

# Utiliser le CPAN en environnement hostile

Journées Perl 2009

2009-06-12

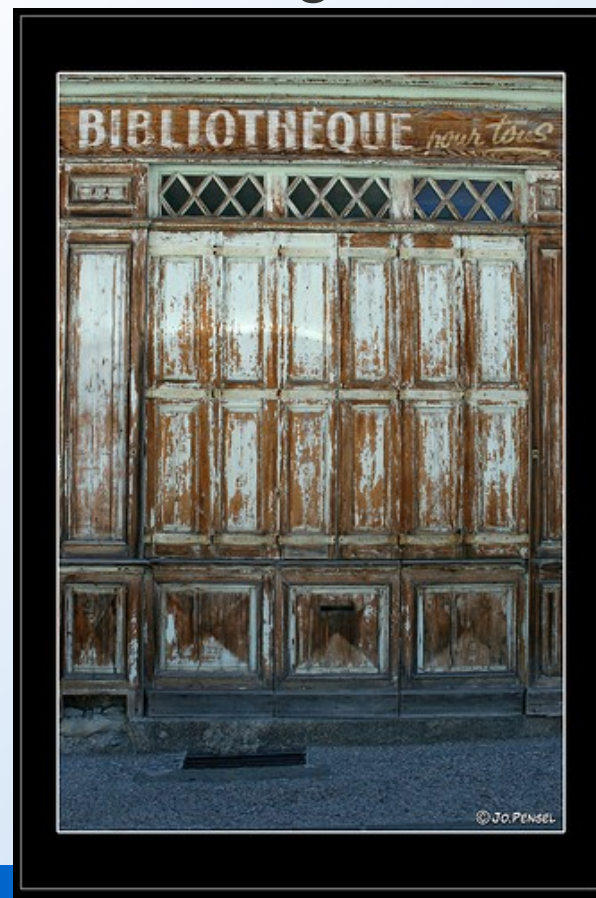


**Olivier Mengué**  
Paris.pm  
dolmen@cpan.org  
<http://o.mengue.free.fr/>

# Utiliser CPAN

---


- 15937 modules
  - Certains sont mêmes utiles !
- Les mises à jour des modules du core
- Des modules testés sur de multiples plateformes, grâce aux CPAN Testers
- POD2::FR : la traduction des docs Perl
  - Merci Paul !



- Vous donne pour chaque module :
  - Les dernières versions
    - Récente ? Module maintenu ?
    - voir backpan pour les autres
  - La doc en ligne (perldoc)
  - La navigation dans la distribution
    - Changes : les modifications
    - t/ : la suite de test. Quelle couverture ?
  - Les bugs ouverts sur rt.cpan.org
    - Activité récente de la part du mainteneur ?
    - Bugs sur votre plateforme ?
  - Les rapports CPANTesters
    - donnent rapidement une idée de la portabilité du module
  - Les dépendances

Olivier 'dolmen' Mengüé > POE-Component-Schedule-0.01 [permalink](#)

