Qt dans l'Enseignement

Les grandes lignes…

digia

Traduction française et Adaptation des supports pour l'enseignement

http://www.qt.io/training-materials/
http://www.qt.io/qt-essentials-widget-edition/

digia 2

Qu'est-ce que Qt ?

"Qt est une librairie de développement multiplateforme écrite en C++."

- C++ de base mais adaptation pour d'autres
  - Python, Ruby, C#, etc.
- Multiplateforme pour tous les aspects
  - Interfaces, bases de données, XML, Web,
  - multimedia, réseau, OpenGL, types de base...

Qu'est-ce que Qt?

- C++ avec macros + introspection

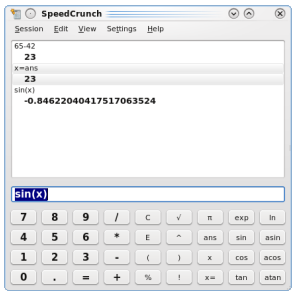
```
foreach (int value, intList) { ... }
```

```
QObject *o = new QPushButton;
o->metaObject()->className(); // returns "QPushButton"
```

```
connect(button, SIGNAL(clicked()), window, SLOT(close()));
```

Objectifs de Qt

- Code unique quelle que soit la plateforme
- Applications avec « look and feel » natif

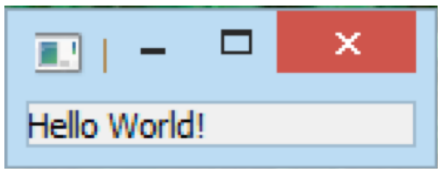


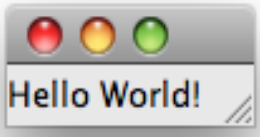
- API facile à (ré-)utiliser, haute productivité, outils simples à utiliser

La communauté Qt

- http://www.qt.io
- http://www.qt.io/developers/
- http://qt.developpez.com/
- http://www.qtfr.org/
- etc.

Exemple : Hello World





Hello World : le code

```
#include <QApplication>
#include <QLabel>

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```

Simplicité du code

Hello World : le code

```
#include <QApplication>
#include <QLabel>

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```


Hello World : le code

```
#include <QApplication>
#include <QLabel>

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```

Instanciation obligatoire

Autre accès : pointeur qApp

Hello World : le code

```
#include <QApplication>
#include <QLabel>

int main( int argc, char **argv )
{
    QApplication app( argc, argv );
    QLabel l( "Hello World!" );
    l.show();
    return app.exec();
}
```

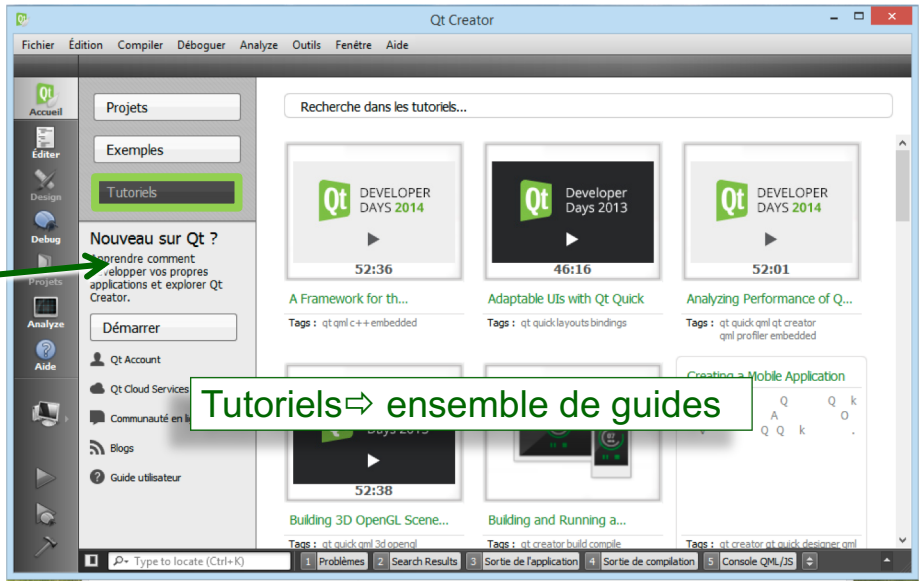
Widget sans parent ⇨ affiché dans une fenêtre

Hello World : le code

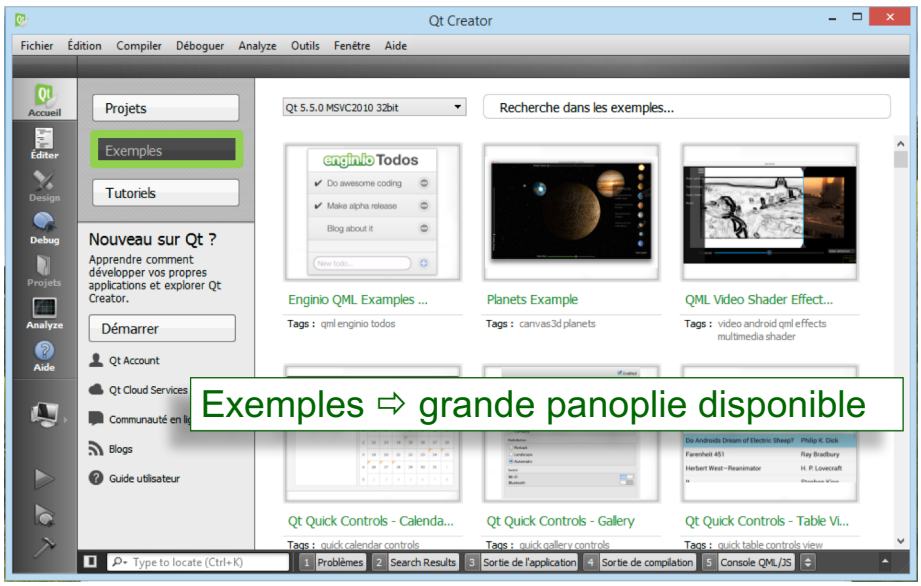
```
#include <QApplication>
#include <QLabel>

