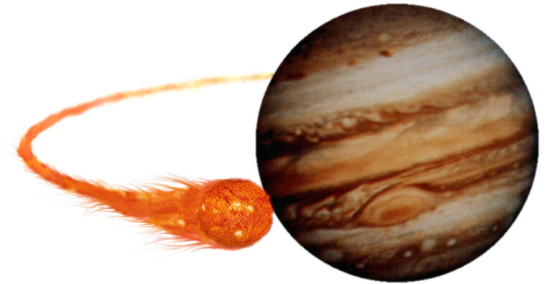


# Perl5::Spartanic

Für Perl auf Android und anderen  
spartanischen Plattformen

Ralf Peine

[www.jupiter-programs.de](http://www.jupiter-programs.de)



# **1. Motivation**

# **Experiment**

Perl Module und Programme auf dem  
Android Tablet entwickeln.

**Das sollte doch nicht so  
schwierig sein?**

**Leider doch!**

# Motivation

## Perl auf Android

- Keine CPAN-Shell
- Kein Standard-Make
- Kein Standard-C-Compiler verfügbar
- Kaum Unix-Commands verfügbar

# Motivation

## Perl auf Android (2)

- Keine IDE
- Kein ctags
- Kein Emacs (mein Lieblingseditor)
- Vim als einziger professioneller Editor

**2. Was nun ?**



Was nun?

Module ohne Compile und Make finden

Auf CPAN existiert leider kein Attribut für:

- Modul muss kompiliert werden
- Modul benötigt unbedingt Make

```
use Perl5::Spartanic;
```

Was nun?

Perl5::Spartanic (2)

Mit "use Perl5::Spartanic" legt man  
folgende Eigenschaften für sein Modul  
fest:

# Was nun?

## Perl5::Spartanic (3)

- Modul muss **NICHT** kompiliert werden
- Modul benötigt **KEIN** Make
- Modul verwendet **AUSSCHLIESSLICH**  
Spartanic-Module
- und **PERL-CORE** Module

```
use Perl5::Spartanic::Strict;
```

# Was nun?

## Perl5::Spartanic::Strict

- Modul muss **NICHT** kompiliert werden
- Modul benötigt **KEIN** Make
- Modul verwendet **AUSSCHLIESSLICH PERL-CORE** Module

# **3. Spartan Lib**

# Perl Lib

## Hilfreiche Core-Module

**Archive::Tar;**

**IO::Compress::Gzip;**



# Spartanic Lib

HTML::Template;	# strict
Trace::Log;	# strict
Alive;	# strict
PQL::Cache;	# strict
Report::Porf;	# strict
Scalar::Validation;	# strict
Perl5::MetaInfo::DB;	

# 4. Spartan IDE

## Perl Module/Class Overview

Mon Feb 1 08:23:36 2016

31 modules / classes listed

# Spartanic Lib

PerlVers	#subs	Name (Methods)	Doc	Location	Namespace	#Line	File
	8	<a href="#">Alive</a>	<a href="#">SPRT</a>	SPRT		12	{SPRT}/Alive.pm
	20	<a href="#">DB</a>	-	SPRT		66	{SPRT}/Log/Trace.pm
	0	<a href="#">Template</a>	<a href="#">SPRT</a>	SPRT	<a href="#">HTML</a>	1	{SPRT}/HTML/Template.pm
	32	<a href="#">Template</a>	<a href="#">SPRT</a>	SPRT	<a href="#">HTML</a>	1059	{SPRT}/HTML/Template.pm
	11	<a href="#">COND</a>	-	SPRT	<a href="#">HTML::Template</a>	1045	{SPRT}/HTML/Template.pm
	1	<a href="#">COND</a>	-	SPRT	<a href="#">HTML::Template</a>	3351	{SPRT}/HTML/Template.pm
	1	<a href="#">DEFAULT</a>	-	SPRT	<a href="#">HTML::Template</a>	3292	{SPRT}/HTML/Template.pm
	1	<a href="#">ESCAPE</a>	-	SPRT	<a href="#">HTML::Template</a>	3372	{SPRT}/HTML/Template.pm
	0	<a href="#">FAQ</a>	<a href="#">SPRT</a>	SPRT	<a href="#">HTML::Template</a>	3	{SPRT}/HTML/Template/FAQ.pm
	1	<a href="#">JSESCAPE</a>	-	SPRT	<a href="#">HTML::Template</a>	3381	{SPRT}/HTML/Template.pm
	2	<a href="#">LOOP</a>	-	SPRT	<a href="#">HTML::Template</a>	1041	{SPRT}/HTML/Template.pm
	2	<a href="#">LOOP</a>	-	SPRT	<a href="#">HTML::Template</a>	3299	{SPRT}/HTML/Template.pm
	1	<a href="#">NOOP</a>	-	SPRT	<a href="#">HTML::Template</a>	3363	{SPRT}/HTML/Template.pm
	3	<a href="#">PRINTSCALAR</a>	-	SPRT	<a href="#">HTML::Template</a>	3400	{SPRT}/HTML/Template.pm
	1	<a href="#">URLESCAPE</a>	-	SPRT	<a href="#">HTML::Template</a>	3390	{SPRT}/HTML/Template.pm
	1	<a href="#">VAR</a>	-	SPRT	<a href="#">HTML::Template</a>	3285	{SPRT}/HTML/Template.pm
	4	<a href="#">Trace</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Log</a>	1	{SPRT}/Log/Trace.pm
	30	<a href="#">Cache</a>	<a href="#">SPRT</a>	SPRT	<a href="#">PQL</a>	11	{SPRT}/PQL/Cache.pm
5.10	19	<a href="#">DB</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Perl5::MetaInfo</a>	13	{SPRT}/Perl5/MetaInfo/DB.pm
	0	<a href="#">Spartanic</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Perl5</a>	19	{SPRT}/Perl5/Spartanic.pm
	2	<a href="#">Porf</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Report</a>	16	{SPRT}/Report/Porf.pm
	24	<a href="#">Framework</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Report::Porf</a>	53	{SPRT}/Report/Porf/Framework.pm
	107	<a href="#">Simple</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Report::Porf::Table</a>	47	{SPRT}/Report/Porf/Table/Simple.pm
	6	<a href="#">AutoColumnConfigurator</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Report::Porf::Table::Simple</a>	25	{SPRT}/Report/Porf/Table/Simple/AutoColumnConfigurator.pm
	10	<a href="#">CsvReportConfigurator</a>	<a href="#">SPRT</a>	SPRT	<a href="#">Report::Porf::Table::Simple</a>	45	{SPRT}/Report/Porf/Table/Simple/CsvReportConfigurator.pm

