

```

#!/usr/bin/perl

use strict ;
use warnings ;

my $RDME = 'README' ;
my $LAST = '/dist/META/ -- How to update META/ROOT' ;

my $prog = substr $0, rindex ( $0, '/' ) + 1 ;
my $Usage = <<USAGE ;
Usage: $prog [-v] [-q] [-d] [-f] args
option v : be verbose
option q : be quiet
option d : show debug info
option f : action ; otherwise dry-run
USAGE
sub Usage { die "$_[0]$Usage" ; }
sub Error { die "[error] $prog: $_[0]\n" ; }
sub Warn { warn "[warn] $prog: $_[0]\n" ; }

# usage: &GetOptions(ARG,ARG,..) defines $opt_ID as 1 or user spec'ed value
# usage: &GetOptions(\%opt,ARG,ARG,..) defines $opt{ID} as 1 or user value
# ARG = 'ID' | 'ID=SPC' | 'ID:SPC' for no-arg, required-arg or optional-arg
# ID = perl identifier
# SPC = i|f|s for integer, fixedpoint real or string argument

use Getopt::Long ;
Getopt::Long::config ( 'no_ignore_case' ) ;
my %opt = ( ) ; Usage '' unless GetOptions
( \%opt, qw(v q d f) ) ;
Usage "Arg count\n" unless @ARGV == 0 ;

$opt{v} ||= $opt{d} ;

@ARGV = ( $RDME ) ;

my $OUT = "$RDME.html" ;
my $TMP = "$OUT.tmp" ;
my @RM = ( ) ;
while ( <> )
{ last if /^$LAST/ ;
  s/</&lt;/g ;
  s/>/&gt;/g ;
  push @RM, $_ ;
}
pop @RM ;
open TMP, '>', $TMP or Error "can't write $TMP ($!)" ;
print TMP "<pre>\n" ;
print TMP $_ for @RM ;
print TMP "</pre>\n" ;
close TMP ;
if ( -f $TMP )
{ rename $TMP, $OUT or Error "can' rename $TMP $OUT ($!)" ; }

```