type light 3

user manual

version 3.0

CR8 Software Solutions
www.cr8software.net

Table of Contents

Introduction

Type light 3.0 is a freeware, but fully functional, OpenType font editor/typeface designer. With Type light 3.0 you can create, edit or convert OpenType TrueType (.ttf) and OpenType PostScript (.otf) fonts. Type light 3.0 supports a basic toolbox of drawing tools and can open fonts containing up to 65535 glyphs (characters)

Type 3.0 full version has additional features, including: extra drawing tools (knife, freehand draw, glue, transform), ability to edit all OpenType names and parameters, kerning, bitmap tracing, automatic composite creation, OpenType features and automated Action scripts.

Type 3.0 full version is avaiable here: **www.cr8software.net/type.html**

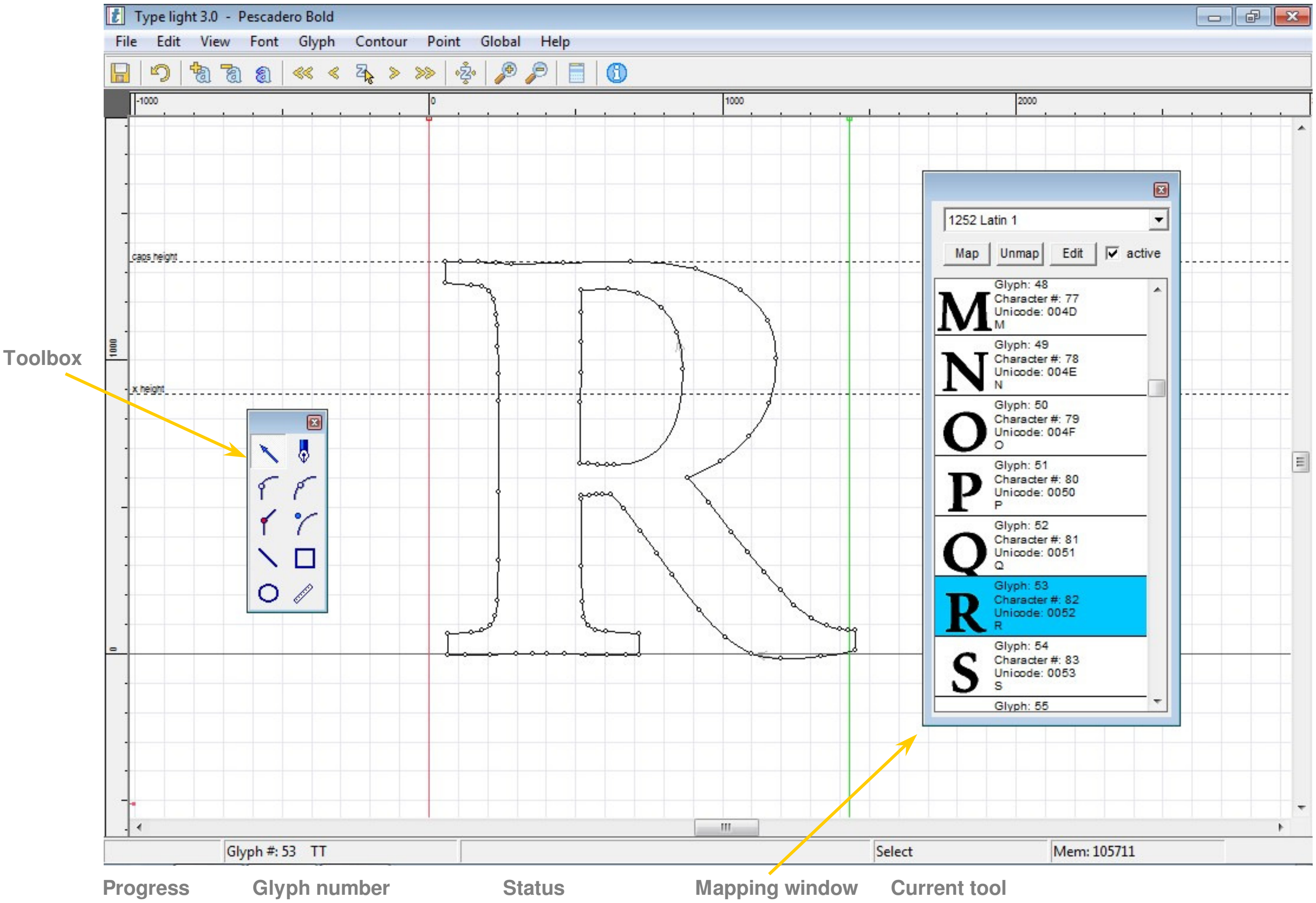
Allan Murray, June 2011

CR8 Software Solutions
www.cr8software.net

Support

For support email: **allanmurray@cr8software.net**
or visit the discussion forum: **http://cr8.proboards.com**

Navigation


Type light 3.0 - Pescadero Bold
File
Edit
View
Font
Glyph
Contour
Point
Global
Help
-1000
0
1000
2000
caps height
x height
1000
1252 Latin 1
Map
Unmap
Edit
☑ active
Glyph: 48 Character #: 77 Unicode: 004D M
Glyph: 49 Character #: 78 Unicode: 004E N
Glyph: 50 Character #: 79 Unicode: 004F O
Glyph: 51 Character #: 80 Unicode: 0050 P
Glyph: 52 Character #: 81 Unicode: 0051 Q
Glyph: 53 Character #: 82 Unicode: 0052 R
Glyph: 54 Character #: 83 Unicode: 0053 S
Glyph: 55
Glyph #: 53 TT
Select
Mem: 105711
Toolbox
Progress
Glyph number
Status
Mapping window
Current tool

To navigate around the edit window (main display area), use the **scroll bars**, the **cursor keys,** or **right click** on a clear area to grab and drag the window.

