TTFLIB

COLLABORATORS			
	TITLE : TTFLIB		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		April 14, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

TTFLIB

1.1 main

```
                ttf.library version 0.7.5

A truetype font engine for Amiga computers.

  Reinstall warning - Fonts installed with previous versions
  of ttf.library, version 0.6.3 and below  must be reinstalled with
  version 0.7 or above for proper operation.
```


The most current publicly available version of this software

can be found at the following web address:

      http://home.sprynet.com/sprynet/ragriffi/

Send comments, suggestions, and bug reports to:

  Richard Griffith
  ragriffi@sprynet.com

1.2 ttf.library Overview

Overview

ttf.library is a truetype compatible font engine for Amiga OS.
It functions in a manner compatible with the outline font engine
standard established by Commodore with the bullet.library engine
for compugraphic format fonts. This means that Amiga applications
which use normal system fonts are now able to use truetype fonts.

1.3 Credits and Legal Issues

Credits and Notes

The ttf.library and related programs were made possible by
the outstanding achievement known as the FreeType Project.
For more information, visit http://www.freetype.org/

Also, Amish S. Dave's type1.library (which does for postscript
type1 fonts what ttf.library does for truetype) served as an
invaluable inspiration and an initial guide for the rather
poorly documented amiga outline font engine format.

Legal

I am not a lawyer, nor do I want to be, so expect plain
english here.

First, this library comes with NO WARRANTY.

ttf.library is free, think of it as my gift to the faithful.
I do not restict its use and/or distribution, however, some
of the technology used is covered by a separate license, see
the
FreeType license
 for details. I do ask that if you
distribute it, please don't try and call it your own work. I
know better, you know better, and I'm sure karma will get you.
If you sell it, be advised you will have stiff competition,
as it will continue to be available free. (As a side note, no

I don't think all software should be free. I make my living as
a programmer, consultant, general geek type, so I do place a
great value on software. This one, though, I feel should be
free. So there...)

Also note that it is virtually impossible to refer to anything
relating to the subject of fonts without the mention of at least
some trademarked names. ttf.library asserts NO CLAIM to ANY sort
of trademark whatsoever. The following list of trademarks is
provided, but may not be complete:

Amiga is a trademark of Amiga, Inc.
TrueType is a tradmark of Apple Computers, Inc.
Microsoft® and Windows® are registered trademarks of Microsoft
Corporation
M$ and Windoze are derisive terms, unregistered by any party
to the best of my knowledge.

1.4 FreeType Project LicenseSystem Requirements

The FreeType Project LICENSE

Copyright 1996-1998 by
David Turner, Robert Wilhelm, and Werner Lemberg

Introduction:

The FreeType Project is distributed in several archive packages;
some of them may contain, in addition to the FreeType font engine,
various tools and contributions which rely on, or relate to, the
FreeType Project.

This license applies to all files found in such packages, and
which do not fall under their own explicit license. The license
affects thus the FreeType font engine, the test programs,
documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic and IJG
(Independent JPEG Group) licenses, which all encourage inclusion
and use of free software in commercial and freeware products
alike. As a consequence, its main points are that:

- We don't promise that this software works. However, we'll be
  interested in any kind of bug reports. ("as is" distribution)

- You can use this software for whatever you want, in parts or
  full form, without having to pay us. ("royalty-free" usage)

- You may not pretend that you wrote this software. If you use
  it, or only parts of it, in a program, you must acknowledge
  somewhere in your documentation that you've used the FreeType

code. ("credits").

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products, provided that all warranty or liability claims are assumed by the product vendor.

Legal Terms:

0.Definitions:

Throughout this license, the terms "package", "FreeType Project", and "FreeType archive" refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the "FreeType project", be they named as alpha, beta or final release.

"You" refers to the licensee, or person using the project, where "using" is a generic term including compiling the project's source code as well as linking it to form a "program" or "executable". This program is referred to as "a program using the FreeType engine".

This license applies to all files distributed in the original FreeType archive, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType project is copyright (C) 1996-1998 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

1. No Warranty:

THE FREETYPE ARCHIVE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

As you have not signed this license, you are not required to accept it. However, as the FreeType project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType project, you indicate that you understand and accept all the terms of this license.

