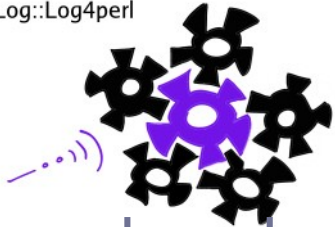


```
use Log::Log4perl qw(:easy);
```

Easy Logging - Eine Einführung

Log::Log4perl



Log::Log4perl ist ein Perl-Port des aus der Java-Welt stammenden Log4j-Systems, das auf **vier** Säulen steht:

- **Logger** lösen Log-Nachrichten aus,
- **Prioritäten** bestimmen, ob sie durchkommen,
- **Layouts** formatieren die Nachrichten und
- **Appender** leiten diese dann an konfigurierbare Ausgabemedien weiter.



**Prioritäten**

Prioritäten steuern, ob ein Logger die ihm übergebenen Nachrichten tatsächlich weitergibt oder unterdrückt.

Log::Log4perl kennt sechs Prioritäten  
(in aufsteigender Reihenfolge):

- Trace
- Debug
- Info
- Warn
- Error
- Fatal

Das Level TRACE kennzeichnet sehr, sehr feinkörnige Ereignisse, feinkörniger als das nächste Level DEBUG.

Beispiel: `TRACE("$bytes bytes transferred");`

Das Level DEBUG kennzeichnet feinkörnige Ereignisse, die nützlich sind, um eine Applikation zu debuggen.

Beispiel: `DEBUG("HTTP get OK");`

Das Level INFO kennzeichnet grobkörnige Ereignisse, die über den Fortschritt des Programms informieren.

Beispiel: INFO("Starting up");



Das Level WARN kennzeichnet Ereignisse, die auf kleine Fehler hinweisen. Das Programm kann aber weiterlaufen.


Beispiel: `WARN("HTTP get failed, retrying");`

Das Level ERROR kennzeichnet Ereignisse, die auf größere Fehler hinweisen. Das Programm kann aber oft noch weiterlaufen.

Beispiel: `ERROR("Out of retries!");`

Das Level FATAL kennzeichnet Ereignisse, die auf schwerwiegende Fehler hinweisen. Das Programm wird meist beendet werden müssen.

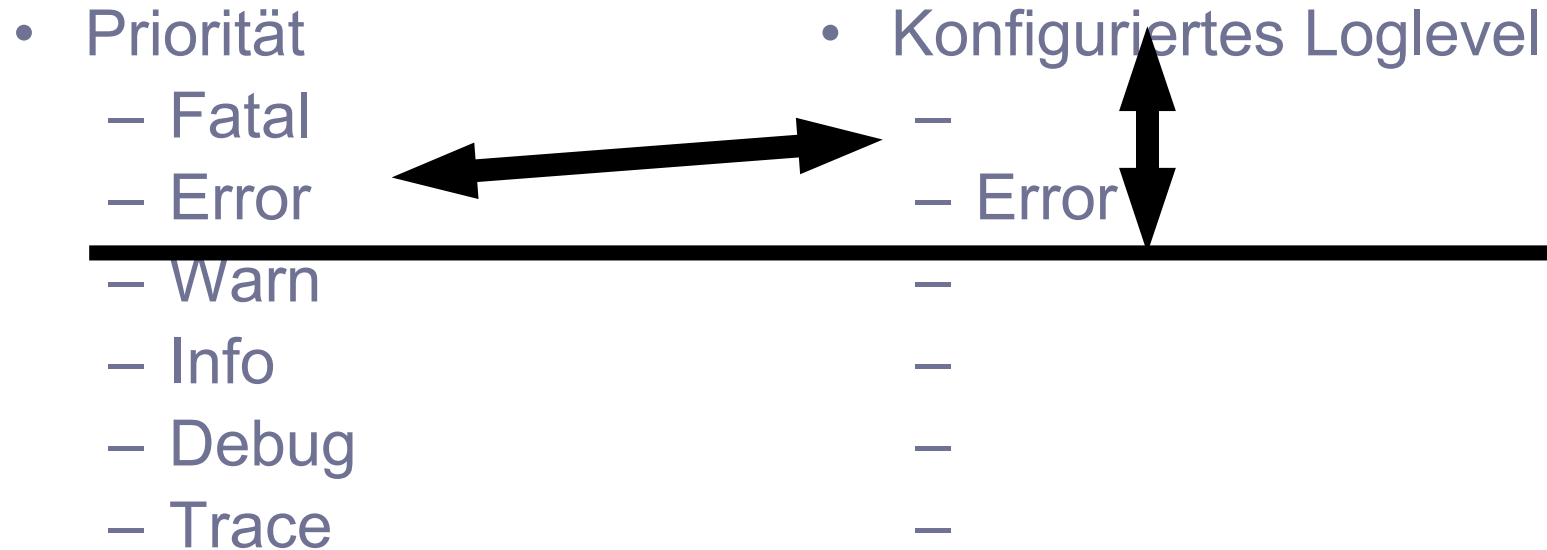
Beispiel: FATAL("Panic! Shutting down.");

