```
#!/bin/sh

# New prettyp: Based on prettyp for Luis Fernandes <elf@ee.ryerson.ca>

# Original Header:

# prettyp: grind source code files & generate .ps files by Luis
# Fernandes <elf@ee.ryerson.ca> originally from a script by K. Clowes
# <kclowes@ee.ryerson.ca>

# Patch by <lannert@lannert.rz.uni-duesseldorf.de> that allows files
# to contain directory components (which are stripped away before
# processing)

# $Id: prettyp.src,v 1.3 2002/05/09 00:45:12 kamath Exp $

TMPDIR="/tmp/TG$$.dir"
CURDIR=`pwd`

# This "helps" TeX find it's files.
TEXMF="/usr/local/share/texmf"
TEXMFCNF="$TEXMF/web2c/texmf.cnf"
export TEXMF TEXMFCNF

# Make sure TeX can be found
PATH=$PATH:/usr/local/share/bin

# Remove the tgrind directory we create.
trap 'cleanup' 1 2 15

ME=`basename $0`

# Simple Usage message
```

usage

```
usage() {
    EV=$1
    MSG=$2

    if [ ! -z "$MSG" ]
    then
        echo "$MSG" 1>&2
    fi

    cat <<EOF 1>&2

Usage: $ME [-chknw] [-l lang] [-H header] file ...

Options:
    -c          Create color output (requires dvips!)
    -H header   Use header for header text
    -h          This help message
    -k          Keep temporary directory (in /tmp/TG\$\$.dir)
    -l lang     Force language to lang
    -n          Number each line
```

```
    -w          Wrap lines using 'fold'
EOF
    exit $EV
}

# Cleanup any stray files
```

cleanup

```
cleanup() {
    if [ ! -z "$TMPDIR" ] && [ -d $TMPDIR ]
    then
        if [ -z "$KEEP" ] || [ $KEEP -eq 0 ]
        then
            rm -rf $TMPDIR
        else
            : # echo "Preserving $TMPDIR"
        fi
    fi
    return
}

# Print a message, cleanup, and exit
```

die

```
die() {
    MSG=$1

    echo "$MSG" 1>&2
    cleanup
    exit 1
}

# The meat of the script. Grind a file.
```

grind

```
grind() {
    PATHFILE=$1
    FILE=`basename $PATHFILE`
    DIRFILE=`dirname $PATHFILE`

    if [ ! -f $PATHFILE ]
    then
        echo "File $FILE not found!"
        return
    fi

    cleanup
    if [ ! -d $TMPDIR ]
    then
        mkdir $TMPDIR
        if [ $? -ne 0 ]
        then
            die "Unable to make temporary directory"
        fi
    fi

    if [ -z "$LANGUAGE" ]
    then
```

```
        LANGUAGE="`lang $PATHFILE`"
    fi

    expand -8 $PATHFILE | $WRAP > $TMPDIR/$FILE

    # The reason for this bizarreness is because HEADER may
    # have multiple items in it, and so we need to pass
    # -h "$HEADER" if HEADER is non-null.  The only other way to do
    # this is setting FOO="-h \"$HEADER\"", and then
    # eval'ing $FOO, but that would expand other things in $HEADER...

    (cd $TMPDIR
     if [ -z "$HEADER" ]
     then
        tfontedpr -l$LANGUAGE $COLOR $LINES $FILE > ${FILE}.tex
     else
        tfontedpr -l$LANGUAGE -h "$HEADER" $COLOR $LINES $FILE > ${FILE}.tex
     fi

     tex ${FILE}.tex
     dvips -q -t letter -f ${FILE}.dvi > ${FILE}.ps
    )
    cp ${TMPDIR}/${FILE}.ps ${PATHFILE}.ps
}

# Defaults
COLOR=
LINES=
LANGUAGE=
HEADER=
WRAP=cat
KEEP=0

while getopts cwnkl:hH: a
do
    case $a in
        # These are vfontdpr options, mostly
        c) COLOR="-c";;
        n) LINES="-N";;
        l) LANGUAGE="$OPTARG";;
        h) usage;;
        H) HEADER="$OPTARG";;
        k) KEEP=1;;
        w) WRAP=fold;;
        \?) usage 1;;
    esac
done
shift `expr $OPTIND - 1`

for i in "$@"
do
    grind "$i"
done
```

```
cleanup

exit 0
```