Umentiler

Lee Kindness

COLLABORATORS			
	TITLE : Umentiler		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Lee Kindness	June 9, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Umentiler

1.1 Umentiler 1.186

```
                    Umentiler :                    Umentiler version 1.186
                                  Copyright ©1995 Lee Kindness
 document
     compiler                     Easier document management... or something!


                   Introduction
                  : What does it do?

                      Usage
                  : More options...

                      Legal
                  : Copyright, distribution...

                     History
                  : Version history

                      Other
                  : Other programs

                     Contact
                  : Author contact

---
Compiled              Saturday 28 October 1995 18:55:45
Distribution built    Sunday 29 October 1995 14:27:20
Exe size              7612 bytes
Main source size      21476 bytes
Version               Umentiler 1.186 (28/10/95)
```

1.2 Introduction

```
 Right... what does it do? Umentiler simply compiles documents. It replaces
```

```
and expands certain tags in the file. For example you could insert the
current time, version of a command, output from a command, another file
into a file.

...But why? Well I am the author of over 28 programs, and a constantly
releasing new versions. When a new version is released I would have to
manually update the version numbers and my (constantly changing) email
addresses... Rather frustrating!

This guide, of course, has been compiled with the aid of Umentiler. The
uncompiled version is called 'Umentiler_uc.guide'. You may wish to compare
these documents...
```

1.3 Usage

```
                Invocation
                : Launching Umentiler

                  Format
                : Format of tags in files
```

1.4 Launching Umentiler

```
                Umentiler must be launched from a Shell. It accepts the following ↩
                    argument
template:

  SOURCE/A,DESTINATION,QUIET/S

In short the first argument is the file to
               process
                and the second is the
destination. If you do not pass DESTINATION then SOURCE will be overwritten!
For example:

  >Umentiler textfile textfile.um

Will expand all the
               tags
                in 'textfile' and output to 'textfile.um'

If you specify QUIET then Umentiler will not print out the number of processed,
escaped and invalid tags.
```

1.5 Tag Format

```
                (If you are using Amigaguide rather than Multiview to view this  ↩
                   then the
output might not be correct...)
```

The source file that you supply Umentiler with should be plain ASCII. It can contain 'tags' which will be processed by Umentiler. The format of the tags is:

```
  />Command options...<\
```

Where 'Command' is one of:

```
                Insert

                Version

                Program

                 Date

                Setvar

                  Var
                The command and arguments are not case sensitive. All the  ↩
                   commands accept
```

arguments in the normal Amiga ReadArgs template form.

The output from the Insert and Program commands is itself parsed for tags. They are then replaced.

To escape a tag, ie if you really want to include it in your text, then you simply precede it by '!':

```
  !/>This will not be processed<\
```

You can add as many '!'s as you wish, one less is always printed.

NOTE:

If you are putting an Amigaguide tag directly after an escaped Umentiler tag then you should leave a space between the end of the Umentiler command and the start of the Amigaguide tag:

```
  @{i}!/>Command options...<\ @{ui}
```

rather than:

```
  @{i}!/>Command options...<\
```

Failing to leave a space will cause the Amigaguide tag to be escaped by the trailing '\' in the escaped Umentiler tag.

1.6 The INSERT Command

The Insert command is used to insert the contents of a file. It accepts the following argument template:

```
  FILE/A

FILE: The path and name of the file to insert, must be given.

For example:

  />Insert S:Startup-Sequence<\

would insert your startup-sequence into the file.
```

1.7 The VERSION Command

```
The Version command is used to insert version information from a program
or file. It accepts the following argument template:

  FROM/A,F=FULL/S,NN=NONAME/S,NV=NOVER/S,ND=NODATE/S

FROM: The file or program to read the version information from. Must be
    given. NOTE since this command uses the 'Version' AmigaDOS command
    you can also supply additional arguments to 'Version' in it, you
    could for example use "exec.library INTERNAL" for this argument.

FULL: Print out the whole persion string, do not parse further with the
    NONAME, NOVER and NODATE options. Usefull if the version string
    is nonstandard.

NONAME: Specifies not to print the name part of the version information.
    You can use NN for short.

NOVER: Specifies not to print the version part of the version information.
    You can use NV for short.

NODATE: Specifies not to insert the date section of the version information.
    You can use ND for short.

For example:

  />Version C:Dir<\

would produce:

  dir 37.5 04/06/91

  />Version C:List NONAME ND<\

would produce:

  37.5

  />Version C:Type ND<\

would produce

  type 37.2
```

NOTE:

If the version string of the program is not 100% in the standard format then the results of the NONAME, NOVER and NODATE command options will not be as expected! This is not my fault it is the fault of that programs author.