- Priorität
    - Fatal
    - Error
    - Warn
    - Info
    - Debug
    - Trace
  - Konfiguration
    - Fatal
    - Error
    - Warn
    - Info
    - Debug
    - Trace
- 

## Prioritäten - Geschwätzige Konfiguration

- Priorität
    - Fatal
    - Error
    - Warn
    - Info
    - Debug
    - Trace
  - Konfiguriertes Loglevel
    - 
    - 
    - 
    - 
    - 
    - Trace
- 
-

## Prioritäten - Schweigsame Konfiguration



The image features a dark blue background with a large, light blue cross shape centered on it. The cross is composed of several rectangular blocks. The word "Appender" is written in white, bold, sans-serif font, positioned on the right arm of the cross. The overall design is minimalist and modern.

**Appender**

Appender sind Datensenken, in die Log-Nachrichten geschrieben werden.

Als Datensenke lässt sich von der einfachen Bildschirmausgabe (STDERR) über Logdateien, Syslog, SMS- oder E-Mail-Benachrichtigung, bis hin zu Datenbanken, so ziemlich alles verwenden, was man sich als Ausgabe-Medium vorstellen kann.



- Log::Log4perl Appender (Auswahl)
  - Log::Log4perl::Appender::Screen
  - Log::Log4perl::Appender::File
  - Log::Log4perl::Appender::Socket
  - Log::Log4perl::Appender::DBI
  - Log::Log4perl::Appender::Synchronized
  - Log::Log4perl::Appender::RRDs

- Log::Dispatch stellt weitere Appender zur Verfügung
  - Log::Dispatch::ApacheLog
  - Log::Dispatch::DBI
  - Log::Dispatch::Email
  - Log::Dispatch::Email::MIME Lite
  - Log::Dispatch::File
  - Log::Dispatch::FileRotate
  - Log::Dispatch::Screen
  - Log::Dispatch::Syslog
  - Log::Dispatch::Tk



Layouts

Layouts formatieren die Log-Nachrichten und legen den Informationsumfang fest.

Log-Nachrichten können in einer printf()-ähnlichen Syntax einfach an die eigenen Bedürfnisse angepasst (formatiert) werden.

## Layouts - Formatierungsanweisungen

- **%c** Category of the logging event.
- **%C** Fully qualified package (or class) name of the caller
- **%d** Current date in yyyy/MM/dd hh:mm:ss format
- **%F** File where the logging event occurred
- **%H** Hostname (if Sys::Hostname is available)
- **%I** Fully qualified name of the calling method followed by the callers source the file name and line number between parentheses.
- **%T** A stack trace of functions called
- **%L** Line number within the file where the log statement was issued
- **%m** The message to be logged
- **%M** Method or function where the logging request was issued
- **%n** Newline (OS-independent)
- **%p** Priority of the logging event
- **%P** pid of the current process
- **%r** Number of milliseconds elapsed from program start to logging
- **%%** A literal percent (%) sign



**Betriebsmodi**

Log::Log4perl kennt zwei Betriebsmodi:

- **Standard Mode**
  - Sehr sehr viele fein granulierte Optionen (FMTEYEWTK)
- **Easy Mode, auch Stealth-Logger genannt**
  - komfortables, einfaches Loggen, eben EASY.