To zoom in and out, use the **magnify** and **reduce buttons**, the **+ key** and **- key**, or the mouse scroll wheel.

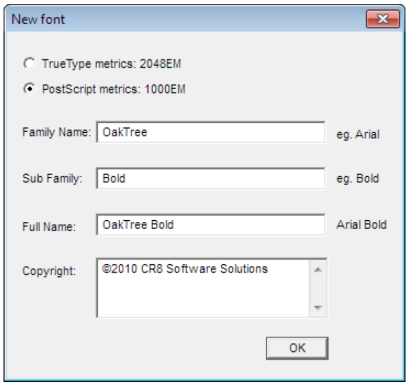
The red (p1) and green (p2) vertical lines mark the width of the glyph (see Appendix II). Move these by dragging at the top of each line, or select **glyph metrics** from the **glyph menu**.

The dashed horizontal lines (optional - use **preferences** from the **view menu**) are: **ascent**, **descent**, **caps-height** and **x-height** (see Appendix II). Anything that is not between the ascent and descent line will be 'clipped' (do not confuse with **ascender** and **descender)**. The values of these parameters can be set using **metrics** from the **font menu**.

Creating a new font

Select **new** from the **file menu**. You will be prompted for the names of your new font. All fields are required. These are:

Family Name:	eg Arial
Sub Family:	eg Regular or Bold etc.
Full Name:	= Family Name + Sub Family (eg Arial Bold). If it's a Regular font, then just put the Family name (eg Arial)
Copyright:	Your copyright details.



New font

○ TrueType metrics: 2048EM
● PostScript metrics: 1000EM

Family Name: OakTree eg. Arial
Sub Family: Bold eg. Bold
Full Name: OakTree Bold Arial Bold
Copyright: ©2010 CR8 Software Solutions

The first four glyphs will be automatically created and mapped (see Appendix II for more details). The first glyph always represents the **undefined** character – normally an empty box-like symbol. You can edit the symbol, but you cannot map this glyph to anything (mapping to the undefined character is equivalent to unmapping).