int main
{
    QAppl
    QLabel
    l.show();
    return app.exec();
}
```

- Lance la boucle d'attente des événements (interaction utilisateur, réseau ou chronomètre)
- Arrêt quand dernière fenêtre fermée

Démonstration QtCreator

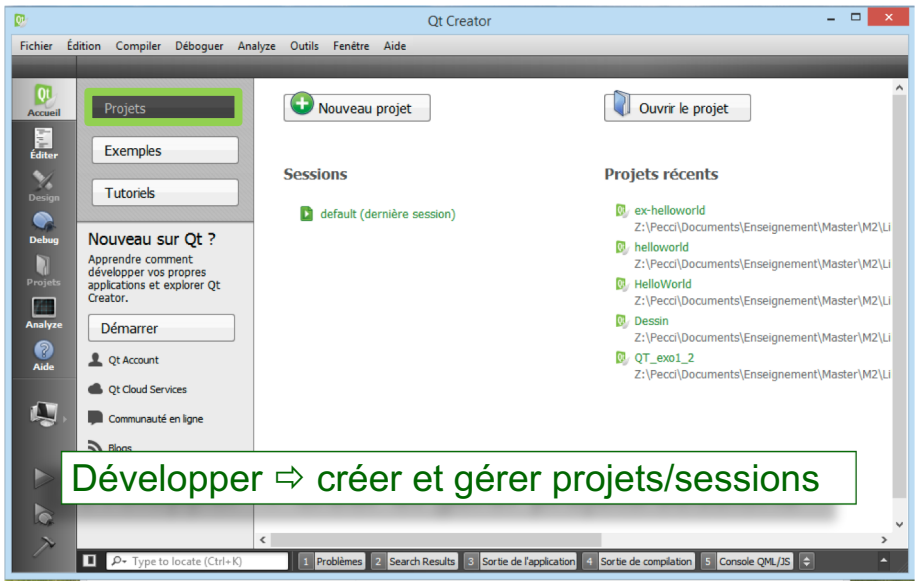


Tutoriels⇨ ensemble de guides



Exemples ⇨ grande panoplie disponible

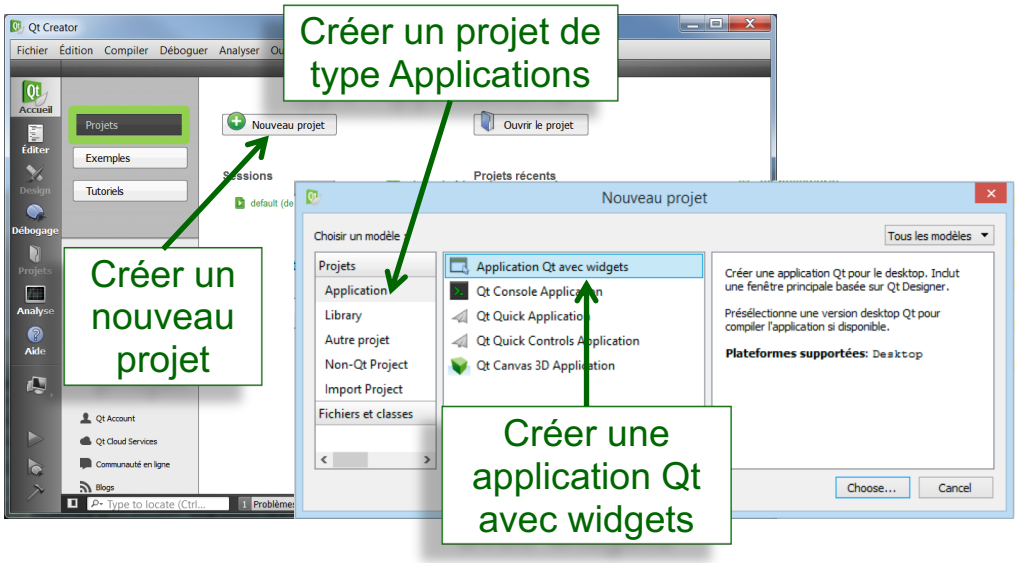
digia 15



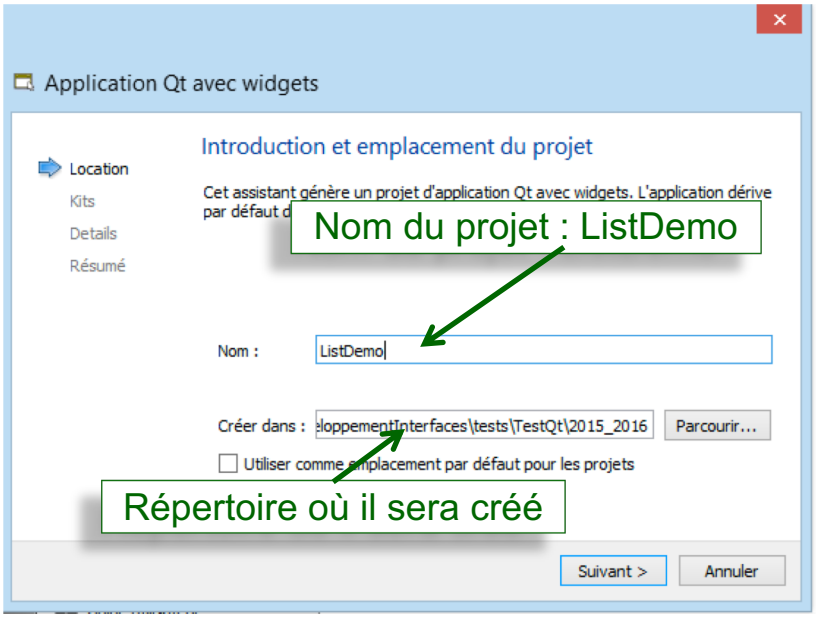
Développer ⇨ créer et gérer projets/sessions

digia 16

Créer un nouveau projet

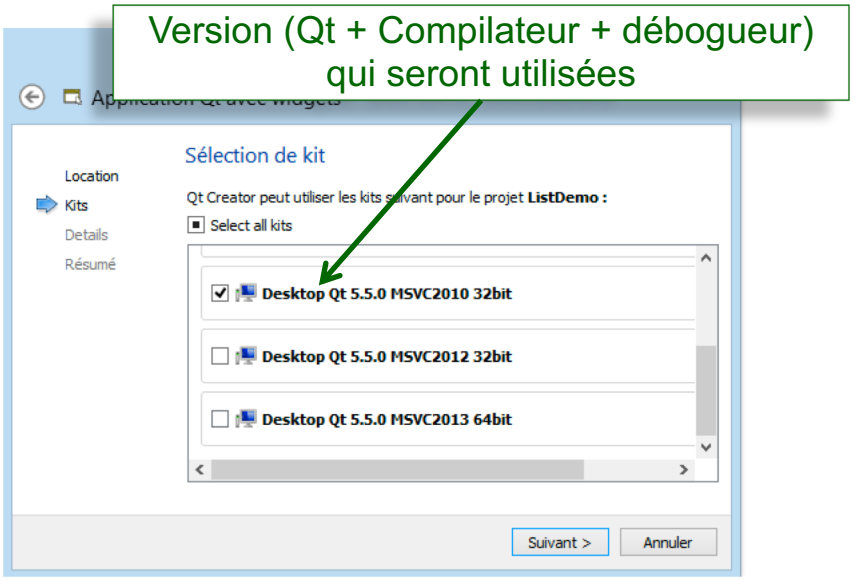


Créer un nouveau projet



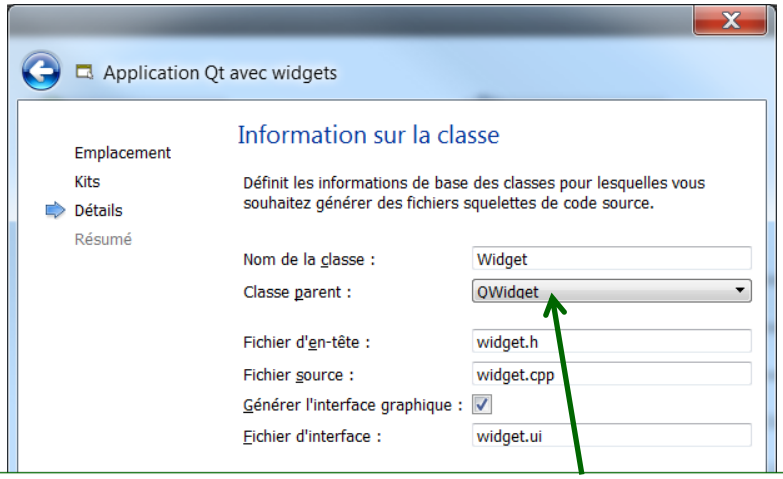
digia 19

Créer un nouveau projet



digia 20

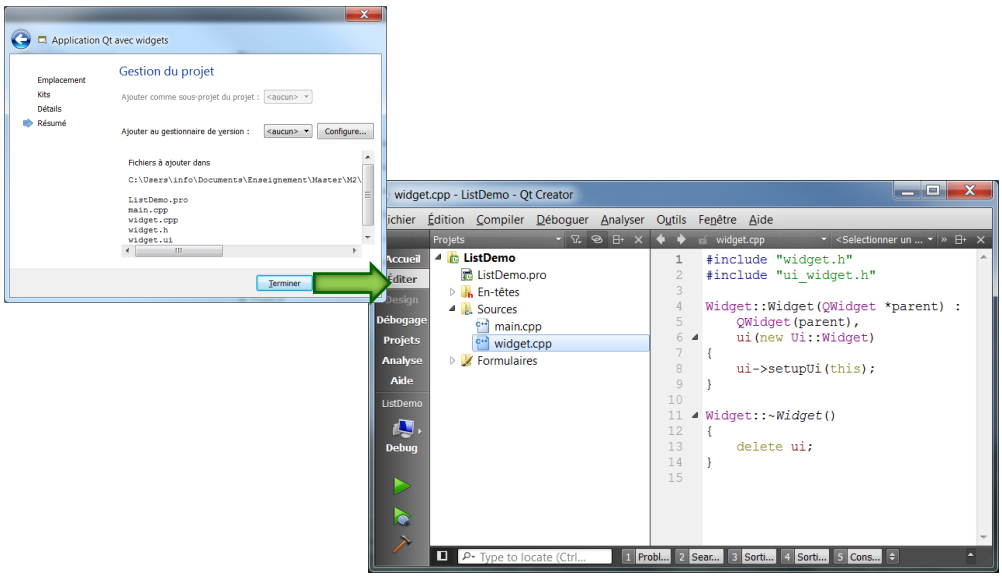
Créer un nouveau projet



Choisir une application basée par exemple sur une widget Qwidget.
Autres templates possibles:
- Application avec une fenêtre principale, QMainWindow
- Application avec une boîte de dialogue, QDialog

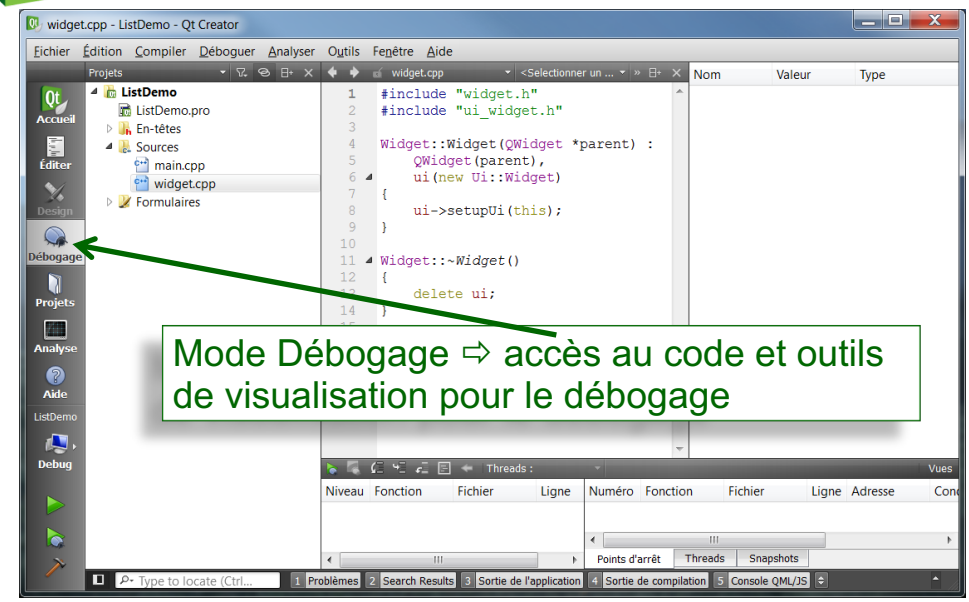
digia 21

Créer un nouveau projet

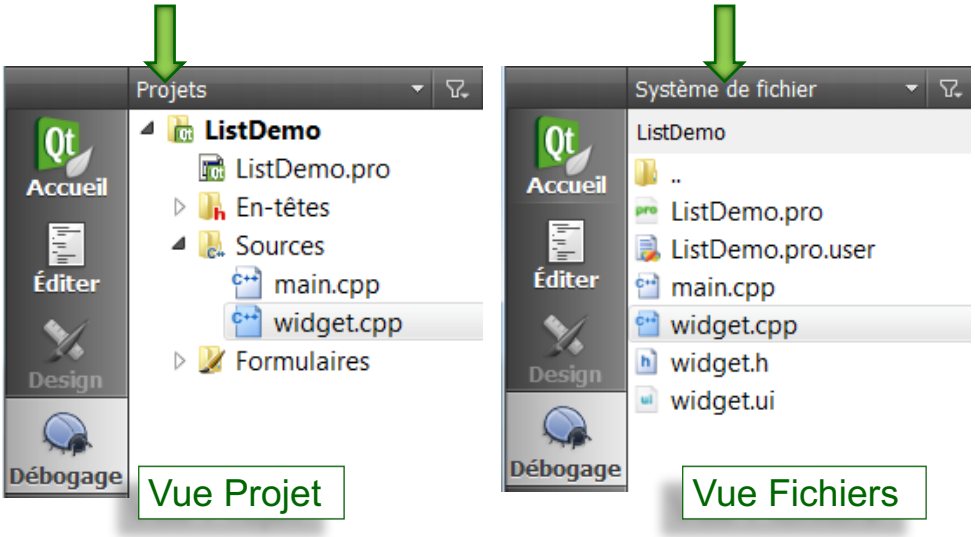


digia 22

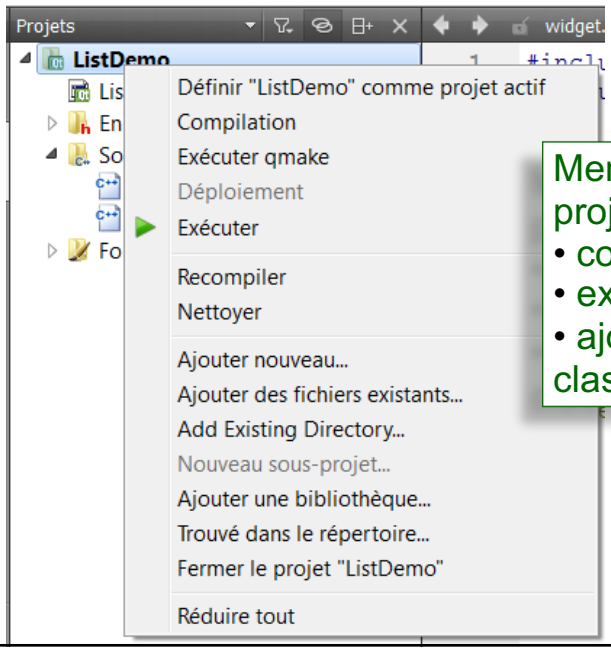
Projet : mode Debug



Gestion du projet

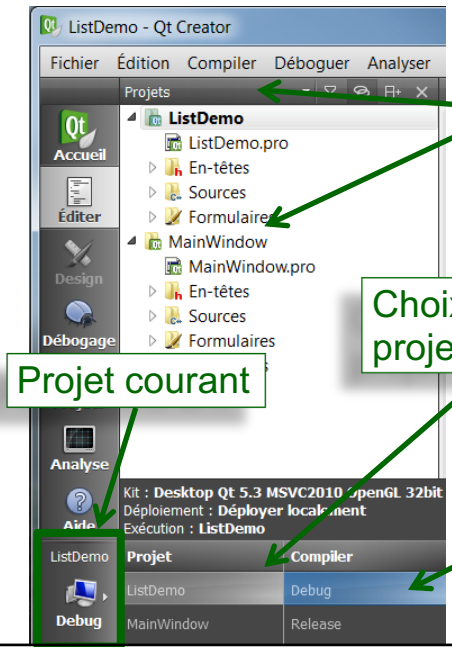


Gestion du projet



Menu contextuel du projet:
- compilations
- exécution