## Easy Mode (Stealth-Logger)

Grundlagen



```
#!/usr/bin/perl  
use strict;  
use warnings;  
  
use Log::Log4perl qw/:easy/;  
Log::Log4perl->easy_init();
```

## Easy Mode mit Konfigurationsdatei initialisieren

```
#!/usr/bin/perl  
use strict;  
use warnings;  
  
use Log::Log4perl qw/:easy/;  
Log::Log4perl->init( 'l4p.conf' );
```

- Level
  - TRACE
  - DEBUG
  - INFO
  - WARN
  - ERROR
  - FATAL
- Convenience: Log and ...
  - LOGWARN
  - LOGDIE
  - LOGEXIT
  - LOGCARP
  - LOGCLUCK
  - LOGCROAK
  - LOGCONFESS

- Level
  - \$TRACE
  - \$DEBUG
  - \$INFO
  - \$WARN
  - \$ERROR
  - \$FATAL

Log::Log4perl->easy\_init();  
initialisiert einen einfachen Logger.

Der Appender ist vom Typ SCREEN, d.h. die Log-Nachrichten erscheinen auf STDERR.

Das Layout der Nachrichten wird auf das Format  
"%d %m%n" gesetzt.

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

use Log::Log4perl qw/:easy/;
Log::Log4perl->easy_init();

INFO("Starte $0");

print "Hallo Welt\n";

INFO("Beende $0");
```

```
$ ./HalloWelt.pl
```

```
YYYY/MM/DD hh:mm:ss Starte HalloWelt.pl
```

```
Hallo Welt
```

```
YYYY/MM/DD hh:mm:ss Beende HalloWelt.pl
```

## Loglevel im Easy Mode konfigurieren

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

use Log::Log4perl qw/:easy/;
Log::Log4perl->easy_init( {
    level => $DEBUG,
} );
```



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

use Log::Log4perl qw/:easy/;
Log::Log4perl->easy_init( {
    level => $DEBUG,
    layout =>
        '%d [%M] %F %L> %m%n',
    } );
```

```
YYYY/MM/DD hh:mm:ss [main::] HalloWelt02.pl  
11> Starte HalloWelt02.pl  
Hallo Welt  
YYYY/MM/DD hh:mm:ss [main::] HalloWelt02.pl  
15> Beende HalloWelt02.pl
```

## Ausgabemedium im Easy Mode konfigurieren

```
use Log::Log4perl qw/:easy/;  
Log::Log4perl->easy_init( {  
    level => $DEBUG,  
    layout =>  
    '%d [%M] %F %L> %m%n',  
    file =>  
    '>> test.log',  
} );
```

## Stealth-Logger in Modulen

```
package Person;
use strict;
use warnings;
sub new { bless( {} , shift ) ; }
sub vorname {
    my $self = shift;
    $self->{'Vorname'} = shift if @_;
    warn "Vorname nicht definiert\n"
        unless $self->{'Vorname'};
    return $self->{'Vorname'};
}
1 ;
```

## Einfaches Treiberprogramm ohne Log::Log4perl

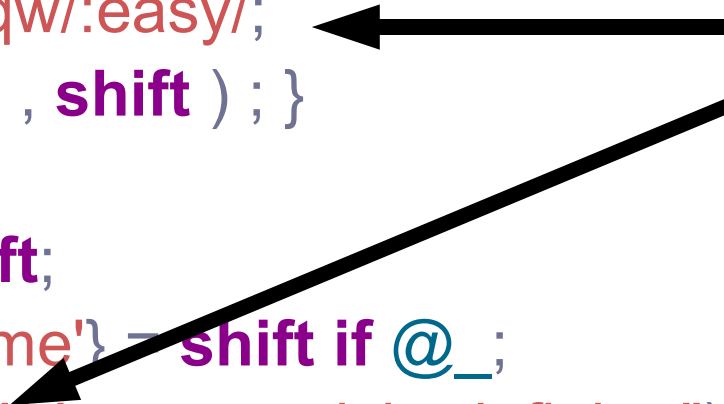
```
#!/usr/bin/perl
use strict;
use warnings;
use Person;

my $person = Person->new();

$person->vorname('Hans');
$person->nachname('Meier');
```

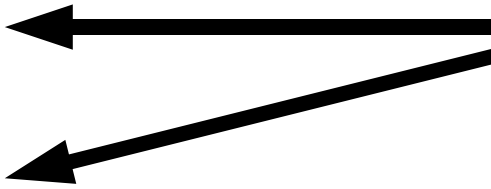
## Einfaches Package mit Log::Log4perl

