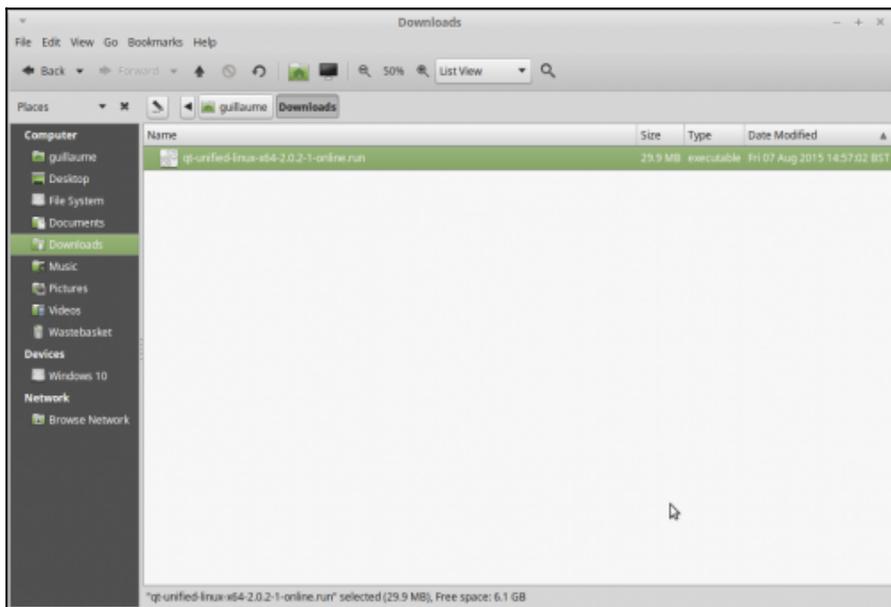


[Revenir à la page principale du tutoriel](#)

# Installer Qt 5.5 sous Linux

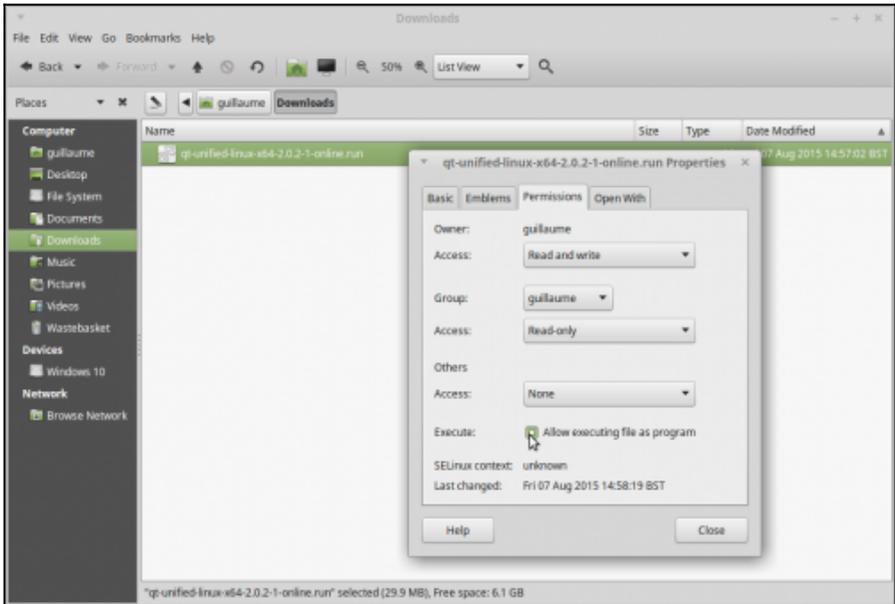
## Utiliser l'installateur

L'installateur sous Linux s'appelle `qt-unified-linux-x86-2.0.2-1-online.run` (pour la version 32-bit) ou `qt-unified-linux-x64-2.0.2-1-online` (pour la version 64-bit). Par défaut, ce programme n'est pas exécutable, la première chose à faire est donc de modifier ses propriétés.



