

# Créer un exécutable à partir de sources Perl

par djibril ([Site personnel](#))

Date de publication : 2011-07-10

Dernière mise à jour : 2012-02-24

Le but de cet article est de vous montrer comment installer et utiliser des outils ou modules nécessaires pour créer un fichier exécutable (.exe) à partir de vos sources Perl.

---

1 - Introduction.....	3
2 - Outils et modules nécessaires.....	3
2-A - perlcc.....	3
2-B - Perl2Exe.....	3
2-C - PerlApp.....	3
2-C-1 - Installation.....	4
2-C-2 - Utilisation.....	4
2-D - Module PAR.....	9
2-D-1 - Installation du module.....	9
2-D-2 - Utilisation du module.....	16
3 - Liens utiles.....	20
4 - Conclusion.....	21
5 - Remerciements.....	21

## 1 - Introduction

Pourquoi vouloir créer un fichier exécutable (un .exe sous Windows) Perl ? Nous pourrions répondre à rien si nous exécutons nos programmes Perl sur notre machine, si nous ne partageons pas nos programmes... Néanmoins, il est toujours intéressant de savoir concevoir un fichier exécutable au sens binaire pour les raisons suivantes :

- vous souhaitez partager vos beaux programmes avec des collègues ou des amis n'ayant pas Perl installé sur leur machine ;
- regrouper un ensemble de scripts Perl en un unique programme exécutable ;
- partager vos programmes Perl en souhaitant ne pas divulguer le code source, même si de toute façon, il est toujours possible de le pirater et le lire. La meilleure protection reste dans ce cas une licence.

Cet article vous explique comment s'y prendre à travers l'utilisation d'outils et modules Perl disponibles sur le site du CPAN.

## 2 - Outils et modules nécessaires

Il existe différents outils et modules permettant de créer un fichier exécutable Perl : **perlcc**, **perl2exe**, **PerlApp** et le module **PAR**.

### 2-A - perlcc

Présent uniquement dans les versions Perl 5.8, le programme **perlcc** permet, en ligne de commande, de créer un exécutable à partir d'un script Perl. Malheureusement, l'exécutable généré n'est pas garanti d'après son auteur. De plus, il s'avère qu'il ne fonctionne pas bien et n'est pas recommandé en production. Étant resté à un stade très expérimental, il n'est plus maintenu à ce jour. C'est d'ailleurs pour cette raison qu'il a été retiré des récentes versions de Perl (5.10, 5.12...). Vous comprendrez bien que je ne vous le recommande pas !

### 2-B - Perl2Exe

C'est un outil qui se lance en ligne de commande. Il ne se trouve pas dans le CORE de Perl et ce n'est pas un module CPAN. C'est un programme développé par la société « IndigoStar » et est téléchargeable sur leur [site web](#).

Perl2Exe permet la création de programmes autonomes ne nécessitant pas d'interpréteur Perl. De plus, votre code Perl sera ainsi caché. Il peut générer des exécutables pour les systèmes d'exploitation Windows, Linux, Solaris... Il permet également de créer des programmes utilisant Tk sans console. Autre point important, il est possible de générer du code multiplateforme. Par exemple, vous pouvez générer du code pour Solaris à partir d'une machine Linux ou Windows. Par contre, un exécutable conçu pour une machine Windows ne fonctionnera que sous Windows.

Il reste important de préciser que cet outil n'est **pas gratuit**. Vous pouvez obtenir une licence d'évaluation de trente jours. Je ne vous en dirai pas plus car pour le moment, après installation, je n'ai pas réussi à le faire fonctionner. À la première utilisation, j'obtiens le message d'erreur suivant :

```
Warning: Can't locate Encode/ConfigLocal.pm
```

### 2-C - PerlApp

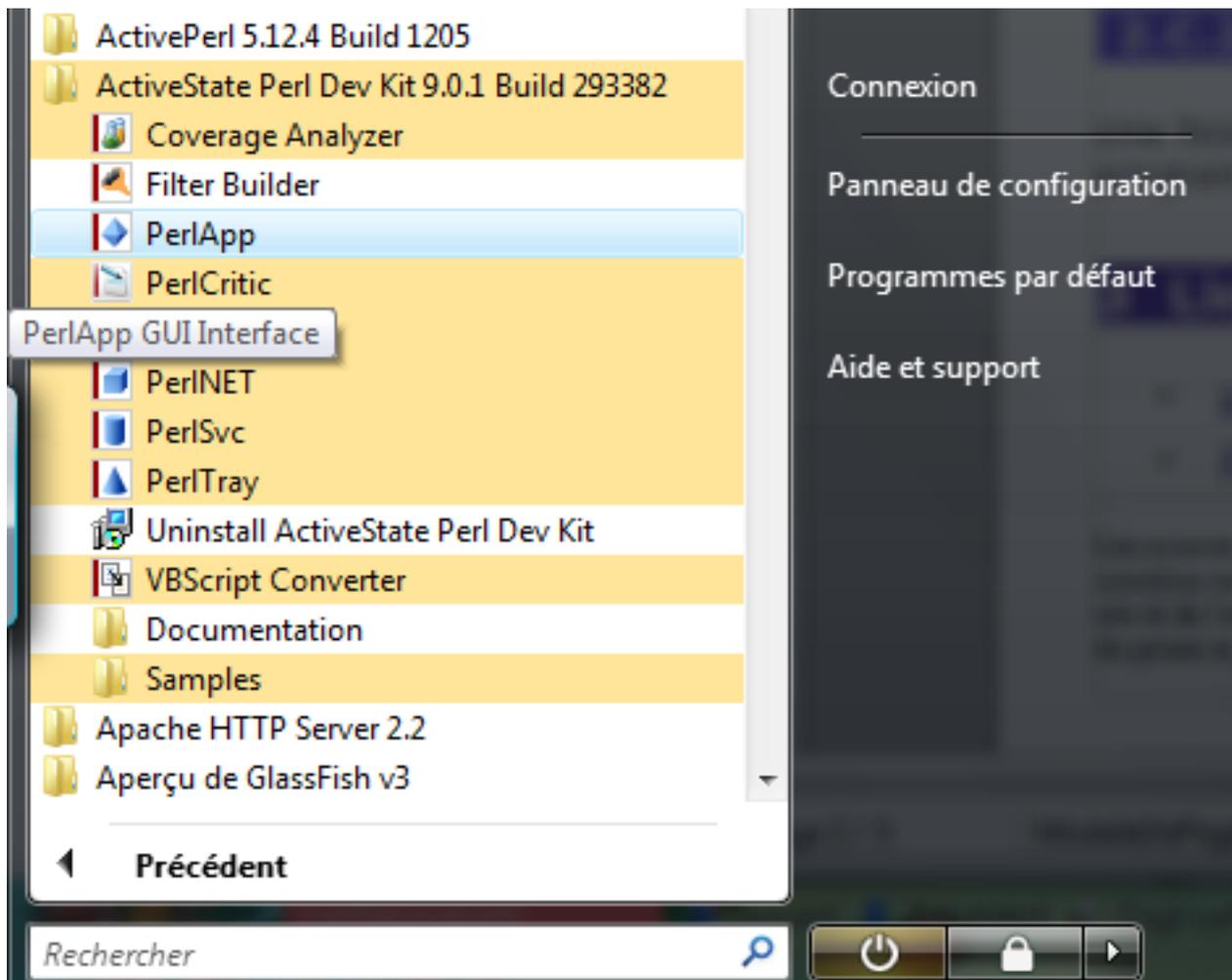
Cet outil est inclus dans l'utilitaire <http://www.activestate.com/perl-dev-kit>. Ce dernier est développé et proposé sur le site d'**ActiveState**. Son rôle est de fournir plusieurs outils (notamment PerlApp) permettant aux développeurs Perl de créer et déployer des applications. Mais une fois de plus, cet outil est payant. Il existe néanmoins une version **d'évaluation de 21 jours**.