- ajouter de nouvelles classes/fichiers

Gestion du projet



2 projets ouverts

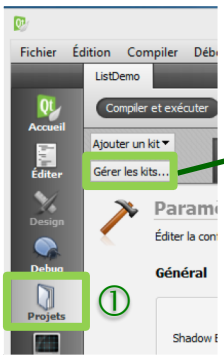
Choix du projet courant

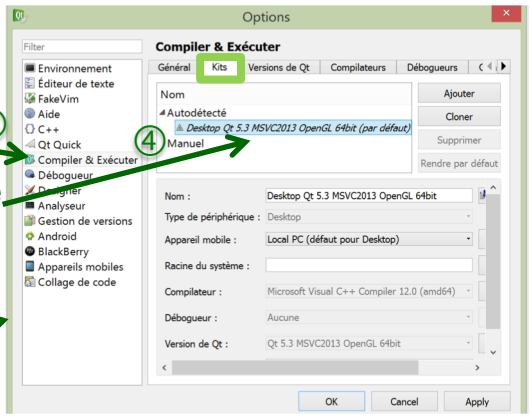
Projet courant

Choix du mode de compilation : debug ou release

Utiliser un kit de développement

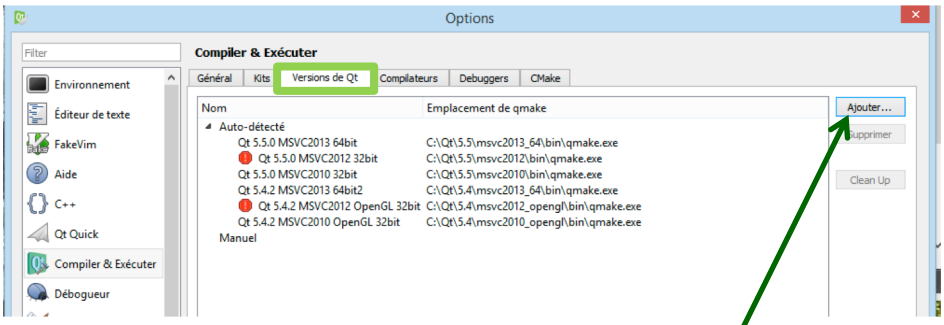
- Vérifier si détecté
  - Projets | Gérer les kits…
    - Compiler & Exécuter
  - Choisir le kit autodétecté





Si kit pas détecté

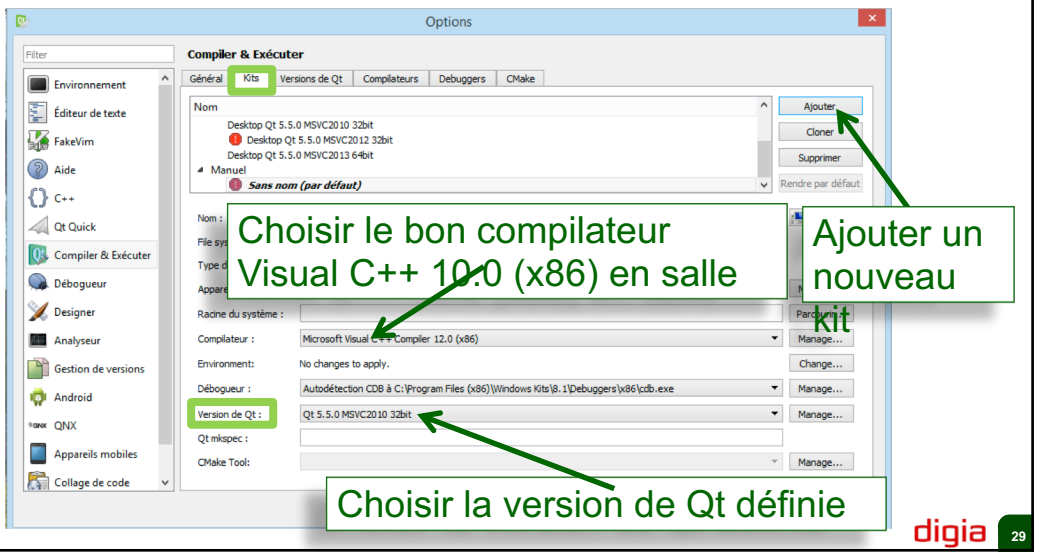
- Définir la version de Qt



En ajouter un si pas auto-détecté
A priori dans C:\Qt5.5\msvc2010\bin\qmake.exe

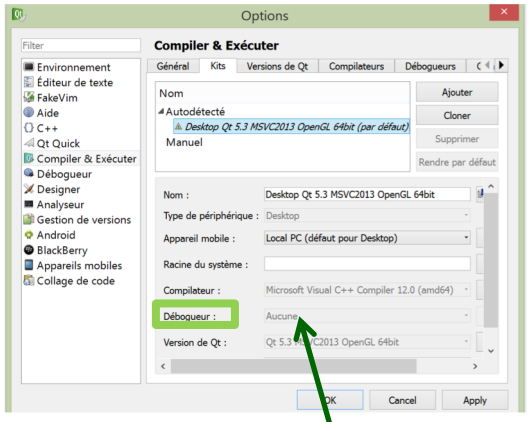
Si kit pas détecté

- Créer le kit à partir de la version de Qt définie



Utiliser débogueur Windows

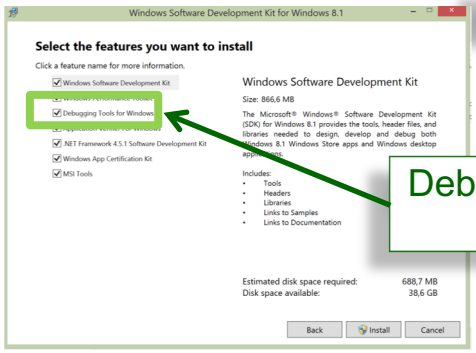
- Vérifier s'il est détecté



Utiliser débogueur Windows

- Si pas détecté : fichier cdb.exe doit être installé
  - Quitter QtCreator
  - Télécharger le kit de développement logiciel
    - Windows SDK pour votre version de Windows

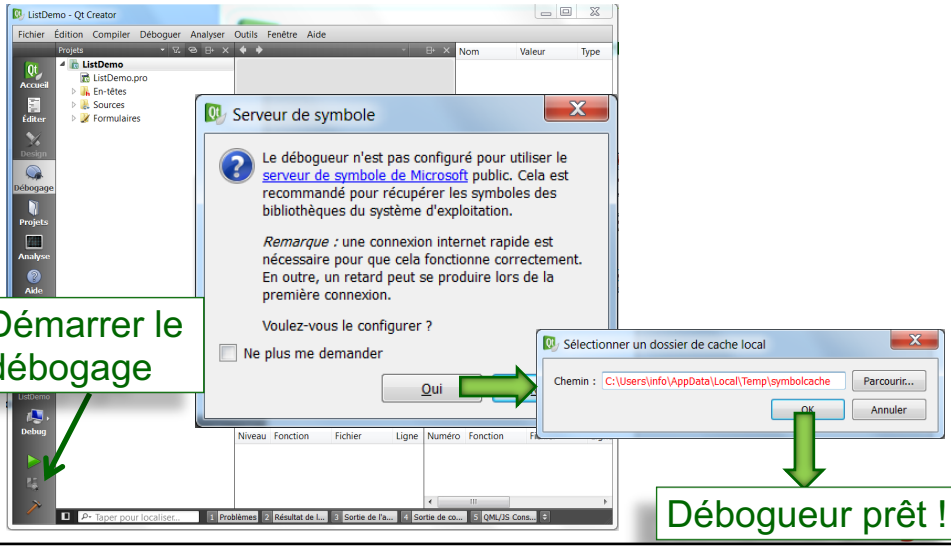
http://www.microsoft.com/fr-fr



Debugging Tools for Windows doit être coché

Configurer le débogueur

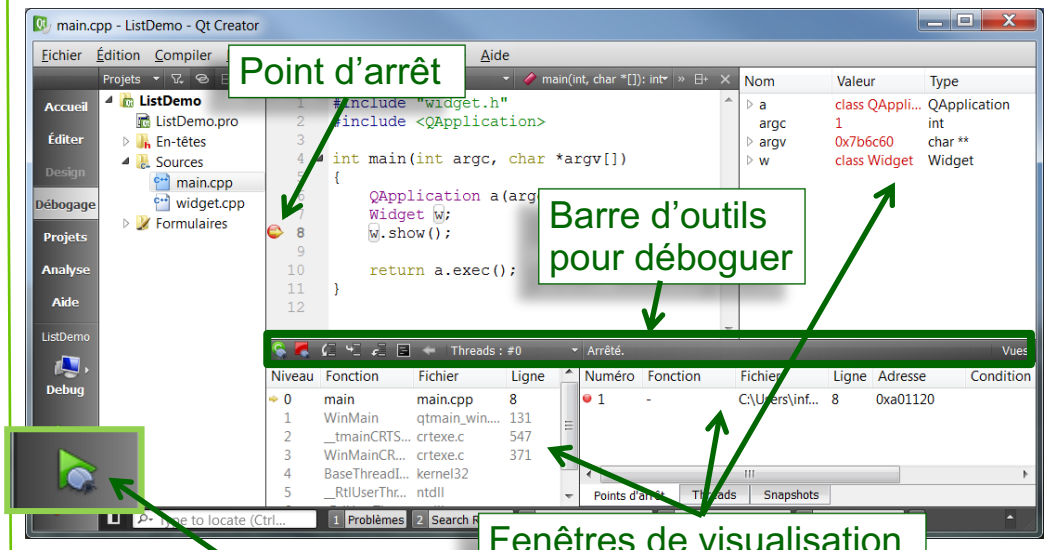
- Démarrer le débogage du projet



Démarrer le débogage

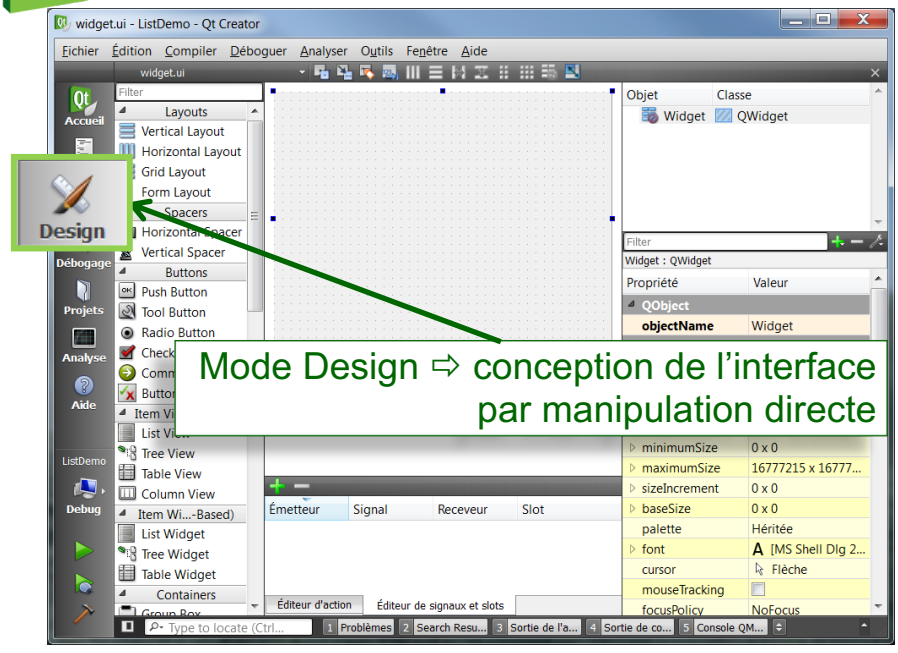