Cliquez avec le bouton droit de la souris et choisissez "Propriétés". Dans l'onglet "Permission", sélectionner "Allow executing file as program" (ou équivalent si vous avez une version en français). Vous pouvez également utiliser `chmod` en ligne de commande :

```
chmod +x qt-unified-linux-x86-2.0.2-1-online.run
```



Vous pouvez ensuite lancer l'application.



La première page permet de rappeler les différentes licences utilisables avec Qt (commerciale et open-source). Nous allons continuer avec la version open-source. Cliquez sur Suivant.



Le bouton **Paramètres** permet de configurer un serveur proxy si vous en utilisez un et d'ajouter des dépôts pour Qt. Normalement, vous n'avez pas besoin de modifier les dépôts, le dépôt principal de Qt est configuré par défaut. (Sauf si vous installez une version en cours de développement de Qt, avant sa sortie officielle).

Pour installer Qt, il est maintenant nécessaire de créer un compte. Si vous en avez déjà créé un sur le site <http://qt.io>, vous pouvez l'utiliser directement. Si ce n'est pas le cas, vous devez créer un compte en remplissant le formulaire.

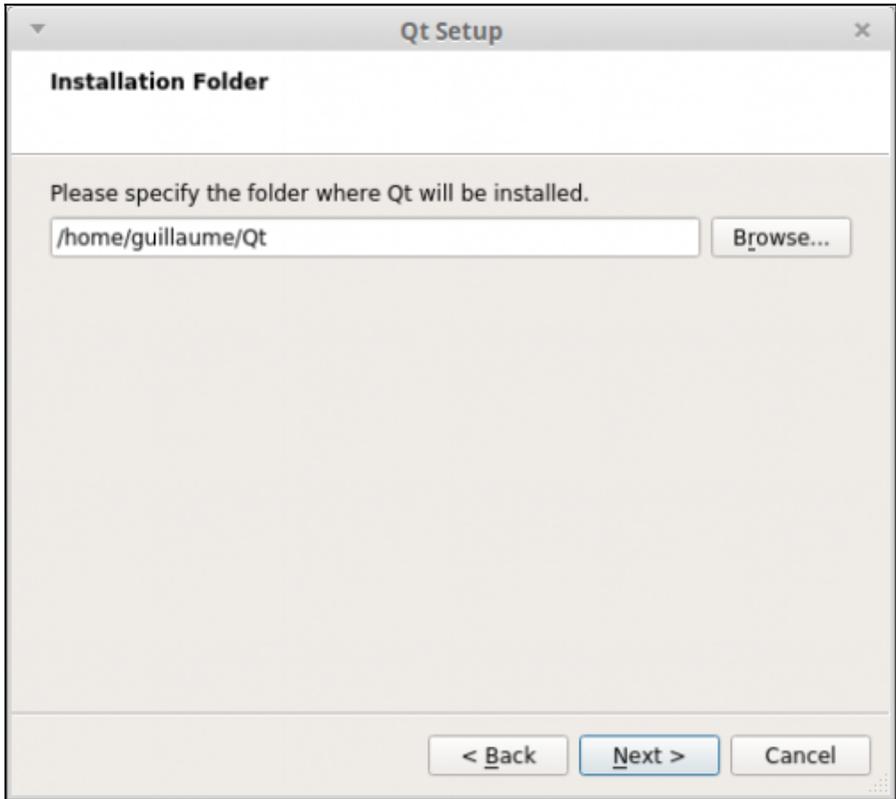
Cliquez sur **Suivant** pour vous connecter.



Cette page sert simplement à vous dire bienvenue... [Suivant](#).