### POE-Component-Schedule-0.01

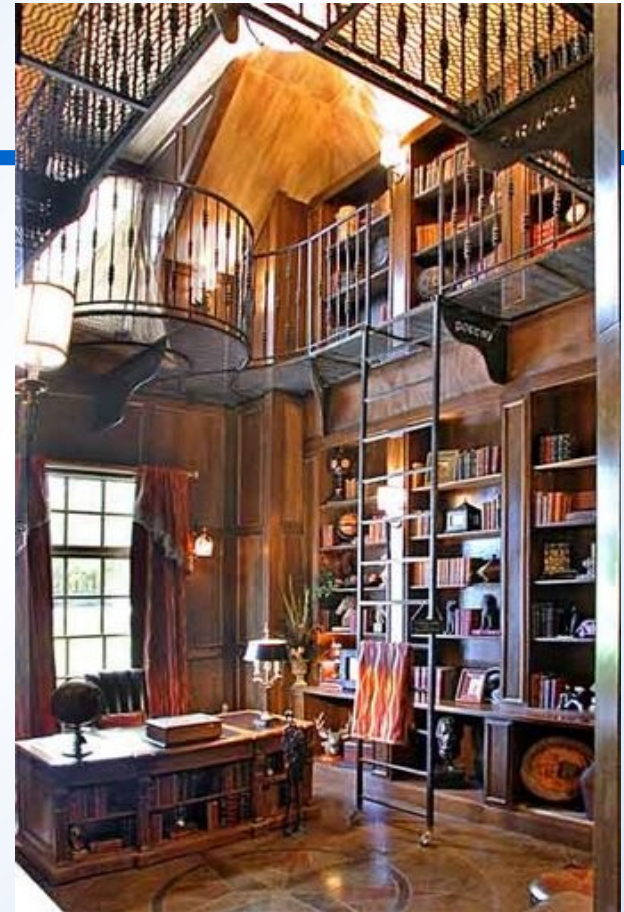
<b>This Release</b>	POE-Component-Schedule-0.01	<a href="#">[Download]</a>	05 Jun 2009	
		<a href="#">[Browse]</a>		
<b>Links</b>	<a href="#">[ Discussion Forum ]</a> <a href="#">[ View/Report Bugs ]</a> <a href="#">[ Website ]</a> <a href="#">[ Dependencies ]</a> <a href="#">[ Other Tools ]</a>			
<b>Repository</b>	<a href="http://poe-component-schedule.googlecode.com/svn/trunk/">http://poe-component-schedule.googlecode.com/svn/trunk/</a>			
<b>CPAN Testers</b>	PASS (49) NA (11) <a href="#">[ View Reports ]</a> <a href="#">[ Perl/Platform Version Matrix ]</a>			
<b>Rating</b>	☆☆☆☆ (0 Reviews) <a href="#">[ Rate this distribution ]</a>			
<b>License</b>	Perl ( <a href="#">Artistic</a> and <a href="#">GPL</a> )			
<b>Special Files</b>	<a href="#">Build.PL</a> <a href="#">MANIFEST</a> <a href="#">Makefile.PL</a> <a href="#">Changes</a> <a href="#">META.yml</a> <a href="#">README</a>			

#### Modules

<a href="#">POE::Component::Schedule</a>	Schedule POE events using DateTime::Set iterators	0.01
--	---	------

# CPAN.pm

- Le shell CPAN standard pour
  - Chercher
  - Télécharger
  - Extraire
  - Construire
  - Tester
  - Installer
- Utilise :
  - des outils externes : wget, curl, unzip, gzip, tar, patch
  - ou du Perl : Archive::Tar, Compress::Zlib...
- Alternative : CPANPLUS
  - Mais pas de distroprefs :(



# En environnement hostile ?

---

- Un vieux Perl
- Une plateforme exotique
- Des droits restreints
- Un accès réseau difficile

# Vieux perl

---

- 5.6
  - Très hostile !
- 5.8
  - Courant
- 5.10
  - Le rêve !
  - Module::Build inclus
- Vieux CPAN.pm
- Vieux ExtUtils::MakeMaker
- Pas de Module::Build



# Une plateforme exotique

---

- Peu de support de la communauté
  - Pas de cpantesters
- Pas de compilateur C
  - Trop hostile !
- Outils manquant pour utiliser CPAN.pm
  - Pas de curl, wget
  - Pas de LWP
  - Pas de GNU tar
    - Problème avec certaines archives (long chemins)
- Exemples :
  - Unix propriétaires
- Windows n'en fait pas partie grâce à StrawberryPerl

# Des droits restreints

---

- Sur Unix, pas root
  - Impossible d'installer dans les destinations standard :
    - /usr/lib/perl5
    - /usr/lib/perl5/\$arch
    - /usr/local/lib
    - ...