2. Redistribution:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistribution of source code must retain this license file (LICENSE.TXT) unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. All copyright notices must be preserved in all copies of source files.

- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType team in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType code, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

3. Advertising:

The names of FreeType's authors and contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: "FreeType Project", "FreeType Engine", "FreeType library", or "FreeType Distribution".

4. Contacts:

There are two mailing lists related to FreeType:

- freetype@lists.lrz-muenchen.de

  Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you're looking for support, start in this list if you haven't found anything to help you in the documentation.

- freetype-devel@lists.lrz-muenchen.de

  Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

- http://www.physiol.med.tu-muenchen.de/~robert/freetype.html

  Holds the current FreeType web page, which will allow you to download our latest development version and read online

```
      documentation.


  You can also contact us individually at:

    David Turner      <turner@email.enst.fr>
    Robert Wilhelm    <robert@physiol.med.tu-muenchen.de>
    Werner Lemberg    <wl@gnu.org>


                           --- The End ---
```

1.5 System Requirements

```
Requirements

OS3.0 or higher
68030 or higher processor

It would be possible to make a plain 68000 version, but so far
no one has asked.
```

1.6 Installation

```
Installation

Install-ttflib is a standard installer script, just double
click the icon.

If you prefer a manual installation, here is a breakdown
of the primary components:

File         Installation instructions
------------ ------------------------------------------------------
ttf.library  copy to your LIBS: drawer
ttfmanager   copy to your favorite tool drawer (i.e. sys:utilities)
ttfinstall    "   "   "      "      "     "    (optional)
ttflist       "   "   "      "      "     "    (optional)
ttfcp         "   "   "      "      "     "    (optional)
ftview        "   "   "      "      "     "    (optional)
```

1.7 Using ttf.library

```
               Usage

The following tools are available:

                ttfmanager
                make truetype fonts available to system
```

ttfinstall
alternate (CLI) tool to install fonts

ttflist
list names and other details from font files

ttfcp
codepage tool for non ECMA-Latin1 usage

ftview
preview a font

1.8 Using ttfmanager

ttfmanager

To use a truetype font, it must first be installed to some part of the system FONTS: drawer. Installation is a bit different than with the compugraphic intellifont program. First, outlines (the actual truetype font files) are NOT copied to the font directory. Instead only the ".font" and ".otag" files are created in the selected directory, and those reference the outlines at the same locations from which they were installed. This makes it much easier to use large font libraries directly from CD-ROM, for example. The font files used by the bullet.library in some cases are actually modified by the installation process, making the copying a requirement. In contrast, ttf.library uses unmodified files, so there is no reason to restrict your choices in disk space management. If you really want a "_ttfoutlines" directory in FONTS:, create it, and copy your font files there before the installation.

The second major difference is that the fonts may be installed to any drawer, not just a FONTS: component. (Note that to USE a font in many programs, the drawer must be assigned to FONTS:, but this allows you to selectively add the required font path components as needed.) This allows for a rotating assignment for FONTS: which is practically required with any large font collection. (Image how long it would take to load the standard system font requester if you had 6000 fonts actually available!)

Also see:

Quick Instructions

Main Window

Options Window

Preview Window

Information Window

Keyboard Controls

Workbench Options

Shell Options

1.9 Quick Font Installation

Quick Instructions

To install truetype fonts, enter a source directory (where the ttf files are located) in the source directory box. Clicking the 'set' button to the right will bring up a standard file requester to allow you to choose the directory. A list of the internal names of all installable truetype fonts in the source directory will be generated in the Available Fonts list.

Set the Destination Drawer in a similar fashion.

To install a single font, click on it in the Available Fonts list. If you desire, you can change the name in the Font Name box. Then click the 'Install' button.

To install all of the fonts in the Available Fonts list, click on the 'Install All' button. All fonts installed this way will get default names and options.

Please note that ttfmanager will OVERWRITE the .font and .otag files if they already exist, thus installing two fonts with the name 'CoolNewFont' will result in only one available font.

1.10 ttfmanager Main Window

ttfmanager Main Window

ttfmanager features a multi-window interface, providing only as much as you decide you want to see. The main window provides the most basic functions, selecting and installing truetype fonts. Closing the main window exits ttfmanager.

