



Programmer en Ruby/Qt avec KDevelop 3.x et QtDesigner

Kdevelop est devenu un environnement de développement multi-langages très agréable à utiliser. Les gros bugs des versions < 3.2.0 m'avaient poussé à utiliser Eric3, l'IDE dédié à Python et à Ruby. Je ne le regrette pas car Eric3 est devenu, lui aussi, très agréable depuis sa version 3.7. Je reste attentif à son évolution, notamment dans sa prise en charge de Ruby qui n'était pas complète dans la version 3.7.0.

Installation

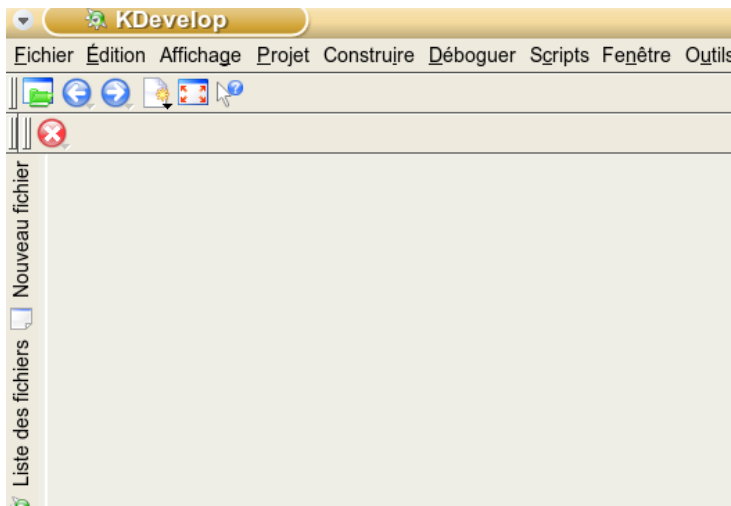
Ruby existe pour toutes les plate-formes. Son installation, dans sa version la plus récente 1.8, ne pose aucun problème particulier. Pour Linux ou BSD, vous le trouvez dans les dépôts de votre distribution. Pour Windows, téléchargez-le et installez-le comme n'importe quel logiciel à partir du site : <http://www.ruby-lang.org/en/20020102.html>

Kdevelop n'existe que pour les plate-formes Unix/Linux/BSD, de préférence dans l'environnement KDE. On le trouve dans les dépôts des diverses distributions Linux ou BSD. J'utilise la **version 3.3** sous **Suse 9.3**.

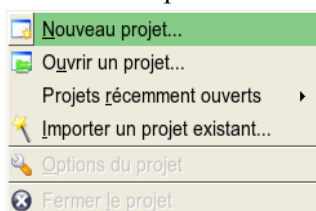
La liaison entre **Ruby** et **Qt** existe sous diverses appellations dans les distributions Linux. Par exemple, **Suse 9.x** utilise le paquet **kdebindinds** qui assure les liaisons entre Qt et différents langages comme C++, Python, Ruby, etc. Sous **Debian** et **Ubuntu**, on trouve la liaison spécifique **Qt/Ruby** sous l'appellation **libqt0-ruby.deb**

Création d'un application

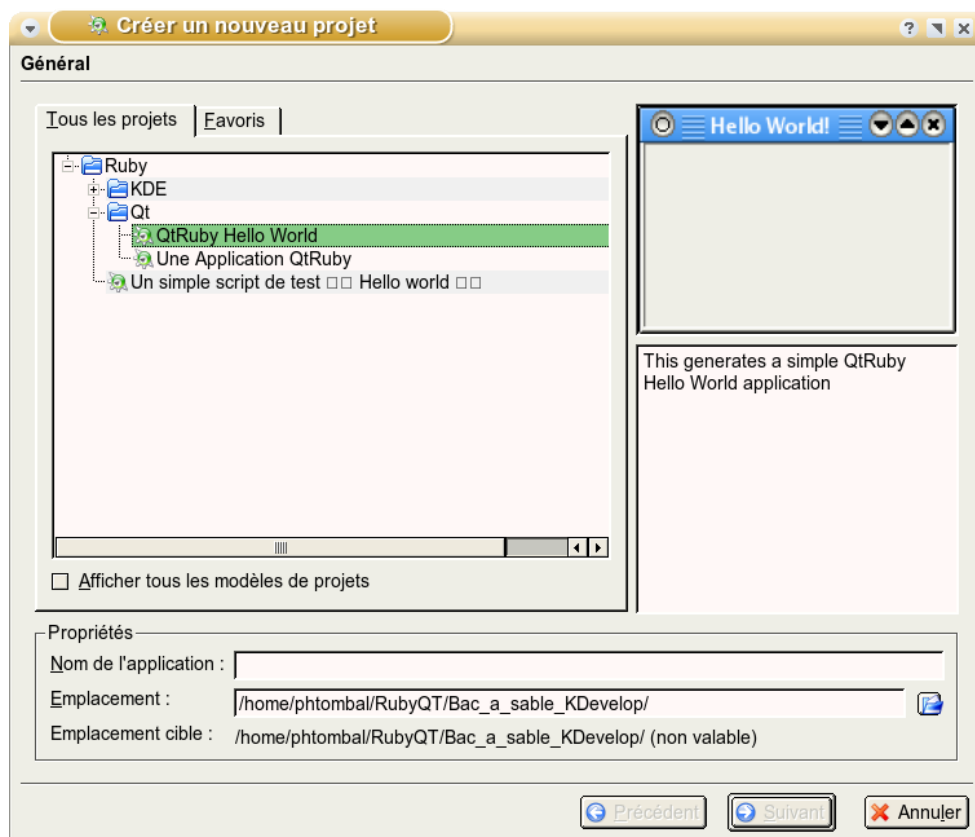
Kdevelop vient de s'ouvrir... Il est tout nu !