# Accès réseau difficile

---



## Accès réseau difficile

---

- CPAN.pm a besoin :
  - de l'index des auteurs
  - de l'index des distributions
  - des distributions
- Sans Internet :
  - CPAN::Mini, pour construire votre miroir sur un CD
- Internet mais via proxy
  - `export http_proxy=http://proxy:8080/`
  - Dans le shell CPAN : `o conf http_proxy`
- Pas d'accès au proxy ou pas de DNS
  - Établir un tunnel SSH vers le proxy pour faire passer les requêtes par votre poste

## Comment faire sans LWP, wget, curl, lynx ?

---

- Utiliser un mini agent HTTP émulant l'interface wget/curl/lynx
- Shell CPAN :
  - o conf lynx
  - o conf wget
  - o conf curl

# mini-lynx.pl

```
# Olivier Mengué
use strict;
use warnings;
use IO::Socket;

unless (@ARGV == 2 && $ARGV[0] eq '-source') { die "usage: $0 -source <URL>" }

my $proxy_url = $ENV{http_proxy};
my ($proxy_host, $proxy_port) = $proxy_url =~ m|http://([^:/]*):(\d+)/?$|;

sub GET($)
{
    my $url = shift;
    my $remote = IO::Socket::INET->new( Proto    => "tcp",
                                       PeerAddr => $proxy_host,
                                       PeerPort => $proxy_port,
                                       )
        or die "cannot connect to http proxy on $proxy_host";
    binmode($remote);
    $remote->autoflush(1);
    print $remote "GET $url HTTP/1.0\r\n\r\n";
    my $status_line = <$remote>;
    if ($status_line !~ m|^HTTP/1\.[01] (\d+) (.*)\r|) {
        close $remote;
        print STDERR "Invalid HTTP response!\n";
        return 500;
    }
    my $status = $1+0;
    my $message = $2;

    while ( <$remote> ) { chomp; last if $_ eq "\r"; }
    local $/ = \8192;
    while ( <$remote> ) { print }
    # TODO Handle redirections
    close $remote;
    return $status;
}

exit(GET($ARGV[1]) != 200);
```

## Win32 : curlIE.wsf

---

- Implémente l'interface de cURL en JavaScript
- Utilise le moteur d'IE
  - Donc les paramètres de proxy
  - Permet l'authentification NTLM transparente
- <http://code.google.com/p/curlie/>

# Construire un environnement de build

---

- Configurer CPAN.pm
- Mettre à jour CPAN.pm et ses dépendances

# Configurer CPAN.pm

---

- o conf lynx / curl / wget
- o conf urllist <http://cpan.enstimac.fr/>
- o conf tar
  - Pour préférer GNU tar
- o conf applypatch
  - Éviter sous Win32 !
- o conf commit



## Rappel : installation d'une distrib CPAN

---

- **Makefile.PL (ExtUtils::MakeMaker, Module::Install)**

```
perl Makefile.PL ...  
make  
make test  
make install
```

- **Build.PL (Module::Build)**

```
perl Build.PL ...  
./Build  
./Build test  
./Build install
```

## Configurer CPAN.pm : o conf makepl\_arg

---

- Paramètres de génération du Makefile pour ExtUtils::MakeMaker
- Pour paramétrer le chemin d'installation :
  - INSTALL\_BASE, mais pour EU::MM récent :-(
  - PREFIX : obsolète, ne marchera jamais, dicit MSCHWERN
    - <http://schwern.org/talks/PREFIX/>
  - Donc on doit tout mettre explicitement

## Configurer CPAN.pm : o conf makepl\_arg

---

- INSTALL\_BASE
- INSTALLARCHLIB
- INSTALLSITEARCH
- VENDORARCHEXP
- VENDORLIBEXP
- SITEARCHEXP
- SITELIBEXP
- INSTALLPRIVLIB
- INSTALLBIN
- INSTALLSCRIPT
- INSTALLMAN1DIR, INSTALLMAN3DIR
- INSTALLHTMLDIR (ancien EUMM)
- INSTALLHTML1DIR, INSTALLHTML3DIR (récent)