Available Fonts

Shows all the available fonts in the Font Source directory. To select a font for further processing, click on it in this list.

Font Source

The location of the actual font file. The Set button will bring up a standard file requester, from which a suitable directory can be chosen. This gadget can also accept a wildcard pattern to

match only specific files. For example, CD0:A#?.ttf will restrict the files shown in the Available Fonts list to only those fonts where the filename starts with 'A' and ends in ".ttf".

See
Workbench Options
to set a default value for this field.

Font Destination

The directory into which the .font and .otag files which allow a font to be used by the system will be placed by Install. The Set button allows choosing this from a standard file requestor.

See
Workbench Options
to set a default value for this field.

New Font Name

ttfmanager will fabricate a name for a font from information within the font file itself, and place it here. To change the name before installing, overtype it here.

Install

Installs the currently selected truetype font as an Amiga font named as specified in New Font Name, in the directory specified by Font Destination, using any options specified in the options window.

Install All

Installs all the fonts in the Available Fonts list into the Font Destination directory. All fonts will be given default names and options.

Preview

Opens the
Preview
window.

Info

Opens the
Font Information
window.

Options

Opens the
Options
window.

The message area across the bottom of the window displays any status or error messages.

1.11 ttfmanager Options Window

ttfmanager Options Window

Encodings

Displays all encoding tables available in the currently selected font. ttfmanager will attempt to identify an appropriate unicode table. If the chosen table is incorrect, click on the desired one to change it.

Selected/Raw/Offset

If desired, you may choose to bypass the encoding table and access the glyphs of a font directly. The raw option maps glyphs in order, beginning at the specified Offset, to ASCII printable charaters only (values from 33 to 126.) This is intended for use with dingbat and clipart fonts, or for accessing otherwise unreachable glyphs in large fonts for artistic purposes only.

Code Page/Use/Set

This cluster of gadgets allows a specific codepage to be associated with a specific font. In the string entry gadget, specify the full path of the desired codepage. The Set button brings up a standard file requestor which can be used to pick a codepage file. For more information on the codepage format, see
ttfcp
.

The Use gadget turns the font specific codepage on and off. If on, the compiled codepage information is stored directly in the .otag file created by an Install operation. The source codepage file itself is never referenced by the ttf.library. The .otag codepage, if specified, overrides the default one, either built-in or set with ttfcp.

Note that these options are NOT reset by selecting a new face from the Available Fonts list, so, by setting the codepage once, a series of fonts can be installed using it.

New Font Name

This is a second copy of the field of the same name on the

main window
, placed here for convienience sake. Note that you can create multiple Amiga fonts from a single truetype file, each with its own options, but remember to give each a unique name here.

Font Attributes

The following attributes are used by the diskfont.library, or similar font mapper, to determine the attributes of the outline font. The default setting are usually adequate, but if you feel that ttfmanager has misidentified a font, they may be changed.

Slant Style - Controls Italic attribute
Weight - Controls the Bold attribute
Width - Controls the Extended (wide) attribute
Serif - Seems to be a comment(?)
Spacing - Controls Monospaced (fixed pitch) attribute

Sizes

A list of sizes which will be listed for this font in the system available fonts list. Use the Add, Del, and size entry gadget to modify this list. See
Workbench Options
to
change the defaults used here.

Stretch

Perhaps the most visually significant switch on this screen. By default, ttf.library will fit a fonts entire bounding box into the amiga sized box, which for screen fonts is typically 'size' pixels high. This has the unfortunate side effect of making smaller sizes quite illegible. Checking the Stretch box will eliminate most of this adjustment, and smaller sizes will be more readable, but there is a price for this. Many irregularities, mainly a jumping baseline, will result. You may or may not be satisfied with the results.

