

Perl Mongers im Ruhrgebiet



Ruhr . pm

Abdeckungsdest auf Grundlage von Devel::Cover

Autor: Lars Dieckow

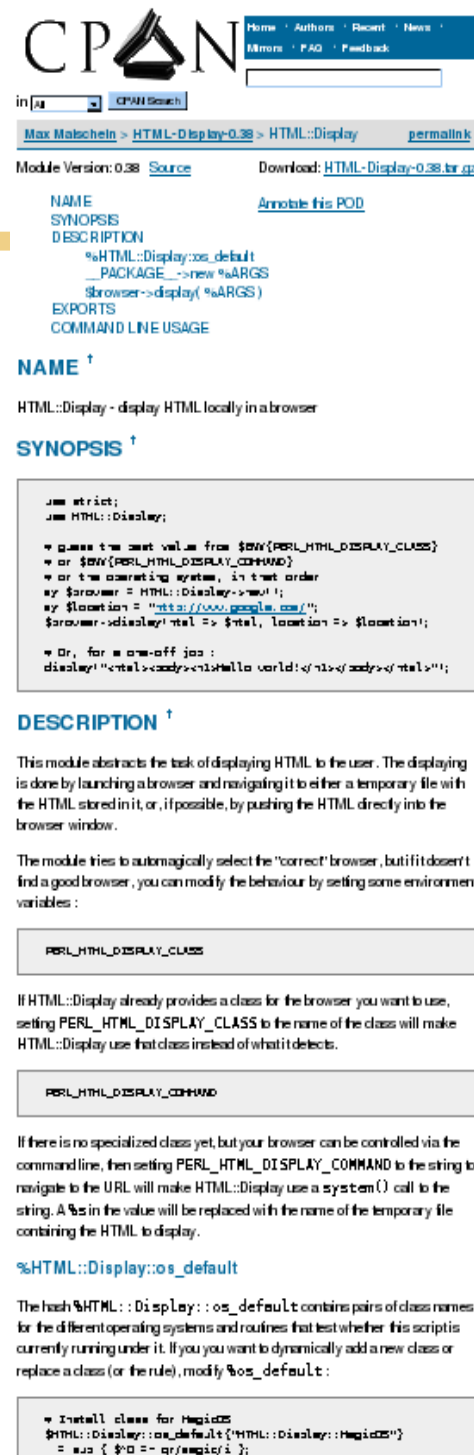
E-Mail: lars AT ruhr.pm.org

Datum: 12. September 2007

<http://ruhr.pm.org/>

Tests

- prüfen
- perlnewmod: läuft wie beabsichtigt
- stimmt Programmcode mit der Doku überein?



CPAN [Home](#) [Authors](#) [Recent](#) [News](#)
[Mirrors](#) [FAQ](#) [Feedback](#)

in [ja](#) CPAN Search

[Max Malschein](#) > [HTML-Display-0.38](#) > [HTML::Display](#) [permlink](#)

Module Version: 0.38 [Source](#) Download: [HTML-Display-0.38.tar.gz](#)

[NAME](#) [Annotate this POD](#)

SYNOPSIS

```
DESCRIPTION
    %HTML::Display::os_default
    __PACKAGE__->new %ARGS
    $browser->display( %ARGS )
EXPORTS
COMMAND LINE USAGE
```

NAME [↑]

HTML::Display - display HTML locally in a browser

SYNOPSIS [↑]

```
use strict;
use HTML::Display;

# guess the best value from $ENV{PERL_HTML_DISPLAY_CLASS}
# or $ENV{PERL_HTML_DISPLAY_COMMAND}
# or the operating system, in that order
my $browser = HTML::Display->new();
my $location = "http://www.google.com/";
$browser->display($html, $location => $location);

# Or, for a one-off job:
display($html->body->html, $location => $location);
```

DESCRIPTION [↑]

This module abstracts the task of displaying HTML to the user. The displaying is done by launching a browser and navigating it to either a temporary file with the HTML stored in it, or, if possible, by pushing the HTML directly into the browser window.