The next two glyphs are mapped to certain **control characters,** and the last is mapped to the **space** character. This glyph must be left blank, but you can change its width to match the spacing of your font.

Press the **new glyph button** to start creating a new font.

or

Double click on one of the characters in the **mapping window** and say **yes** to the prompt – a new glyph mapped to that character will be created.

You may now proceed to create glyphs using the range of tools available in the **toolbox.**

Common Issues

The following are a list of tips to help you avoid some common issues that can occur when creating or editing fonts.

- Do not open font files for editing from the Windows font folder. Copy them to another folder before opening them.
- Don't save font files directly into the Windows fonts folder (they will not be installed correctly).
- The correct way to install fonts is to drag the font file (save it somewhere else first) into the Windows font folder.
- If you are editing an installed font, then you will need to uninstall the original or give the new font a different font **family name** before you can install the new font.
- Make frequent backups (always good practice) of your font files, preferably as .gfs files, during the font development process.
- When a font does not work as expected, it is often a **naming** (conflicting font names) or **encoding** (make sure codepage Latin 1252 is active) issue.
- You should only have up to four different fonts that have the same font **family name** –regular, bold, italic and bold italic.
- If you change the name of a font, make sure that you change the advanced names also. Some programs will use advanced names (eg unique font name) to distinguish fonts, so they need to be unique.
- If you change the name of a font, make sure that you change the names for Macintosh (Roman) (otherwise the font may not work if installed on an Apple Mac) and for other active languages.
- Avoid creating glyphs with overlapping contours. The glyph will display correctly on a TrueType (.ttf) font (but is not recommended) but will have a white space in the overlapping region on a PostScript (.otf) font.

Definitions

Glyph – Glyphs are the shapes and symbols that you design. They normally represent characters or components of characters. A font contains a list of glyphs indexed by a glyph ID. They can be in an arbitrary order, but creating them in character set order is normal practice, and has some advantages.

Character – Characters are the basic symbols that are used to represent a language. The letter A is a character in the Latin Alphabet for example.

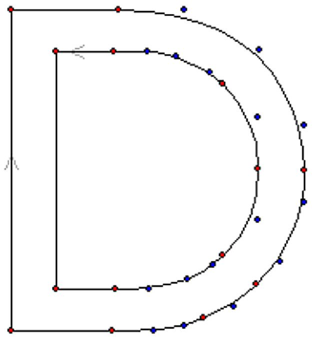
Mapping – Mapping is the process of connecting glyphs to characters. Glyph ID #36 may be assigned to represent the letter A character in a certain font for example. A single glyph can be mapped to more than one character, and (using OpenType features) several glyphs can also be used to represent the same character.

Unicode characters – Unicode is an international character encoding system that assigns a code to every character for most of the worlds language systems. A **Unicode script** is a range containing related characters. For example the Basic Latin script is the unicode range 0000 to 007F.

Character set – A Character set is the group of characters used to represent a particular language. Single byte character sets (**Code pages**) can contain up to 256 characters. The characters can represent different languages depending on the encoding used. Fonts can contain more than one Code page, which can be selected from the font selection menu on most programs. (Western = Latin 1252) .

TrueType points view

Select **TT points** from the **view menu**. This is the native TrueType format and can only be used with a glyph made of TrueType curves (see Appendix III). A TrueType font stores glyphs as a series of contours made of points – either on-curve (red) or off-curve (blue).



The gray arrows on each contour indicate the contour direction. The arrow is between point one and point two (see Appendix II regarding correct contour direction).

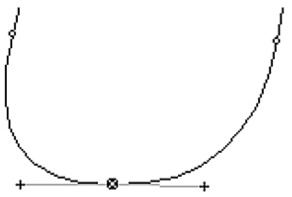
Point one must always be an on-curve point.