## Configurer CPAN.pm : o conf makepl\_arg

---

- Et ça ne suffit pas !
- <https://rt.cpan.org/Ticket/Display.html?id=45600>  
« INSTALLHTML[13]DIR are not based on \$(PREFIX) »
- <https://rt.cpan.org/Ticket/Display.html?id=45816>  
« Can't install if existing .packlist is not readable »

## Configurer CPAN.pm : o conf makepl\_arg

---

```
perl -MConfig -e '$p=$ARGV[0];  
$libarch="$p/lib/perl5/$Config{archname}";  
print qq{INSTALL_BASE=$p INSTALLARCHLIB=$libarch  
INSTALLSITEARCH=$libarch VENDORARCHEXP=$libarch  
VENDORLIBEXP=$p/lib/perl5 SITEARCHEXP=$libarch  
SITELIBEXP=$p/lib/perl5  
INSTALLPRIVLIB=$p/lib/perl5 INSTALLBIN=$p/bin  
INSTALLSCRIPT=$p/bin INSTALLMAN1DIR=$p/man/man1  
INSTALLMAN3DIR=$p/man/man3 INSTALLHTMLDIR=none  
INSTALLHTML1DIR=none INSTALLHTML3DIR=none} '
```

## Configurer CPAN.pm : Module::Build

---

- o conf prefer\_installer MB
- o conf buildpl\_arg --install\_base *chemin*

# Mon .shrc/.kshrc/.bashrc

```
makepl_arg() {
    perl -MConfig -e '$p=$ARGV[0];$libarch="$p/lib/perl5/$Config{archname}";print qq{INSTALL_BASE=$p
INSTALLARCHLIB=$libarch INSTALLSITEARCH=$libarch VENDORARCHEXP=$libarch VENDORLIBEXP=$p/lib/perl5
SITEARCHEXP=$libarch SITELIBEXP=$p/lib/perl5 INSTALLPRIVLIB=$p/lib/perl5 INSTALLBIN=$p/bin
INSTALLSCRIPT=$p/bin INSTALLMAN1DIR=$p/man/man1 INSTALLMAN3DIR=$p/man/man3 INSTALLHTMLDIR=none
INSTALLHTML1DIR=none INSTALLHTML3DIR=none}' "$1"
}

mbuildpl_arg() {
    echo "--install_base $1"
}

cpan_conf() {
    [[ -x /usr/local/bin/tar ]] && echo 'o conf tar "/usr/local/bin/tar"'
    cat <<-EOF
o conf auto_commit 0
o conf makepl_arg "$(makepl_arg "$1")"
o conf mbuildpl_arg "$(mbuildpl_arg "$1")"
o conf lynx "$HOME/build-env/bin/mini-lynx.pl"
o conf prefs_dir "$HOME/build-env/distroprefs"
o conf patches_dir "$HOME/build-env/distroprefs"
o conf urllist http://cpan.enstimac.fr/
o conf urllist
o conf term_is_latin 0
o conf commit
EOF
}

alias makepl='perl Makefile.PL $(makepl_arg $HOME/perl)'
alias mbuildpl='perl Build.PL $(mbuildpl_arg $HOME/perl)'
alias cpan-conf='cpan_conf "$HOME/perl"'
```



# Mise à jour de l'environnement de build

---

- ExtUtils::MakeMaker
- Test::Harness
- Test::More
- File::Spec
- IO::Tty
- IO::Handle
- version
- List::MoreUtils
- Scalar::Util
- IO::Pty
- IO::Tty
- Expect
- YAML
- YAML::Syck
- CPAN

## Mise à jour de l'environnement de build

---

- ExtUtils::CBuilder
- ExtUtils::ParseXS
- Module::Build
  
- Archive::Tar
- Compress::Zlib
- LWP

## Mise à jour de l'environnement de build

---

- Pour mémoriser l'ordre et le reproduire, utilisez des Bundle
- Mettre votre dépôt de bundles dans PERL5LIB
- Créer :
  - Bundle::whatever::Build1
  - Bundle::whatever::Build2
  - ...
- Je sépare les étapes, pour quitter le shell CPAN :
  - par précaution pour éviter les effets de bord suite aux montées de version
  - pour reconfigurer CPAN.pm avec les nouvelles options que les modules installés connaissent maintenant
    - Ajouter INSTALL\_BASE pour makepl\_arg
    - Ajouter prefs\_dir, patches\_dir
    - ...