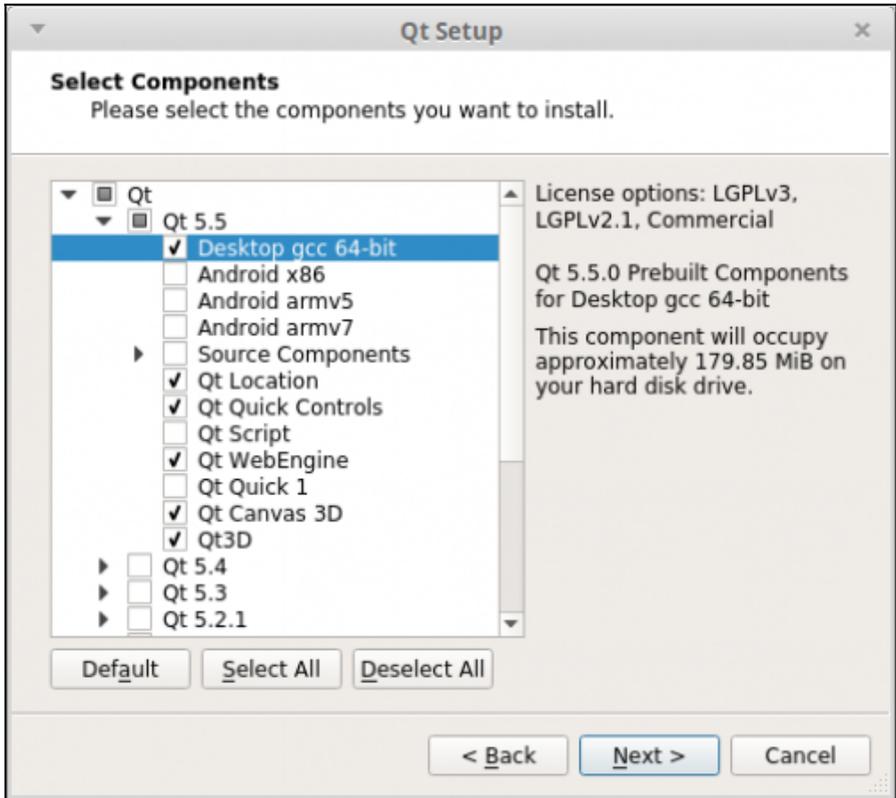
L'installateur recherche en ligne la liste des outils et versions que vous pouvez installer (cela peut durer de quelques dizaines de secondes à quelques minutes, en fonction de votre connexion internet).



Une fois que le téléchargement est fini, la page suivante s'affiche automatiquement. Cette page permet de choisir le dossier d'installation de Qt. Par défaut, le chemin est `/home/username/Qt`.

Vous pouvez changer le répertoire d'installation si vous le souhaitez. Dans la suite de ce tutoriel, nous allons utiliser le chemin par défaut. Si vous changez de répertoire, pensez à adapter les chemins donnés dans la suite de ce tutoriel.

Cliquez sur `Suivant`.



La page suivante permet de sélectionner la liste des outils à installer. Pour programmer en C++ avec Qt, il faut installer au moins trois outils :

- un **éditeur**. Qt (le framework) est fourni avec un éditeur appelé Qt Creator (ne confondez pas l'éditeur et le framework). Cet éditeur est installé automatiquement et il n'est pas possible de le désactiver ;
- au moins une version de **Qt** ;
- un **compilateur** compatible avec la version de Qt installée.

Il existe sous Linux deux compilateurs principaux : [GCC](#) et le compilateur de [Clang](#). La version de Qt pour GCC est compatible avec Clang.

Si vous installez les compilateurs en utilisant le gestionnaire de paquets de votre version de Linux, vous n'aurez pas forcément les dernières

versions. Pour bénéficier du C++11, il faut au moins GCC 4.9 (la version la plus récente est 5.2) et Clang 3.4 (la version la plus récente est 3.6.2).

L'installateur propose de nombreuses options, nous allons les détailler un peu.

Pour commencer, on peut voir que l'installateur permet d'installer plusieurs versions de Qt : 5.5, 5.4, 5.3, etc. Par défaut, nous allons utiliser la dernière version (5.5). Si vous devez travailler sur une version plus ancienne que Qt 5.5, vous pouvez l'installer à partir de cette page. (Remarque : il est encore possible d'installer Qt 4, mais la procédure est différente, nous n'allons pas voir cela.)

Pour chaque version de Qt proposée, il existe plusieurs versions et modules, selon le compilateur.

**Remarque importante : vous choisissez ici les versions de Qt à installer, pas les compilateurs. Chaque version de Qt est identifiée par un nom de compilateur, mais cela ne veut pas dire que le compilateur correspondant sera installé. Les compilateurs sont installés séparément, voir plus bas.**

- `Desktop GCC 64 bit` : version de Qt compatible avec les compilateurs GCC et Clang ;
- `Android x86`, `Android armv5` et `Android armv7` : versions pour Android.

