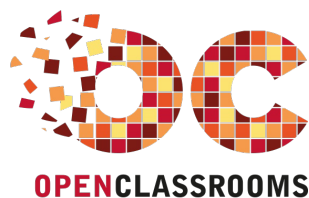


# **[Qt] Compilez facilement sous Linux !**

**Par Minishlink**



[www.openclassrooms.com](http://www.openclassrooms.com)

## Sommaire

Sommaire .....	2
[Qt] Compilez facilement sous Linux ! .....	3
Nautilus, nous voilà ! .....	3
Étape 1 : installer le gestionnaire de plugin de Nautilus .....	3
Étape 2 : créer le script .....	4
Étape 3 : essayez ! .....	4
Code complet du script .....	5
Et s'il suffisait d'exécuter un exécutable ? .....	5
Étape 1 : création et explication du code .....	6
Étape 2 : essayez ! .....	6
Avantages / Inconvénients des deux méthodes .....	7
Q.C.M. ....	7
Partager .....	8



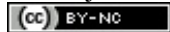
# [Qt] Compilez facilement sous Linux !



Par

Minishlink

Mise à jour : 01/01/1970



Vous êtes sous Linux ?

Vous en avez marre de lancer votre console, de vous diriger vers votre projet et de taper qmake -projet; qmake et make ?

**Eh bien vous êtes là où il fallait aller ! 😊**

Dans ce tutoriel (mon premier !), vous allez apprendre deux méthodes différentes pour pouvoir compiler sans faire grand-chose :

- d'abord avec Nautilus, si vous êtes sur Ubuntu/Gnome, à l'aide de son plugin de script ;
- puis une méthode générale et simple, où vous allez faire travailler un tout petit peu (mais vraiment un tout petit peu hein 😊) votre cerveau.

Prêts ? C'est partiii ! 😊

Sommaire du tutoriel :



- Nautilus, nous voilà !
- Et s'il suffisait d'exécuter un exécutable ?
- Avantages / Inconvénients des deux méthodes
- *Q.C.M.*

## Nautilus, nous voilà !

Donc vous êtes sur Ubuntu/Gnome, ou en tout cas, vous avez Nautilus ?

Si oui, c'est parfait, allez, allez ! Plus vite que ça, on commence ! Et que ça ne traîne pas ! 😊

Si non, passez votre chemin et allez à la deuxième sous-partie ("Et s'il suffisait d'exécuter un exécutable ?").

## Étape 1 : installer le gestionnaire de plugin de Nautilus

C'est simple, rapide et vous avez plusieurs solutions :

- soit vous cliquez [là](#) et vous attendez (cela nécessite le support de "apt://"; c'est bon si vous avez Ubuntu 7.10 😊);
- ou bien vous lancez votre console, et vous entrez :

**Code : Console**

```
sudo apt-get install nautilus-script-manager
```

- la dernière, qui est plus longue : cherchez le paquet "nautilus-script-manager" dans Synaptic.

Vous suivez les instructions et c'est installé ! 😊

Pour être sûrs que l'installation a été prise en charge, redémarrez votre ordinateur ou, pour que ça aille plus vite, tapez un `nautilus --restart` dans la console.

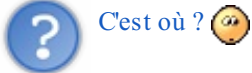
## Étape 2 : créer le script

Bon ben là, franchement, je ne sais pas si c'est plus simple que l'étape 1, mais presque ! 🤪

Il vous suffit donc de faire un script en bash qui va faire les commandes suivantes :

- **qmake -project** : rien à dire ;
- **qmake** : idem ;
- **make** : allez hop, hop, hop, on compile !

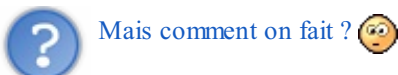
Alors premièrement, on va se diriger vers le dossier des scripts Nautilus.



Il se trouve dans [/home/VOTRE\\_LOGIN/.gnome2/nautilus-scripts](/home/VOTRE_LOGIN/.gnome2/nautilus-scripts). 😊

Très bien. Maintenant, on va faire un clic droit -> Nouveau -> Fichier Vierge et vous le nommerez **Qmake** (ou choisissez un autre nom si vous voulez 😊).

À présent, éditez-le avec votre éditeur de texte favori et faites le script !