- Tout commence par la création d'un nouveau projet : **menu Projet -> Nouveau projet...**



La création d'un projet est indispensable si vous voulez profiter de toutes les facilités de la prise en charge de Ruby : compilation des formes (.ui) provenant de QtDesigner, lancement de l'application principale, regroupement de tous vos fichiers en vue de publication, etc. Et surtout, quand vous serez revenu de vos courses, il suffira d'ouvrir le fichier du projet (.kdevelop) pour retrouver votre environnement comme vous l'aviez laissé !



Seul le langage Ruby apparaît dans cette interface de démarrage du projet. Si j'avais coché l'option « **Afficher tous les modèles de projets** », j'aurais vu apparaître la liste complète des langages pris en charge.

Je choisis le modèle « **QtRuby Hello World** ». C'est le plus simple... Cela donne un petit fichier qui va se contenter de lancer mon application.

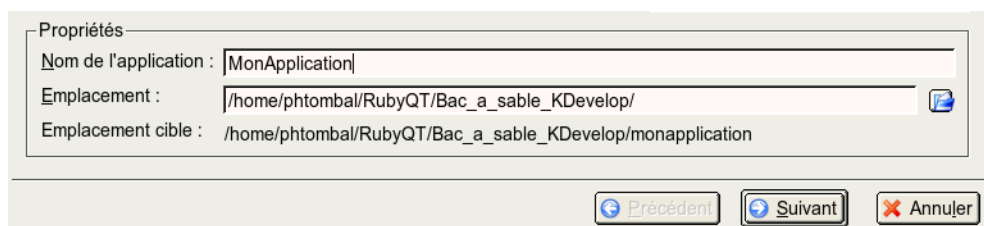
Si vous voulez voir un exemple d'application complète avec fenêtre décorée de ses menus classiques (Fichier, Edition, etc.) fonctionnels, choisissez « **Une Application QtRuby** ». Ce n'est toutefois pas la bonne méthode pour créer ce genre d'interface. Il vaut mieux la créer à partir de QtDesigner (ce qui permet de la modifier plus facilement) et d'y ajouter les fonctionnalités des menus en copiant/collant le code de cet exemple.

