μ**ls_eng**

Matteo Cortese

	COLLABORATORS		

	TITLE : μls_eng		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Matteo Cortese	January 2, 2023	

REVISION HISTORY			

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

μls_eng

1.1 μls Manual

```
                                              $\mathrm{\mu}$ls 1.3

                          by Matteo Cortese

  $\mathrm{\mu}$ls  (ie  micro-ls)  is a fast and optimized alternative to the  ←
      beloved
Dir  and  list commands.  It has been speficically designed for ergonomy,
with the shell-user in mind.




                Introduction

                Distribution

                Requirements & installation

                Usage

                Notes
```

1.2 Introduction

```
                $\mathrm{\mu}$ls  is  inspired by the UNIX command "ls",  but  ←
                    has features which are
peculiar  to the Amiga philosophy,  in the first place the modest size of
the executable, its ability to be made resident and the speed of use.  On
the  other  hand,   $\mathrm{\mu}$ls inherits from the UNIX world the ease of  ←
    combinig
options  together,   as it supports those "-ldx..." switches UNIX wizards
love  so  much.  So  I  am proud to say that $\mathrm{\mu}$ls takes "The best of ←
     both
worlds".
```

You want a list of $\mathrm{\mu}$ls's features, don't you? So here it is:

-~Two   formats   are   available:
                short~format
                 (multicolumn)   and

                long~format
                . (one file per line).

-~Everything is alphabetically sorted and dirs always come before files.

-~Dirs can be shown in white, to emulate the "visual impact" of file
    requesters.

-~$\mathrm{\mu}$ls can adapt its output to the current console width.

-~You can tell the type (dir, link...) of each entry from the
                suffix
                    appended to the entry's name.

-~You can hide or filter out some files. By default $\mathrm{\mu}$ls hides ↩
    icons and
     backups.

-~You can use wildcards.

-~You can use both Amiga (template) and UNIX style (eg "-ld") for
    options.

-~For every XPK-compressed files, $\mathrm{\mu}$ls shows packed and unpacked ↩
    sizes,
     method and whether a password is required.

-~$\mathrm{\mu}$ls is pure and quick.

1.3 Distribution

            This program is PUBLIC DOMAIN or, if you prefer, FREEWARE. ↩
                You can
copy, use and modify it as you like. If you want to sell it (should you
find someone fool enough to buy it) you can.

  If you want to suggest some enhancements, please contact the author. If
you want to talk about Amiga programming, contact the author. If you are
not satisfied with this proggy, hit the first passer-by and leave the
author in peace. The author will not be responsible for psychic desease
caused by eccessive use of this program. If you are still reading, you
definitively need a psychiatrist. If you need a psychiatrist, do not
contact the author.

  The
                source

is included.

                          Matteo Cortese
                          str. Torino 65
                       10024 Moncalieri (TO)
                              ITALY

               email: matteocortese@hotmail.com
                     fidonet: 2:334/21.56

         http://cclix1.polito.it/~s75735/index.html

1.4 Requirements & installation

                    $\mathrm{\mu}$ls requires AmigaOS 3.0 (V39). If you have at ←
                    least 68020 you can use
the more optimized version $\mathrm{\mu}$ls_020.   The XPK or "-x" option ←
   will be
ignored if $\mathrm{\mu}$ls cannot find the xpkmaster.library.

  No special installation is required.

  On my A1200 (with a little UNIX flavour) I have copied $\mathrm{\mu}$ls ←
     to the
SYS:bin directory,   with the name "ls" and the
                 h bit
                 set, while the
User-Startup defines the alias "ll" as "ls -l".

1.5 Usage

                This command can be lauched from the shell only and ←
                  accepts the
parameters in both Amiga and UNIX style. The standard template is:

  DIR/M,P=PATTERN/K,ITSELF/S,FILES/S,DIRS/S,ALL/S,LONG/S,
SS=SHELLSIZE/S,WHITE/S,XPK/S

but please note that the "/M" after DIR is a "trick" to handle UNIX-style
options and that you cannot examine more than one directory.

  DIR indicated the name of the files or the directory to be examined.
You can use standard AmigaDOS wildcards. If omitted, $\mathrm{\mu}$ls ←
   examines the
current directory.