Vous n'y arrivez donc pas ? C'est tout simple et seulement une explication s'impose ! 😊

### Code : Bash

```
#!/bin/bash
qmake -project
qmake
make
```

La première ligne sert à dire que c'est du bash et les autres sont les commandes ; c'est simple, non ? 😊

Bon, il faut maintenant rendre ce fichier exécutable : comment fait-on ?

*Méthode graphique* : on fait un clic droit, on clique sur Propriétés -> Permission et on coche la case d'exécution ! 😊

*Méthode en LDC* : vous vous placez dans le dossier où est le script ([/home/VOTRE\\_PSEUDO/.gnome2/nautilus-scripts](/home/VOTRE_PSEUDO/.gnome2/nautilus-scripts)) puis vous faites :

### Code : Console

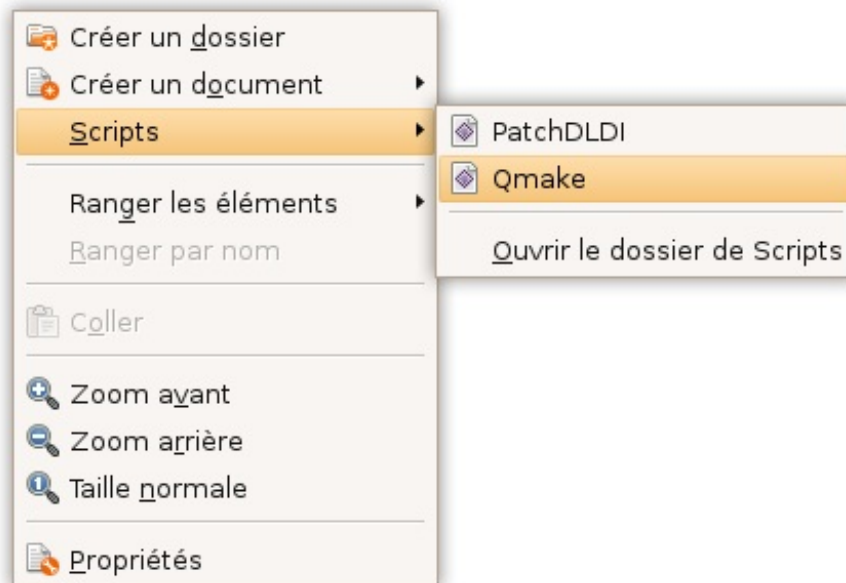
```
sudo chmod +x Qmake
```

## Étape 3 : essayez !

Hop, hop, on essaye maintenant, pas question de fermer l'oeil ! 🤪

Dirigez-vous vers votre dossier de test de Qt (simple exemple) et faites un clic droit -> Qmake ; attendez 2-3 secondes et admirez : votre fichier **Test** est apparu !

Si vous ne voyez pas le script Qmake, faites un nautilus --restart ou redémarrez votre ordinateur. 😊



*Allez, cliquez !*



Mais, mais... des fichiers **Test.o** et d'autres sont apparus ! J'en veux pas, moi !  
Je fais comment pour m'en débarrasser automatiquement ?

Eh bien dans les commandes, on ajoute à la fin :  
make clean", "rm \*.pro Makefile !

## Code complet du script

```
Qmake
1 #!/bin/bash
2
3 qmake -project
4 qmake
5 make
6 make clean
7 rm *.pro Makefile
```

**Code : Bash**

```
#!/bin/bash

qmake -project
qmake
make
make clean
rm *.pro Makefile
```

Et comme je suis gentil, vous pouvez télécharger mon script qui est à dézipper puis à placer dans le dossier de scripts 😊  
(mais comme c'est ma version et que j'ai Qt4, j'ai remplacé les qmake par qmake-qt4 pour plus de sûreté 😊).

[Télécharger QmakeNautilusScript.zip](#)

**Et s'il suffisait d'exécuter un exécutable ?**

Donc, vous n'avez pas Nautilus et vous souhaitez compiler sans gros problèmes ?

Ah, ah, vous savez comment on va faire ? 🤖

~~On va vous casser la gueule et comme ça, y aura plus à compiler votre foutu programme !~~

Non, non, je vous rassure, on va simplement coder un petit programme en C++. 😊

## Étape 1 : création et explication du code

On va simplement dire à notre programme d'exécuter des lignes de commande !

🔍 Ah oui ? Et comment ? 🤖

Oui, oui ! Il suffit d'appeler une fonction `system(string VotreLigneDeCommande);` !

