

T1ASCII

NAME

`t1ascii` – convert PostScript Type 1 font from binary to ASCII

SYNOPSIS

t1ascii [*-l length*] [*input* [*output*]]

DESCRIPTION

t1ascii converts Adobe Type 1 font programs in PFB (binary) format to PFA (hexadecimal) format. If the file *output* is not specified output goes to the standard output. If the file *input* is not specified input comes from the standard input.

OPTIONS

--line-length=num, -l num

Set the maximum length of encrypted lines in the output to *num*. (These are the lines consisting wholly of hexadecimal digits.) The default is 64.

--warnings, -w

Warn when the input font contains lines longer than 255 characters. Long lines don't strictly conform to Adobe's Document Structuring Conventions, and may cause problems with older software.

SEE ALSO

t1binary(1), t1unmac(1), t1mac(1), t1asm(1), t1disasm(1)

Adobe Type 1 Font Format

AUTHORS

Lee Hetherington (ilh@lcs.mit.edu)

Eddie Kohler (ekohler@gmail.com)

Ported to Microsoft C/C++ Compiler and MS-DOS operating system by Kai-Uwe Herbing (herbing@netmbx.netmbx.de).

T1ASM

NAME

t1asm – assemble PostScript Type 1 font

SYNOPSIS

t1asm [-a|-b] [-l *length*] [*input* [*output*]]

DESCRIPTION

t1asm assembles Adobe Type 1 font programs into either PFA (hexadecimal) or PFB (binary) formats from a human-readable form. If the file *output* is not specified output goes to the standard output. If the file *input* is not specified input comes from the standard input.

t1asm tokenizes the charstring data and performs eexec and charstring encryption as specified in the “black book,” *Adobe Type 1 Font Format*.

The input must have a line of the form

/-l{string currentfile exch readstring pop}executeonly def

which defines the command, in this case ‘-l’, that is to start charstring data. It is an error not to define such a command. Another common name for this command is ‘RD’.

After the start of the **Subrs** array in the input, all open braces ‘{’ not in a comment begin a charstring. Such a charstring is terminated by the next non-comment close brace ‘}’. Within such a charstring, only comments, integers, and valid charstring commands are allowed. Valid charstring command names can be found in *Adobe Type 1 Font Format* and other documents describing the newer Type 2 opcodes. The format within a charstring is unimportant as long as integers and commands are separated by at least a one whitespace (space, tab, newline) character. Note that within charstrings, comments are discarded because they cannot be encoded.

OPTIONS

--pfa, -a

Output in PFA (ASCII) format.

--pfb, -b

Output in PFB (binary) format. This is the default.

--block-length=num, -l num

PFB only: Set the maximum output block length to *num*. The default length is as large as memory allows.

--line-length=num, -l num

PFA only: Set the maximum length of encrypted lines in the output to *num*. (These are the lines consisting wholly of hexadecimal digits.) The default is 64.

EXAMPLES

```
% t1asm Utopia-Regular.raw > Utopia-Regular.pfb
```

```
% t1asm -a Utopia-Regular.raw > Utopia-Regular.pfa
```

SEE ALSO

t1disasm(1), t1ascii(1), t1binary(1), t1unmac(1), t1mac(1)

Adobe Type 1 Font Format is available free from Adobe as a PDF file. http://partners.adobe.com/asn/developer/PDFS/TN/T1_SPEC.PDF

The Type 2 Charstring Format, also available from Adobe as a PDF file, describes the newer Type 2 operators, which are also used in some multiple-master Type 1 fonts like Adobe Jenson and Kepler. <http://partners.adobe.com/asn/developer/PDFS/TN/5177.Type2.pdf>

AUTHORS

Lee Hetherington (ilh@lcs.mit.edu)

Eddie Kohler (ekohler@gmail.com)

T1BINARY

NAME

t1binary – convert PostScript Type 1 font from ASCII to binary

SYNOPSIS

t1binary [**-l** *length*] [*input* [*output*]]

DESCRIPTION

t1binary converts Adobe Type 1 font programs in PFA (hexadecimal) format to PFB (binary) format. If the file *output* is not specified output goes to the standard output. If the file *input* is not specified input comes from the standard input.

OPTIONS

--block-length=*length*, **-l** *length*

Set the maximum length of sections in PFB output to *length*. The default length is as large as memory allows.

SEE ALSO

t1ascii(1), **t1unmac(1)**, **t1mac(1)**, **t1disasm(1)**, **t1asm(1)**

Adobe Type 1 Font Format

AUTHORS

Lee Hetherington (ilh@lcs.mit.edu)

Eddie Kohler (ekohler@gmail.com)

Ported to Microsoft C/C++ Compiler and MS-DOS operating system by Kai-Uwe Herbing (herbing@netmbx.netmbx.de).

T1DISASM

NAME

t1disasm – disassemble PostScript Type 1 font

SYNOPSIS

t1disasm [*input* [*output*]]

DESCRIPTION

t1disasm disassembles Adobe Type 1 font programs in either PFA (hexadecimal) or PFB (binary) formats into human-readable form. If the file *output* is not specified output goes to the standard output. If the file *input* is not specified input comes from the standard input.

t1disasm performs eexec and charstring decryption as specified in the “black book”, *Adobe Type 1 Font Format*. Additionally, the charstring binary tokens are expanded into human-readable text form, using the names given in the black book and later documents describing Type 2 opcodes.