  With the PATTERN keyword you can limit the list to only those entries
that match a certain pattern, typically to hide some types of files. If
omitted, $\mathrm{\mu}$ls uses the default (hardwired) pattern:

```
   ~(#?(.info|.bak|,b)|.#?)
```

ie: it hides icons (#?.info), backups (#?.bak and #?,b) and files
starting with a dot (.#?). The ALL switch cancels whatever pattern set.

   The remaining options are boolean switches and have a corresponding
flag in the standard UNIX style. If you prefer the latter, please
remember that more flags can be specified in a single option, like "-lw"
instead of "-l~-w" and that flags are case-sensitive. Note also that
unlike UNIX commands $\mathrm{\mu}$ls accepts the options in any order.

   Valid options are:

ITSELF/S
-d or -i
    show informations about the directory DIR instead of its contents. It
    is of no use if used in the short format. It is identical to the "-d"
    option of the UNIX command "ls".

DIRS/S
-D
    only show the directories,   which includes hardlinks to dirs and
    softlinks.

FILES/S
-F
    only show the files. Please note that in the UNIX commmand "ls", the
    option "-F" has a different meaning.

ALL/S
-a or -A
    cancel whatever pattern was set, either via the PATTERN keyword or
    the default one.
LONG/S
-l
    use the
                long~format
              . If omitted $\mathrm{\mu}$ls uses the
              short~format
              .

SS=SHELLSIZE/S
-s
    set the output width (in the
                short format
) according to the width
    of the current console window. If omitted the width defaults to 78
    chars per row.

WHITE/S
-w
    show the directories in white.

XPK/S
-x
```

```
every  file  is  analysed:   if it is a xpk-compressed file $\mathrm{\mu}$ls  ←
    shows
additional info such as the compression method, etc.  This option, no
matter   if   there  actually  are  compressed  files,    forces  the
```

```
        long~format
   .
```

```
        Short format

        Long format

        Suffix
```

1.6 Short format

```
        In  the  short  format,   the list is split into columns whose  ←
            width is
dynamically determined,  according to the longest name in the list and to
the width of the current console window, if the SHELLSIZE option is used.
The  entries are sorted by column (à la UNIX),  not by row (like Dir) and
all the directories are listed before the files. The name of the dirs can
be followed by a
        suffix
        that indicates the type of that entry.

  EG:
```

```
archives/     obj/       tmp#
bin/       sources/      working/
includes/     tests/
abort.c    main.c     x.o
Art.iff    startup.o
az       x.c
```

1.7 Long format

```
        In the long format each entry takes an entire line. Various  ←
            details are
shown:

  cppppppppp ssssssss ddddddddd ttttt n...nx
```

The first character tells the type of the entry:

```
  - = file
  d = directory
  D = hardlink to a directory
  l = symbolic link (softlink)
  r = root directory
  p = pipe
  x = XPK file (only with -x or XPK)
```

   Then come the protection bits (p), the filesize (s), the date (d) and
time (t) of the last modification and lastly the name (n), optionally
followed by the
                 suffix
                    .

   When you specify the -x or XPK option, the compressed files are shown
in a slightly modified format:

```
   cppppppppp ssssssss/uuuuuuuu mmmme n...nx
```

The actual size (s) of the file is followed by the size of the
uncompressed file (u), the four characters that identify the method (m)
used optionally followed by an asterisk (e) if a password is required
touncompress that file.

1.8 Suffix

   Each entry can be followed by a character which indetifies the type of
that entry:

```
  / = directory
  # = hardlink to a directory
  @ = symbolic link (softlink)
  : = root directory
  | = pipe
```

The files and the hardlinks to a file do not have any suffix.

1.9 Notes

                  h bit

                  Source

1.10 h bit

Many Amiga users think that the h flag stands for "hidden" (doing so they tremendously confuse the AmigaDOS protection bits with the MS-DOS attributes) and that it is not used by the system. That's not true! The h flag stands for "hold" and its function is that of a "delayed Resident": the executable flagged with "h" will be made resident when it will be called the first time.

1.11 Source

The included source is very badly commented, in a mixture of English and Italian and so very hard to understand. However you should not have trouble if what you want is just to change the hardwired default pattern or -perhaps- the color of the directories...