

mprename Version: 1.6

Daniel Eric Smith
e-mail:desmith@jsu.edu

January 3, 2005

Abstract

Documentation for using the Perl program `mprename`. `mprename` will rename the files output running MetaPost on the file `file.extension` from `file.#` to `filename.extension`, where you put the new name `filename.extension` in the MetaPost file. You can also process multiple files at the same time.

1 Introduction

You may need to change the first line of the program if your Perl compiler is not in `/usr/bin/perl` or if you are using a non-Unix type system. It is usually found in `/usr/bin/perl` or `/usr/local/bin/perl` on Unix type systems. On non-Unix systems you should make the first line `#!perl`. I stole this from *Learning Perl*, by Schwartz and Phoenix; 3rd Ed, O'Reilly.

2 Usage

The program `mpremane` is invoked by

```
mprename [switches] FILE0.mp [FILE1.mp] [FILE2.mp] ...
```

There are only two switches:

- `-nolog` Suppresses writing the log file.
- `-mpost` Runs MetaPost on each file.

The order of the switches is not important.

You must supply at least one MetaPost file to `mprename`. There is no maximum number of MetaPost files you can process (unless the number of file names is so large that you run out of memory). The MetaPost file name(s) must contain a 'dot'. So the following are valid file names: `file.mp,file.,file.whatever`. But the file name `file` is not valid. If you do not include a 'dot' in the file name, `file mprename` will think that the name of the postscript files are `fil.#`. I may try to fix this, but probably not.

3 Syntax

After each `beginfig` statement you put `%file=filename.extension`. The `filename.extension` is what you want to rename that image file as. You cannot put any other text on that line. If you do not put a `%file=filename.extension` on a `beginfig(#)`; line the program will just skip that line and leave the file name as `file.#`.

The following is an example of a file called `picture.mp` that contains the “code” for two images: a Venn diagram and a circle.

```
beginfig(1);%file=venndiag.eps
  MetaPost instructions
endfig;

beginfig(5); % file = circle.ps
  MetaPost instructions
endfig;

end
```

I tried to anticipate where you might put spaces. So the following should both work: `%file=circle.eps` and `% file = cirlce.eps`.

4 Log File

The log file produced by `mprename` is named `file.mprlog` and contains the following:

1. The name of the file that `mprename` was run on.
2. The arguments passed to `mprename`.
 - (a) The switches passed to `mprename`
 - (b) The MetaPost files passed to `mprename`.
3. The date `mprename` was run.
4.
 - (a) Each time `mprename` see a line `beginfig(%)%file=filename.extension` it prints: I see the line: `beginfig(%)%file=filename.extension`
 - (b) If the file `file.#` exists it tells you what it renamed that file to. If the file `file.#` does not exist it prints an error message.

You can turn off writing the log file by passing the `-nlog` switch to `mprename`.

5 Code

This is the source code for the Perl program mprename.

```
#!/usr/bin/perl
#####
#Program: mprename
#Author: Daniel Eric Smith #
#E-mail: desmith@jsu.edu #
#Date: Jan 03, 2005 #
#Version: 1.6 #
#####
#The strict pragma forces variable declaration and some other things
use strict;
#The warnings pragma enables all warnings
use warnings;
#number of arguments passed to mprename
#and then the number of elts in @switches or @mpfiles
my $argnum=@ARGV;
#assigns arguments to my array @args
my @args=@ARGV;
chomp(@args);
#determines if a log file will be written (default: write log file)
my $logbool=1;
#determines if MetaPost will be run on each file (default: don't run)
my $mpostbool=0;
#array of switches passes to mprename
my @switches;
#array of MetaPost files passes to mprename
my @mpfiles;
#location of the . in the MetaPost file name
my $wheredot;
#holds the MetaPost file name without the .mp
my $filenamesub;
#holds the name of the log file
my $logfile;
#time variable for log file
my $tm;
#holds MetaPost file lines while processing
my $line;
#name of the old postscript file
my $oldfile;
#name of the new postscript file
my $newfile;
#if variable needed to debug use this
my $debug
```

```

#DEBUG line
#print "DEBUG: The number of arguments passed equals ",scalar @ARGV,"\n";
#print "DEBUG: The value of argnum is $argnum\n";
#print "DEBUG: The arguments passed are @ARGV\n";
#DEBUG end

#Checks if any arguments are passed
if($argnum==0)
{
  #If no arguments are passed ask what to do:
  print "Give me a MetaPost file(s) (and some switches) to process:\n\n";
  &print_usage;
  print "Enter something:";
  $line=<STDIN>;
  chomp $line;
  #If still no input exit program
  if($line eq "")
  {
    print "\nYou still have not given me something to do, so I quit!\n";
    print "But please try again.\n\n";
    exit;
  }
  #Converts the string $line into an array splitting at whitespaces
  @args = split /\s+/, $line;
  #DEBUG line
  #print "DEBUG: You typed: $line\nAnd I have args=(@args)\n";
  #DEBUG end
}
#run through each element in the array
foreach (@args)
{
  if($_ eq "--help")
  {
    #Prints the usage of this command (a subroutine)
    &print_usage;
    exit;
  }
  elsif($_ =~ m/^\w+/#checks if a switch
  {
    #how many elements in array @switches
    $argnum=$#switches;
    #adds new element to @switches
    $switches[$argnum+1]=$_;
    #DEBUG lines
    #$debug=$argnum+1;
  }
}