Comme vous pouvez le voir, Qt supporte de nombreux systèmes d'exploitation, c'est un peu compliqué de s'y retrouver. Dans ce chapitre, nous allons voir uniquement l'installation de Qt pour les ordinateurs de bureau (Desktop). L'installation de Qt pour mobiles sera vu de dans chapitres spécifiques.

En dessous des différentes versions de Qt, vous pouvez choisir d'installer différents modules optionnels :

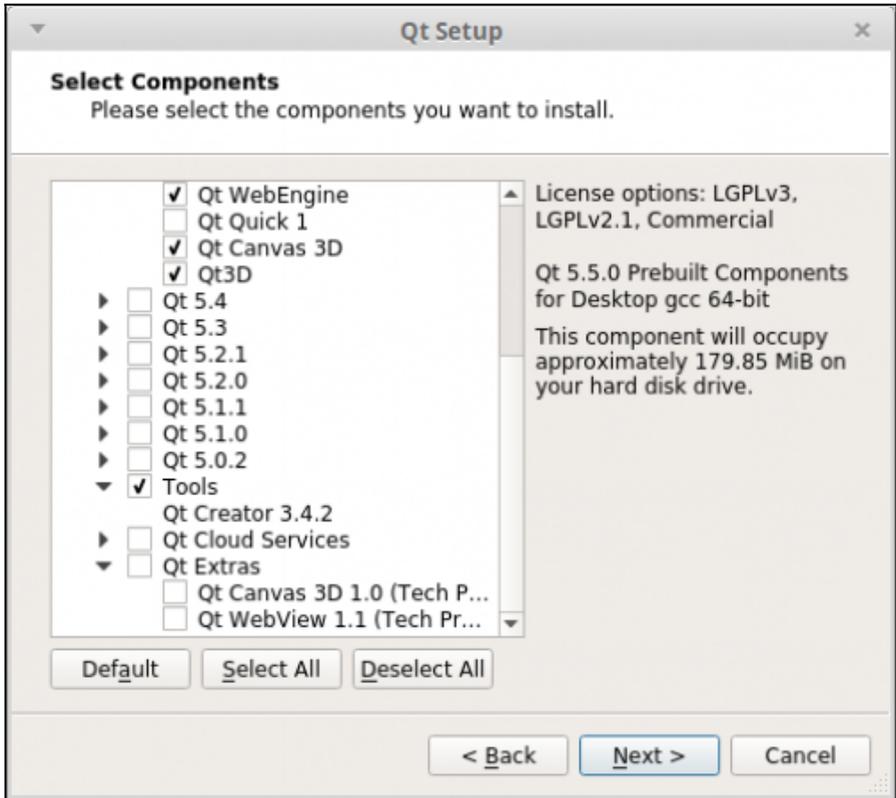
- `Source Components` : permet d'installer le code source de Qt. Si vous débutez, cela ne vous servira pas, mais si vous êtes un développeur C++/Qt un peu plus expérimenté, cela permet

d'analyser comment Qt est conçu ;

- `Qt Location` : fonctionnalités de positionnement (GPS) ;
- `Qt Quick Controls` : éléments d'interface graphique (bouton, dialogues, etc.) pour Qt Quick ;
- `Qt Script` : déprécié, ne pas utiliser (sauf besoin spécifique) ;
- `Qt WebEngine` : moteur Web, pour afficher des pages internet ;
- `Qt Quick 1` : déprécié, ne pas utiliser (sauf besoin spécifique) ;
- `Qt Canvas 3D` : moteur 3D simple similaire à WebGL en JavaScript ;
- `Qt 3D` : moteur 3D simple.

Je vous conseille d'installer les modules correspondant à la capture d'écran précédente (Qt pour GCC, Qt Location, Qt 3D, Qt Canvas 3D, Qt Quick Controls et Qt WebEngine. Vous n'aurez pas besoin de tous ces modules pour commencer, mais cela vous permettra d'explorer ce que propose Qt et de vous amuser un peu).