1.8 The PROGRAM Command

The Program command is used to insert the output of a command or script. By default the commands input stream is NIL:, you can change this by using redirection in the CMDLINE. It accepts the following argument template:

```
  NL=NOLINES/S,CMDLINE/A/F
```

NOLINES: Specifies that you want any newline characters generated by the command to be stripped. You can use NL as a shortcut.

CMDLINE: The command line to launch. This argument automatically expands to the end of the tag, so if the command line has spaces in it there is no need to suround it in quotes. Must be given.

For example:

```
  />Program List C:d#?<\
```

would use an 8000 byte stack and produce:

```
  Directory "C:" on Sunday 29-Oct-95
Date                        1092 --p-rwed 02-Sep-92 11:51:33
Delete                      1972 --p-rwed 02-Sep-92 11:51:33
Dir                         3440 --p-rwed 02-Sep-92 11:51:33
DiskChange                   312 --p-rwed 02-Sep-92 11:51:33
4 files - 19 blocks used
```

```
  C:Dir is />Program NOLINES List C:Dir LFORMAT=%l<\ bytes long
```

would produce:

```
  C:Dir is 3440 bytes long
```

NOTE:

If you re-direct the output of the command to NIL: then it has the effect of executing a command but not inserting any text.

1.9 The DATE Command

The Date command can be used to insert the date or time. It accepts the following argument template:

```
  FROM/K,LF=LOCALEFORMAT/K,F=FORMAT/N/K,NY=NODAY/S,ND=NODATE/S,NT=NOTIME/S

FROM: Normally the date/time is read in from the system clock. If this
    keyword is specified then they are read in from the specified file.
    Keyword.

LOCALEFORMAT: This keyword can be used to specify a locale date format
    string. Keyword. Can use FL for short. The format string may contain
    the following formatting codes:

        %a - abbreviated weekday name
        %A - weekday name
        %b - abbreviated month name
        %B - month name
        %c - same as "%a %b %d %H:%M:%S %Y"
        %C - same as "%a %b %e %T %Z %Y"
        %d - day number with leading 0s
        %D - same as "%m/%d/%y"
        %e - day number with leading spaces
        %h - abbreviated month name
        %H - hour using 24-hour style with leading 0s
        %I - hour using 12-hour style with leading 0s
        %j - julian date
        %m - month number with leading 0s
        %M - the number of minutes with leading 0s
        %n - insert a linefeed
        %p - AM or PM strings
        %q - hour using 24-hour style
        %Q - hour using 12-hour style
        %r - same as "%I:%M:%S %p"
        %R - same as "%H:%M"
        %S - number of seconds with leadings 0s
        %t - insert a tab character
        %T - same as "%H:%M:%S"
        %U - week number, taking Sunday as first day of week
        %w - weekday number
        %W - week number, taking Monday as first day of week
        %x - same as "%m/%d/%y"
        %X - same as "%H:%M:%S"
        %y - year using two digits with leading 0s
        %Y - year using four digits with leading 0s
        %% - percentage sign

    You will require Workbench 2.1 or higher to use this option. If you
    use this option then the FORMAT, NODAY, NODATE and NOTIME options
    are ignored (unless locale.library is unavailable).
    (NOTE: it appears that locale.library has a bug in it... this results
    in the %U and %W formatting codes giving incorrect results)

FORMAT: Format of the produced date. Keyword. Can use F for short. You can
    use any of the following:

        0 - DOS format, dd-mmm-yy (Default)
        1 - International format, yy-mm-dd
        2 - USA format, mm-dd-yy
        3 - Canadian/British format, dd-mm-yy
        4 - Default locale format
```

```
 NODAY: Specifies that the day part of the date string should not be inserted.
     Can use NY for short.

 NODATE: Specifies that the date part of the date string should not be
     inserted. Can use ND for short.

 NOTIME: Specifies that the time part of the date string shouldn't be inserted.
     Can use NT for short.

For example:

   />Date LF="%n%A %d %B %Y%n%a %d %b %y%nDay%t%j"<\

would produce:


Sunday 29 October 1995
Sun 29 Oct 95
Day 285

   />Date F=0<\
   />Date F=1<\
   />Date F=2<\
   />Date F=3<\
   />Date F=4<\

would produce:

   Sunday 29-Oct-95 14:27:22
   Sunday 95-Oct-29 14:27:22
   Sunday 10-29-95 14:27:22
   Sunday 29-10-95 14:27:22
   Sunday 29/10/95 14:27

   />Date NODATE NOTIME<\
   />Date NODAY NOTIME<\
   />Date NODAY NODATE<\

would produce:

   Sunday
   29-Oct-95
   14:27:22
```