# Spartanic IDE

## HTML-Infozeile

1	<u>Cache</u>	<u>SPRT</u>	SPRT	<u>LOG</u>
30	<u>Cache</u>	<u>SPRT</u>	SPRT	<u>PQL</u>
10	nn	nnnn	nnnn	nnnnnn

# Spartanic IDE

## Subs (top)



## PQL::Cache

[Home](#)

{SPRT}/PQL/Cache.pm

Mon Feb 1 08:23:37 2016

Documentation: [SPRT](#)

[30 Methods/Subs have been found in .pm file](#)

[Modules/Classes are used by this one.](#)

[Modules/Classes are using this one.](#)

# Spartanic IDE

## Subs (Liste)

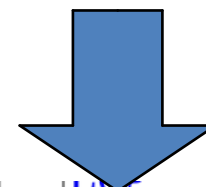


### Methods / Subs of Cache

30 methods/subs have been found by the parser in the \*.

ID	#Line	***	Name
34	595		_find_matches
35	696		_handle_where
13	80		_implement_compare_subs
11	59		_init
26	386		_insert
14	209		_prepare_cache
33	564		_select_columns
28	437		_update_index_key
29	458		build_index_string
15	230		clear
30	481		delete

# Spartanic IDE Namespace



30	<a href="#">Cache</a>	<a href="#">SPRT</a>	SPRT	<a href="#">PQL</a>
10	DD	DDDT	DDDT	DDDT

## PQL

### Home

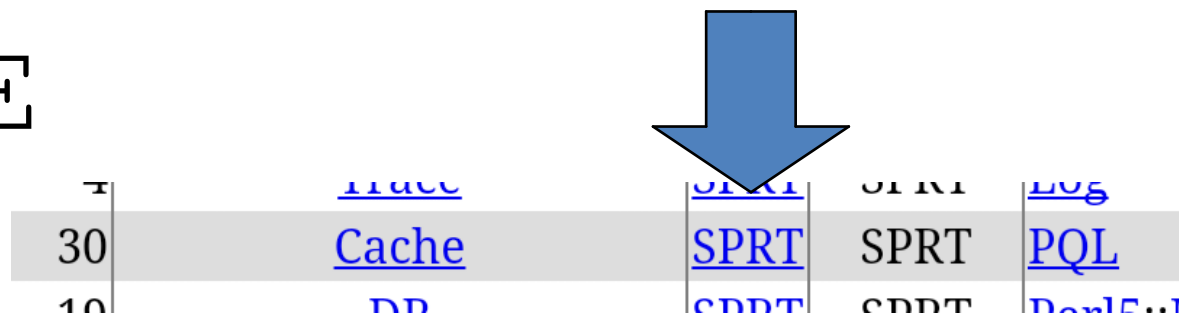
## Modules Of Namespace

Generated: Mon Feb 1 08:23:37 2016

Modules listed: 1

PerlVers	#subs	Name (Methods)	Doc	Location	Namespace	#Line	File
	30	<a href="#">Cache</a>	<a href="#">SPRT</a>	SPRT	<a href="#">PQL</a>	11	{SPRT}/PQL/Cache.pm

# Spartanic IDE Manpage



## NAME

PQL::Cache - A mini in memory database using pure perl and PQL - PerlQueryLanguage -

PQL is similar to DBIx::Class API, so switching to DBIx and SQL databases is possible.

It uses pure Perl (no compilation needed), so you can run it from a USB stick.

## VERSION

This documentation refers to version 0.800 of PQL::Cache

## SYNOPSIS

```
use PQL::Cache qw(:all);
```

### Create Cache

```
my $cache = PQL::Cache->new();
$cache->set_table_definition
    ('person',
     {
         keys    => ['ID'],
         columns => [pname => surname => birth => gender => 'perl_level']
     });
```



**Danke!**

# Wozu validieren ?

```
sub add {  
    my $sum = 0;  
    for (@_) { $sum += $_; }  
    return $sum;  
}
```

# Fazit

# Vergleich Fazit (1)

Für Moose

- `MooseX::Params::Validate`
- `Type::Params`

**Danke!**