Premier programme KDE

Par

Date de publication : 5 mai 2011

Dernière mise à jour : 1 septembre 2011

Votre premier programme se doit de saluer le monde avec un amical "Hello World", n'est-ce pas ? Pour cela, on va utiliser une KMessageBox et personnaliser un de ses boutons.

Commentez

I - Résumé..... 3
II - Le code..... 3
III - Remerciements..... 4

I - Résumé

Pour obtenir plus d'information à propos d'une classe, Konqueror propose un raccourci. Par exemple, pour obtenir des informations à propos de KMessageBox, on tape kde:kmessagebox dans Konqueror pour être dirigé vers la documentation.

On peut utiliser KDevelop pour ses projets. Ce programme propose des fonctionnalités très utiles comme la complétion de code, un accès facile à la documentation de l'API ou le support du débogage. Cependant, il faudra toujours éditer les fichiers CMake manuellement.

II - Le code

Tout le code nécessaire se trouvera dans un unique fichier, main.cpp. Voici son contenu :

```
#include <KApplication>
#include <KAboutData>
#include <KCmdLineArgs>
#include <KMessageBox>
#include <KLocale>

int main (int argc, char *argv[])
{
    KAboutData aboutData(
                         // Le nom du programme utilisé en interne.
                         "tutorial1",
                         // Le nom du catalogue de message.
                         // Si nul, le nom du programme est utilisé à la place.
                         0,
                         // La chaîne de caractères contenant le nom du programme à afficher.
                         ki18n("Tutorial 1"),
                         // La chaîne contenant la version du programme.
                         "1.0",
                         // Une courte description de ce que l'application fait.
                         ki18n("Displays a KMessageBox popup"),
                         // La licence sous laquelle le code est distribué.
                         KAboutData::License_GPL,
                         // Copyright
                         ki18n("(c) 2007"),
                         // Un texte optionnel affiché dans la boîte À propos.
                         // Il peut contenir toutes informations désirées.
                         ki18n("Some text..."),
                         // La page web du programme.
                         "http://example.com/",
                         // L'adresse courriel à utiliser pour rapporter des bogues.
                         "submit@bugs.kde.org");
```

```
    KCmdLineArgs::init( argc, argv, &aboutData );
    KApplication app;
    KGuiItem yesButton( i18n( "Hello" ), QString(),
                        i18n( "This is a tooltip" ),
                        i18n( "This is a WhatsThis help text." ) );
    KMessageBox::questionYesNo( 0, i18n( "Hello World" ),
                                i18n( "Hello" ), yesButton );
    return 0;
}
```

La première partie de code spécifique à KDE dans ce programme est KAboutData. C'est la classe qui est utilisée afin de mémoriser des informations à propos du programme telles qu'une courte description, qui en sont les auteurs, ainsi que sous quelle licence il est distribué. Quasiment toutes les applications KDE devraient utiliser cette classe.

Ensuite, KCmdLineArgs. C'est la classe qui sera utilisée pour spécifier des paramètres de ligne de commande pour, par exemple, lancer le programme avec un fichier particulier. Cependant, dans ce tutoriel, on l'initialise simplement avec l'objet KAboutData, juste créé, afin de supporter les paramètres --version et --author.

On crée alors un objet KApplication. Cela doit être fait une et une seule fois dans chaque programme, car cela est nécessaire pour des choses telles que la gestion de l'internationalisation.

On a désormais effectué toutes les initialisations nécessaires à KDE, on peut donc commencer à faire des choses intéressantes avec l'application. On va créer une boîte de dialogue mais en personnalisant un de ses boutons. Pour cela, on utilise un objet KGuiItem. Le premier argument du constructeur de cette classe est le texte qui apparaîtra dans l'élément (dans notre cas, un bouton). On a alors la possibilité de définir une icône, mais on n'en veut pas, on utilise alors simplement QString(). Puis, on définit l'infobulle (le texte qui apparaît lorsque l'on passe au-dessus d'un élément) et enfin le texte "Qu'est-ce que c'est ?" (auquel on accède à partir d'un clic droit ou avec Shift-F1).

Maintenant, on peut créer la boîte de dialogue en appelant la fonction KMessageBox::questionYesNo(), qui, par défaut, crée une boîte de message avec un bouton "Oui" et un bouton "Non". Le deuxième argument est le texte qui apparaîtra au-dessus des boutons dans la boîte de message. Le troisième est l'intitulé de la fenêtre et enfin le dernier argument définit le KGuiItem pour le bouton "Oui", c'est-à-dire le KGuiItem yesButton juste créé.

Tous les textes visibles par l'utilisateur sont passés par la fonction i18n(), ceci est nécessaire afin que l'interface graphique soit traduisible.

En ce qui concerne le code, tout est bon. Il ne reste plus qu'à le compiler et à l'essayer.

III - Remerciements

Merci à **Thibaut Cuvelier**, **Claude Leloup** et **Maxime Gault** pour leur relecture !