Débogueur prêt !

Projet : débogage



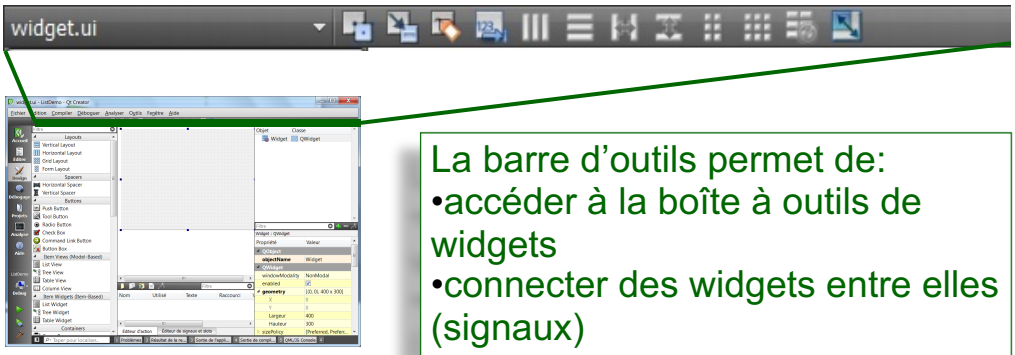
digia 33

Projet : mode Design



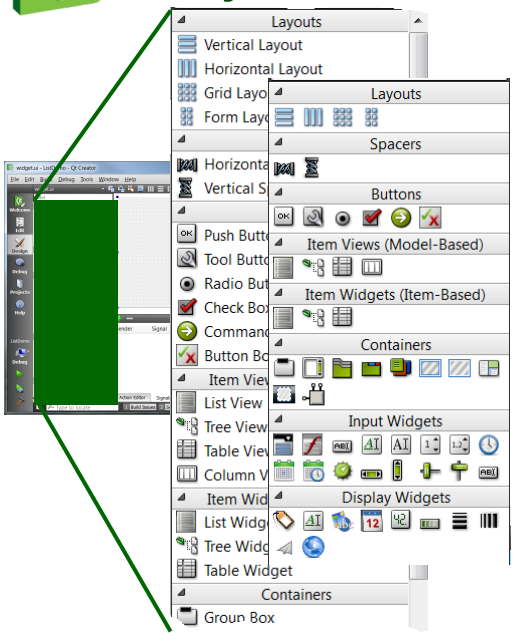
digia 34

Projet : mode Design

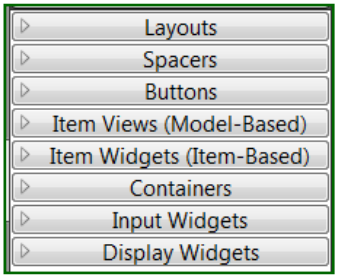


La barre d'outils permet de:
- accéder à la boîte à outils de widgets
- connecter des widgets entre elles (signaux)
- connecter des libellés amies (« buddies ») à d'autres widgets
- définir l'ordre de tabulation
- placer/aligner des widgets

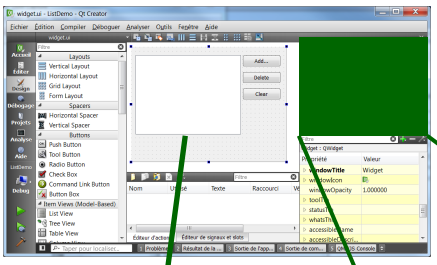
Projet : mode Design



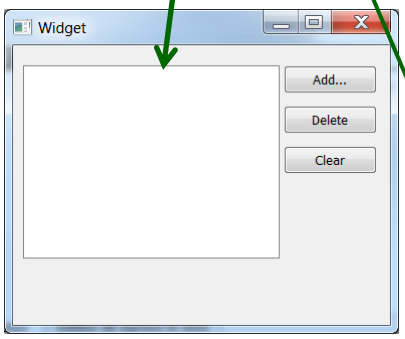
La boîte des widgets
- liste les widgets disponibles par catégorie
- 3 vues possibles (menu contextuel)



Projet : mode Design



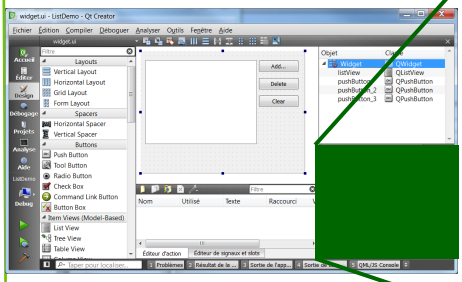
La vue arborescente ⇨hiérarchie des widgets sur le formulaire



Objet	Classe
Widget	QWidget
listWidget	QListWidget
pushButton	QPushButton
pushButton_2	QPushButton
pushButton_3	QPushButton

Projet : mode Design

La fenêtre des propriétés ⇨propriété du/des widgets sélectionnés sur le formulaire

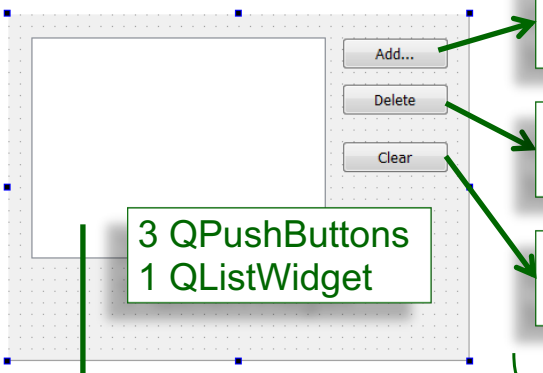


Filtre

Widget : QWidget

Propriété	Valeur
windowTitle	Widget
windowIcon	
windowOpacity	1.000000
toolTip	
statusTip	
whatsThis	
accessibleName	
accessibleDescri...	

Projet : mode Design