Dans les propriétés, j'entre le Nom de l'application : **MonApplication**

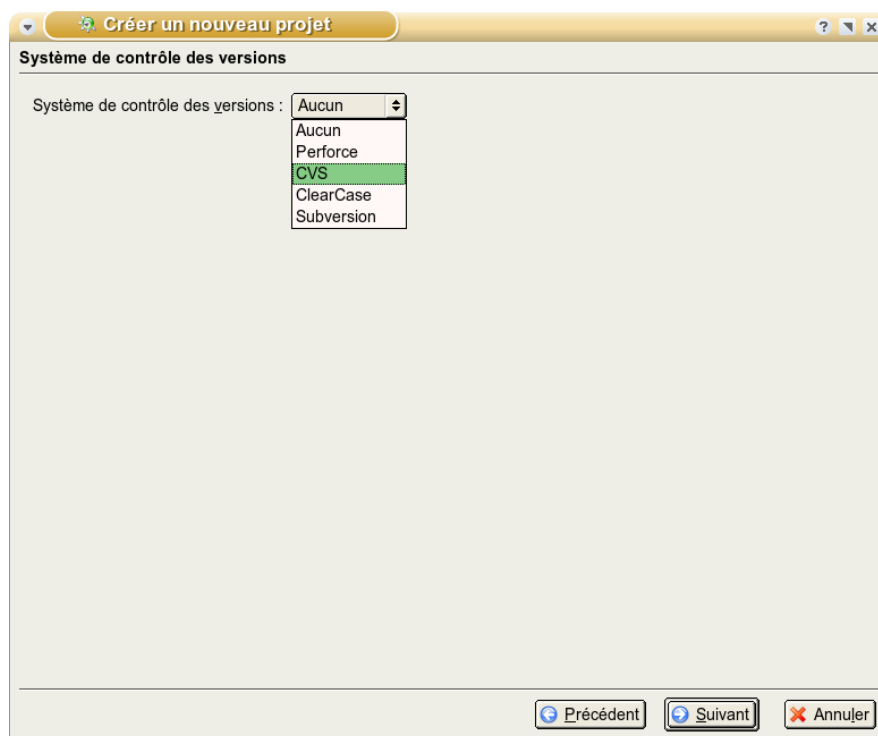
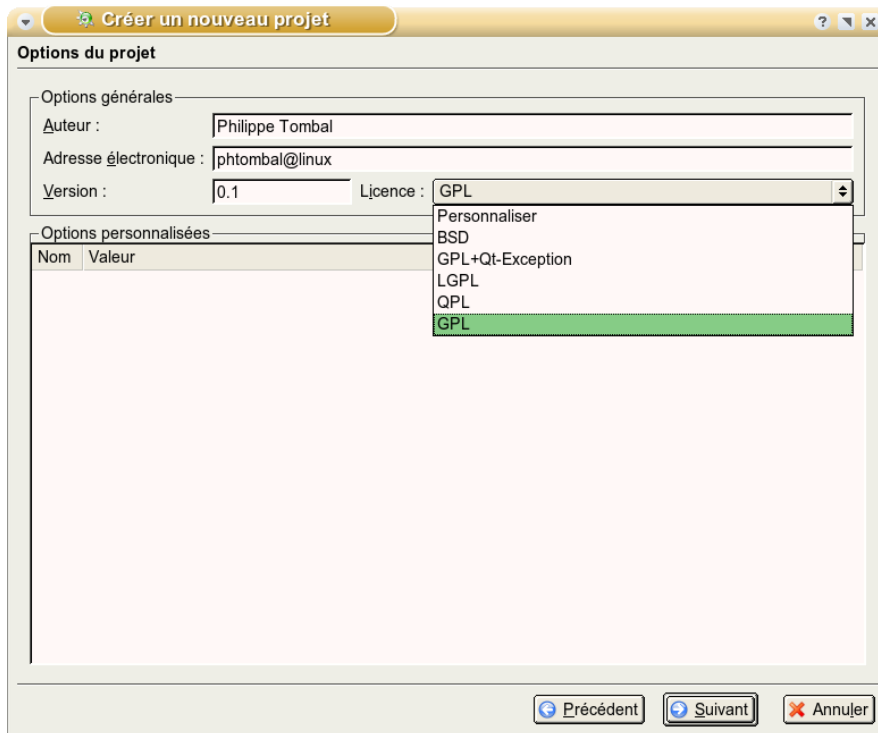
Je choisis un emplacement pour mon projet : un répertoire existant ou un répertoire que je crée à cet effet.

Ici, j'ai créé un répertoire appelé **Bac_a_sable_KDevelop**

Vous remarquerez que Kdevelop transforme automatiquement le nom de votre application pour en faire un nom de répertoire tout en minuscules.



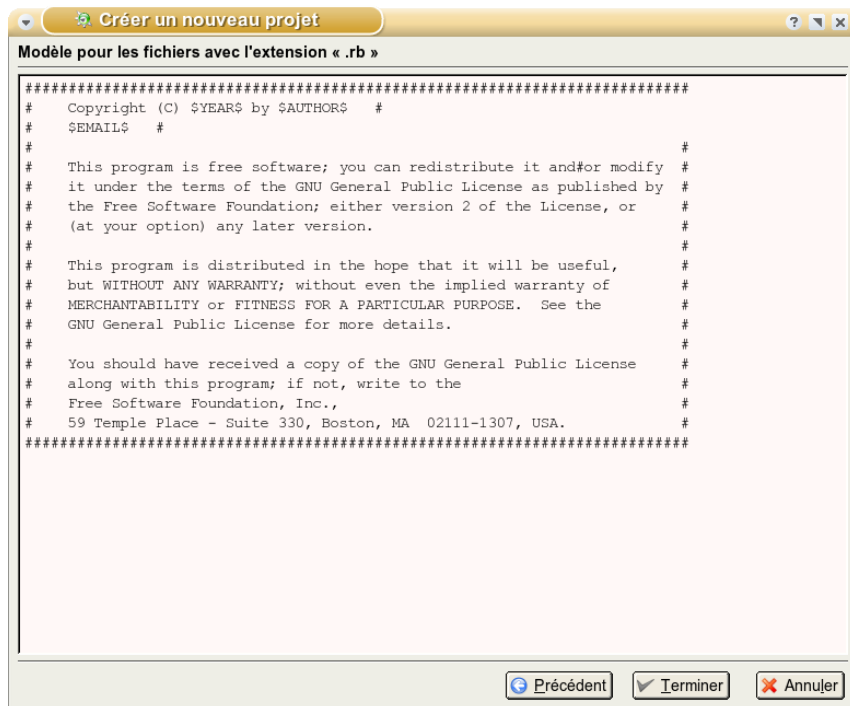
Les dialogues suivants vous demandent des informations sur vous et le type d'application que vous développez en vue d'une publication : **licence** et **système de révision**