## 2-C-1 - Installation

Téléchargeons la version d'évaluation. Vous avez le choix d'un msi en fonction de votre plate-forme Windows, Linux, Mac Os sous 32 ou 64 bits. Une fois le choix effectué, téléchargez-le et installez-le. C'est aussi simple qu'installer Perl.

## 2-C-2 - Utilisation

Une fois l'installation effectuée, il n'y a pas d'icône sur le bureau. C'est donc à vous de le chercher dans les programmes (cherchez par exemple « ActiveState Perl Dev Kit 9.0.1 ») puis cliquez sur PerlApp.



Lorsque vous cliquez sur PerlApp, vous avez une fenêtre qui s'ouvre et il ne vous reste plus qu'à paramétrer, choisir les options qui vous intéressent afin de créer votre exécutable : nom de l'exécutable, information sur l'exécutable (licence, nom de la société...), choisir si vous souhaitez inclure ou exclure l'interpréteur Perl... il y a le choix ! Rien de plus simple qu'un exemple avec un code Perl/Tk dont le but est d'ouvrir une fenêtre et vous saluer.

Programme test.pl dans C:\test.pl

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;
use utf8;

# Création de la fenêtre
my $fenetre = new MainWindow(
    -title => 'Première fenêtre Tk',
```

## Programme test.pl dans C:\test.pl

```
-background => 'white',
);

# Taille minimale de ma fenêtre
$fenetre->minsize( 300, 100 );

my $message_accueil = "Bonjour tout le monde\n\nWelcome dans le monde magnifique de Perl/Tk et PerlApp
\n\n";

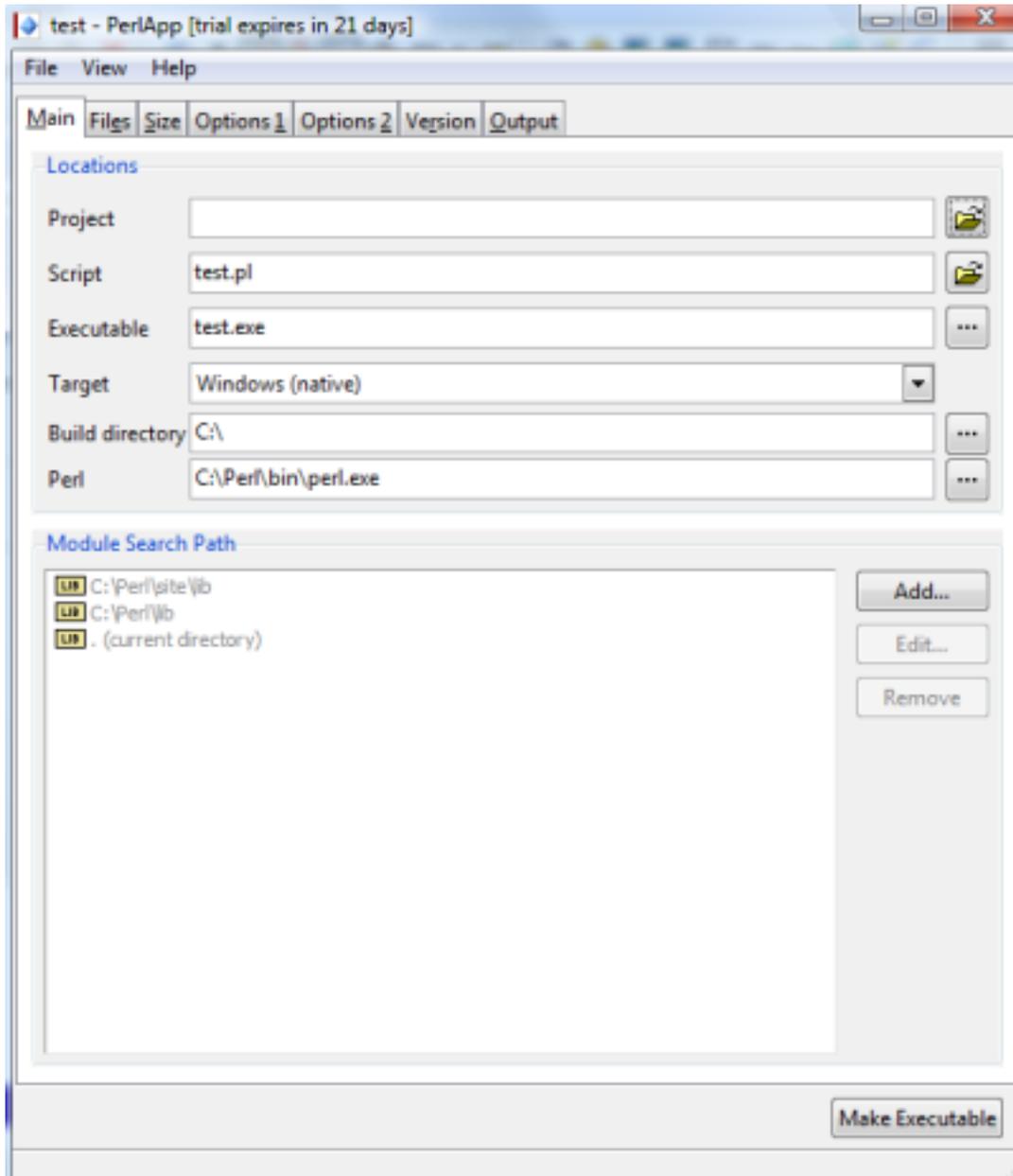
# Affichage d'un texte
my $label_accueil = $fenetre->Label(
    -text      => $message_accueil,
    -background => 'white',
)->pack();

# Affichage d'un bouton pour fermer la fenêtre
my $bouton = $fenetre->Button(
    -text      => 'Ferme la fenêtre',
    -command => sub { exit; },
)->pack();

