêtre



Étape n°3 : déclaration des widgets de la classe MaFenetre

```
class MaFenetre : public QWidget
{
    Q_OBJECT

private:
    QLineEdit    *valeur; // par pointeurs
    QLabel       *resultat;
    QLabel       *unite;
    QComboBox    *choixConversion;
    QPushButton  *bConvertir;
    QPushButton  *bQuitter;

public:
    MaFenetre(QWidget *parent=0);
    ~MaFenetre();
};
```

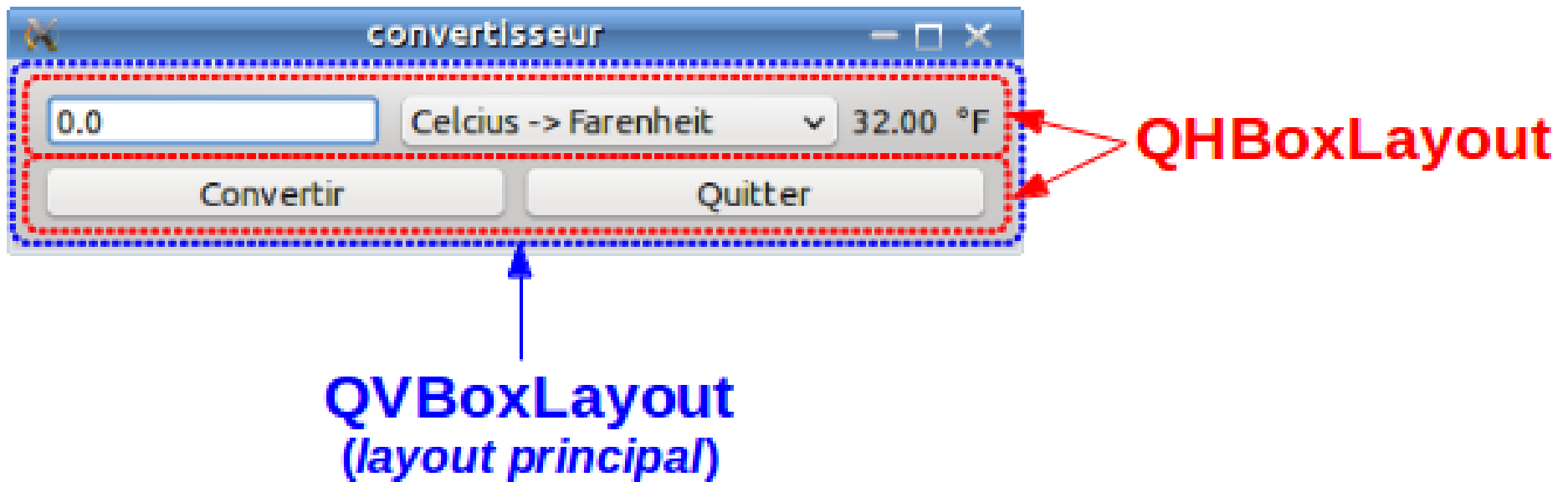

Étape n°4 : instantiation des widgets de la classe MaFenetre

```
// Dans le constructeur de la fenêtre
MaFenetre::MaFenetre(QWidget *parent) : QWidget(parent)
{
    // Mes widgets enfants :
    valeur          = new QLineEdit(this);
    resultat        = new QLabel(this);
    unite           = new QLabel(this);
    choixConversion = new QComboBox(this);
    bConvertir      = new QPushButton("Convertir", this);
    bQuitter        = new QPushButton("Quitter", this);

    //...
}
```



Exemple : le positionnement des widgets dans la fenêtre



Étape n°5 : positionnement des widgets

```
MaFenetre::MaFenetre(QWidget *parent) : QWidget(parent)
{
    // Mes widgets enfants :
    //...

    // Positionnement par layout :
    QHBoxLayout *hLayout1 = new QHBoxLayout;
    QHBoxLayout *hLayout2 = new QHBoxLayout;
    QVBoxLayout *mainLayout = new QVBoxLayout;
    hLayout1->addWidget(valeur);
    hLayout1->addWidget(choixConversion);
    hLayout1->addWidget(resultat);
    hLayout1->addWidget(unite);
    hLayout2->addWidget(bConvertir);
    hLayout2->addWidget(bQuitter);
    mainLayout->addLayout(hLayout1);
    mainLayout->addLayout(hLayout2);
    setLayout(mainLayout);
}
```