Note that Preview will show the 'stretch' size, but it will not properly illustrate the imperfections which may result. For the curious, the shifting baselines are the result of the diskfont.library trying to fit the glyphs into the pixel height box. If the ascent is too high, the bit map is pushed down, resulting in a low baseline. If the descent is too low, the bit map is moved up. Interestingly, ttf.library does not explicitly tell diskfont.library how to locate the baseline, as it infers it instead from the glyph descriptions.

```
Test - not yet implemented


Info

 Opens the
                Font Information
                 window.


Preview

 Opens the
                Preview
                 window.


Install

 Installs the currently selected truetype font as an Amiga font
 named as specified in New Font Name, in the directory specified
 by Font Destination (on the
                main window
                ) using any specified
 options.


The message area across the bottom of the window displays any
status or error messages.
```

1.12 ttfmanager Preview Window

```
                ttfmanager Preview Window

This optional window will display a sample of the currently
selected font. Resizing the window will redisplay the sample
in a size determined by the new window size.

See
                Workbench Options
                 to set the default string used for the preview.

Preview also has several
                 keyboard control
                 options:

ESC or DEL    - clears the current preview string
Backspace     - remove the last character from the preview string
any character - adds the character to the preview string
Cursor Left   - Apply kerning to preview display
Cursor Right  - Turn off kerning in preview display
```

1.13 ttfmanager Keyboard Controls

```
ttfmanager Keyboard Controls

The following keys are available in any ttfmanager window:

Cursor Down   - Select the next font from the Available Fonts list
Cursor Up     - Select the previous font from the Available Fonts list
```

1.14 ttfmanager Font Information Window

```
ttfmanager Font Information Window

This optional window shows Name and Copyright information, as
well as various statistics and metrics from the currently
selected font.
```

1.15 ttfmanager Workbench options

```
ttfmanager Workbench options

ttfmanager supports the following ToolTypes in its icon:

SOURCE
  Specifes the intial Font Source in which to look for truetype
  font files. Example: SOURCE=Work:myfonts

FONTS
  Specifes the intial Font Destination, a directory in which the
  .otag and .font files for an intalled truetype font will be
  placed. Example: FONTS=Sys:ttfonts

SIZES
  Specifes the default Sizes as seen on the options page. This
  specifies the font sizes which will be reported in the system
  AvailFonts list for use in font requestors. Up to 20 sizes may
  be specified in a comma separated list. Example:
  SIZES=10,12,16,24,32,48,64,72,98,122

PREVIEW
  Specifes the string used for the Preview option Example:
  PREVIEW=Every Good Boy Does Fine

Additional arguments, such as shift-clicked icons, will be used
as the Font Source. For example, select a drawer on the workbench,
hold down the shift key and double click the ttfmanager icon,
and ttfmanager will use the drawer as the initial Font Source.
Also, a project icon associated with a truetype font file could
```

refer to ttfmanager as its default tool, invoking ttfmanager
when clicked, which would set the Font Source to the single file.
Note that the project file does not have to be a font file. Any
ToolTypes in the project icon are evaluated as well.

1.16 ttfmanager Shell options

ttfmanager Shell options

The current command line interface will change soon

FROM/A,TO/A,SIZES/F

Example: ttfmanager work:coolnewfonts fonts: sizes=12,24,36,48

1.17 Using ttfinstall

ttfinstall

Note:
ttfmanager
is now available and will perform font
installation from the Workbench environment. This tool is
included for those who want CLI and scripting power. For
an overview of font installation, including differences
from the system intellifont program, see
ttfmanager
.

The command line for installing a truetype font looks like
the following:

```
ttfinstall fontfile.ttf sys:fonts
```

Note that 'fontfile.ttf' is the actual file to be installed, and
'sys:fonts' can be any existing directory. In addition, a file
name pattern can be specified for 'fontfile.ttf', allowing the
installation of several fonts at once. For example:

```
ttfinstall cd0:funky/#?.ttf work:ttfonts
```

Would install all the font found in the cd0:funky drawer to the
ttfonts drawer on volume work:

Once a truetype font is installed on the fonts: path, it should
be possible to select that font from any decent font requester.