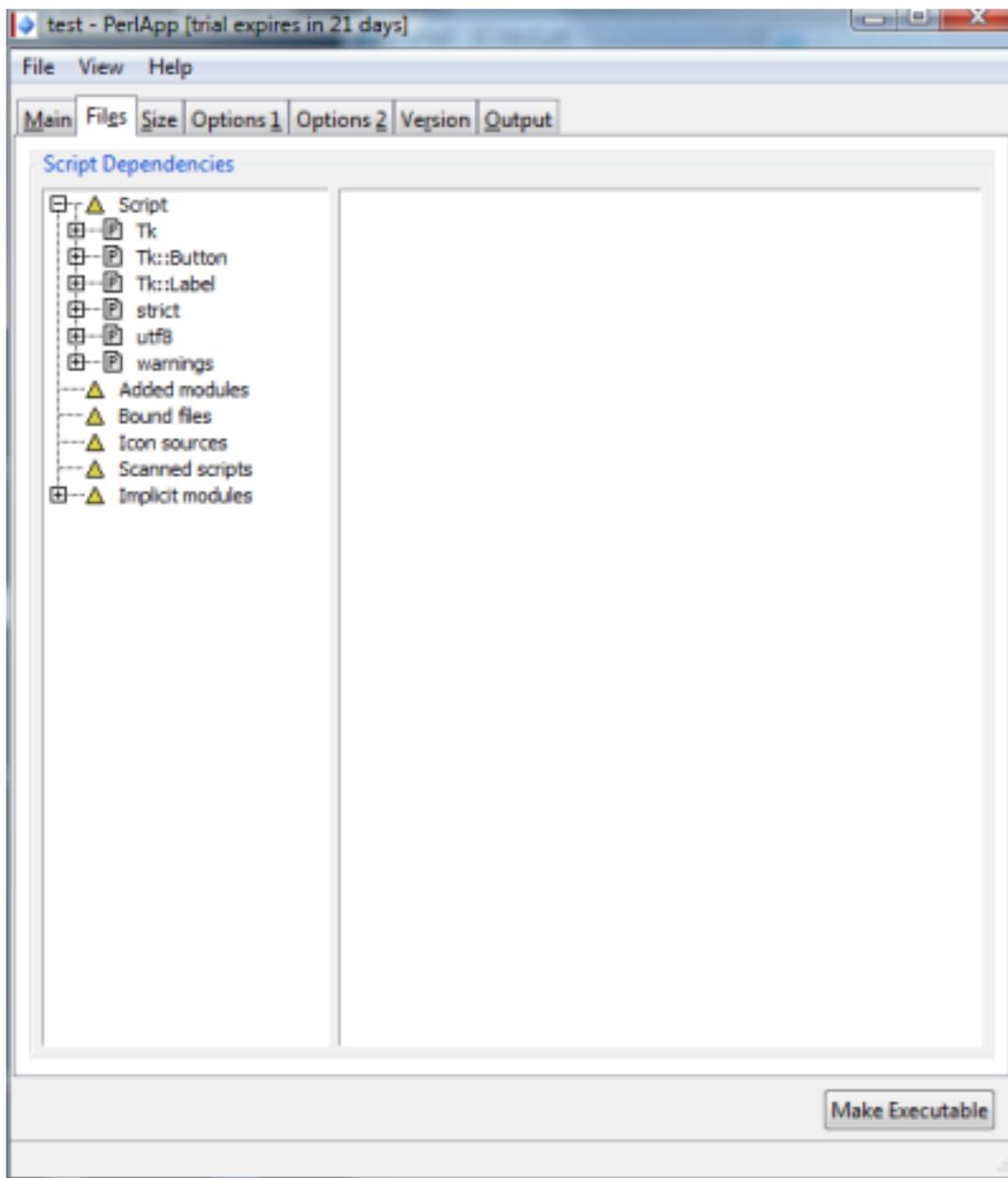
MainLoop;
```

Ouvrons PerlApp et configurons-le !

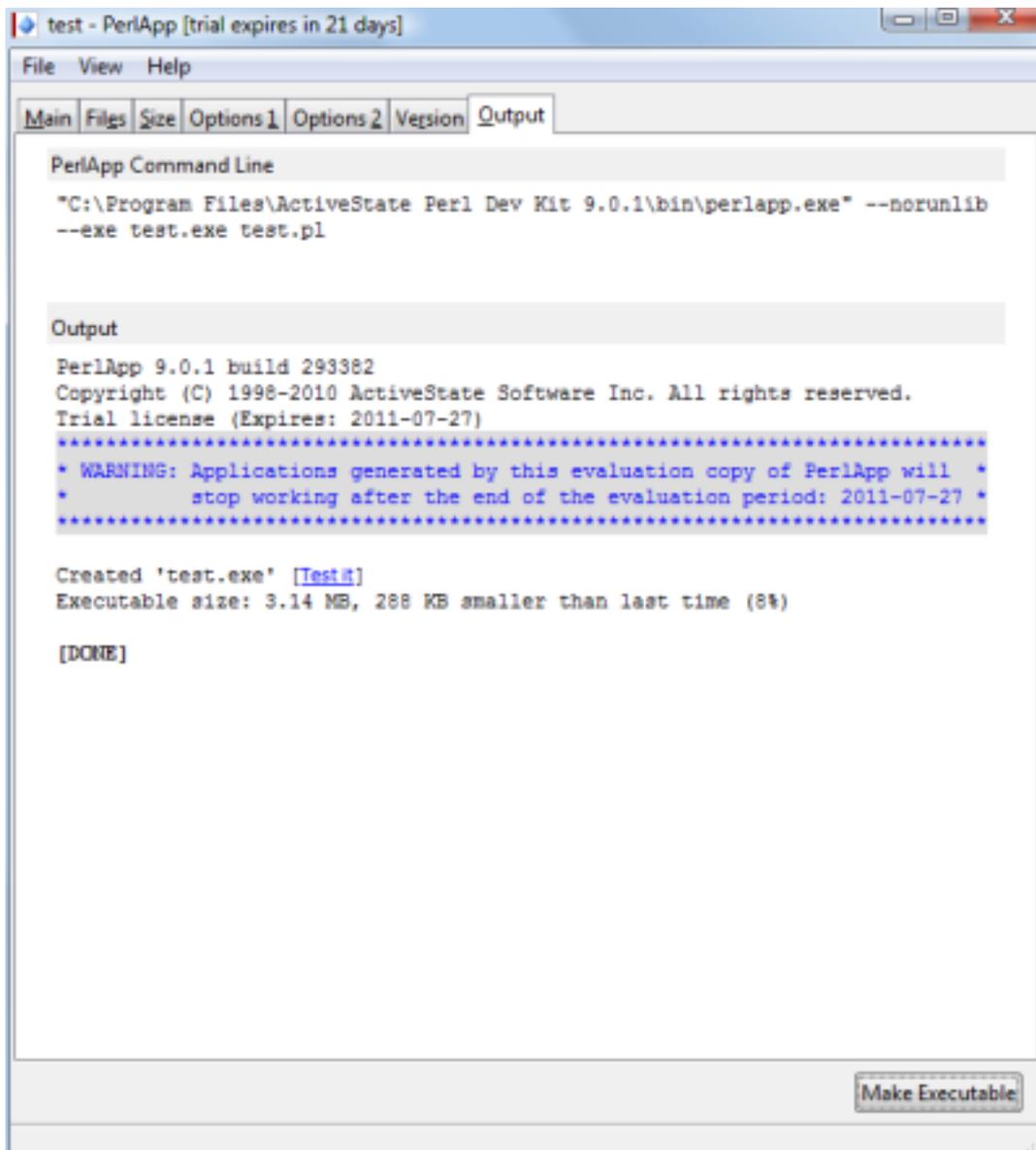
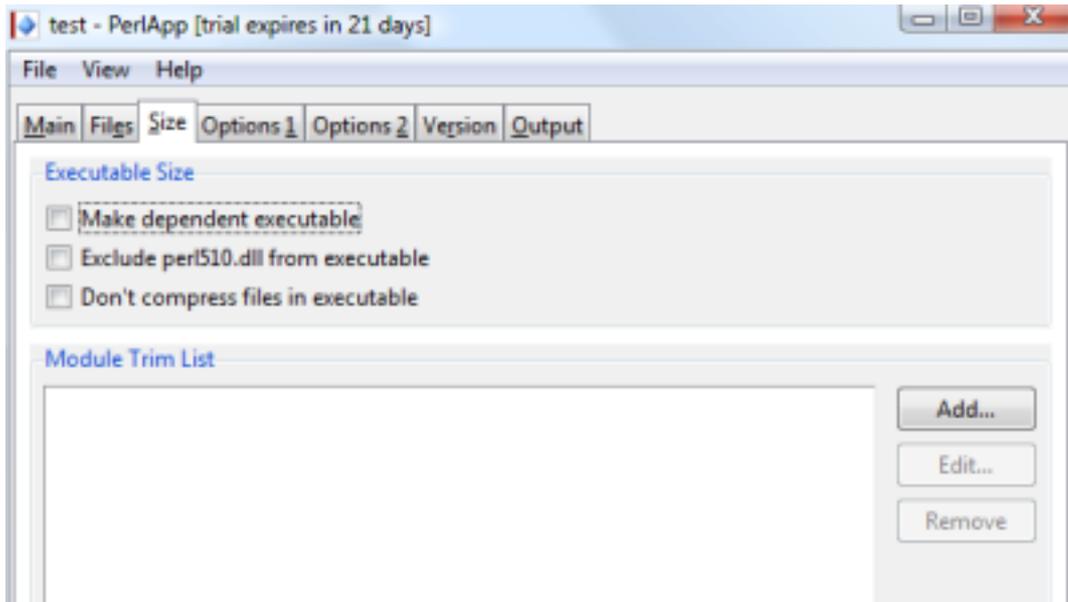
L'onglet « main » permet de spécifier son programme Perl ou son projet. Vous pouvez également modifier le nom de l'exécutable, rajouter des répertoires du @INC.

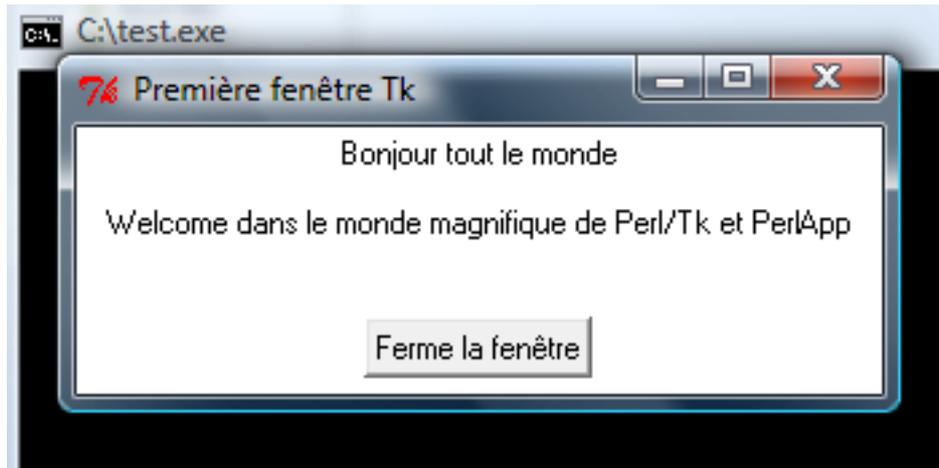


Lorsque vous sélectionnez votre programme Perl, PerlAPP scanne le programme pour y rechercher toutes les dépendances. Vous le voyez dans l'onglet « Files ».



Quelquefois, vous aurez besoin d'ajouter vous-même des modules, mais ce n'est pas compliqué. Il suffit de faire un clic droit sur « Added modules » et d'ajouter le nom de son module. Vous pouvez exclure la dll de Perl dans l'onglet « Size ». Cela permet de moins alourdir l'exécutable, mais dans ce cas, Perl devra être installé sur la machine hôte. Pour rajouter des informations à propos de votre exécutable (numéro de version, nom de votre société...), tout est dans l'onglet Version. Une fois votre configuration terminée, vous n'avez plus qu'à cliquer sur le bouton « Make Executable ». C'est vraiment très simple ! Voici quelques captures d'écran des configurations et de l'exécution de notre programme généré.





Voilà notre programme ! Tout fonctionne à merveille ! On constate qu'il y a une fenêtre DOS derrière, nous aurions pu empêcher de l'afficher en cochant « Hide console (GUI executable) » dans l'onglet « Options 2 ». Maintenant à vous de jouer, mais c'est payant !