objectName = addButton
text = Add…

objectName = deleteButton
text = Delete

objectName = clearButton
text = Clear…

3 QPushButtons
1 QListWidget

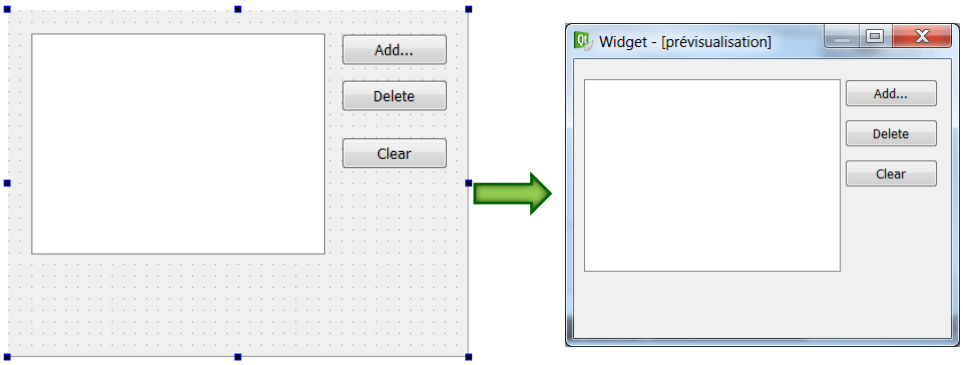
en utilisant…
- *la fenêtre des Propriétés*
- *Le menu contextuel du QPushButton*

selectionMode= MultiSelection

digia 39

Prévisualisation de l'interface

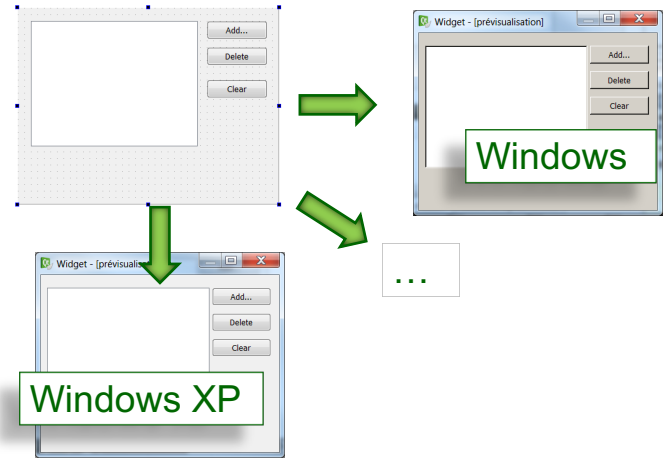
- Vue rapide de l'interface ⇨ pas de compilation
  - pas de réactions personnalisées sur les widgets
  - Outils|Form Editor|Prévisualisation…



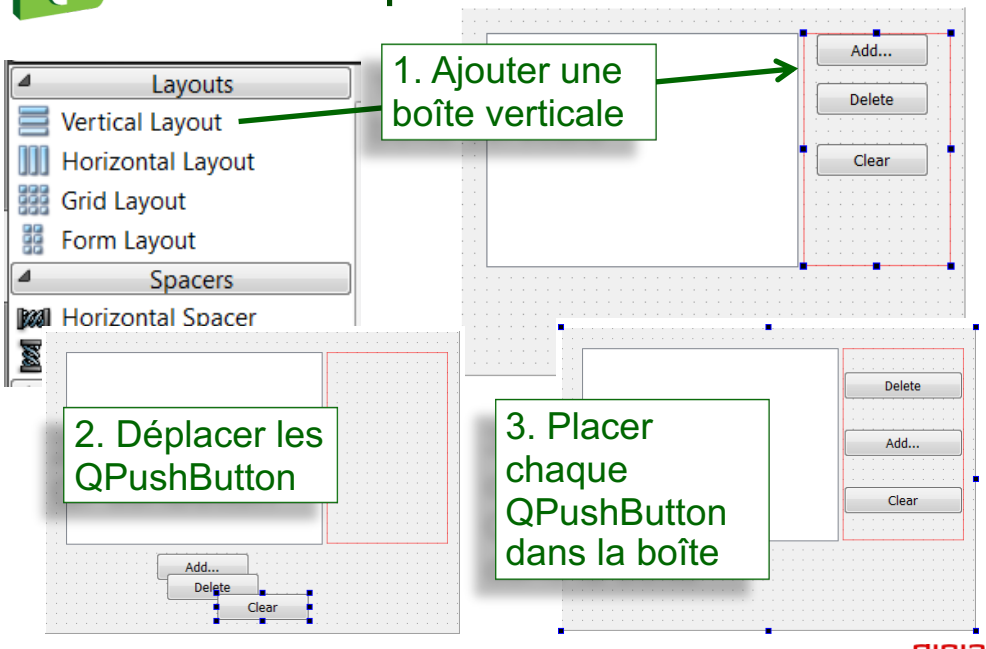
digia 40

Prévisualisation de l'interface

- Prévisualisation sous différents styles
  - Outils|Form Editor|Aperçus dans

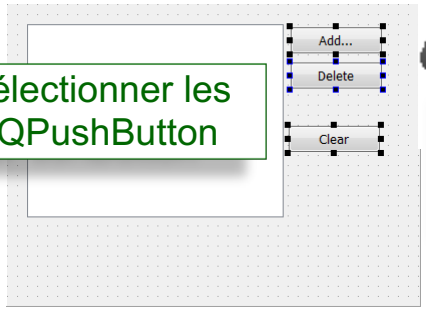


Outils de placement

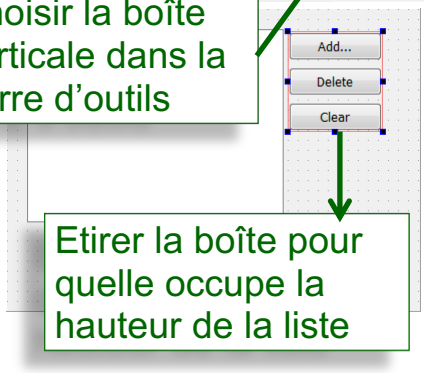


Outils de placement

Sélectionner les 3 QPushButton



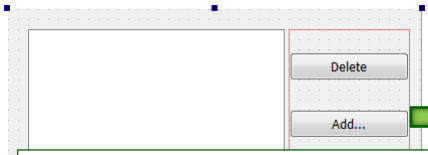
Choisir la boîte verticale dans la barre d'outils



Etirer la boîte pour quelle occupe la hauteur de la liste

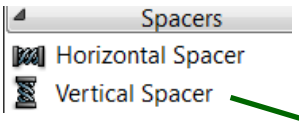
Autre solution…

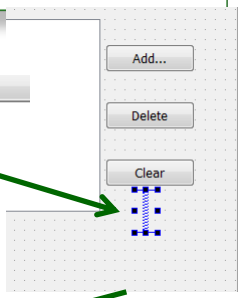
Outils de placement

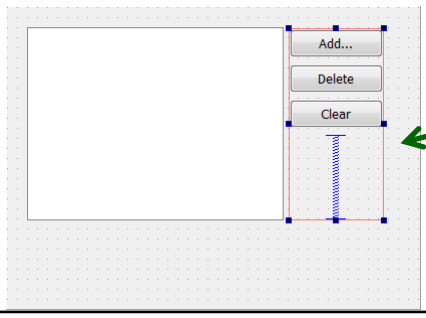


Supprimer la boîte avec son Menu contextuel : Mettre en page|Casser la mise en page

Ajouter un espace sous les boutons







