
Subject: Re: HELP files (generation of?)

Posted by [jbm](#) on Sun, 07 Aug 1994 13:07:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article 1fr@sun4.bham.ac.uk, sjt@xun8.sr.bham.ac.uk (James Tappin) writes:

> Does anyone know of a utility which will take a collection of files or a
> file with multiple topics and an indication of where the topic changes
> (say a set of extracted routine documentation headers) and generate a help
> file readable by MAN_PROC?
>
> I think it would be quite easy to do, but if it's already been done then
> there's no point re-inventing the wheel.
>
> --
> James Tappin, School of Physics & Space Research
> University of Birmingham
> sjt@xun8.sr.bham.ac.uk
> "If all else fails--read the instructions!"

Yes! I have just such a shell script, if you are running under UNIX and have 'nawk'.

It assumes that each procedure has a set of help comments before the procedure which begin after a line starting ';' and run until a line starting ';-'. (Standard comment delimitation in IDL.)

The program takes a listing of the programs and outputs a help file into a filename given as a command line argument. It is quite smart about removing paths from the program filenames before converting the filenames into a help topic (eg. if a filename was /usr/local/lib/idl/utills/plot.pro, then the help topic would be 'PLOT'.) The calling sequence if you name it 'make_help' is:

> make_help output_filename program_filenames

If you don't put the '.help' suffix on the output filename, it will append it. As an example I might use this program to make a help file for the above utills directory by typing:

> make_help utills /usr/local/lib/idl/utills/*.pro

This would make a file named 'utills.help' in my current directory which contains the help information for those programs in /usr/local/lib/idl/utills/ which contain help type comments.

I note that it gives an error about not being able to find your output file. Don't worry about that, it's just trying to pass that file as input to the nawk program. Not a fatal error.

Try it and let me know if it works, or doesn't work for you. Good luck!

```
ls -1 $* | nawk '
BEGIN {
  NPROG = 0L;
```

```

NCHAR    = 0L;
CHARS[1] = 0L;
}

function upper(STR, I,LOW,CHAR,LOWER,UPPER) {
  UPPER = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  LOWER = "abcdefghijklmnopqrstuvwxyz";

  while (match(STR,/ [a-z]+/))
    for (I=RSTART;I<RSTART+RLENGTH;++I) {
      LOW = substr(STR,I,1);
      if (CHAR = index(LOWER,LOW))
        gsub(LOW,substr(UPPER,CHAR,1),STR);
    }
  return STR;
}

function append(STR1,STR2) {
  return STR1 STR2;
}

(substr($0,length($0)-3,4) == ".pro") { #if it is a program use it
  ++NPROG;
  NAMES[NPROG] = upper(substr($0,1,length($0)-4)) ;
  while (match(NAMES[NPROG],"/"))
    NAMES[NPROG] = upper(substr(NAMES[NPROG],RSTART+1,length($0)-RSTART)) ;
  while (getline LINE < $0 > 0) {
    if (substr(LINE,1,2) == ";;") {      # find the beginning of the comments
      TEXT[NPROG] = sprintf("%s\n",LINE);
      NCHAR += length(sprintf("%s\n",LINE));
      INTEXT = 1;
    } else if (INTEXT) {
      if (substr(LINE,1,2) == ";;-") {   # the end of the comments?
        TEXT[NPROG] = append(TEXT[NPROG],sprintf("%s\n",LINE));
        NCHAR += length(sprintf("%s\n",LINE));
        CHARS[NPROG+1] = NCHAR;
        INTEXT = 0;
      } else {
        # add the line to the comments without the semi-colon
        TMP_STR = substr(LINE,2,length(LINE)-1);
        TEXT[NPROG] = append(TEXT[NPROG],sprintf("%s\n",TMP_STR));
        NCHAR += length(sprintf("%s\n",TMP_STR));
      }
    }
  }
}
if (length(TEXT[NPROG]) == 0)    # if the program is not documented delete it.
  --NPROG;
close($0);
}

```

```

END { #write arrays to temporary files.
  TITLE_BAR = OUTFILE ;
  while (match(TITLE_BAR,"/"))
    TITLE_BAR = substr(TITLE_BAR,RSTART+1,length(TITLE_BAR)-RSTART);
  if (substr(OUTFILE,length(OUTFILE)-4,5) != ".help") {
    OUTFILE = append(OUTFILE,".help");
    TITLE_BAR = upper(TITLE_BAR);
  } else {
    TITLE_BAR = upper(substr(TITLE_BAR,1,length(TITLE_BAR)-5));
  }
  printf("%%TITLE:%s\n",TITLE_BAR) > OUTFILE;
  printf("%%8d\n",NPROG) >> OUTFILE;
  for (I=1;I<=NPROG;++I)
    printf("%%-20s%d\n",NAMES[I],CHARS[I]) >> OUTFILE;
  for (I=1;I<=NPROG;++I)
    printf("%%s",TEXT[I]) >> OUTFILE;
}
' OUTFILE=$1 -

```

Jason McPhate (jbm@pha.jhu.edu) The Johns Hopkins University
 Dept. of Physics and Astronomy