Please note that ttfinstall will OVERWRITE the .font and .otag
files if they already exist, thus installing two fonts with
the name 'CoolNewFont' will result in only one available font.
It is possible to rename both the .font and .otag files once
they have been created, so multiple fonts with the same internal

```
name can be used if you work at it. ttfmanager has an option
to change the name when installing individual fonts.

Note: switches EXACT, TYPO, and DESIGN are no longer valid.
```

1.18 Using ttflist

```
ttflist

ttflist works a lot like the normal system 'list' command, but
is designed to display readable names along with the file name.
Also, it will optionally provide a great deal of perhaps useful
information from a font file. The command line would look like:

  ttflist [pattern] [all] [verbose]

where:

 pattern  is an optional amigados file pattern or
          drawer name. By default it is "#?.(ttf|ttc)"

 all      is an optional switch to list matching drawer
          contents as well. Note that the pattern specified
          must match the drawer names to be searched. When
          all is specified the default pattern is "#?".

 verbose  gives a lengthy, and probably ignorable, report for
          each font found.

example:  ttflist cd0: all
```

1.19 Using ttfcp and alternate codepages

```
ttfcp and alternate codepages

ttfcp is a command line tool to install codepage tables that
allow the use of ttf.library on systems which do not use the
default ECMA Latin-1 character set. ttfcp uses simple text files
to describe the mapping of the 256 possible Amiga character set
positions to any unicode characters. The text file format is
quite simple:

- Each mapping line contains two numbers separated by space
  and/or tab characters. These numbers may be in decimal,
  octal, or hexidecimal. (decimal numbers must start with
  a digit 1-9, octal with a "0", and hexidecimal with "0x")
  The first number is the Amiga character set position being
  mapped, and must be between 0 and 255. The second number
  is the unicode character to be assigned to the position.
- A "#" character begins a comment
- any line without two valid numbers as the first non space
```

```
  items are ignored without warning
- any character positions not mapped will be set to 0

Sample mapping file lines: ---------------------------------
# full line comment
0x20  0x0020
32 32      # functionally identical to the previous line
325 0x0042 # this line would be ignored (index>255)
49  72659  # this line would be ignored (unicode>0xFFFF)
0xA1  0x040E  # CYRILLIC CAPITAL LETTER SHORT U
0xB6  0x0386  # GREEK CAPITAL LETTER ALPHA WITH TONOS
------------------------------------------------------------

Note: A variety of mapping tables suitable for use with ttfcp
      are available at ftp.unicode.org in Public/MAPPINGS.

ttfcp translates the mapping table into a more efficient form
for use by the library. The command line is:

  ttfcp mapfile [ENV] [TO outfile]

where:

 mapfile  is the required mapping file as described above.

 ENV      is a switch which will cause the translated table
          to be placed in the "ttfcodepage" environment
          variable. (And copied to ENVARC:)

 TO file  allows writing the translated table to "file"

example: ttfcp 8859-2.TXT ENV

If neither ENV or TO options are specified, no translated
table is written, but an normalized ascii version of the
input is displayed.

To enable alternate codepage support:

1) obtain or create a mapping file
2) use ttfcp with the ENV option to install the mapping file

These steps need only be performed once. The next use of the
ttf.library will begin using the new codepage mapping. To
remove codepage mapping delete the file "ttfcodepage" from
ENV: and ENVARC:.

For codepage support to work correctly, the fonts used must
contain the glyphs needed to represent the characters of
the chosen codepage. Also, the font must have a suitable
unicode encoding table.
```

1.20 Using ftview

```
ftview

ftview is a truetype font display program from the freetype
project test suite, with very minimal 'amigaizing'. Its
command line format is:

  ftview [-g] [-r res] pointsize fontfile ...

where

 -g        is an optional grey scale (smoothing) rendering.
           Smoothing looks very wonderful, but is not available
           through ttf.library. Also note that ftview opens on
           the default public screen, and if that screen does
           not have sufficient available or matching pen colors,
           it may look worse. The -g option uses five distinct
           grey levels.

 -r res    specifies an optional resolution to use in rendering
           the qlyphs of the font. 'res' is in Dots Per Inch.
           Note this doesn't change the display resolution
           only the value used for font rendering.

 pointsize is a required argument specifying the point size
           to display.

 fontfile  is one or more truetype font files to view

Example: ftview 30 flyingp.ttf

While the font is displayed, options to change many rendering
characteristics are available by keystroke or standard amiga
menu selections.
```