## 2-D - Module PAR

Enfin une solution gratuite ! Le module **PAR::Packer** permet de créer un exécutable à partir de programmes Perl en une ligne de commande. Une fois ce module installé, il faut utiliser l'utilitaire **pp** en ligne de commande ou **tkpp** qui ouvre une fenêtre Tk. Les lignes de commandes permettant d'utiliser pp sont vraiment très simples. Ce module Perl permet de créer un exécutable, d'insérer une icône ou des modules dans le fichier binaire, d'inclure (par défaut) ou d'exclure l'interpréteur Perl, spécifier les informations de l'exécutable...

La plus grande difficulté de ce module n'est pas son utilisation, mais son installation. Nous allons dans cet article essayer de répertorier les solutions pour mener à bien notre projet.

**N.B.** Si vous générez un exécutable sous une plate-forme Linux, il ne sera fonctionnel que sur une plate-forme Linux. Si l'exécutable est généré sous Windows, il ne pourra fonctionner que sous Windows.

### 2-D-1 - Installation du module

Avant de parler installation, listons la plupart des erreurs rencontrées par les utilisateurs novices ou non. Elles apparaissent au moment de l'utilisation de pp (**pp -o test.exe test.pl**) ou à l'exécution de l'exécutable généré sous une machine Windows. Par exemple :

#### Liste d'erreurs fréquemment rencontrées