Although this is the correct representation of a TrueType glyph, it is not necessarily the easiest way to design glyphs, nor is it the most common method used in other design applications.

Nodes view

Select **nodes** from the **view menu**. Another (better) way to represent a glyph is using a series of nodes (black circles) – always on the curve, and with control points (crosses). The line between the control point and the node (gray) is always at a tangent to the curve. Simply move the position of the control points to change the shape of the curve.*

- Straight lines will have no control points.
- If the two control points and the node are all in a line, then the curve at the node is called **smooth**.
- If the curve is **smooth** <u>and</u> the control points are equal distance from the node, then the curve at the node is called **symmetric**.
- If the control points are **linked**, then moving one, will also move the other so that the curve remains smooth. If the control points are **free**, then they may be moved independently.
- Symmetric curves are always **linked**.
- Use the point menu (or right click) to change the type of curve: **smooth(free)**, **smooth(linked)**, or **symmetric(linked)**.



Symmetric

---

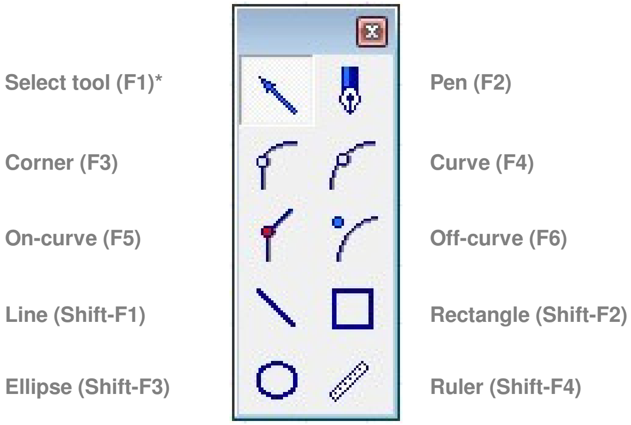
* Other font editors only allow this type of editing with PostScript curves – Type light 3.0 also allows you to edit TrueType curves in this manner, but limits are imposed to restrict the curve to a TrueType curve.

The Toolbox

The **toolbox** contains the tools that you will need for creating and drawing glyphs.

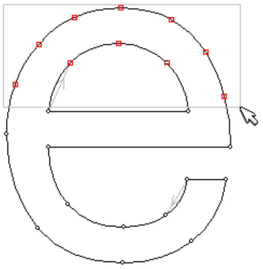
Using different tools, you can move points and contours, draw and manipulate lines and curves, create shapes, and measure distances.

*The keyboard shortcut keys for the Toolbox are the function keys F1-F6.



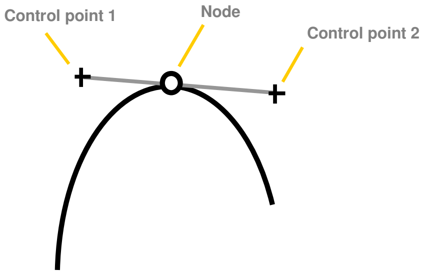
The Select Tool

You can use the **select tool** to highlight and move points around. By dragging a rectangle over the points or nodes, you can select multiple points. Pressing **Shift** at the same time allows you to add more points to your selection. By pressing **Ctrl** at the same time, you can select an entire contour. This method of selection allows you to move multiple points (move one of the red squares to move all of them), perform transformations (see below) and contour operations (using the **contour menu**).



Selecting a point by clicking on it allows you to move a single point, and gives you access to the options on the **points menu** – either from the main menu or by clicking the right mouse button.

If you are in **nodes view mode** then when you select a node, the control points belonging to that node will be displayed. The control points (crosses) dictate the degree of (and are at a tangent to) the curve.



If you are editing a glyph with TrueType curves, moving a conrol point may also affect neighboring nodes. PostScript curves do not have this limitation.