Remettre une boîte verticale

Gestion des interactions

Clic sur bouton Clear ⇨ supprimer tous les éléments dans la liste

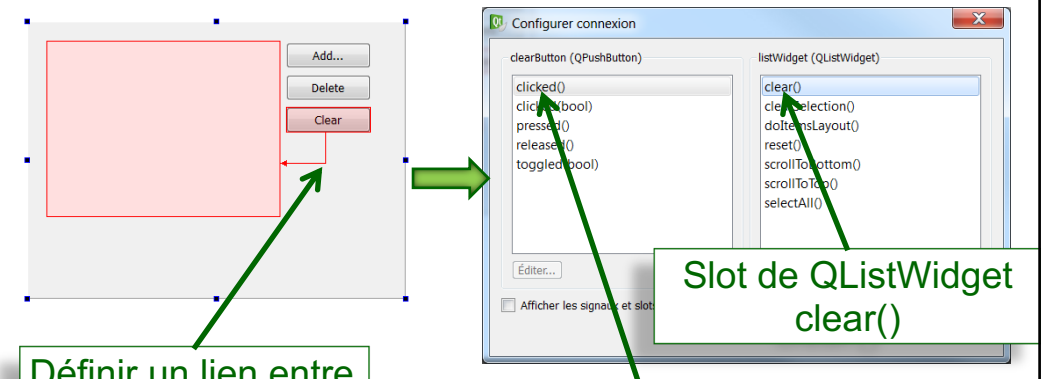
Signal ⇨ intervention

Slot ⇨ action

Cas slot prédéfini

Gestion des interactions

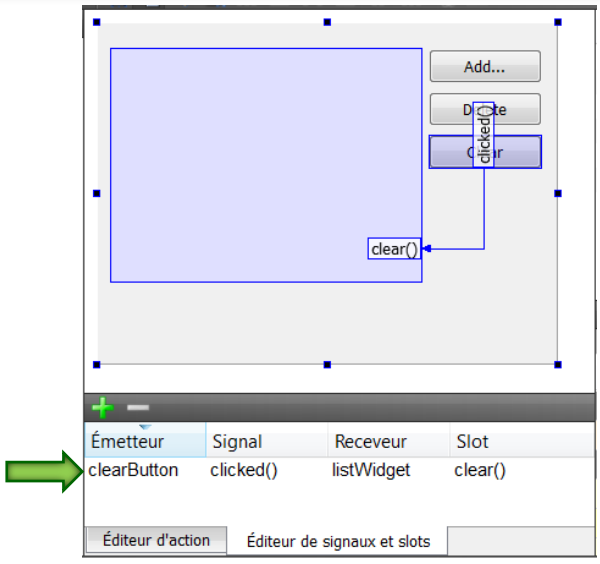
Clic sur bouton Clear ⇨ supprimer tous les éléments dans la liste



Connexion signal ⇨ slot

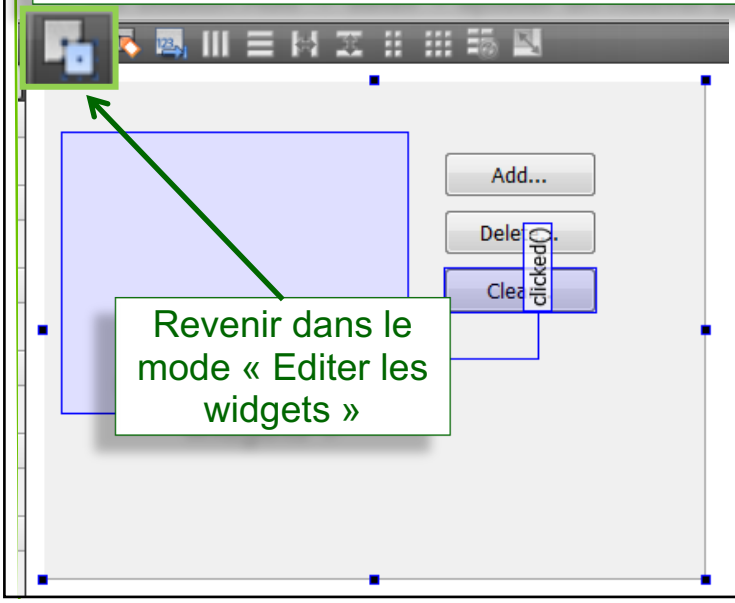
Gestion des interactions

Clic sur bouton Clear ⇨ supprimer tous les éléments dans la liste



Gestion des interactions

Clic sur bouton Add ⇨ saisir + ajouter un nouvel élément dans la liste

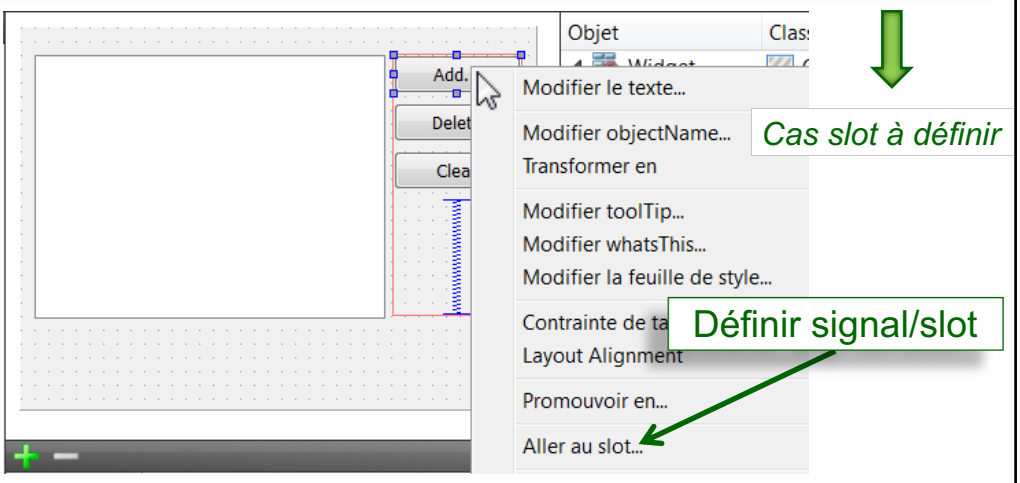


Revenir dans le mode « Editer les widgets »

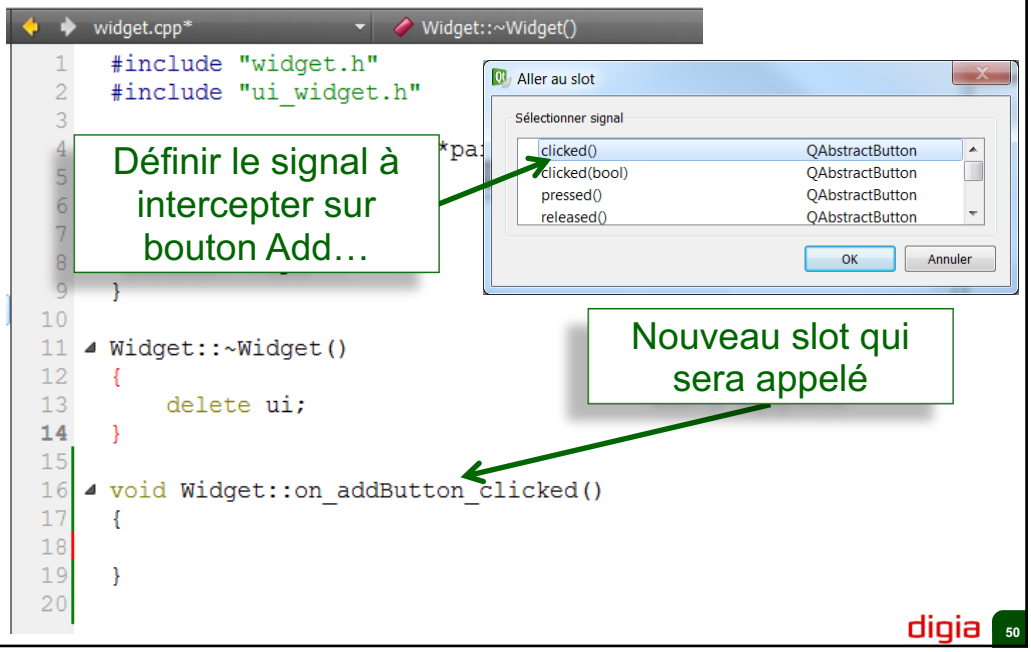
Cas slot à définir

Gestion des interactions

Clic sur bouton Add ⇨ saisir + ajouter un nouvel élément dans la liste

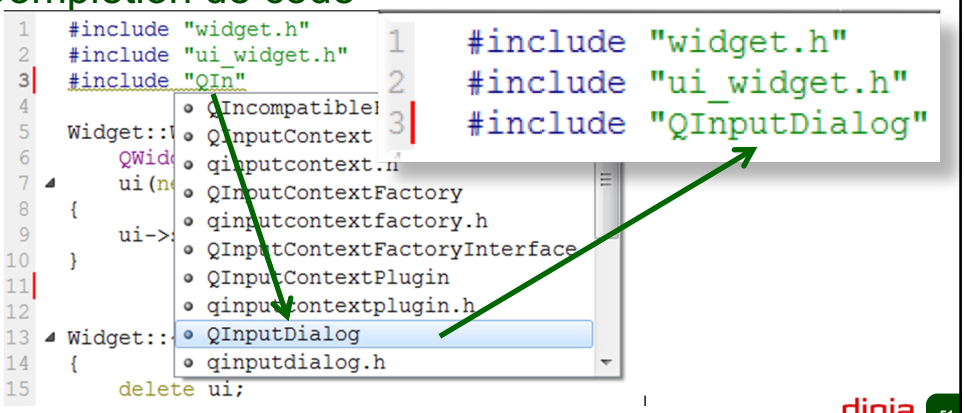


Gestion des interactions



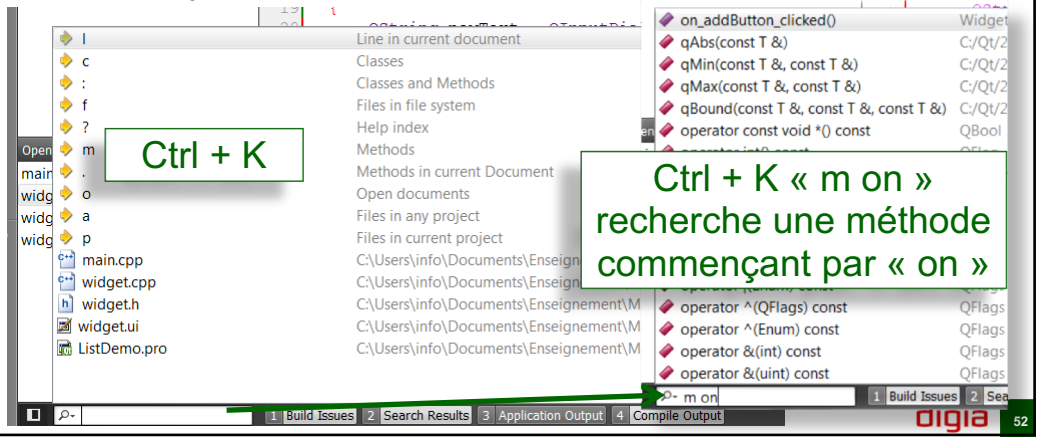
L'éditeur de code

- F4 : navigation fichiers entête/implémentation
- F2 : idem F4 avec localisation sur une méthode
- Complétion de code



L'éditeur de code

- Ctrl + K : recherche et localisation
  - Classes, méthodes, fichiers, ligne, etc.
  - Exemple : localiser le slot à définir



L'éditeur de code