```

C:\>pp -o test.exe test.pl
'pp' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
=====
C:\>tkpp
'tkpp' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.
=====
Le point d'entrée de procédure Perl_croak_xs_usage est introuvable dans la bibliothèque de liaisons
dynamique perl510.dll.
=====
C:\Perl\site\bin>pp -o C:\test.exe C:\test.pl
Set up gcc environment - 3.4.5 (mingw-vista special r3)
Can't load 'C:/Perl/lib/auto/IO/IO.dll' for module IO: load_file:La procédure spécifiée est introuvable
  at C:/Perl/lib/XSLoader.pm line 70.
  at C:/Perl/lib/IO.pm line 11
Compilation failed in require at C:/Perl/lib/IO/Handle.pm line 266.
BEGIN failed--compilation aborted at C:/Perl/lib/IO/Handle.pm line 266.
Compilation failed in require at C:/Perl/lib/IO/Seekable.pm line 101.
BEGIN failed--compilation aborted at C:/Perl/lib/IO/Seekable.pm line 101.
    
```

### Liste d'erreurs fréquemment rencontrées

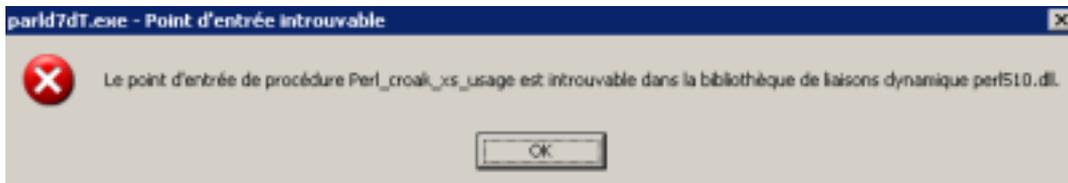
```

Compilation failed in require at C:/Perl/lib/IO/File.pm line 133.
BEGIN failed--compilation aborted at C:/Perl/lib/IO/File.pm line 133.
Compilation failed in require at -e line 448.
pp: Failed to extract a parl from 'PAR::StrippedPARL::Static' to file 'parluQPapwM.exe' at C:/Perl/site/lib/PAR/Packer.pm line 1155, <DATA> line 1.
=====
Le point d'entrée de procédure Perl_sv_2iv_flags est introuvable dans la bibliothèque de liaisons dynamique perl58.dll

Can't load 'C:/Perl/lib/auto/IO/IO.dll' for module IO: load_file:La procédure spécifiée est introuvable at C:/Perl/lib/XSLoader.pm line 64.
at C:/Perl/lib/IO.pm line 11
Compilation failed in require at C:/Perl/lib/IO/Handle.pm line 263.
BEGIN failed--compilation aborted at C:/Perl/lib/IO/Handle.pm line 263.
Compilation failed in require at C:/Perl/lib/IO/Seekable.pm line 101.
BEGIN failed--compilation aborted at C:/Perl/lib/IO/Seekable.pm line 101.
Compilation failed in require at C:/Perl/lib/IO/File.pm line 133.
BEGIN failed--compilation aborted at C:/Perl/lib/IO/File.pm line 133.
Compilation failed in require at -e line 405.
C:/Perl/site/bin/pp: Failed to extract a parl from 'PAR::StrippedPARL::Static' to file 'parlOrCgfpQ.exe' at C:/Perl/site/lib/PAR/Packer.pm line 1149, <DATA> line 1
=====

pp -o test2.exe test2.pl

Compress::Zlib object version 1.22 does not match oader/0x8b1100/DynaLoader.pm line 225.
Compilation failed in require at /loader/0x8b1100/
BEGIN failed--compilation aborted at /loader/0x8b1
Compilation failed in require at par.pl line 252.
=====
C:/toto.exe n'est pas une application Win32 valide
    
```



Si vous avez rencontré des difficultés d'installation ou d'utilisation de pp, vous avez sans doute été confronté à l'une des erreurs listées ci-dessus.

Les premières erreurs sont simples à comprendre et signifient que Windows ne reconnaît pas la commande pp ou tkpp. En fait, lorsque vous installez Perl, deux variables d'environnement pointant vers les répertoires bin de Perl sont placées dans votre « PATH ». Ainsi, lorsque le module PAR::Packer est installé, les programmes pp et tkpp se trouvent dans ce répertoire et Windows ou Linux s'y retrouve. Mais si le module n'est pas correctement installé, pp et tkpp sont introuvables. Pour le reste, c'est uniquement un problème de compilation, de compatibilité entre la version du module installé et votre version de Perl.

- **Pourquoi l'installation de PAR::Packer pose des soucis, notamment sous Windows ?**

Sous Windows et avec ActivePerl (version que j'utilise), les modules s'installent généralement via l'utilitaire ppm. L'un de ces grands avantages est la gestion de toutes les dépendances. Mais pour son bon fonctionnement, il faut qu'il ait à sa disposition plusieurs « repositories ». Malheureusement, en ce qui concerne le module **PAR::Packer**, il y a un souci de version entre celle disponible à ce jour sur le CPAN (1.009) datant du 26 Mars 2011 et celle dans les dépôts. Afin de vous illustrer la complexité et la raison de la difficulté d'installation de ce module via ppm, voici une liste des dépôts disposant de ce module :

Dépôt	Version d'ActivePerl	Version de PAR::Packer
ActiveState	5.8 build 818, 5.8.8 build 825	<b>Aucune</b>

	5.10 build 1000, 5.12	
<b>Bribes</b>	5.6, 5.8.6, 5.8.8 et build 820, 822 5.10, 5.10.1 build 1006	<b>0.991</b>
<b>trouchelle</b>	5.8 5.10 5.12	<b>1.009</b> <b>aucune</b> <b>0.991</b>
<b>Uwinnipeg (ou theoryx)</b>	5.8 5.8.8 5.10.xx 5.12.xx	<b>0.976</b> <b>0.973</b> <b>0.977</b> <b>aucune</b>

Vous remarquerez que les versions du module diffèrent de la dernière disponible sur le CPAN ( **1.009** ). Cela explique les nombreuses complications d'installation.

Donc malheureusement, pour installer la toute dernière version du module PAR::Packer, je vous conseille de le compiler vous-même.

- **Installation du module PAR::Packer**

Afin de tester l'installation, nous allons suivre pas à pas les instructions et tester l'installation du module sur une machine **Windows Vista sous ActivePerl 5.12.4** . Sachez que vous pouvez faire les mêmes manipulations pour une version 5.8 ou plus de Perl. Je dirais même qu'elle pourrait être utilisée pour StrawberryPerl.

Le bon réflexe avant de se lancer est de consulter le **README** du module. Voici les commandes adéquates assez classiques :

```
perl Makefile.PL
make          # ou 'nmake' ou 'dmake' pour Win32
make test
make install
```

Pour les utilisateurs Windows, nous avons besoin d'un **nmake** téléchargeable sur le site de **Microsoft** .

a) Supposons que nous n'ayons pas de connexion Internet

Il faudra de toute façon télécharger le fichier compressé du module « PAR-Packer-1.009.tar.gz » puis trouver le nmake, grâce au lien ci-dessus. Téléchargez le nmake15.exe ! Copiez-le dans un répertoire du PATH : C:\WINDOWS\System32 ou autre. Personnellement, je le mets dans le répertoire bin de Perl (C:\Perl\bin). Ensuite, il faut se placer à l'intérieur du répertoire pour l'exécuter.

```
C:\>cd C:\Perl\bin
C:\Perl\bin>nmake15.exe

PKSFY (R)  FAST!  Self Extract Utility  Version 2.04g  02-01-93
Copr. 1989-1993 PKWARE Inc. All Rights Reserved. Shareware version
PKSFY Reg. U.S. Pat. and Tm. Off.

Searching EXE: C:/PERL/BIN/NMAKE15.EXE
Inflating: NMAKE.ERR
Inflating: NMAKE.EXE
Inflating: README.TXT

C:\Perl\bin>
```

Nous disposons maintenant d'un fichier NMAKE.EXE. En lisant le README.TXT créé, on remarque qu'il correspond à une version 32 bits et qu'il correspond à la version 1.5.