EXAMPLES

```
% t1disasm Utopia-Regular.pfb Utopia-Regular.raw
% t1disasm Utopia-Regular.pfa Utopia-Regular.raw
```

In **Subrs** entries in Utopia-Regular.raw will look like

```
dup 5 {
  8 111 vstem
  -12 128 hstem
  707 -20 hstem
  return
} |
```

and the **CharStrings** entries like

```
/exclam {
  58 242 hsbw
  6 callsubr
  5 4 callsubr
  63 707 rmoveto
  -54 0 -5 -22 4 -45 rrcurveto
  40 -431 rlineto
  29 hlineto
  42 431 rlineto
  4 45 -5 22 -55 0 rrcurveto
  closepath
  6 4 callsubr
  -719 vmoveto
  243 callsubr
  endchar
} |-
```

SEE ALSO

t1asm(1), **t1ascii(1)**, **t1binary(1)**, **t1unmac(1)**, **t1mac(1)**

Adobe Type 1 Font Format is available free from Adobe as a PDF file: http://partners.adobe.com/asn/developer/PDFS/TN/T1_SPEC.PDF

The Type 2 Charstring Format, also available from Adobe as a PDF file, describes the newer Type 2 operators, which are also used in some multiple-master Type 1 fonts like Adobe Jenson and Kepler: <http://partners.adobe.com/asn/developer/PDFS/TN/5177.Type2.pdf>

AUTHORS

Lee Hetherington (ilh@lcs.mit.edu)
Eddie Kohler (ekohler@gmail.com)

TIMAC

NAME

t1mac – translate a PFA or PFB PostScript Type 1 font into Macintosh format

SYNOPSIS

t1mac [**--macbinary** | **--applesingle** | **--appledouble** | **--binhex** | **--raw**] [**--filename** *name*]
[*input* [*output*]]

DESCRIPTION

t1mac reads a PFA (hexadecimal) or PFB (binary) PostScript Type 1 font file and generates an equivalent Macintosh Type 1 font file. The output file can be in MacBinary II, AppleSingle, AppleDouble, or BinHex format, or it can be a raw resource fork. The default is MacBinary II; use an option to choose a different format. If the *output* file is not specified output goes to the standard output.

WARNING: The output of **t1mac** is not sufficient to use the font, since Macintoshes can't read raw Type 1 fonts. You will need to create a font suitcase containing bitmap fonts if you do not have such a suitcase for the font already. **t1mac** cannot help you do this.

OPTIONS

--raw, -r

Indicates that output should be a raw resource fork.

--macbinary

Indicates that output should be in MacBinary I or II format. This is the default.

--applesingle

Indicates that output should be in AppleSingle format.

--appledouble

Indicates that output should be in AppleDouble format.

--binhex

Indicates that output should be in BinHex 4.0 format.

--filename=*name*, -n *name*

Sets the Macintosh filename of the output font to *name*. The default is to construct the filename from the font's name using established Macintosh conventions. This option is not useful when output is a raw resource fork.

SEE ALSO

t1unmac(1), **t1ascii(1)**, **t1binary(1)**, **t1asm(1)**, **t1disasm(1)**

AUTHORS

Eddie Kohler (ekohler@gmail.com)

T1UNMAC

NAME

t1unmac – translate a Mac PostScript Type 1 font into PFA or PFB format

SYNOPSIS

t1unmac [-a|-b] [-r] [*input* [*output*]]

DESCRIPTION

t1unmac extracts POST resources from a Macintosh PostScript font file and creates a PFA (hexadecimal) or PFB (binary) font file. The file *input* should be in MacBinary I or II, AppleSingle, AppleDouble, or BinHex format, or it can be a raw resource fork. If the file is a raw resource fork, you need to give the ‘--raw’ option; otherwise **t1unmac** should automatically figure out what kind of file you have. If the file *output* is not specified output goes to the standard output.

OPTIONS

--pfa, -a

Output in PFA (ASCII) format.

--pfb, -b

Output in PFB (binary) format. This is the default.

--raw, -r

Indicates that the input is a raw resource fork.

--macbinary

Indicates that the input is in MacBinary I or II format.

--applesingle

Indicates that the input is in AppleSingle format.

--appledouble

Indicates that the input is in AppleDouble format.

--binhex

Indicates that the input is in BinHex 4.0 format.

--block-length=num, -l num

PFB only: Set the maximum output block length to *num*. The default length is as large as memory allows.

--line-length=num, -l num

PFA only: Set the maximum length of encrypted lines in the output to *num*. (These are the lines consisting wholly of hexadecimal digits.) The default is 64.

EXAMPLES

On Mac OS X, you can use **t1unmac** to translate a font into PFA or PFB format as follows:

```
% t1unmac --raw FONTFILENAME/..namedfork/rsrc > OUTPUT
```

SEE ALSO

t1mac(1), **t1ascii(1)**, **t1binary(1)**, **t1asm(1)**, **t1disasm(1)**

AUTHORS

Lee Hetherington (ilh@lcs.mit.edu)

Eddie Kohler (ekohler@gmail.com)

Ported to Microsoft C/C++ Compiler and MS-DOS operating system by Kai-Uwe Herbing (herbing@netmbx.netmbx.de).