```
void Widget::on_addButton_clicked()
{
    QString newText = QInputDialog::getText(this, "Enter text", "Text:");
        if( !newText.isEmpty() )
            ui->listWidget->addItem(newText);
}
```

Aide à la saisie des paramètres de la méthode

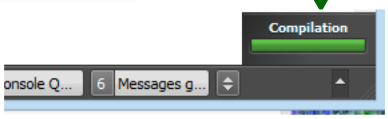
- Compiler, puis exécuter

Exécution (+ compilation si nécessaire)

Exécution en mode débogage

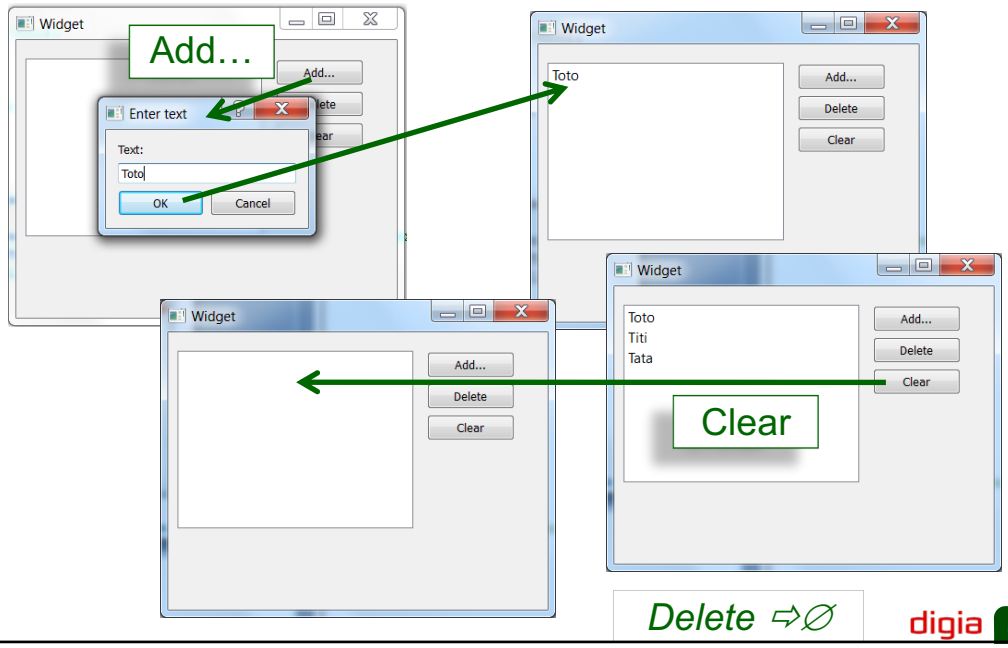
Compilation

Barre de progression pendant la compilation



digia 53

Exécution



Delete ⇨∅

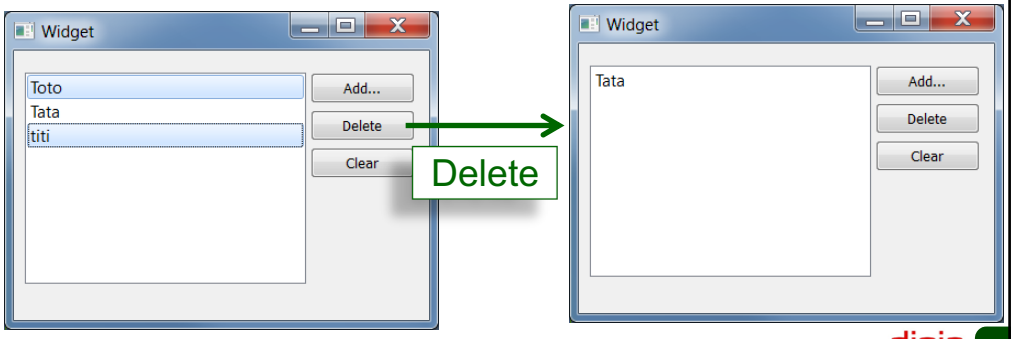
digia 54

Gestion du bouton delete

Clic sur bouton Delete ⇨ supprimer les éléments sélectionnés

Cas slot à définir