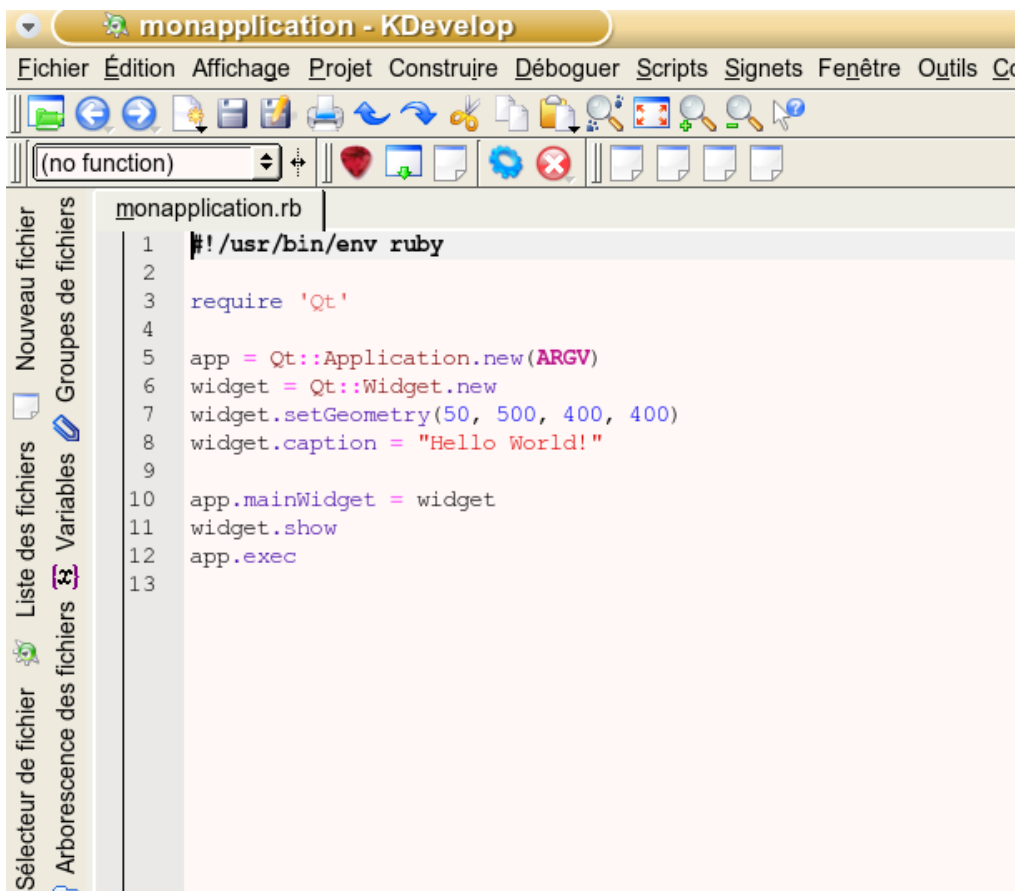
Si vous avez l'intention de publier votre oeuvre sur <http://freshmeat.net/> et de permettre à d'autres programmeurs d'améliorer votre code, informez-vous sur le fonctionnement du contrôle de révision **CVS** qui permet de suivre l'évolution des différentes versions en cours de modification : <http://www.nongnu.org/cvs/> Dans mon exemple, ici, j'ai choisi **Aucun**.