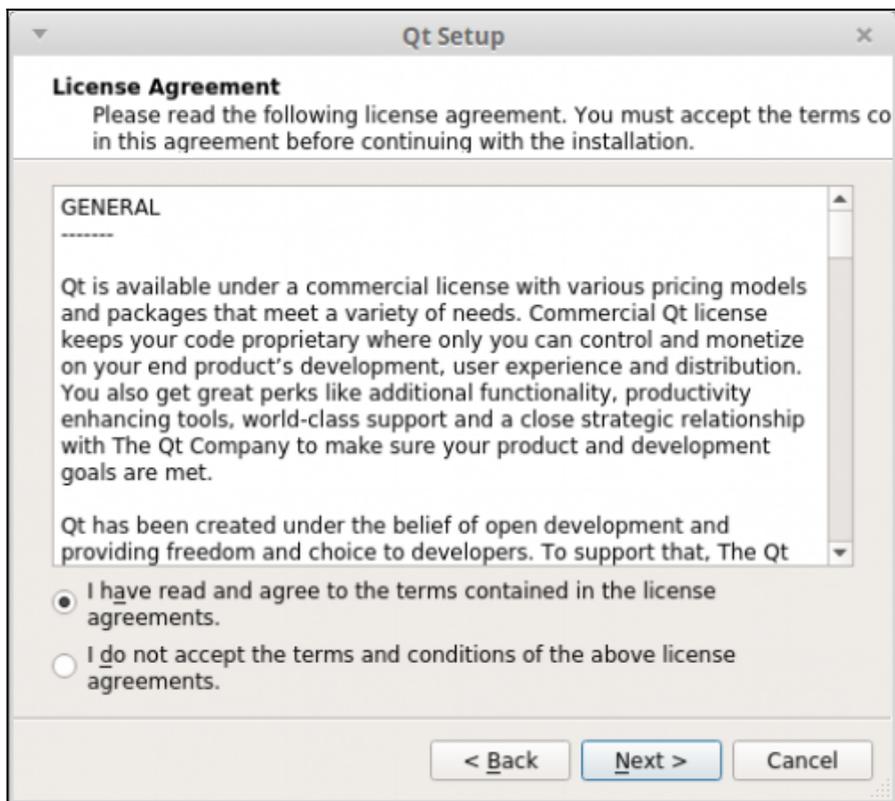
En dessous du choix des versions de Qt et des modules additionnels, vous pouvez sélectionner l'installation des outils externes.



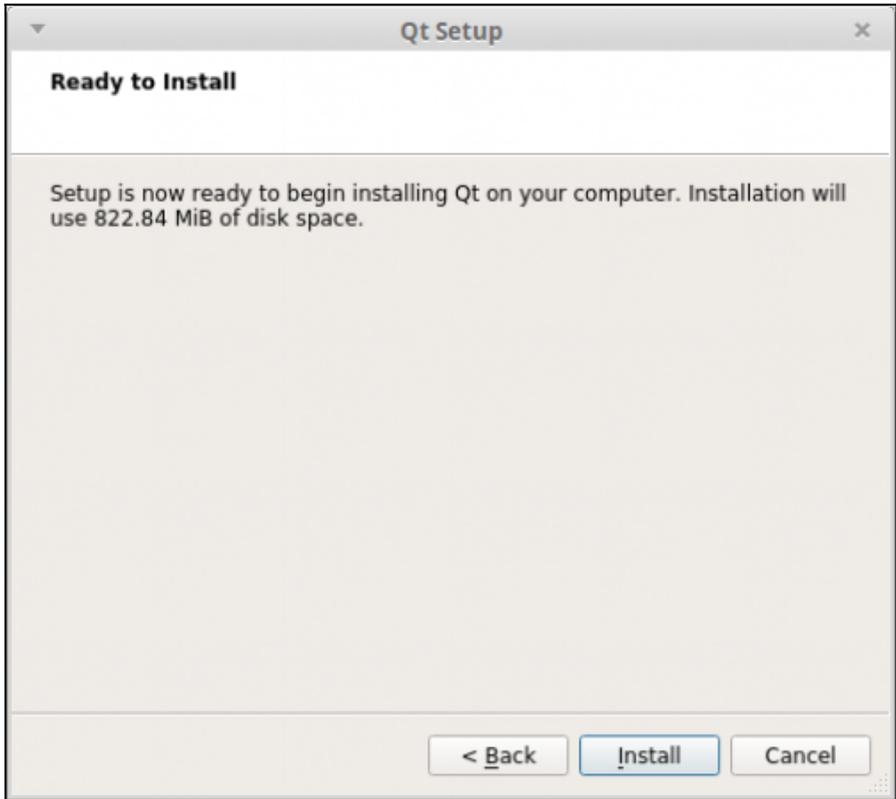
Qt Creator est l'éditeur fournit avec Qt. Il n'est pas possible de ne pas l'installer. (Même si vous souhaitez utiliser un autre éditeur, on utilisera Qt Creator dans ce tutoriel, pour tester si l'installation s'est bien passée).