```
NMAKE.EXE 1.50 is a the version of NMAKE that ships with Visual C++
```

```
2.X. It is used to build external projects. It is a 32 bit version of NMAKE that was designed to run on Windows NT version 3.5. It has been extensively used on Windows 95 and requires a 32 bit operating system. It will not work on Windows 3.1.
```

```
NMAKE v1.5's commands are backwards is compatible with NMAKE v1.4.
```

```
Please see your online documentation for details on how to use NMAKE.
```

```
Files included in NMAKE15.EXE:
```

```
NMAKE.EXE    Nmake version 1.50
NMAKE.ERR    Error text file for Nmake.exe. This must be included
              in the same directory as Nmake.exe
README.TXT   This file.
```

Maintenant, mettons-nous dans le répertoire décompressé du module PAR::Packer que nous avons préalablement téléchargé chez un ami et exécutons nos commandes d'installation.

Dernière étape très importante, il est nécessaire d'avoir un compilateur, par exemple MinGW, GCC pour CygWin ou Microsoft Visual C++ utilisé pour ActivePerl. Sans compilateur, inutile de continuer, nous ne pourrions pas concevoir de fichier binaire exécutable via pp.

Installons maintenant un compilateur :

- gcc (gcc-4.6.0-32.exe)

J'ai téléchargé ce fichier et l'ai installé dans le répertoire C:\gcc.

Maintenant, je ferme ma console DOS et la rouvre afin d'avoir mon PATH à jour grâce à gcc. Lançons ensuite le Makefile !

#### Makefile.PL

```
C:\>cd PAR-Packer-1.009

C:\PAR-Packer-1.009>perl Makefile.PL
Set up gcc environment - 4.6.0

It looks like you don't have either nmake.exe or dmake.exe on your PATH,
so you will not be able to execute the commands from a Makefile. You can
install dmake.exe with the Perl Package Manager by running:

    ppm install dmake

Warning: prerequisite Getopt::ArgvFile 1.07 not found.
Warning: prerequisite Module::ScanDeps 1.01 not found.
Warning: prerequisite PAR 1.000 not found.
Warning: prerequisite PAR::Dist 0.22 not found.
Warning: prerequisite Parse::Binary 0.04 not found.
Warning: prerequisite Win32::Exe 0.14 not found.
Prototype mismatch: sub main::prompt: none vs ($;$) at C:/Perl/lib/ExtUtils/MakeMaker.pm line 202
Note (probably harmless): No library found for -lkernel32
Note (probably harmless): No library found for -luser32
Note (probably harmless): No library found for -lgdi32
Note (probably harmless): No library found for -lwinpool
Note (probably harmless): No library found for -lcomdlg32
Note (probably harmless): No library found for -ladvapi32
Note (probably harmless): No library found for -lshell32
Note (probably harmless): No library found for -lOLE32
Note (probably harmless): No library found for -lOLEAUT32
Note (probably harmless): No library found for -lnetapi32
Note (probably harmless): No library found for -luuid
Note (probably harmless): No library found for -lws2_32
Note (probably harmless): No library found for -lmpr
Note (probably harmless): No library found for -lwinmm
Note (probably harmless): No library found for -lversion
```

### Makefile.PL

```
Note (probably harmless): No library found for -lodb32
Note (probably harmless): No library found for -lodbccp32
Note (probably harmless): No library found for -lcomctl32
Note (probably harmless): No library found for -lmsvrt
Writing Makefile for par.exe
Writing Makefile for PAR::Packer
```

On remarque qu'il y a des dépendances à installer au préalable. Cela nous rend la tâche assez difficile.

De plus, assez étrangement, il ne trouve pas notre fichier nmake.exe. Même en le mettant dans le répertoire C:\WINDOWS\System32, le message d'erreur est identique. Il est donc préférable d'abandonner la solution d'installation sans Internet. Le jour où il y aura un package du module installable via ppm, il sera possible de prévoir et organiser une installation sans Internet.

#### b) Supposons que nous avons une connexion Internet

Au lancement du Makefile de PAR::Packer, il détectera automatiquement que nous n'avons pas de nmake et de compilateur et essaiera de nous les installer.

```
C:\PAR-Packer-1.009>perl Makefile.PL

It looks like you don't have a C compiler on your PATH, so you will not be
able to compile C or XS extension modules. You can install GCC from the
MinGW package using the Perl Package Manager by running:

    ppm install MinGW

Fetching 'PAR-Packer-1.009-MSWin32-x86-multi-thread-5.12.4.par' from www.cpan.org... Fetching failed:
No compiler found, won't generate 'script/parl.exe'!
The required 'nmake' executable not found, fetching it...
Fetching 'Nmake15.exe' from download.microsoft.com... done!
Warning: prerequisite Getopt::ArgvFile 1.07 not found.
Warning: prerequisite Module::ScanDeps 1.01 not found.
Warning: prerequisite PAR 1.000 not found.
Warning: prerequisite PAR::Dist 0.22 not found.
Warning: prerequisite Parse::Binary 0.04 not found.
Warning: prerequisite Win32::Exe 0.14 not found.
Writing Makefile for PAR::Packer
```

Le Makefile a noté que nous n'avons pas de nmake et compilateur. Il nous l'a donc installé et nous demande d'installer un compilateur gcc à travers MinGW. De plus, il détecte qu'il y a plusieurs modules à installer. Cela nous oblige une fois de plus à installer les modules prérequis les uns à la suite des autres. Nous allons passer par l'utilitaire **CPAN** (comme sous Linux/UNIX) qui est également capable de gérer les dépendances.

#### - méthode CPAN

Comme précisé dans l'article « [Installation des modules Perl CPAN](#) », afin de disposer de la coloration syntaxique toujours appréciée, installons les modules **Term::ANSIColor** et **Win32::Console::ANSI** à travers ppm (bien évidemment, nous aurions pu l'installer via CPAN).

```
C:\>ppm install Term-ANSIColor
No missing packages to install

C:\>ppm install Win32-Console-ANSI
No missing packages to install

C:\>
```

Les deux modules sont déjà présents dans notre version d'ActivePerl 5.12.4. Installons maintenant PAR::Packer !

Au lancement de **cpan** , ce dernier nous installe le compilateur gcc de MinGW (version 5.1.4.1) et le dmake 4.11.2008107.

#### cpan

```
C:\>cpan

It looks like you don't have a C compiler and make utility installed. Trying
to install dmake and the MinGW gcc compiler using the Perl Package Manager.
This may take a a few minutes...

Downloading MinGW-5.1.4.1...done
Downloading dmake-4.11.20080107...done
Unpacking MinGW-5.1.4.1...done
Unpacking dmake-4.11.20080107...done
Generating HTML for MinGW-5.1.4.1...done
Generating HTML for dmake-4.11.20080107...done
Updating files in site area...done
1070 files installed

Please use the `dmake` program to run commands from a Makefile!

Set up gcc environment - 3.4.5 (mingw-vista special r3)

cpan shell -- CPAN exploration and modules installation (v1.9600)
Enter 'h' for help.

cpan>
```

Installons maintenant PAR::Packer :

#### Installation de PAR::Packer

```
cpan> install PAR::Packer
Going to read 'C:\Perl\cpan\Metadata'
Database was generated on Fri, 24 Jun 2011 18:27:17 GMT
Fetching with LWP:
http://ppm.activestate.com/CPAN/authors/01mailrc.txt.gz
Error reading 'C:\Perl\cpan\FTPstats.yml': Alert: While trying to 'parse' YAML file
'C:\Perl\cpan\FTPstats.yml'
with 'YAML::XS' the following error was encountered:
Usage: YAML::XS::LibYAML::Load(yaml_sv) at C:\Perl\lib\YAML\XS.pm line 48.