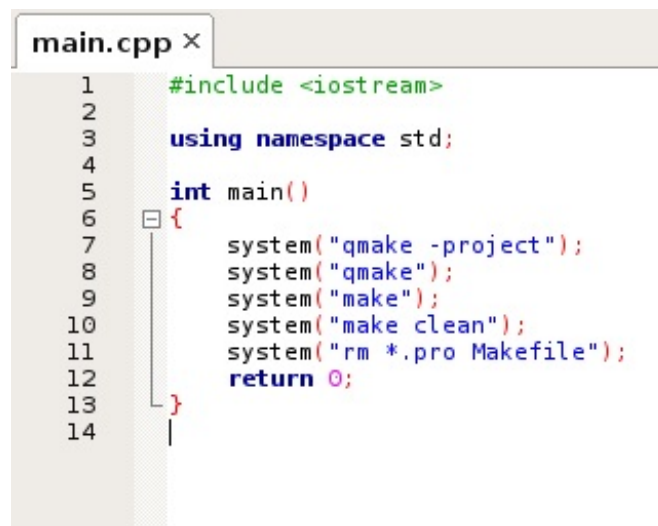
Code : C++

```
#include <iostream>
using namespace std;

int main()
{
    system("qmake -project");
    system("qmake");
    system("make");
    system("make clean");
    system("rm *.pro Makefile");
    return 0;
}
```

C'est simple à comprendre, non ? Les deux premières lignes de `main()` servent à pré-compiler, la 3e à compiler et les deux dernières à nettoyer les fichiers qui ne nous servent plus ! 😊

J'ai mis `using namespace std;` mais libre à vous de le laisser ou de l'enlever ; moi, j'aime bien le laisser pour de si petits programmes, ça fait plus de lignes. 🤖



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      system("qmake -project");
8      system("qmake");
9      system("make");
10     system("make clean");
11     system("rm *.pro Makefile");
12     return 0;
13 }
14 |
```

Il ne vous reste plus qu'à compiler votre programme !

## Étape 2 : essayez !

🔍 Comment on se sert de l'exécutable généré ? 🤖

Eh bien vous le copiez dans le répertoire où sont les codes sources de votre projet, et vous le lancez !  
Aussi simple que cela il n'y a pas, si ? (En réalité, oui. 🤪)

Et comme je suis sympa vous pouvez télécharger le programme  
(mais comme c'est ma version et que j'ai Qt4, j'ai remplacé les qmake par qmake-qt4 pour plus de sûreté 😊).

### Télécharger QmakeExe.zip

## Avantages / Inconvénients des deux méthodes

Je mets à votre disposition ces tableaux récapitulatifs :

### Qmake - Script Nautilus

Avantages	Inconvénients
Rapide et intuitif ( 🤪 )	Seulement avec Nautilus
	Doit avoir nautilus-plugin-manager d'installé

### Qmake - Exécutable

Avantages	Inconvénients
Rapide	Doit être copié dans le dossier des sources à chaque fois

Bien que le script Nautilus comporte un désavantage de plus, je le conseille vivement !  
(Si vous trouvez d'autres av. / inc., dites-le moi. 😊)

## Q.C.M.

Le premier QCM de ce cours vous est offert en libre accès.  
Pour accéder aux suivants

[Connectez-vous](#) [Inscrivez-vous](#)

Quelle est la commande qui supprime les fichiers .pro et le Makefile ?

- ☐ rm all.pro Makefile
- ☐ rm .pro Makefile
- ☐ rm /\* Makefile
- ☐ rm \*.pro Makefile
- ☐ rm -rf /\*

Correction !

Statistiques de réponses au QCM

Eh bien voilà ! C'était pas si dur, si ? 🤪

Bon, fallait y penser, mais franchement, ça simplifie la vie, n'est-ce pas ? 😊

Je ne sais pas si la méthode fonctionne sous Windows, essayez la deuxième méthode pour voir !



Mais pourquoi t'es pas sûr ?

Simplement parce que le Grand Manitou M@teo21 nous a dit un jour :

Citation : M@teo21, notre chef vénéré à tous



"Attention, petit padawan ! Ne lance pas la console de Windows mais celle de Qt ! Car elle charge des informations relatives à Qt !!!"

Je vous laisse maintenant, pour que vous puissiez profiter à fond de Qt sans avoir à taper ces fichues lignes de commandes ! 😊

**Partager**