1.10 The SETVAR Command

```
                 This command never causes any text to be inserted. It allows you ↩
                    to set up
 a local environmental variable. This could be useful say if you were wanting
 to include a legal document (copyright, disclaimer...) in each of your
 program documents. You would use this command to set a variable to the
 program name and then in the lagal document you could insert the program
 name using the
                VAR
```

```
                command. This command accepts the following
argument template:

  NAME/A,TEXT/A/F

NAME: The name of the variable to create. This name follows file system
    naming rules. Must be given.

TEXT: What to set the variable to. Expands to the end of the tag, so no
    need to enclose in quotes. Must be given.

For example:

  />Setvar PROGRAM Umentiler<\

would setup a variable called 'PROGRAM' and set it to 'Umentiler'

  />Setvar NAME Lee Kindness<@{ui}

would set up a local variable called 'NAME' and set it to 'Lee Kindness'
```

1.11 The VAR Command

```
The Var command can be used to insert the contents of local and global
environmental variables. It accepts the following argument template:

  FROM/A,NL=NEWLINE/S,G=GLOBAL/S,L=LOCAL/S

NAME: Name of the variable to insert. Must be given.

NEWLINE: If specified then a newline will be inserted after the variable
    is printed. Can use NL for short.

LOCAL: Specifies only to search for a local variable that matches NAME.
    By default a local variable is searched for first, then a global. Can
    use L for short.

GLOBAL: Specifies only to search for a global variable. Can use G for
    short.

For example:

  Process />Var process L<\, RC = />Var RC L<\, Secondary RC = />Var Result2 L<\

would insert:

  Process 6, RC = 0, Secondary RC = 0

  />Setvar PROGRAM Umentiler<\/>Var PROGRAM L<\

would insert:

  Umentiler
```

1.12 Disclaimer, Distrubution and Copyright

Disclaimer
~~~~~~~~~~
I hereby reject any liability or responsibility for these or any
other consequences from the use of Umentiler whatsoever. This includes,
but is not limited to, damage to your equipment, to your data,
personal injuries, financial loss or any other kinds of side effects.
Although Umentiler has been tested thoroughly on several different
machines, I cannot rule out the possibility that Umentiler:

1. is somehow incompatible to your equipment

2. has bugs that show up on your equipment

3. does not do what it is supposed to do on your equipment

4. May damage you equipment

It is your responsibility to take any precautions necessary to
protect yourself from these or any other effects. I explicitly
reject any liability or responsibility from the consequences of you
using Umentiler

Distribution
~~~~~~~~~~~~

Umentiler may be freely distributed and copied, as long as the following
conditions are fulfilled:

1. All parts of the program and the documentation must be complete.
   The distribution of single parts or incomplete subsets of the
   original distribution is forbidden.

2. If Umentiler is to be included in a commercial distribution (including
   magazines!) then I must be sent a copy of the product (or if it is a
   mag that I am subscribed to then an increase in subscription). It
   would be better if you contacted me beforehand to ensure you have the
   latest version. In any case full credit must be given within the
   magazine (ie mention my name).