The Pen Tool

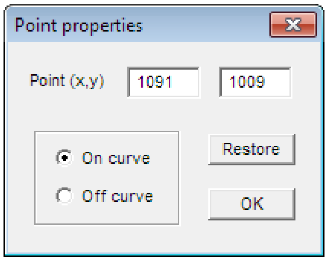
The **pen tool** will draw smooth curves. Left mouse click to add a node, and move the mouse before you release the button to set the control points. The curves will be symmetric about the node.

If you are editing a glyph with TrueType curves, then when you release the button the curve will convert to a TrueType curve (with additional nodes).

The node and points tools

These four tools work as follows:

- If no point or node is currently selected, a you will start a new contour
- If the last point or node on a contour is selected, then you will add to the current contour
- If you click on the first point or node of a contour, the contour will 'close'
- If the mouse pointer is over a curve or line (**[line]** is displayed in the status bar), the new point or node will appear in the middle of the contour.



Point properties

Point (x,y) 1091 1009

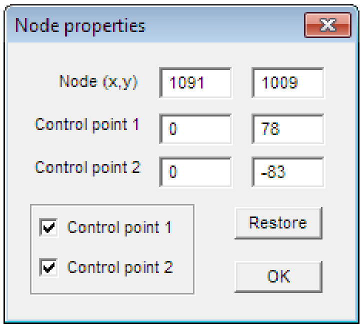
● On curve

○ Off curve

The **corner tool** and **curve tool** allow you to adjust the degree of curve. If you move the mouse before you release the button, you can change the position of the control points.

The **off-curve tool** functions only when editing a glyph with TrueType curves.

Points and nodes can be precisely positioned by using **properties** from the **points menu** (or right click, **proporties**).



Node properties

Node (x,y) 1091 1009

Control point 1 0 78

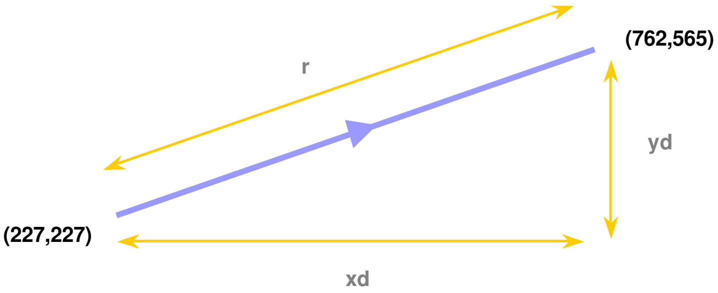
Control point 2 0 -83

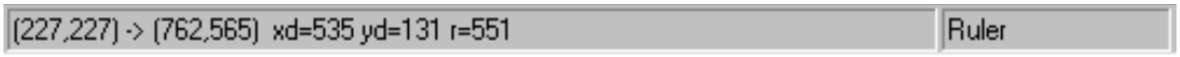
☑ Control point 1

☑ Control point 2

The ruler

The **ruler tool** can be used to measure distances. Click at the start position (or node) and drag to the end position (or node). Various measurements will appear in the status bar:





```
(227,227) -> (762,565)  xd=535 yd=131 r=551
```

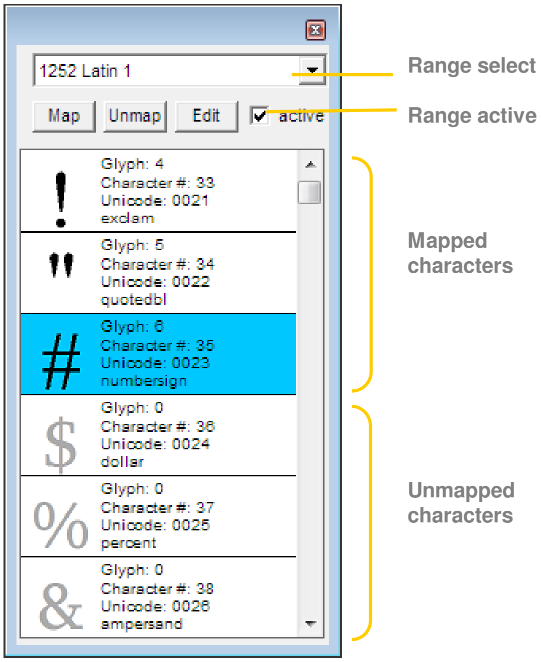
The Mapping Window

The **mapping window mode button** will sequence through the different modes available. You can also change the mode using **preferences** from the **view menu**:

1 - Code Pages (character sets)
2 - Unicode Scripts (a range of unicode values)
3 - Glyph List (all mapped and unmapped glyphs)

Modes 1 and 2 allow you to map glyphs to unicode characters (uni0000 to uniFFFF). If you used the **new glyph button** to create a glyph, then you will need to map it manually: Select a character (it will highlight blue), then press the **map button**. The glyph in the edit window will be mapped to the character, and it will show in black.

Use **range select** to select which characters are displayed, and the **range active box** to indicate which ranges are used in your font. For example, to make a Western character set font, select the **1252 Latin 1 page range**, and map glyphs to these characters. Then check the **range active box**.



Glyph: 4
Character #: 33
Unicode: 0021
exclam

Glyph: 5
Character #: 34
Unicode: 0022
quotedbl

Glyph: 6
Character #: 35
Unicode: 0023
numbersign

Glyph: 0
Character #: 36
Unicode: 0024
dollar

Glyph: 0
Character #: 37
Unicode: 0025
percent

Glyph: 0
Character #: 38
Unicode: 0026
ampersand

Font paremeters

An OpenType font contains a whole lot of information – names, font descriptions, parameters and metrics specific to your font. Type light 3.0 allows you to set and change most of these.

If you are new to font design, and are creating a basic font, then you can keep the default values and use the automatic settings. See Appendix II for a better understanding of various parameters.

Names

When you create a new font, you are prompted for certain name information. This information is specific to just one language – US English. Your font can contain name information for other languages too. Use the **names** from the **font menu** to add more names.

Select the language at the bottom of the window, and then check the box to the left to make this language active. The default names will be the same as the US English ones, so uncheck the boxes to change the name entries if required.

The **advanced button** allows you to enter various other names. The most important ones are: **Unique ID [3]** and **PostScript name [6]**. If you change the name of your font then you must change these also – or make them blank. If these fields are invalid or blank, then Type light 3.0 will generate valid names when you save the font.

For an ordinary Latin font, name information is normally only entered for **US English** and **Macintosh Roman** languages (these two must always be present).

Font metrics

Use **metrics** from the **font menu** to set your font's metrics. These are parameters describing certain measurements specific to your font. Some of these are self explanatory like **underline thickness** and **italic angle**. The most important ones to understand are:

EM Unit size:	Usually set to 2048 (TrueType or OpenType TT), or 1000 (OpenType PS). This is used to calculate the point size when the font is displayed.
WinAscent:	top (anything above this may be clipped)
WinDescent:	bottom (anything below this may be clipped)
Caps height:	uppercase character height
'x' height:	lowercase character height (height of a lowercase 'x')

If you have checked the option **show horizontal markers** (using **parameters** from the **view menu**) then the last four parameters above will be visible in the main edit window as horizontal lines.

Font description

Use **description** from the **font menu** to set the parameters that describe the font: weight, width, version number, italic, bold and mono-spaced. If you set the font to **Mono-spaced** then the advance width of all glyphs (except glyph number 1 - which must have an advanced width of zero), will be constant. Advanced description parameters can only be altered using Type 3.0 full version.

Composite Glyphs

A composite glyph has no contours or points of its own, but is made up of other glyphs. Creating a composite glyph is as simple as copying the component glyphs into a blank glyph. The composite is displayed as a blue outline, and although you can move and transform the components (as if you were manipulating contours) you cannot move individual points unless you first **decompose** the glyph. The **decompose button** is sixth from the left on the toolbar.