Pour la licence, je suppose que nous sommes tous d'accord pour choisir la GPL ?
<http://www.gnu.org/copyleft/gpl.html>

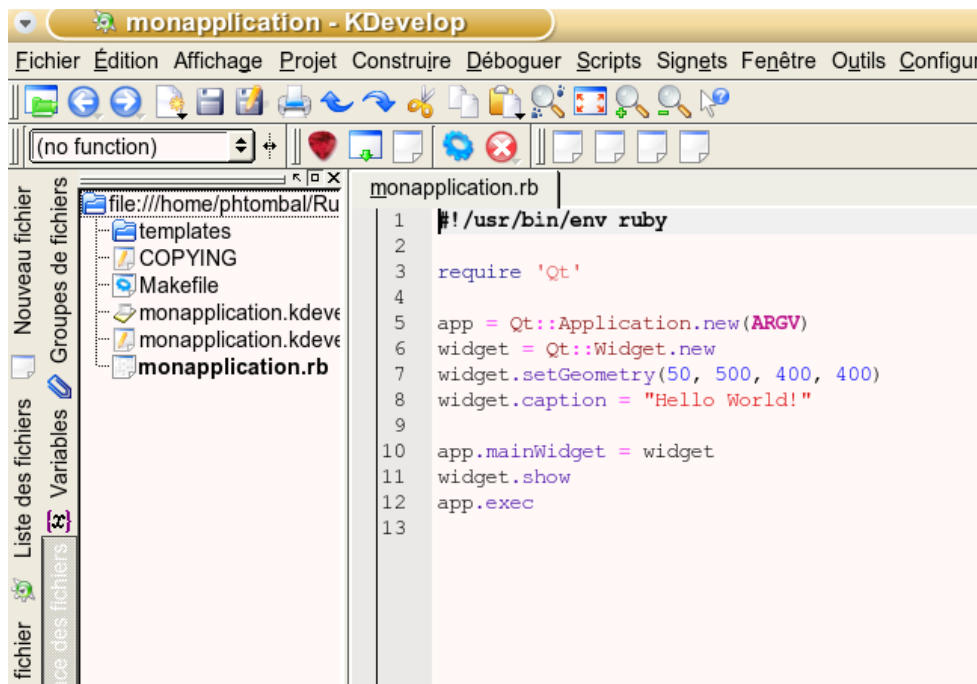
Il n'y a plus qu'à terminer...



Ce qui nous donne, comme prévu, le **monapplication.rb** contenu dans le répertoire **monapplication** devenu sous-répertoire de **Bac_a_sable_KDevelop**.



Si vous cliquez sur l'onglet « **Arborescence des fichiers** » dans la bordure gauche, vous voyez apparaître la liste des fichiers que Kdevelop a créé pour vous (nous reviendrons sur leur signification) :



Maintenant, commence les choses sérieuses :

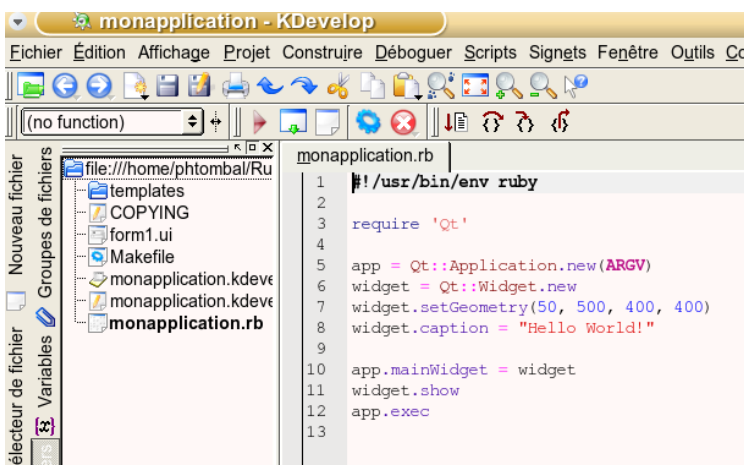
- Je crée une interface à partir de **QtDesigner**. Pour cela j'ouvre **QtDesigner** indépendamment de Kdevelop et je crée mon interface avec les connexions éventuelles entre mes widgets, etc.

⊗ Si vous ne voyez pas de quoi je cause, allez voir mon tutoriel « **Introduction à QtDesigner** ».

- Je sauve mon interface (par défaut **form1.ui** si je ne change pas le nom de ma forme principale) dans le répertoire **monapplication** du projet Kdevelop.

👁 Je sais qu'il est possible de créer une forme directement à partir de Kdevelop. Pour cela, il suffit de créer un fichier avec l'extension .ui (**Fichier --> Nouveau --> form1.ui**) . Kdevelop ouvre alors QtDesigner dans sa fenêtre à lui. Je trouve que ce n'est pas très pratique d'utiliser QtDesigner coincé comme cela dans un autre environnement. Je préfère donc passer de Kdevelop à QtDesigner à partir de ma barre des tâches. Question de goût...