```
package Person;
use strict;
use warnings;
use Log::Log4perl qw/:easy/;
sub new { bless( {} , shift ) ; }
sub vorname {
    my $self = shift;
    $self->{'Vorname'} = shift if @_;
    WARN("Vorname nicht definiert")
        unless $self->{'Vorname'};
    return $self->{'Vorname'};
}
1;
```



## Einfaches Treiberprogramm mit Log::Log4perl

```
#!/usr/bin/perl
use strict;
use warnings;
use Log::Log4perl qw/:easy/;
Log::Log4perl->easy_init( {
    level  => $DEBUG,
    layout => '%d [%M] %F %L> %m%n',
    file   => '>> test.log',  } );
use Person;
my $person = Person->new();
$person->vorname('Hans');
$person->nachname('Meier');
```





Im Paket legt der/die Programmierer(in) fest, was mit welcher Priorität geloggt werden kann.

Die Konfiguration erfolgt stets im Anwendungsprogramm.



**Konfigurationsdateien**

## Vorteile Konfigurationsdateien

- Konfigurationsdateien trennen den Code vom Logging-Level
- Konfigurationsdateien trennen den Code vom Logging-Layout
- Konfigurationsdateien trennen den Code vom Logging-Appender
- Konfigurationsdateien können unabhängig vom Programm editiert werden,
  - z.B. von Administratoren
  - zur Laufzeit des Programms

## Einfaches Treiberprogramm mit Konfigurationsdatei

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

```
use strict;
```

```
use warnings;
```

```
use Log::Log4perl qw/:easy/;
```

```
Log::Log4perl->init( 'l4p.conf' )
```



```
    INFO("Starte $0");
```

```
use Person;
```

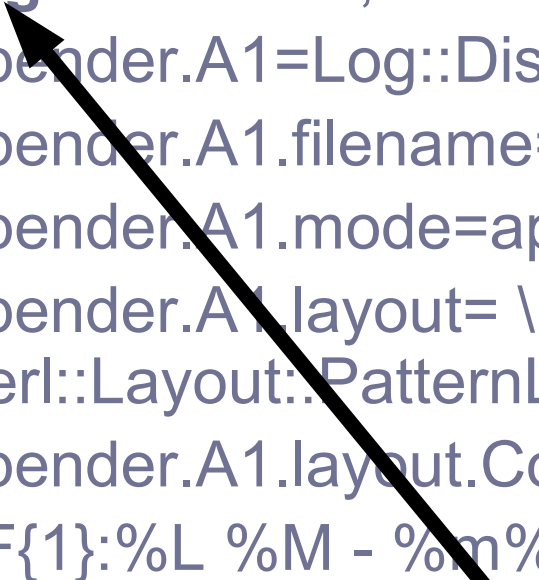
```
my $person = Person->new();
```

```
$person->vorname('Hans');
```

```
$person->nachname('Meier');
```

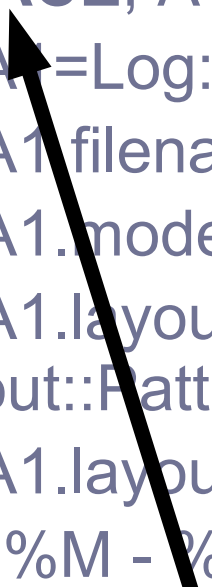
```
log4perl.logger=TRACE, A1
log4perl.appender.A1=Log::Dispatch::File
log4perl.appender.A1.filename=./person.log
log4perl.appender.A1.mode=append
log4perl.appender.A1.layout= \
Log::Log4perl::Layout::PatternLayout
log4perl.appender.A1.layout.ConversionPattern= \
%d %p> %F{1}:%L %M - %m%n
```

```
log4perl.logger=TRACE, A1  
log4perl.appender.A1=Log::Dispatch::File  
log4perl.appender.A1.filename=./person.log  
log4perl.appender.A1.mode=append  
log4perl.appender.A1.layout= \  
Log::Log4perl::Layout::PatternLayout  
log4perl.appender.A1.layout.ConversionPattern= \  
%d %p> %F{1}:%L %M - %m%n
```



**Name der Kategorie (Root Logger)**

