

Perl Tipps

Wolfgang Kinkeldei

Warum Perl?

- ausdrucksstark
- getestet
- umfangreich
- universell
- schnell
- u.v.m.

Woher bekommen?

- Mac OS X: vorinstalliert
- Linux: vorinstalliert
- *BSD: vorinstalliert
- Windows: <http://strawberry.com>
- sonst: selbst kompilieren
<http://www.perl.org/get.html>

Warum selbst installieren?

- Erfordernisse von Modulen
- Einheitlichkeit verschiedener Systeme
- Features neuer Versionen
- Geschwindigkeit
- zum Testen

die Hardcore Variante

□ Quelltexte herunterladen

```
$ cd perl-5.14.2
```

```
$ sh Configure -de
```

```
$ make
```

```
$ make test
```

```
$ make install
```

Ja Aber...

- evtl. höhere Privilegien erforderlich für:
 - Perl Installation
 - Modul Installation
- Installation „míscht“ sich ins System
 - z.B. nach `/usr/local`

Installation in \$HOME?

□ Theoretisch ganz einfach*

\$ sh Configure -Dprefix=/some/dir

□ Aber es geht noch einfacher
mit perlbrew

* wenn die vielen gestellten Fragen nicht wären,
aber das steht ja nur im Kleingedruckten :-)

***brew

□ homebrew (diverse Software):

<http://mxcl.github.com/homebrew>

□ RVM (Ruby):

<https://github.com/wayneeseguin/rvm>

□ perlbrew:

<http://www.perlbrew.pl>

□ pythonbrew:

<https://github.com/utahta/pythonbrew>

perlbrew

□ Installieren

```
$ curl -k -L http://xrl.us/perlbrewinstall | bash
```

```
$ ~/perl5/perlbrew/bin/perlbrew init
```

□ Vor jedem Gebrauch – oder in ~/.profile

```
$ . ~/perl5/perlbrew/etc/bashrc
```

□ Fertig zum Einsatz!

perlbrew nutzen

- verfügbare Versionen anzeigen

```
$ perlbrew available
```

- neues Perl installieren

```
$ perlbrew install perl-5.14.2
```

- Dauerhaft oder in dieser Shell nutzen:

```
$ perlbrew switch perl-5.14.2
```

```
$ perlbrew use perl-5.14.2
```

Stolperfalle Scripte

❑ So geht es schief:

```
#!/usr/bin/perl
```

❑ So klappts!

```
#!/usr/bin/env perl
```

Sicherheit?

- Warum funktioniert perlbrew?

```
$ echo $PATH
```

```
/Users/xxx/perl5/perlbrew/bin:/usr/bin:/bin:/usr/  
sbin:/sbin:/usr/local/bin:/usr/X11/bin
```

- Perlbrew Pfad steht am Anfang!

Perl Module installieren

□ Installationsquelle: CPAN

<http://search.cpan.org> | <http://metacpan.org>

□ Der Klassiker: cpan

□ Alternativ: cpanp

□ Flexibel und einfach: cpanm

cpanm - Installation

- ohne Admin Rechte

```
$ curl -L http://cpanmin.us | perl - App::cpanminus
```

- Oder: via perlbrew

```
$ perlbrew install-cpanm
```

- Ab dann benutzbar

```
$ cpanm Catalyst::Runtime
```

```
$ cpanm /pfad/zu/Xxx-0.01.tar.gz
```

cpanm - Besonderheiten

□ üblicherweise sehr wenig Meldungen

□ Projekt-Abhängigkeiten auflösen

```
cpanm --installdeps .
```

□ Zielverzeichnis für Module angebbar

```
cpanm -L /ziel/pfad Modul::Name
```

□ und viel mehr

```
cpanm --help
```

minicpan

- Bisher: CPAN Module werden von Spiegel-Servern geladen
- Warum nicht selbst Spiegel sein?
 - > 12 GB für alle Module
 - Aber: nur 2 GB für letzte Versionen

minicpan benutzen

- Installieren, z.B. mit cpanm

cpanm CPAN::Mini

- Neueste Versionen vom Spiegel laden

minicpan -l /mein/minicpan -r <http://cpan.noris.de>

- Bei cpanm benutzen

cpanm --mirror /mein/minicpan --mirror-only ...

POD Documentation

```
/Users/wolfgang/proj/MyHomeBinaries — less — 80x24
NAME
DBIx::Class::ResultSet - Represents a query used for fetching a set of
results.

SYNOPSIS
my $users_rs = $schema->resultset('User');
while( $user = $users_rs->next) {
    print $user->username;
}

my $registered_users_rs = $schema->resultset('User')->search({ regist
ered => 1 });
my @cds_in_2005 = $schema->resultset('CD')->search({ year => 2005 }->a
ll());

DESCRIPTION
A ResultSet is an object which stores a set of conditions representing
a query. It is the backbone of DBIx::Class (i.e. the really
important/useful bit).

No SQL is executed on the database when a ResultSet is created, it just
stores all the conditions needed to create the query.
```

```
ResultSet.pdf (Seite 1 von 23)
Vorherige Nächste Zoomen Bewegen Text Auswählen Anmerken Seitenleiste Suchen nach

DBIx/Class/ResultSet.pm

NAME
DBIx::Class::ResultSet - Represents a query used for fetching a set of results.

SYNOPSIS
my $users_rs = $schema->resultset('User');
while( $user = $users_rs->next) {
    print $user->username;
}

my $registered_users_rs = $schema->resultset('User')->search({ regist
ered => 1 });
my @cds_in_2005 = $schema->resultset('CD')->search({ year => 2005
})->all();

DESCRIPTION
A ResultSet is an object which stores a set of conditions representing a query. It is the backbone of
DBIx::Class (i.e. the really important/useful bit).

No SQL is executed on the database when a ResultSet is created, it just stores all the conditions
needed to create the query.

A basic ResultSet representing the data of an entire table is returned by calling resultset on a
DBIx::Class::Schema and passing in a Source name.

my $users_rs = $schema->resultset('User');

A new ResultSet is returned from calling search on an existing ResultSet. The new one will contain all
the conditions of the original, plus any new conditions added in the search call.
```