The module tries to automatically select the "correct" browser, but if it doesn't find a good browser, you can modify the behaviour by setting some environment variables:

PERL_HTML_DISPLAY_CLASS

If HTML::Display already provides a class for the browser you want to use, setting PERL_HTML_DISPLAY_CLASS to the name of the class will make HTML::Display use that class instead of what it detects.

PERL_HTML_DISPLAY_COMMAND

If there is no specialized class yet, but your browser can be controlled via the command line, then setting PERL_HTML_DISPLAY_COMMAND to the string to navigate to the URL will make HTML::Display use a system() call to the string. A %s in the value will be replaced with the name of the temporary file containing the HTML to display.

%HTML::Display::os_default

The hash %HTML::Display::os_default contains pairs of class names for the different operating systems and routines that test whether this script is currently running under it. If you want to dynamically add a new class or replace a class (or the rule), modify %os_default:

```
# Install class for Mac OS
$HTML::Display::os_default{"HTML::Display::MacOS"}
= sub { $?D == qr/macos/i };
```



Ruhr . pm

Devel : : Cover

- wie gründlich sind die Tests?
- Coverage - Abdeckung:
 - Anweisungen
 - Verzweigungen
 - Bedingungen
 - Subroutinen



Ruhr.pm

mit Module::Build

```
~/Acme-My-Module$ perl Build.PL  
~/Acme-My-Module$ ./Build testcover
```



Ruhr.pm

mit ExtUtils::MakeMaker

```
~/ .cpan/build/HTML-Display-0.37-D18Qx1$ \  
cover -delete ; \  
HARNESS_PERL_SWITCHES=-MDevel::Cover make test ; \  
cover
```

```
Deleting database /home/daxim/.cpan/build/  
HTML-Display-0.37-D18Qx1/cover_db
```

```
PERL_DL_NONLAZY=1 /usr/bin/perl \  
"-MExtUtils::Command::MM" "-e" "test_harness(0, 'blib/lib', \  
  'blib/arch')" t/*.t  
t/00-HTML-Display-use.....ok  
t/01-HTML-Display-TempFile-share.....ok  
|  
|  
|
```



Ruhr.pm

ctd.

```

|
|
|
-----
File                stmt    bran    cond    sub    pod    time    total
-----
...ib/HTML/Display/Common.pm    n/a    n/a    n/a    n/a    n/a    n/a    n/a
.../lib/HTML/Display/Dump.pm    n/a    n/a    n/a    n/a    n/a    n/a    n/a
inc/IO/Catch.pm                48.6   16.7   n/a    42.9   n/a    2.4    43.9
lib/HTML/Display.pm            100.0   n/a    n/a    100.0  n/a    95.2   100.0
lib/HTML/Display/TempFile.pm    100.0   n/a    n/a    100.0  n/a    1.2    100.0
...HTML/Display/Win32/OLE.pm    100.0   n/a    n/a    100.0  n/a    1.1    100.0
Total                          76.5   16.7   n/a    63.6   n/a    100.0  70.6
-----

```

```

Writing HTML output to /home/daxim/.cpan/build/HTML-Display-0.37-D18Qx1/
                                cover_db/coverage.html ...

```



Ruhr.pm

ctd.

```
use Test::Exception;  
use lib 'inc';  
use IO::Catch;
```

```
dies_ok {  
    tie *STDOUT, 'IO::Catch';  
} 'tie without variable';
```



Ruhr.pm

t/02-IO-Catch.t

```
use strict;
use Test::More tests => 5;
use Test::Exception;
use lib 'inc';
use IO::Catch;
use vars qw($_STDOUT_);

dies_ok {
    tie *STDOUT, 'IO::Catch';
} 'tie without variable';

ok tie(*STDOUT, 'IO::Catch', '_STDOUT_'), 'tie';

ok print $_STDOUT_, 'print to tied handle';
ok printf $_STDOUT_, 'printf to tied handle';

ok untie(*STDOUT), 'untie';
```