- Je retourne dans Kdevelop et je constate qu'il a ajouté mon fichier form1.ui sans l'Arborescence de fichiers



Si je clique sur le fichier **form1.ui** dans l'Arborescence, Kdevelop ouvre QtDesigner dans la fenêtre d'édition. On se rend compte tout de suite que ce n'est pas très pratique ! Il vaut mieux donc le fermer (clic droit sur l'onglet du nom de fichier au dessus de la fenêtre d'édition).

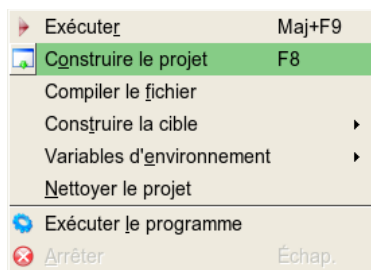
Si vous voulez visualiser le contenu XML du fichier **form1.ui**, il suffit de cliquer droit sur le nom de fichier dans l'Arborescence et sélectionner **Ouvrir avec... --> Ouvrir comme de l'UTF-8**.

Il ne reste plus qu'à traduire ce fichier XML en Ruby.

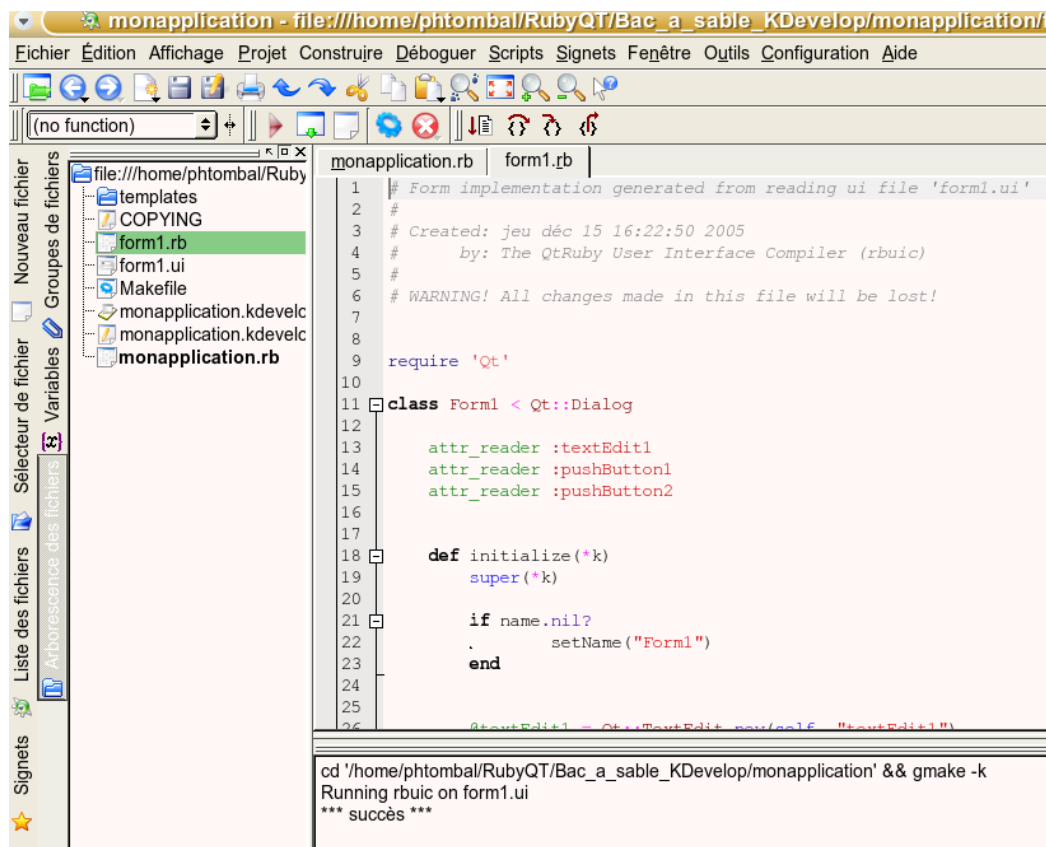
En console, on traduit un fichier .ui en code Ruby par la commande suivante :

```
rbuic form1.ui -o form1.rb
```

Kdevelop le fait pour vous, par la commande « **Construire le projet** » du menu « **Construire** » ou F8.



Par cette commande Kdevelop lance le script **rbuic** sur toutes les formes **.ui** de l'Arborescence. Il faut donc la relancer chaque fois que vous faites une modification de votre interface avec QtDesigner.