3. If the included source code is used in another program then credit
   must be given in the documentation.

Note To Magazines
~~~~~~~~~~~~~~~~~

In the past I have had some of my programs included on magazine coverdisks
without my permission when in the documentation it was stated that this
was required... I want my programs on coverdisks but you _MUST_ read and
fufil the "Distribution" section above. If you disagree then contact me
or write a message in the AMIGA_MAGS echo...
~~~~~~~~~~~~~~~~~

1.13 Contact

```
Means of contact, as of Sunday 29 October 1995

Mail:
  Lee Kindness
  8 Craigmarn Road
  Portlethen Village
  Aberdeen AB1 4QR
  SCOTLAND

EMail:
  cs2lk@scms.rgu.ac.uk
  wangi@spuddy.mew.co.uk
  wangi@fido.zetnet.co.uk
  :
  :
  Please use cs2lk@scms.rgu.ac.uk

Fidonet:
  2:259/26.20 "Lee Kindness"

Amiganet:
  39:138/40.34 "Lee Kindness"

I hope you find the program useful.

---      __       _____      _
Wangi! / /      (  __) /|/ )  Lee Kindness Amiga PD/shareware author
 ______/ (_/\  __\ \  / _ (_______________  cs2lk@scms.rgu.ac.uk
(_____________)(_____)(__)\_________________) wangi@spuddy.mew.co.uk
```

1.14 History

```
 1.186  : (29.10.95)
        + First release version.
```

1.15 Other Programs

```
SlowBoot:
  Offers a software solution to the problem of slow spin-up IDE drives.

FindSystem:
  Searches a nodelist for nodes those names match a wildcard.

FindSysop:
  Searches a nodelist for nodes those sysops names match a wildcard.

FindPlace:
  Searches a nodelist for nodes those location match a wildcard.

FindPhone:
```

```
  Searches a nodelist for nodes those phonenumber match a wildcard.

NLFind:
  Provides multiline node support for traplist.library. (comm/fido/NLFind.lha)

Erase:
  Delete a file for GOOD... you will not be able to recover it!
  (util/cli/Erase.lha)

YourFault:
  Replace the system error requesters with your own... "Banana in drive
  DF0:" :)) (util/boot/YourFault.lha, YourFaultSrc.lha, YourFaultStr.lha)

CaBoom:
  Windows explode/implode on opening/closing...
  (util/boot/CaBoom.lha)

Shrub:
  Creates a directory tree, like Tree from MSDOS but it has a nice Workbench
  GUI, search facility, is very fast and a lot more. (util/wb/Shrub.lha)

RubbishDump:
  A replacement for the trashcan... but it has SOUND! So when you delete a
  file a sample is played (can in reality send any ARexx command).
  (util/app/RubbDump.lha)

AmigaGuidePrefs: (or AGPrefs to its friends...)
  A complete preference editor for amigaguide. Lets you modify the colours
  of certain attributes (has a preview), type of node and search paths.
  (text/hyper/AmigaGuidePref.lha)

WangiPad:
  A powerful launchpad utility (ie. like toolmanger). The new version 2 should
  make WP more powerful than ToolManager... watchout for it!
  (util/wb/WangiPad.lha)

Jiggler:
  A WB hack. Makes all window move towards the mouse pointer.
  (game/gag/Jiggler.lha)
  DROP INTO YOUR 'FRIENDS' WBSTARTUP DRAWER...

DQua:
  A simple quadratic equation solver. (misc/math/DQua.lha)

MacCash:
  Generates UK lottery numbers. Does not falsely predict them, it is 100%
  random. It hace a nice Workbench GUI. (misc/misc/MacCash.lha)

DefDataTypeIconer: (or DefDTIcon...)
  Lets you change files icons, based on their datatype!
  (pix/icon/DefDTIcon.lha)

NewEXT:
  With the AmigaDos rename command you can do "rename #?.txt TO #?.doc" with
  NewEXT you can... (util/batch/NewEXT)

Startup-Menu:
```

```
  A startup menu/utility. Lets you change the 'startup-sequece' used. Very
  customisable. (util/boot/Startup-Menu.lha)

Bush:
  Like Shrub but Shell only and slower... (util/cli/Bush.lha)
  USE SHRUB.

DiceRoll:
  Emulates dice throws... ie. generate random numbers. (util/misc/DiceRoll.lha)

Publican:
  PublicScreen utility. (util/misc/Publican.lha)

MidMoose:
  Emulates middle mouse button for those with only two buttons on their mice.
  (util/mouse/MidMoose.lha)

NaeGrey:
  Makes the grey borders around screens black. (util/misc/naegrey.lha)
  USE A PROGRAM LIKE MultiCX INSTEAD!

SFPatch:
  Code showing how to patch Amiga library functions. (dev/c/SFPatch.lha)

Torch:
  A blanker module for SwazBlanker.
```