# Bundle/Dolmen/Build1.pm

---

```
package Bundle::Dolmen::Build1;
our $VERSION = 0.01;
1;
__END__
=head1 NAME
```

```
Bundle::Dolmen::Build1 - Outils de construction (1)
```

```
=head1 CONTENTS
```

```
Test::Harness
```

```
Test::More
```

```
ExtUtils::MakeMaker 6.51_02
```

```
=cut
```

## CPAN::Shell : exemple de scénario

---

- `install Module`
  - Échec d'un test
- `look Module`
  - `make test VERBOSE=1`
  - `perl -Ib/lib/lib t/test-en-échet.t`
- Rapportez immédiatement le bug, probablement sur `rt.cpan.org`
  - vérifiez dans la doc du module
  - suivez le lien à partir de `http://search.cpan.org/dist/Dist`

# distroprefs

---

- o conf prefs\_dir
- o conf patches\_dir
- [http://search.cpan.org/dist/CPAN/lib/CPAN.pm#Configuration\\_files](http://search.cpan.org/dist/CPAN/lib/CPAN.pm#Configuration_files)  
)
- Un fichier YAML défini les patches à appliquer à une distribution
  - Nombreux critères de matching
    - Version, OS...
  - Nombreuses actions à appliquer
    - Patch
    - Réponses aux questions (Expect)
    - Remplacement de makepl\_args...

## Distropref : exemple INGY.YAML-0.68.yml

---

---

```
comment: "Fix for RT25434: utf8 support"
```

```
match:
```

```
  distribution: "^INGY/YAML-0.68.tar.gz"
```

```
patches:
```

```
  - "DOLMEN/patches/YAML-0.68-INGY-RT25434.patch"
```



## Distroprefs : stockage des patches

---

- DOLMEN/patches/YAML-0.68-INGY-RT25434.patch
- Local : relatif à l'option patches\_dir de CPAN >= 1.94 (toute dernière)
- Sur CPAN :  
<http://cpan.cict.fr/authors/id/D/DO/DOLMEN/patches/YAML-0.68-INGY-RT25434.patch>

# Patcher

---

- Ecrivez un test plus précis, éventuellement, pour isoler le problème, et l'ajouter au bug
- Préparez un patch
- Créez une distropref
- Installez le module, pour vérifier le bon fonctionnement du patch et de la distropref
- Archivez votre patch et votre distropref (CVS, SVN, Git...)
- Ajoutez le patch au bug
- Distribuez votre patch via CPAN



# makepatch / applypatch

- JV/makepatch-2.03.tar.gz
- Pour des patchs endurcis (checksum)
- 
- o conf applypatch
- 
- Nécessite les binaires diff/patch
  - Pour win32, utilisez ceux de [gnuwin32.sf.net](http://gnuwin32.sf.net)
- Win32 :
  - %HOME%/.makepatch :  
-diff "diff -urp -binary"
  - %HOME%/.applypatch :  
-patch "patch -p0 -N -binary"
  - Des bugs, liés aux sauts de ligne (voir [rt.cpan.org](http://rt.cpan.org))



# cpan-prep-patch

---

- Un outil que j'ai commencé à écrire, pas encore publié
- cpan-prep-patch DOLMEN/POE-Component-Schedule-0.01.tar.gz
- Prépare l'environnement pour :
  - coder votre patch
  - générer le patch
  - générer la distropref



# Pistes à explorer

---

- CPANPLUS
  - Quand il y aura les distroprefs
- pip
  - Encapsule CPAN.pm pour automatiser les installations



# Questions ?



**Olivier Mengué**

[dolmen@cpan.org](mailto:dolmen@cpan.org)  
<http://o.mengue.free.fr/>