Stashing away as 'C:\Perl\cpan\FTPstats.yml.1310083042' to prevent further interruptions. You may want
to remove that file later.
Could not rename: Permission deniedGoing to read 'C:\Perl\cpan\sources\authors\01mailrc.txt.gz'
.....DONE
Fetching with LWP:
http://ppm.activestate.com/CPAN/modules/02packages.details.txt.gz
Error reading 'C:\Perl\cpan\FTPstats.yml': Alert: While trying to 'parse' YAML file
'C:\Perl\cpan\FTPstats.yml'
with 'YAML::XS' the following error was encountered:
Usage: YAML::XS::LibYAML::Load(yaml_sv) at C:\Perl\lib\YAML\XS.pm line 48.

Stashing away as 'C:\Perl\cpan\FTPstats.yml.1310083048' to prevent further interruptions. You may want
to remove that file later.
Could not rename: Permission deniedLockfile removed.

C:\>
```

Nous obtenons un message d'erreur qui ressemble à :

**Stashing away as 'C:\Perl\cpan\FTPstats.yml.1310083048' to prevent further interruptions. You may want to remove that file later.**

**Could not rename: Permission denied Lockfile removed.**

Lorsque vous obtenez ce message d'erreur, quel que soit le module que vous installez par le biais du CPAN, ce n'est pas grave ! Il suffit de relancer en utilisant la flèche vers le haut du clavier (parfois il faut relancer deux fois et ça finit par passer). Si malgré diverses tentatives vous n'y arrivez pas, il faudra supprimer le fichier **C:\Perl\cpan\FTPstats.yml** à chaque fois que le message apparaît. Sous Windows Vista, une solution pour éviter ces messages est d'ouvrir une fenêtre DOS via l'option « Exécuter en tant qu'administrateur ».

L'installation des modules dont dépend PAR::Packer est très longue, soyez patient. Il se peut également que vous ayez une erreur d'installation de Win32-EXE essentiel à pp, il faut à tout prix que ce module soit correctement installé avant d'installer PAR::Packer.

- **C'est bien beau tout ça, mais quelle est la bonne méthode à utiliser alors ?**

La meilleure méthode est à mon avis d'utiliser **CPAN** et **ppm**. En lisant le programme Makefile.PL du module PAR::Packer, j'ai listé les modules prérequis. Ensuite, je les ai cherchés via ppm afin de les installer, puis j'installe PAR::Packer via l'utilitaire cpan. Pour faire toutes ces installations, j'ai écrit un programme permettant de lancer l'installation de tous ces modules.

#### Installer tout-en-un

```
#!/usr/bin/perl
#=====
# Author : djibril
# Date   : 22/02/2012
# Main   : Installation du module PAR::Packer et ses dépendances
#=====
use Carp;
use strict;
use CPAN;
use English '-no_match_vars'; # Importer des variables prédéfinies
use ExtUtils::Installed;

my @modules_a_installer_ppm
  = ( 'Term-ANSIColor', 'Module-ScanDeps', 'PAR-Dist', 'PAR', 'Getopt-ArgvFile', 'Parse-Binary', 'IO-Compress', 'Archive-Tar' );

print "=====\nOS : $OSNAME\n=====\n\n";

# Windows System
if ( $OSNAME eq 'MSWin32' ) {
  push @modules_a_installer_ppm, 'Win32-Exe';
  push @modules_a_installer_ppm, 'Win32-Process';
  unshift @modules_a_installer_ppm, 'Win32-Console-ANSI';
  unshift @modules_a_installer_ppm, 'MinGW';

  # Installation de modules via ppm
  foreach my $module (@modules_a_installer_ppm) {
    print "\n\n[ppm install $module]\n\n";
    system "ppm install $module";
  }
  @modules_a_installer_ppm = ('PAR::Packer');
}
else {

  # Other OS : MAC OS, Linux...
  foreach (@modules_a_installer_ppm) { s{-}{:}g; }
  push @modules_a_installer_ppm, 'PAR::Packer';
}

# Installation de modules via CPAN
print "install CPAN and reload cpan\n\n";
CPAN::Shell->install('CPAN'); # Mise à jour du cpan
CPAN::Index->reload;
foreach my $module (@modules_a_installer_ppm) {
  print "- Installation de $module\n";
  CPAN::Shell->install($module);
}

__END__
```

Vous n'avez plus qu'à lancer ce programme ! ppm installera facilement et rapidement les modules puis cpan prendra le relais. Si vous obtenez le message :

**Stashing away as 'C:\Perl\cpan\FTPstats.yml.1310083048' to prevent further interruptions. You may want to remove that file later.**

**Could not rename: Permission denied Lockfile removed.**

N'hésitez pas à relancer le programme plusieurs fois.

## 2-D-2 - Utilisation du module

Le module PAR::Packer maintenant installé correctement, il ne nous reste plus qu'à utiliser pp ou tkpp en ligne de commande. Pour que vous puissiez comprendre facilement, nous allons convertir un programme Perl « test.pl » dont le but est d'afficher une fenêtre Tk qui nous dira bonjour. Le code Perl/Tk est le suivant :

```
#!/usr/bin/perl
use warnings;
use strict;
use Tk;
use utf8;

# Création de la fenêtre
my $fenetre = new MainWindow(
    -title      => 'Fenêtre Tk',
    -background => 'white',
);

# Taille minimale de ma fenêtre
$fenetre->minsize( 300, 100 );

my $message_accueil = "Bonjour tout le monde\n\n"
    ."Welcome dans le monde magnifique de Perl/Tk et pp/tkpp\n\n";

# Affichage d'un texte
my $label_accueil = $fenetre->Label(
    -text      => $message_accueil,
    -background => 'white',
)->pack();

# Affichage d'un bouton pour fermer la fenêtre
my $bouton = $fenetre->Button(
    -text      => 'Ferme la fenêtre',
    -command => sub { exit; },
)->pack();