pod2pdf

Installieren

```
cpanm pod2pdf
```

Und konvertieren

```
pod2pdf /Pfad/Zu/Modul.pm > /pfad/zu/modul.pdf
```

selbst CPAN Autor werden

- Warum?
 - andere profitieren von meinem Code
- Wie?
 - anmelden bei: <http://pause.perl.org>
- Dann eigene Module hochladen
 - per http upload

CPAN Module erstellen

- erfordert Einhaltung einiger Regeln
- von Hand schwierig
- es gibt einige „Helferchen“
 - Module::Starter
 - Extutils::ModuleMaker
 - Dist::Zilla

Dist::Zilla

□ Zentraler Angelpunkt:

dist.ini Datei

□ Steuerung über

ein Kommando

\$ dzil ...

```
dist.ini
name = Catalyst-View-ByCode
version = 0.15
author = Wolfgang Kinkeldei <wolfgang@kinkeldei.de>
license = Perl_5
copyright_holder = Wolfgang Kinkeldei

repository = http://github.com/wki/Catalyst-View-ByCode

[@Classic]

[AutoPrereqs]

[Prereqs]
Catalyst::Runtime = 5.80
Devel::Declare = 0.005007
HTML::Entities = 1.35
HTML::Tagset = 3.10
List::Util = 1.21
MIME::Base64 = 3.07
Moose = 1.0
MooseX::AttributeHelpers = 0.23
UUID::Random = 0.04
YAML = 0.68

Test::Builder = 0.86
Test::Builder::Tester = 1.18
Test::Harness = 3.14
Test::More = 0.86
Test::Exception = 0.27
FindBin = 1.49
|
```

Dist::Zilla benutzen

- Neues Modul beginnen

`dzil new My::Module::Name`

- .tar.gz Datei bauen

`dzil build`

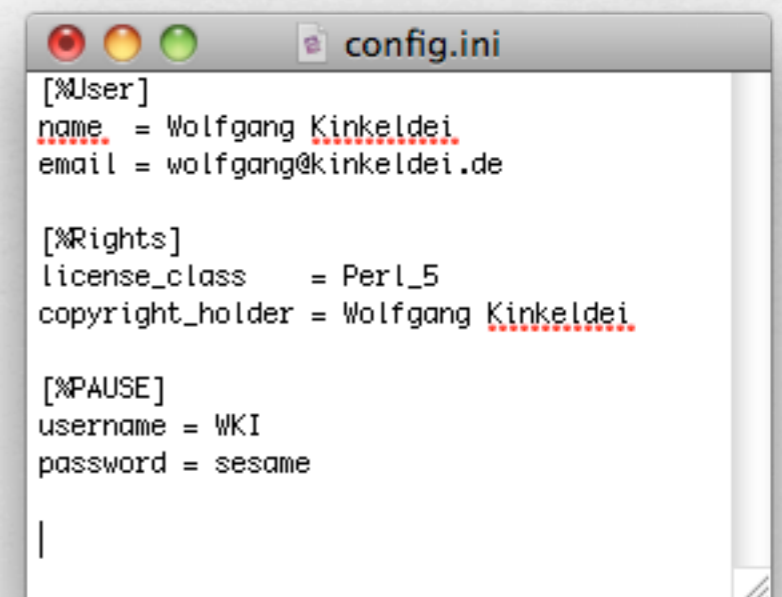
- Tests ausführen

`dzil test`

- CPAN Upload

`dzil release`

`~/ .dzil/config.ini`



```
config.ini
[User]
name = Wolfgang Kinkeldei
email = wolfgang@kinkeldei.de

[Rights]
license_class = Perl_5
copyright_holder = Wolfgang Kinkeldei

[PAUSE]
username = WKI
password = sesame
|
```

__END__

```
' ^_^ ' .print[split//, "" .(($/=q|Cms)+-03467:;<=l)=~tr!C-z -B! -z!)x  
'( o.o )' .$/]->[hex]foreach split qr<>,qq+1ecd039ad65b025b8063475b+ll  
' > ^ < ' .q<!-- Wolfgang Kinkeldei - wolfgang at kinkeldei dot de -->
```


__END__

```
$ cat <<EOF | perl
```

```
' ^_^ ' .print[split//,"".(($/=q|Cms)+-03467:;<=l)=~tr!C-z -B! -z!)x  
'( o.o )' .$/]->[hex]foreach split qr<>,qq+1ecd039ad65b025b8063475b+ll  
' > ^ < ' .q<!-- Wolfgang Kinkeldei - wolfgang at kinkeldei dot de -->
```

```
EOF
```

```
Just another Perl hacker
```