1.21 Limits and known problems

```
Limits, warnings

Portions of this software are new, and have had minimal testing.
The performance of this software has been heavily monitored on
the development machine with enforcer, mungwall, poolwatch and
other tools, however, it is still unproven. Please use caution,
and use at your own risk.

Several feature of the bullet library standard have not yet
been implemented (rotation, shearing) as they are not required
for proper font operation in many circumstances. Programs
expecting to use advanced (or undocumented) features of the
bullet.library may not work as expected with ttf.library.
```

1.22 Recent Change History

```
Changes in v0.7.5

  eliminate potential crash when a font file could not be accessed
  kerning support enabled
     note that only one level of kerning is supported, and bullet
     api calls specifying either 'Design' or 'Text' kerning will
     result in the same kerning value.
  corrected width list calculation (apparently broken since 0.7)
  ttfmanager adds some keyboard support, especially in preview

Changes in v0.7.3

  should now support infamous win symbol encoding
  added 'stretch' option for larger glyphs at small sizes
  applied the most current freetype patches


Changes in v0.7.2

  plugged a nasty memory leak in the library
  font specific codepage support enabled


Changes in v0.7.1

  ttfmanager - several cosmetic changes
  ttflist - now reports some kerning table info
  ttf.library
   - fixed a problem with baseline shift on some characters
     at smaller sizes (especially with diacritical marks)
   - some changes in advance width calculations


Changes in v0.7

  Reinstall Warning - some important things have changed in
  the font installation, and all fonts installed with any
  previous version must be re-installed. (Sorry, but this
  should be the last time.)

  ttfmanager

   - preview, font information and options windows added
   - several changes in the .otag files produced by installing,
     adding a complete set of font attribute mappings, fixing
     the space width calculation.

  ttf.library

   - More speed! Typically 1/3 faster on a 040/25MHz.

   - The library now handles internally the correction between
     Amiga font size and actual em size, instead of relying on
     diskfont.library to do the job. This opens the way for
     several future enhancements.
```

```
Changes in v0.6.3

  ttf.library now supports a system wide alternate codepage

  Added ttfcp, a codepage installation tool.

Changes in v0.6.2

  Only the ttf.library has changed in this version.

  The quest for perfection continues - the character advance
  calculations are virtually identical to that common M$ OS
  engine. An issue with non-glyph widths (i.e. the space
  character will be fixed in the next installer version.

  Source of mystery crashes (of other tasks) found and fixed.

Changes in v0.6

  Re-examined the character advance calculations to improve
  spacing at smaller sizes.

  Modified the font install routines to calculate the extremes
  of the ascent and descent by default. Problem fonts (baseline
  shifts or clipped glyphs) should be re-installed with the
  new ttfmanager or ttfinstall version.

  Added a fake version number to ttf.library for WordWorth.

Changes in v0.5

  Added ttfmanager and Install-ttflib.
  Re-examined linkages, often resulting in smaller file sizes.

Changes in v0.4

  WidthLists are now supported.

  Subtle changes to returns of non-existent glyphs to be more
  compatible with bullet, and make smaller (memory wise) fonts
  when called by diskfont.library.

  ttfinstall - correct advance width problem with monospaced fonts

Changes in v0.3

  Sets many more tags in otag file to make some programs much
  happier (FinalWriter, to name one.) As a result, ALL FONTS
  installed with v0.2 or earlier MUST BE RE-INSTALLED. Sorry,
  but the results are usually worth it.

  Mimic bullet.library in handling of glyphless codes, such as
  a space.

  Widths are calculated more appropriately, resulting in better
  display under many conditions.
```

Several other peculiar, but un-enumerated, bugs were squashed.

1.23 The future of ttf.library

Future

Still many improvements in the works for ttfmanager.

Add library support for kerning and transforms

Examining possibilities of using the forthcomming
FreeType 2 engine.

1.24 Frequently Asked Question

FAQ

Can we make bitmaps, please?

A very frequently requested enhancement is for bitmap font generation. It is comming. In the meantime, a font editor that uses diskfont.library to load a font will do the job. Also, Bitline by Georg Steger holds some promise.

How about an 060/PPC version

Once the development of the library has slowed down, support for additional processors is planned. As it is, I spend more time on packaging than development, and additional variations would add to that. A PPC creates some additional concerns, but it is being considered.