Il n'est pas nécessaire d'installer les modules `Qt Cloud Services` et `Qt Extras`.

Cliquez sur suivant lorsque vous avez sélectionné les modules qui vous intéressent.

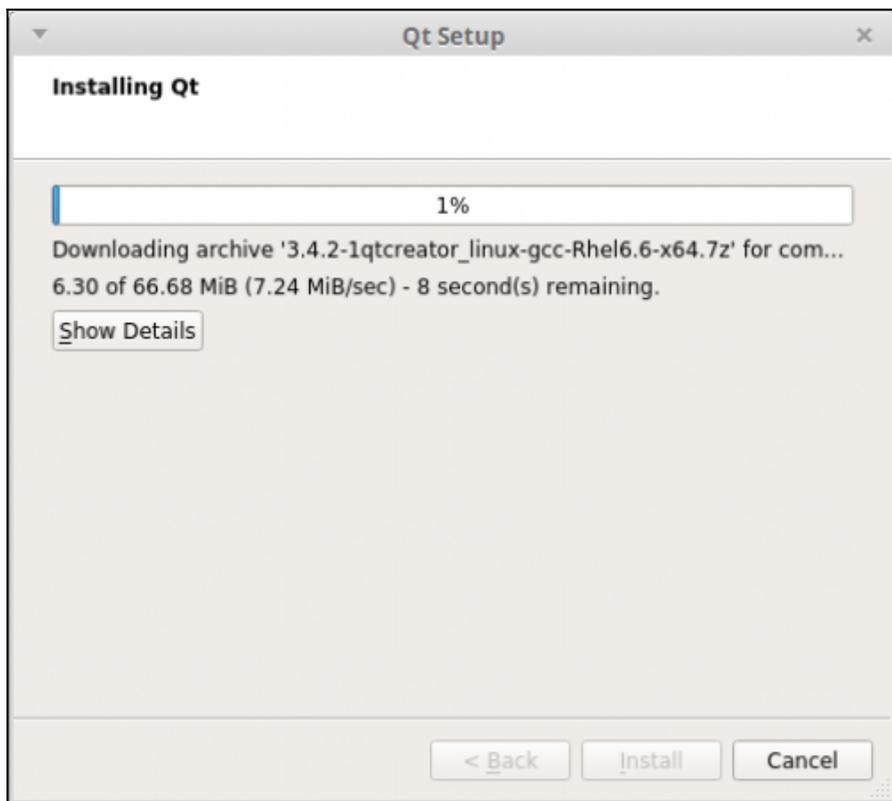


Cette page permet de valider les licences utilisateurs. Acceptez et cliquez sur suivant.



Une fois que tout cela est fait, l'installation est prête à démarrer. Cliquez sur Installation.