MainLoop;
```

Pour utiliser ce programme, vous devrez installer le module Tk. Étudions maintenant les deux utilitaires à notre disposition.

- Utilitaire pp

Tout se joue en ligne de commande. Il y a une multitude d'options possibles afin de paramétrer au mieux la création de votre exécutable. Nous n'allons pas répertorier toutes les options dans cet article, mais je vous montrerai celles qui sont indispensables pour commencer.

Options pp	Explications	Exemples de ligne de commande
-o ou --output=FICHIER	Pour spécifier le nom de notre exécutable. Il génère	pp -o test.exe test.pl pp -o test.out test.pl

	un fichier d'extension <b>.exe</b> sous Windows ou <b>.out</b> .	pp -o test test.pl pp --output=test.exe test.pl
-v ou --verbose[=Nombre]	Mode verbeux (nombre compris entre 1 et 3). Par défaut, nombre=1. Plus le nombre est grand, plus de détails seront affichés.	pp -o test.exe test.pl -v pp -o test.exe test.pl -vv ( ou --verbose=2 ou -v 2) pp -o test.exe test.pl -vvv (ou --verbose=3 ou -v 3)
-V ou --version	Affiche la version et le copyright du programme.	pp -o test.exe test.pl -V
-g ou --gui	Permet de créer un exécutable qui n'ouvrira pas de console Windows en arrière-plan. N'utilisez pas cette option lorsque vous êtes en développement sous peine de ne pas voir les messages d'erreurs ou d'avertissements de votre programme.	pp -o test.exe test.pl -g pp -o test.exe test.pl --gui
-h ou --help	Affiche l'usage basique de pp.	pp -help (ou -h)
-i ou --icon=FICHIER	Associe une icône à votre exécutable. L'icône doit être au format <b>.ico</b> , <b>.exe</b> ou <b>.dll</b> . Pour qu'elle soit bien lisible, elle doit être à la dimension 16x16. Cette option est ignorée de pp sous les systèmes non Windows si l'option -p est utilisée.	pp -o test.exe test.pl -i « C:\dossier1\icone.ico »
-p ou --par	Crée une archive PAR seulement. Le fichier résultant ne sera donc plus un fichier binaire exécutable.	pp -p test.pl (crée un fichier test.par)
-d ou --dependent	Réduit la taille de votre exécutable en n'incluant pas l'interpréteur Perl. Dans ce cas, l'exécutable fonctionnera uniquement sur les plates-formes ayant Perl d'installé car il aura besoin de trouver perl5x.dll ou libperl.so.	pp -o test.exe test.pl -d
-M ou --module	Permet de rajouter des modules dans votre exécutable. Très utile lorsque pp ne réussit pas à scanner tous les modules nécessaires à votre application. Le module doit avoir une extension .pm, .ix ou .al. En fait, inclure par exemple le module Tk::Chart veut dire qu'il ira chercher le module	pp -o test.exe test.pl -M Tk::Chart -M Net::FTP

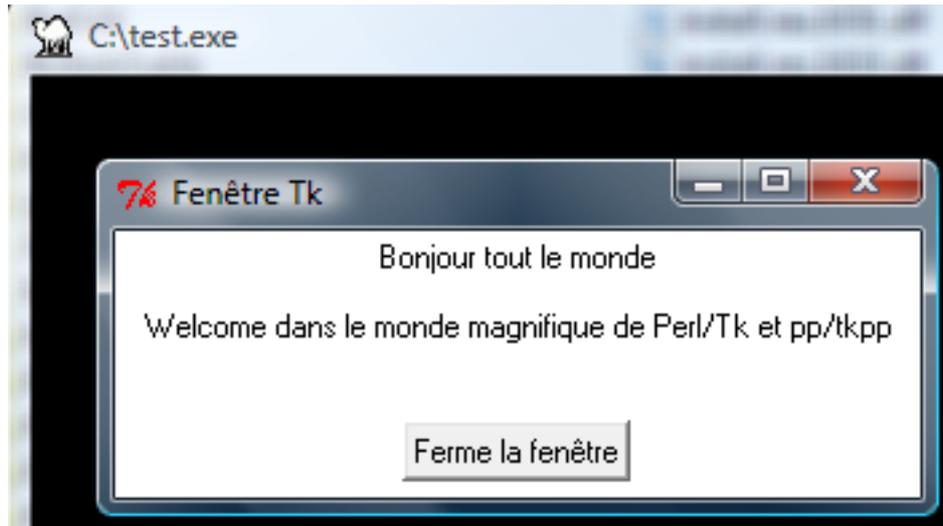
	Tk/Chart.pm pour effectuer les scans.	
-a ou -addfile=FICHIER  RÉPERTOIRE	Ajoute un fichier ou répertoire (de façon récursive tous les fichiers seront inclus) dans le package dans le répertoire / à l'intérieur du package. On peut changer l'emplacement via le séparateur « ; ».	pp -o test.exe test.pl -a fichier1 -a fichier2 (fichier dans le répertoire /) pp -o test.exe test.pl -a « fichier1;/tutu/fichier1 » -a « fichier2;/tutu/fichier2 »
-l ou --link	Ajouter une librairie partagée (dll).	pp -o test.exe test.pl -l « C:\dossier\lib.dll » pp -o test.exe test.pl -l ncurses (signifie -l ncurses.so)
-N ou --info=CLEF=VALEUR	Rajoute des informations propres à l'exécutable sous forme de clef valeur. Sous Windows, les clefs sont les suivantes : <b>Comments</b> , <b>CompanyName</b> , <b>FileDescription</b> , <b>FileVersion</b> , <b>InternalName</b> , <b>LegalCopyright</b> , <b>LegalTrademarks</b> , <b>OriginalFilename</b> , <b>ProductName</b> , <b>ProductVersion</b> .	pp -o test.exe test.pl -N Comments="Mon commentaire" -N CompanyName="Nom compagnie" -N FileDescription="Description" -N FileVersion=7.8.9.1 -N InternalName="Nom interne" -N LegalCopyright="Mon copyright" -N LegalTrademarks="trade" -N OriginalFilename="Nom original" -N ProductName="Nom produit" -N ProductVersion=1.2.3.4
-L ou --log=FICHIER	Les logs de la création du package sont mis dans le fichier log.	pp -o test.exe test.pl -l « erreur.log »

Voilà, vous avez toutes les options nécessaires pour générer un fichier exécutable à partir de votre programme Perl ! Il existe d'autres options, référez-vous à la documentation officielle du module. En ce qui concerne notre programme, voici la commande que nous pouvons exécuter .

Génération de notre exécutable.

```
pp -o test.exe test.pl -v
```

Nous avons choisi de laisser une fenêtre en arrière-plan. Voici le résultat en image :



Magique non ?

- Utilitaire **tkpp**

Comme tkpp ouvre une fenêtre graphique, il va de soi que vous devez installer Tk et d'autres modules Tk pour utiliser cet utilitaire. Il vous le dira ! À ce stade de l'article, vous savez parfaitement comment procéder pour l'installer. La fenêtre fournie par tkpp permet de paramétrer toutes les options du module **PAR::Packer** (depuis la version **1.013**). Voici le rendu de la fenêtre de configuration :

ation

2) Tkpp is a GUI frontend to pp, which can turn perl scripts into stand-alone PAR files, perl scripts or executables.

Options | **Information** | Size | Other Options | Output

Information option

Source file

Output file (--output, --par, --perlscrip)

Icon (--icon)

Log file (--log)

Perl interpreter path (perl)

Perl script path (pp)

Perl dependencies (--compile, --execute, --noscan)

Verbose mode (--verbose)

Run in console Windows. (--gui)

Cache detected dependencies. Creates FILE unless present. This will speed up the scanning process on subsequent runs. (--cachedeps)

**Display command line**

Pour lancer l'ouverture de cet utilitaire, tapez la commande **tkpp** dans une console DOS.

### 3 - Liens utiles

Voici quelques liens utiles cités dans cet article :

- **perlcc**
- **Perl2Exe**
- **PerlApp**
- **PAR**
- **PAR::Packer**
- **pp, tkpp et Tk**
- <http://download.microsoft.com/download/vc15/Patch/1.52/W95/EN-US/nmake15.exe> issu du site de Microsoft

## 4 - Conclusion

Dans cet article, j'espère avoir réussi à vous montrer et expliquer les différentes méthodes permettant de générer un fichier exécutable à partir de sources Perl. Il existe des solutions payantes et gratuites (CPAN). Vous avez pu voir les difficultés d'installation du module Perl PAR::Packer et à présent, vous avez toutes les cartes en main pour l'installer sur votre PC, serveurs...

Si vous avez des remarques, corrections, suggestions ou même des compliments, n'hésitez pas à les faire, c'est toujours tout aussi agréable !

## 5 - Remerciements

Je tiens à remercier **stoyak** et **ClaudeLELOUP** pour la lecture de cet article !