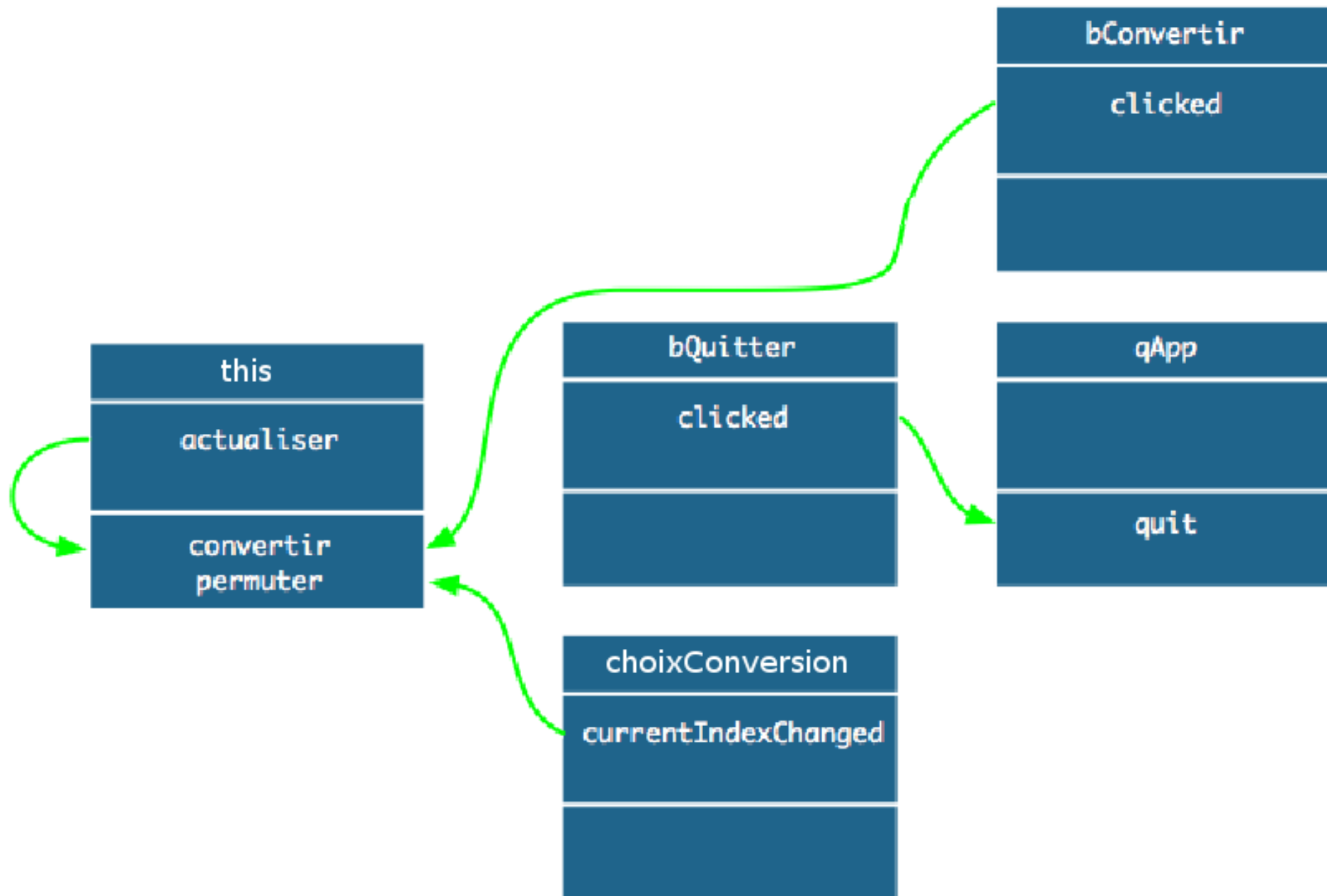
Étape n°6 : initialisation des widgets

```
// Dans le constructeur de la fenêtre
MaFenetre::MaFenetre(QWidget *parent) : QWidget(parent)
{
    // Mes widgets enfants :
    //...
    // Positionnement par layout :
    //...

    // Initialisation des widgets :
    valeur->clear();
    choixConversion->addItem("Celcius -> Farenheit");
    choixConversion->addItem("Farenheit -> Celcius");
    resultat->setText(QString::fromUtf8("--.--"));
    unite->setText(QString::fromUtf8(" °F"));
}
```



Exemple : la connexion des signaux / slots



Étape n°7 : déclaration des signaux et des slots

```
class MaFenetre : public QWidget
{
    Q_OBJECT

private:
    //...

public:
    MaFenetre(QWidget *parent=0);
    ~MaFenetre();

signals:
    void actualiser();

private slots:
    void convertir();
    void permuter(int index);
};
```

Étape n°8 : connexion signal/slot

```
MaFenetre::MaFenetre(QWidget *parent) : QWidget(parent)
{
    //...

    // Connexion signal/slot :
    connect(bConvertir, SIGNAL(clicked()), this, SLOT(convertir()));
    connect(this, SIGNAL(actualiser()), this, SLOT(convertir()));
    connect(choix, SIGNAL(currentIndexChanged(int)), this, SLOT(permuter(
        int)));
    connect(bQuitter, SIGNAL(clicked()), qApp, SLOT(quit()));
}
```



Étape n°9 : définition des slots

```
void MaFenetre::convertir()
{
    switch (choixConversion->currentIndex()) {
        case 0: resultat->setText(QString::fromUtf8("%1").arg(9 * valeur->
            text().toDouble() / 5 + 32, 0, 'f', 2));
            unite->setText(QString::fromUtf8(" °F"));
            break;
        case 1: resultat->setText(QString::fromUtf8("%1").arg(5 * (valeur->
            text().toDouble() - 32) / 9, 0, 'f', 2));
            unite->setText(QString::fromUtf8(" °C"));
            break;
    }
}

void MaFenetre::permuter(int index)
{
    valeur->setText(resultat->text()); /* on permute */
    emit actualiser(); /* on déclenche une nouvelle conversion */
}
```


Étape n°10 : l'application principale

```
#include <QApplication>
#include "mafenetre.h"

int main( int argc, char **argv )
{
    QApplication a( argc, argv );

    MaFenetre w; // rappel : pas de parent = fenêtre !

    w.show();

    return a.exec();
}
```