On constate que le fichier généré **form1.rb** est apparu dans l'Arborescence. On le visualise dans l'éditeur en cliquant simplement sur son nom.

Pour ajouter des fonctionnalités à mon interface, c'est-à-dire entrer mon code, la meilleure méthode (comme en Python) est de créer une sous-classe de la classe **Form1**.

Pour rappel, il est déconseillé de modifier le fichier **form1.rb** généré par rbuic. Cela va de soi puisque si vous modifiez l'interface **form1.ui** avec QtDesigner, ce que vous aurez entré dans **form1.rb** sera balayé par l'action de rbuic.

Donc, le mieux est de créer une sous classe de Form1.

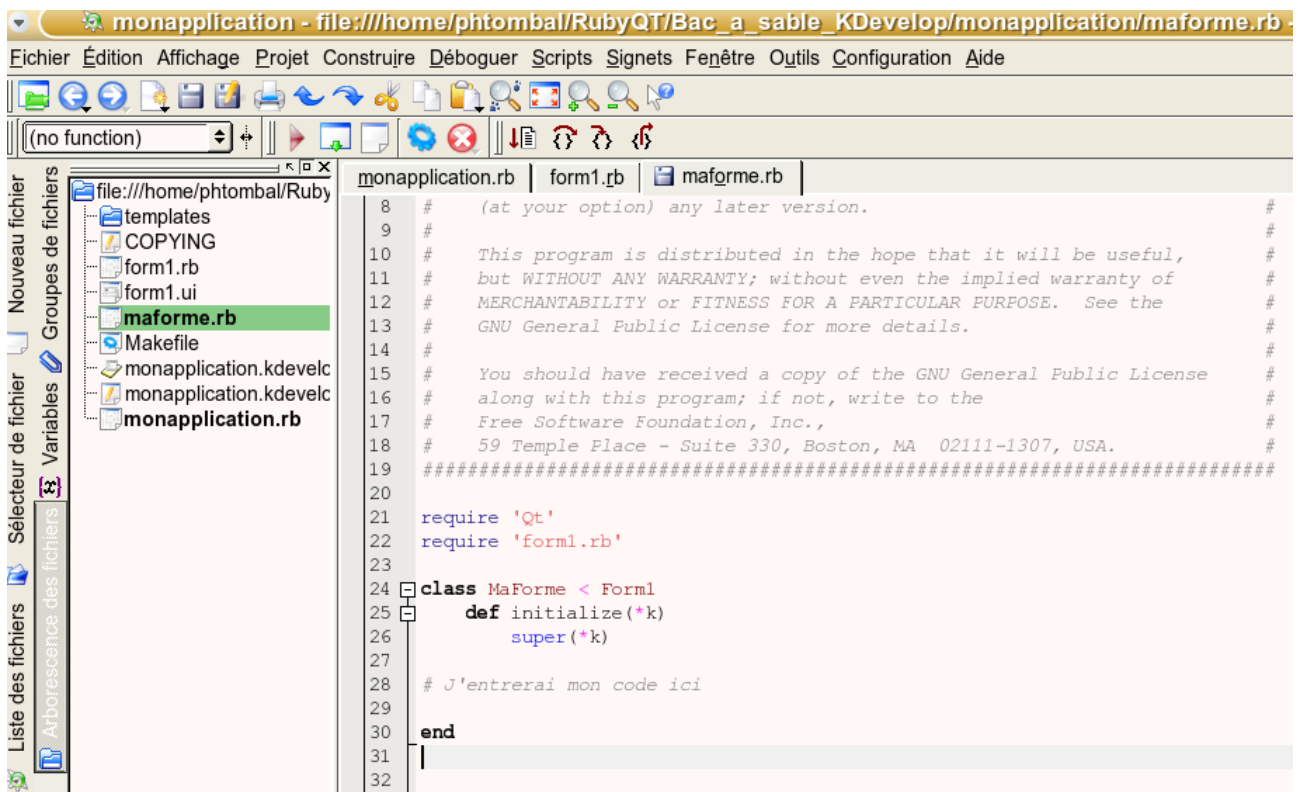
Pour cela, je crée un nouveau fichier que j'appelle, par exemple, **maforme.rb** et j'entre le code suivant :

```
require 'Qt'
require 'form1.rb'

class MaForme < Form1
  def initialize(*k)
    super(*k)

  # J'entrerais mon code ici

end
```



Je peux donc entrer mes définitions de slots correspondant à **Form1** comme méthodes de la classe **MaForme** sans risque.

Je peux également créer dans ce fichier de nouvelles définitions de classes.

Cette méthode est valable pour n'importe quelle application.