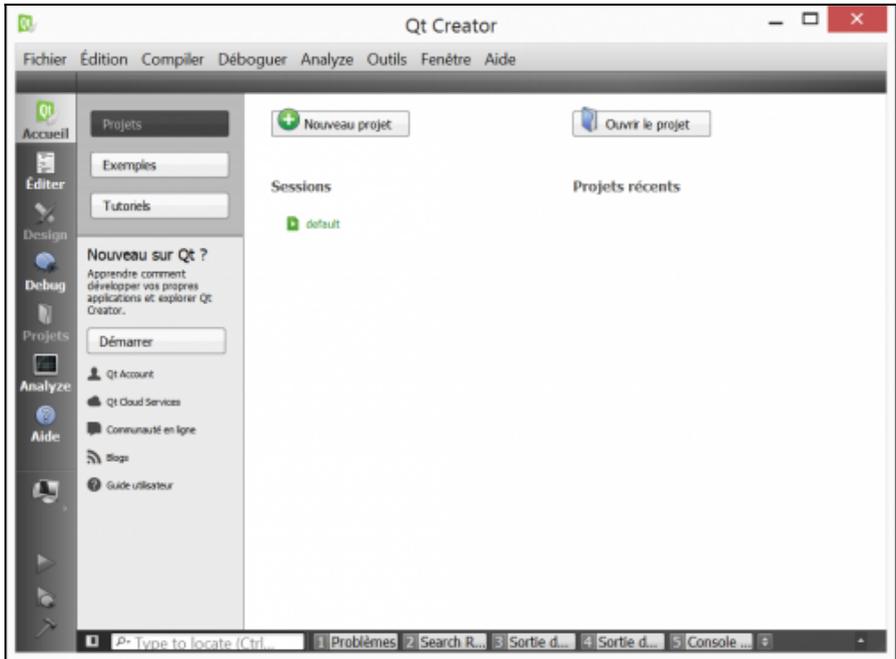
Le téléchargement puis l'installation se lancent. Selon votre connexion et le nombre de paquets que vous installez, cela peut prendre plusieurs minutes à plusieurs heures (si vous souhaitez installer beaucoup de paquets, il est probablement préférable de répéter l'installation plusieurs fois). L'installation sature le processeur, ne vous étonnez pas trop si Linux devient un peu lent pendant ce temps-là. Allez vous balader dehors, il fait beau (la pluie, c'est beau...).



Une fois que l'installation est terminée, la page suivante propose de lancer Qt Creator.

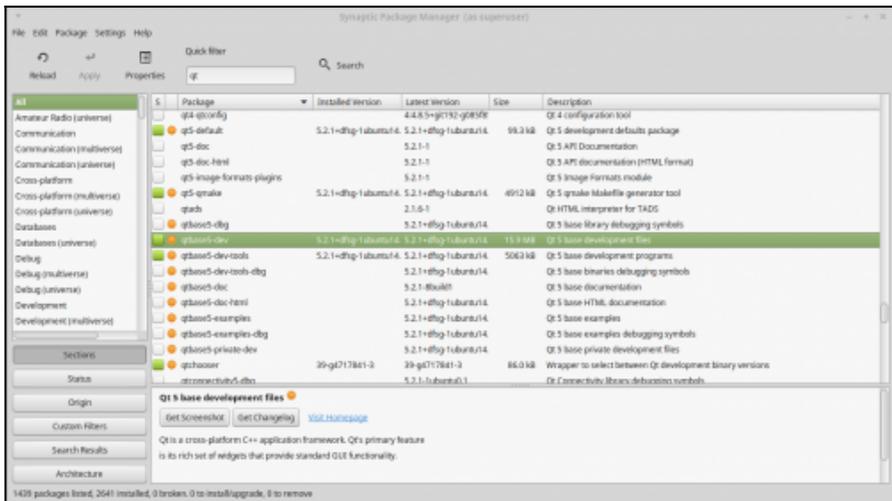


Cliquez sur Terminer. Qt Creator s'ouvre et affiche la page d'accueil.



## Installer en utilisant les dépôts

Beaucoup de distributions Linux proposent Qt 5 directement dans les dépôts, vous pouvez donc commencer par là. Par exemple, avec Ubuntu, lancez `Synaptic` et faites une recherche sur `qtbase5-dev` :



Comme vous pouvez le voir dans la copie d'écran, Ubuntu propose Qt 5, mais pas forcément Qt 5.5 (la version de Qt disponible dépendra de la distribution). Si vous n'avez pas besoin de la dernière version de Qt, vous pouvez installer celle des dépôts.