```
void Widget::on_deleteButton_clicked()
{
    foreach (QListWidgetItem *item, ui->listWidget->selectedItems())
        delete item;
}
```



Gestion du bouton delete

Bouton Delete ⇨ accessible si au moins un élément sélectionné dans la liste

Cas slot défini par programme

```
class Widget : public QWidget
{
    Q_OBJECT

public:
    explicit Widget(QWidget *parent = 0);
    ~Widget();

private:
    Ui::Widget *ui;

private slots:
    void on_deleteButton_clicked();
    void on_addButton_clicked();
    void updateDeleteEnabled();
};
```

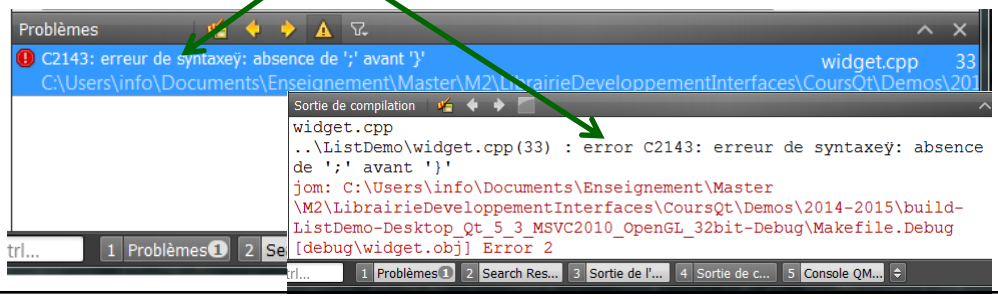
Ajout d'un nouveau slot

Gestion du bouton delete

Définition de l'action du slot ⇨ définir l'accessibilité au bouton Delete

```
void Widget::updateDeleteEnabled()
{
    ui->deleteButton->setEnabled(ui->listWidget->selectedItems().count()!=0)
}
```

Compilation ⇨ erreur



57

Gestion du bouton delete

```
Widget::Widget(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::Widget
{
    ui->setupUi(this);
    connect(ui->listWidget->selectionModel(),
                SIGNAL(selectionChanged(QItemSelection,QItemSelection)),
            this, SLOT(updateDeleteEnabled()));
}
```

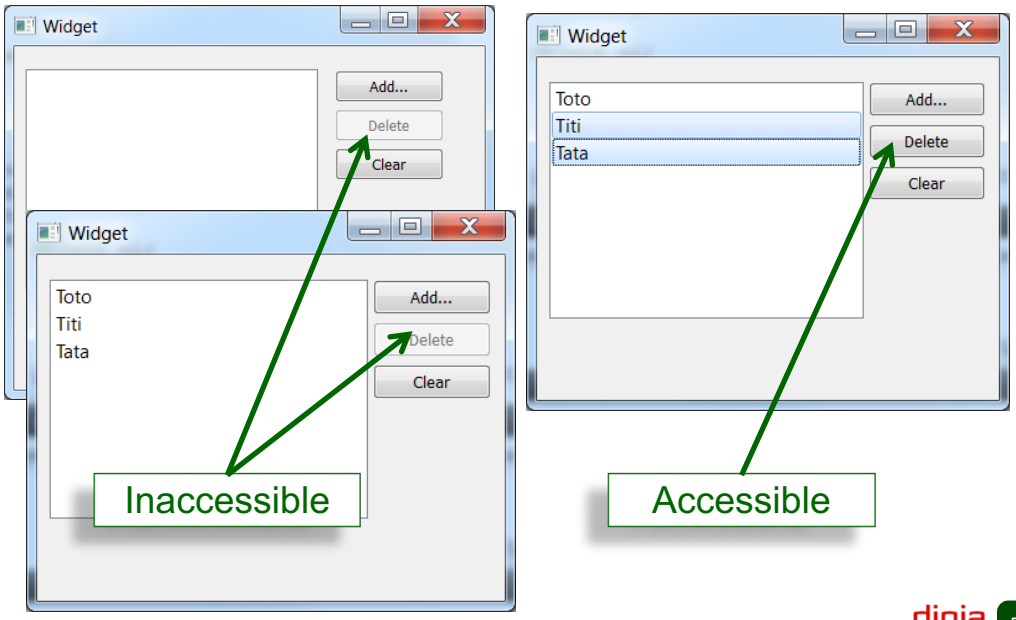
Connexion signal/slot dans le constructeur

Problème : le bouton n'est pas inaccessible au départ ⇨ appeler le slot après la connexion

```
Widget::Widget(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::Widget)
{
    ui->setupUi(this);
    connect(ui->listWidget->selectionModel(),
            SIGNAL(selectionChanged(QItemSelection,QItemSelection)),
            this, SLOT(updateDeleteEnabled()));
    this->updateDeleteEnabled();
}
```

58

Gestion du bouton delete

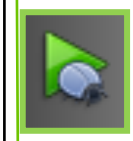


digia 59

Déboguer le programme

```
20
21  void Widget::on_addButton_clicked()
22  {
23      QString newText = QInputDialog::getText(this, "Enter text", "Text:");
24      if (!newText.isEmpty())
25          ui->listWidget->addItem(newText);
26  }
```

Point d'arrêt



Lancer le débogueur