```
log4perl.logger=TRACE, A1  
log4perl.appender.A1=Log::Dispatch::File  
log4perl.appender.A1.filename=./person.log  
log4perl.appender.A1.mode=append  
log4perl.appender.A1.layout= \  
Log::Log4perl::Layout::PatternLayout  
log4perl.appender.A1.layout.ConversionPattern= \  
%d %p> %F{1}:%L %M - %m%n
```



**Log Level: TRACE, DEBUG, .....**

## Konfigurationsdatei (Log Alias Definition)

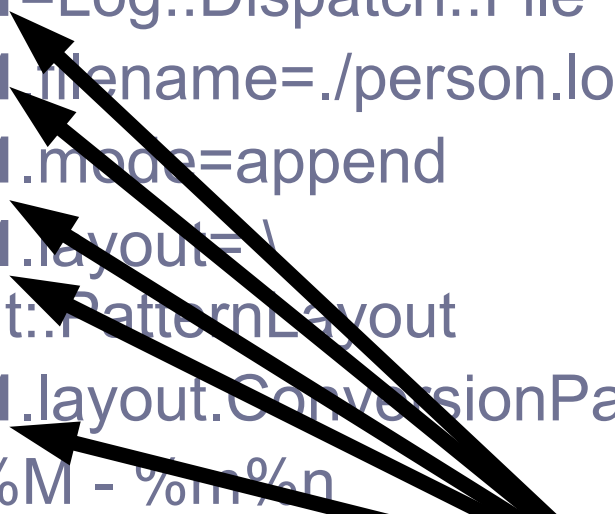
```
log4perl.logger=TRACE, A1  
log4perl.appender.A1=Log::Dispatch::File  
log4perl.appender.A1.filename=./person.log  
log4perl.appender.A1.mode=append  
log4perl.appender.A1.layout= \  
Log::Log4perl::Layout::PatternLayout  
log4perl.appender.A1.layout.ConversionPattern= \  
%d %p> %F{1}:%L %M - %m%n
```

**Log Alias Definition**



## Konfigurationsdatei (Log AliasVerwendung)

```
log4perl.logger=TRACE, A1  
log4perl.appender.A1=Log::Dispatch::File  
log4perl.appender.A1.filename=./person.log  
log4perl.appender.A1.mode=append  
log4perl.appender.A1.layout=  
Log::Log4perl::Layout::PatternLayout  
log4perl.appender.A1.layout.ConversionPattern= \  
%d %p> %F{1}:%L %M - %m%n
```



**Log Alias Verwendung**

## Konfigurationsdatei (Appender festlegen)


```
log4perl.logger=TRACE, A1  
log4perl.appender.A1=Log::Dispatch::File  
log4perl.appender.A1.filename=./person.log  
log4perl.appender.A1.mode=append  
log4perl.appender.A1.layout= \  
Log::Log4perl::Layout::PatternLayout  
log4perl.appender.A1.layout.ConversionPattern= \  
%d %p> %F{1}:%L %M - %m%n
```

**Appender**



## Konfigurationsdatei (Appender konfigurieren)

```
log4perl.logger=TRACE, A1  
log4perl.appender.A1=Log::Dispatch::File  
log4perl.appender.A1.filename=./person.log  
log4perl.appender.A1.mode=append  
log4perl.appender.A1.layout= \  
Log::Log4perl::Layout::PatternLayout  
log4perl.appender.A1.layout.ConversionPattern= \  
%d %p> %F{1}:%L %M - %m%n
```



**Appender konfigurieren**

**Log::Log4perl in CPAN-Modulen**

## CPAN-Module, die Log::Log4perl bereits benutzen

- App-Daemon
- Archive-Tar-Wrapper
- Authen-PAAS
- CPAN-Unwind
- Cache-Historical
- Catalyst-Log-Log4perl
- Config-Patch
- Data-Throttler
- File-Comments
- Gaim-Log-Mailer
- Gaim-Log-Parser
- IPC-Cmd-Cached
- JavaScript-SpiderMonkey
- Jifty
- Log-Dispatch-File-Rolling
- Log-Dispatch-FileRotate
- Log-Log4perl-Layout-XMLLayout
- Log-Statistics
- Mail-DWIM
- Module-Rename
- Nagios-Clientstatus
- Net-Amazon
- Perl-Configure
- Process-MaxSize
- RRDTool-OO
- Rose-DBx-Object-InternalPager
- SQL-Translator
- Sysadm-Install
- Text-Language-Guess
- Text-Scan-License
- Text-TermExtract
- Trash-Park
- Webservice-ReviewBoard
- Workflow
- XML-RSS-FromHTML-Simple
- ... more to come

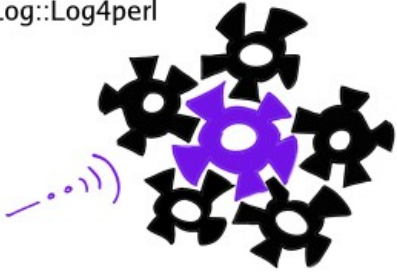
## Log::Log4perl in CPAN-Module einbinden