Il ne reste plus qu'à rendre notre application exécutable. Pour cela, je modifie mon fichier de départ **monapplication.rb** afin qu'il tienne compte des éléments que je viens d'ajouter.

Je transforme donc le code d'origine :

```
#!/usr/bin/env ruby

require 'Qt'

app = Qt::Application.new(ARGV)
widget = Qt::Widget.new
widget.setGeometry(50, 500, 400, 400)
widget.caption = "Hello World!"

app.mainWidget = widget
widget.show
app.exec
```

par le code suivant

```
#!/usr/bin/env ruby

require 'Qt'
require 'maforme.rb'

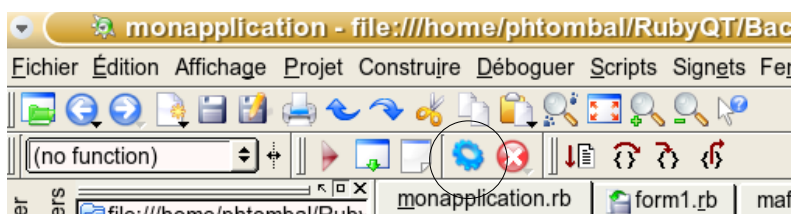
app = Qt::Application.new(ARGV)
widget = MaForme.new

app.mainWidget = widget
widget.show
app.exec
```

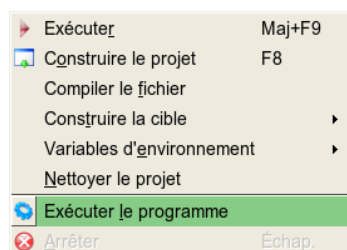
- J'ai ajouté une instruction **require 'maforme.rb'**
- J'ai remplacé **Qt::Widget** par **MaForme**
- J'ai supprimé les lignes correspondant à l'affichage du message « Hello World !»

Il n'y a plus qu'à exécuter l'application :

Par le bouton d'exécution :



ou à partir du menu « **Construire** » :



Et voilà !

Il vous reste à consulter la documentation (onglet « Documentation » à droite de la fenêtre de Kdevelop) pour en savoir plus sur les autres fonctionnalités...

Annexe 1

Trucs et astuces

Comment installer Kdevelop 3.3 sous Novell/Suse 9.3 ?

Sous **Suse 9.3**, la version de Kdevelop incluse dans la distribution est **3.2.x**. Je conseille d'installer la version **3.3** qui corrige pas mal de bugs.

C'est très facile.

Téléchargez dans le répertoire de votre choix la version 3.3 à partir de ce site :

<ftp://ftp.belnet.be/packages/kde/stable/3.5/SuSE/ix86/9.3/>

Avec **Konqueror**, cliquez droit sur le fichier et sélectionnez **Actions --> Installer avec Yast**

ou bien :

Dans une console, tapez :

```
su -  
  
cd /le répertoire où vous avez téléchargé le fichier rpm  
  
rpm -Uvh kdevelop3-3.3.0-5.i586.rpm
```

Comment mettre Kdevelop en français sous Ubuntu ou Kubuntu 5.10

Tapez dans une console (sur une seule ligne) :

```
sudo ln -s /usr/share/locale-langpack/fr/LC_MESSAGES/kdevelop.mo  
/usr/share/locale/fr/LC_MESSAGES/kdevelop.mo
```

En effet, le lien existe en tant que kdevelop3.mo et n'est pas pris en compte par kdevelop.

Annexe 2

Des liens intéressants

La grosse partie de la documentation sur Ruby, Qt, Kdevelop et QtDesigner se trouve dans la documentation interne de Kdevelop. Elle est en anglais.

A noter : ce sont des liens vers des sites externes. Vous devez donc être connecté à Internet pour y accéder.

Je vais renseigner ici les liens en français et/ou les liens vers de la documentation non répertoriée dans Kdevelop.

Sur Ruby :

Le site du tutoriel facile et en français :

<http://www.ruby-doc.org/docs/beginner-fr/xhtml/index.html>

Le site des développeurs français en Ruby :

<http://www.rubyfr.org/>

Le guide (poignant) :

<http://fr.poignantguide.net/>

RubyGarden :

<http://www.rubygarden.org/>

Un exemple de classe Ruby : les fractions

<http://www.rubygarden.org/ruby?Fraction>

Sur Kdevelop et Ruby :

La création d'une application complète en QtRuby :

<http://developer.kde.org/language-bindings/ruby/colortooltutorial/designer-manual-3.html>

La base du Ruby binding avec Qt3 et KDE :

<http://developer.kde.org/language-bindings/ruby/>

Un autre point d'entrée au développement avec Kdevelop :

<http://www.kdevelop.org/doc/technotes/rubyrad.html>