Why are the fonts so tiny? 8 point soandso.font looks
bad. On a PC I can use 7 point arial, but it is just
little squiggles on my amiga. Etc, etc.

A quick lesson of fonts and terms is in order. In more traditional typography, a point is a standard unit of measure, approximately 1/72 of an inch. A point size refers to the size of an imaginary box surrounding the letter 'M', know as the em square. The size of this box in 1/72 inch units is the point size. When Windoze defined its version of point size, it invented a new concept, the virtual display inch (I'm not kidding!) which is "approximately 30 to 40 percent larger" If you consider that the declared resolution of a typical Win display is 96DPI square, you'll find that the math works out to the em square point measurement is approximately the pixel height count of the letter M.

The Amiga however uses 'points' to specify the entire height of the font, not just the letter M. There is no provision, through diskfont.library for a glyph to fall outside of a 'point' high bitmap. As a consequence, a box which will hold the 'M' won't hold a 'Ç' or an 'È' or any other characters. The needed room must come from somewhere, and this means that the sizes must be scaled to fit. Since ttf.library is at the bottom of the chain, the apparently inflated point size must be specified. As a result, requests for 8 point glyphs will usually result in total garbage, as 3-4 'points' get used for any needed ascenders and descenders, leaving little to reflect the characters. 10 is sometimes recognizeable, and 12 starts to be useful for most fonts.

To match an Amiga point size exactly to those of another platform, examine the highest ascender, lowest decender and the letter 'M' on the other platform. The letter 'M' should be approximately the point size pixels tall. Add the additional pixel heights of the high and low extremes to that point height, and you should have a close value for a corresponding Amiga size.

What can I do to make my browser look like a PC?

You can match the font sizes used by default in MSIE or Mozilla on a Winbox. There will be differences in the layout by the browser, but the letters in the words could be made to match in many ways. The biggest problem is in finding the correct sizes to specify, approximate values of which are presented below. You will need to obtain the TimeNewRoman and CourierNew font families. I leave all legal, ethical and tactical aspects of this to the individual. (These are excellent fonts, and they are available without cost, but I have yet to find an Amiga only way to obtain them, and remain entirely within the EULA. Special euro versions are available, too)

Another issue is that the Winbox text display is kerned, with a variable advance width between specific characters when placed next to one another. Even when kerning support is enabled in ttf.library (soon) the system graphic text routines will not use that information. Sorry.

Approximate matching fonts/sizes are as follows:

```
    H1          TimesNewRomanBold/36
    H2          TimesNewRomanBold/28
    H3          TimesNewRomanBold/20
    H4          TimesNewRomanBold/18
    H5          TimesNewRomanBold/14
    H6          TimesNewRomanBold/12

    FONT SIZE=1 TimesNewRoman/12
    FONT SIZE=2 TimesNewRoman/15
```

```
      FONT SIZE=3 TimesNewRoman/18
      FONT SIZE=4 TimesNewRoman/20
      FONT SIZE=5 TimesNewRoman/27
      FONT SIZE=6 TimesNewRoman/35
      FONT SIZE=7 TimesNewRoman/53

      Preformated CourierNew/15
```

Another potentially useful browser trick is to set a codepage to eliminate the dreaded no such character box that appears often where quote marks and apostrophies should appear. Although I think that any web author that allows such rot to appear on a page should at least be publicly chastised, the fact remains that these artifacts are widespread. It seems that M$ 'improved' (in its opinion) ISO 8859-1 to create the 1252 codepage. (See ftp.unicode.org. for a mapping table.) It seems that range from 0x80 to 0x9F, which is non-printing in 8859-1 (a REAL standard) is heavily and arbitrarily populated in CP1252, adding left and right single and double qoutes among other things.

What is this strange version number on ttf.library?

WordWorth wants to open Version 2, and ttf.library is not even to version 1.0 yet. This is a bug in WordWorth. It has no reason to specify a library version at all when opening the library. This is likely a holdover from a very early bullet.library, but it is still WRONG! For the time being, I have faked the version in the library header to pass this test. Version 0.7 appears as version 10.7 to the system. Any future versions will continue to be offset by 10, as long as required.