```

```

        #print "DEBUG: switches[$debug]=$_\n";
        #DEBUG end
    }
else #not a switch must be a MetaPost file
{
    #how many elements in array @mpfiles
    $argnum=$#mpfiles;
    #adds new element to @mpfiles
    $mpfiles[$argnum+1]=$_;
    #DEBUG lines
    #$debug=$argnum+1;
    #print "DEBUG: mpfiles[$debug]=$_\n";
    #DEBUG end
}
}
#DEBUG lines
#print "DEBUG: mpfiles=@mpfiles\n";
#print "DEBUG: scalar mpfiles=@mpfiles\n";
#DEBUG end
#Check to see if any files are passed
unless(@mpfiles)
{
    print "You did not give me any files to process.\n";
    print "Please try again.\n\n";

    #Prints the usage of this command (a subroutine)
    &print_usage;
    exit;
}
#####
#ADD #
#####
#Should check that the files passed exist      #
#if a file does not exist remove it from @mpfiles #
#if none exists print error message and exit    #
#Put check here                                #
#####

#####
#ADD #
#####
#Should check that switches passed are valid #
#print the invalid switch and continue      #
#Put check here                              #
#####

```

```

foreach(@switches)
{
$logbool=0 if($_ eq "-nlog"); #turns off log file
$mpostbool=1 if($_ eq "-mpost"); #run MetaPost
}
#process each MetaPost file
#Note: uses default variable $_
foreach(@mpfiles)
{
#DEBUG line: what is the value of logbool
#print "logbool=$logbool\n\n";
#DEBUG end

#where is the last period in the file name
$wheredot=rindex($_,".");
#grab everything before the last period
$filenamesub=substr($_,0,$wheredot);
#do this if -nlog is not passed
if($logbool)
{
#the log file will be called "filename.mprlog"
$logfilename=$filenamesub . ".mprlog";
print "\nThe log file name is $logfilename\n\n";
unless (open LOG, "> $logfilename")
{
die "Cannot create $logfilename: $!";
}
print LOG "File Name:$_ \n";
print LOG "Arguments passed to mprename:\n";
print LOG "Switches:@switches \n";
print LOG "MetaPost Files:@mpfiles \n";
#use module Time
use Time::localtime;
#get current date/time
$tm = localtime;
#fancy print date/time
printf(LOG "Run on Date: %04d-%02d-%02d\n\n",$tm->year+1900,($tm->mon)+1,$tm->mday);
}
#if -mpost switch passed run MetaPost
if($mpostbool)
{
print "MetaPost output:\n";
if(system "mpost $_")
{
die "\nError:MetaPost unable compile $_\n\n";
}
}
}

```



```

        select LOG;
        print "I could not renamed $oldfile to $newfile.\n";
        print "Check to see if $oldfile exists.\n";
        print "If not you may need to run MetaPost.\n\n";
        select STDOUT;
    }
}
}
}

sub print_usage
{
my $lastslash=rindex($0,"/");
my $progrname=substr($0,$lastslash+1);
print "Usage: $progrname [switches] FILE0.mp [FILE1.mp] [FILE2.mp] ... \n\n";
print "Switches:\n";
print "\t-nlog \tSupress writing the log file(s)\n";
print "\t-mpost \tRun MetaPost on the file(s)\n\n";
print "FILE(s):\n";
print "\tYou must supply at least one MetaPost file.\n";
print "\tEach file must be of the form filename.extension.\n";
print "\tYou may process as many file as you want.\n\n";
}

```