Vous pouvez également installer en ligne de commande, par exemple :

```
sudo apt-get install qtbase5-dev
```

Chaque module de Qt est dans un paquet spécifique. Regarder la liste des paquets dans `Synaptic` (`qtbase5-dev`) contient les fonctionnalités de base, vous pouvez commencer juste avec lui.

En complément, vous aurez besoin d'installer un compilateur C++. Vous pouvez installer GCC ou/et Clang, via synaptic ou en ligne de commande. Il est également possible d'installer d'autres méta-paquets, qui contiennent plusieurs outils de développement, comme les paquets "build-essential" ou "ubuntu-sdk".

Pour tester si vous avez un compilateur fonctionnel et quelle est sa version, vous pouvez taper l'une des lignes suivantes :

```
g++ -v
clang++-3.5 -v
```

Par exemple, chez moi, cela retourne pour GCC les lignes suivantes. Vous pouvez voir à la dernière ligne que c'est la version 4.9.1 de GCC.

```
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/4.9/lto-wr
apper
Target: x86_64-linux-gnu
Configured with: ../src/configure -v
--with-pkgversion='Ubuntu 4.9.1-16ubuntu6'
--with-bugurl=file:///usr/share/doc/gcc-4.9/README.Bugs
--enable-languages=c,c++,java,go,d,fortran,objc,obj-c++
--prefix=/usr --program-suffix=-4.9 --enable-shared
--enable-linker-build-id --libexecdir=/usr/lib
--without-included-gettext --enable-threads=posix
--with-gxx-include-dir=/usr/include/c++/4.9
--libdir=/usr/lib --enable-nls --with-sysroot=/
--enable-clocale=gnu --enable-libstdcxx-debug
--enable-libstdcxx-time=yes --enable-gnu-unique-object
--disable-vtable-verify --enable-plugin
--with-system-zlib --disable-browser-plugin
--enable-java-awt=gtk --enable-gtk-cairo
--with-java-home=/usr/lib/jvm/java-1.5.0-gcj-4.9-amd64/jre
--enable-java-home
--with-jvm-root-dir=/usr/lib/jvm/java-1.5.0-gcj-4.9-amd64
--with-jvm-jar-dir=/usr/lib/jvm-exports/java-1.5.0-gcj-4.9-a
md64 --with-arch-directory=amd64
--with-ecj-jar=/usr/share/java/eclipse-ecj.jar
--enable-objc-gc --enable-multiarch
--disable-werror --with-arch-32=i686 --with-abi=m64
--with-multilib-list=m32,m64,mx32
--enable-multilib --with-tune=generic
--enable-checking=release --build=x86_64-linux-gnu
--host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 4.9.1 (Ubuntu 4.9.1-16ubuntu6)
```

Pour Clang :

```
Ubuntu clang version 3.5.0-4ubuntu2 (tags/RELEASE_350/final)
(based on LLVM 3.5.0)
```

```
Target: x86_64-pc-linux-gnu
Thread model: posix
Found candidate GCC installation:
/usr/bin/../lib/gcc/i686-linux-gnu/4.9
Found candidate GCC installation:
/usr/bin/../lib/gcc/i686-linux-gnu/4.9.1
Found candidate GCC installation:
/usr/bin/../lib/gcc/x86_64-linux-gnu/4.8
Found candidate GCC installation:
/usr/bin/../lib/gcc/x86_64-linux-gnu/4.8.3
Found candidate GCC installation:
/usr/bin/../lib/gcc/x86_64-linux-gnu/4.9
Found candidate GCC installation:
/usr/bin/../lib/gcc/x86_64-linux-gnu/4.9.1
Found candidate GCC installation:
/usr/lib/gcc/i686-linux-gnu/4.9
Found candidate GCC installation:
/usr/lib/gcc/i686-linux-gnu/4.9.1
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.8
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.8.3
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.9
Found candidate GCC installation:
/usr/lib/gcc/x86_64-linux-gnu/4.9.1
Selected GCC installation:
/usr/bin/../lib/gcc/x86_64-linux-gnu/4.9
Candidate multilib: .;@m64
Candidate multilib: 32;@m32
Candidate multilib: x32;@mx32
Selected multilib: .;@m64
```

[Revenir à la page principale du tutoriel](#)