```
package Foo;
use strict;
use warnings;
use Log::Log4perl qw(:easy);

sub foo {
    my $bar = shift @_;
    # ... do something
    DEBUG "Fooing $bar";
}
1;
```

## Log::Log4perl in CPAN-Module einbinden – So einfach geht's!

Log::Log4perl



# Fertig!

Das Modul ist jetzt  
Log::Log4perl enabled.



**Weblinks**



- CPAN – Log::Log4Perl:
  - <http://search.cpan.org/perldoc?Log::Log4perl>
- Log::Log4Perl Projektseite:
  - <http://log4perl.sourceforge.net/>
- Apache log4j (Log4Java) Projektseite:
  - <http://logging.apache.org/log4j/>

- Michael Schilli: Logging nach Bedarf mit Log::Log4perl - Wachsame Schläfer
  - [http://www.linux-magazin.de/heft\\_abo/ausgaben/2](http://www.linux-magazin.de/heft_abo/ausgaben/2)
- brian d foy: Mastering Perl: Logging
  - [http://www252.pair.com/comdog/mastering\\_perl/C](http://www252.pair.com/comdog/mastering_perl/C)
- Michael Schilli: Retire your debugger, log smartly with Log::Log4perl!
  - <http://www.perl.com/pub/a/2002/09/11/log4perl.htr>
- Horshack's Log4perl Page
  - [http://lena.franken.de/perl\\_hier/log4perl.html](http://lena.franken.de/perl_hier/log4perl.html)

- Michael Schilli: Log4perl: the Only Logging System You'll Ever Need - OSCON 2008
  - <http://log4perl.sourceforge.net/l4p.ppt>
- Michael Schilli: Talk at YAPC 2007 (Houston, TX):
  - [http://perlmeister.com/log4perl\\_yapc.html](http://perlmeister.com/log4perl_yapc.html)
- Madmongers - Logging
  - <http://www.madmongers.org/uploads/Mc/oB/McoB>



**Fragen?**

Danke!

Danke!



**About**

- Perl-Programmierer und Systemadministrator seit 1996
- Perl-Trainer seit 2006
- Sozialisation: kleine und mittlere Unternehmen
  - Perl-Erfahrung seit 1996
  - Linux-Erfahrung seit 1996
  - MySQL- Erfahrung seit 1996
  - OpenLDAP Erfahrung seit 1999
  - Openoffice.org Erfahrung seit 2000
  - Solaris Erfahrung seit 2002
  - XML Erfahrung seit 2002

- Homepage:
  - <http://thomas-fahle.de>
- E-Mail:
  - [info@thomas-fahle.de](mailto:info@thomas-fahle.de)
- Personal Weblog
  - <http://thomas-fahle.blogspot.com>
- Online-Profile:
  - [http://www.xing.com/profile/Thomas\\_Fahle](http://www.xing.com/profile/Thomas_Fahle)
  - <http://www.linkedin.com/in/thomasfahle>



## About Thomas Fahle – Weitere Websites

- Perl-Suchmaschine:
  - <http://perl-suchmaschine.de>
- Perl-Howto.de
  - <http://perl-howto.de>



**Sonstiges**

- HP Business templates & images - presentations
  - <http://www.hp.com/sbso/productivity/office/present>
  - <http://www.hp.com/cgi-bin/sbso/productivity/office/>