Glyphs can be stored as composites in a **TrueType font (.ttf)**, but when saving an **OpenType PostScript font (.otf)** the glyphs will be not stored as composites – when you open the font again, they will be decomposed. (Save a copy as a **.gfs file** if you want to alter the glyphs as composites at a later time).

Copy and paste functions

Contours and whole glyphs can be copied and pasted between glyphs:

- Use **copy glyph** from the **edit menu** to copy the current glyph to the clipboard.
- Use right click **copy glyph** from the **mapping window** (glyph list mode only) to copy the selected glyph to the clipboard.
- Use **copy** from the **edit menu** to copy only the selected points to the clipboard.
- When the destination glyph is in the Glyph Window, use **paste** from the **edit menu**. When copying glyphs into a blank glyph, a composite glyph will be created. Press the decompose button if you don't want this.
- The **paste metrics** option from the **edit menu** pastes only the Left Side Bearing and Advance Width from the clipboard.

Hinting

When a font is displayed at a small point size on a low resolution device (eg a screen), several distortions can occur. This happens because there are not enough pixels to accurately represent the rasterized font. Adjustments (called hints) can improve the appearance of the font at low resolutions.

Hints can either be global (apply to all glyphs within the font) or can apply to individual glyphs.

You can use Type light 3.0 to specify **gasp hinting** only. Type 3.0 full version also supports **global hinting** of **PostScript** fonts, which can be auto-hinted using Adobe's free font development kit.

Gasp (grayscale) hinting

Gasp hinting allows you to specify the rasterization technique for a TrueType font when it is rendered on grayscale-capable devices at different sizes.

A typical Gasp table could be:

Font size^	**Rasterization technique**
ppem<=8	grayscale only *
9<=ppem<=19	gridfit only **
20<=ppem	gridfit and grayscale

Select **gasp hinting** from the **font** menu. Enter 8 in the **max ppem** box, select grayscale in the **smoothing selection**, then press **add**. Enter 19 in the **max ppem** box, select gridfit in the **smoothing selection**, then press **add**. Enter 65535 in the **max ppem** box, select grayscale & gridfit in the **smoothing selection**, then press **add** (always enter 65535 as the max ppem for the last entry).

^ ppem = pixels per em. Em for a 72 point font is 1 inch (1pt = 1/72th inch).
So a 72pt font rendered on a 96 dpi (dots per inch) monitor is 96 ppem. (So it follows that 8 ppem would be 6pt on a 96 dpi monitor).

* Note for your computer must also have font smoothing activated for grayscale rendering.

** Gridfit means use TrueType hinting instructions (they must be present in the font).

Technical Information – font options

OpenType Tables

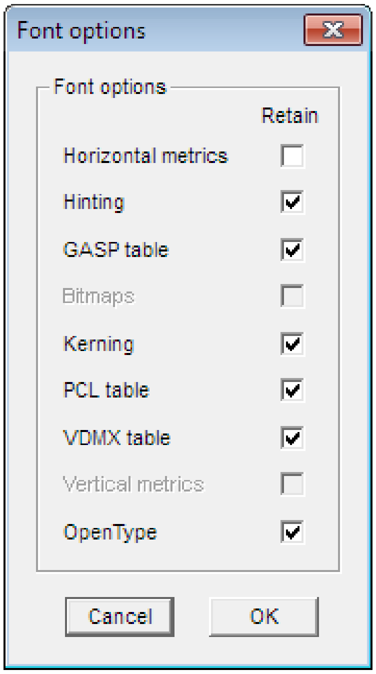
Type light 3.0 creates TrueType and OpenType TT fonts (.ttf) with these tables:

OS/2 **cmap** **glyf** **head**
hhea **hmtx** **loca** **map**
name **post**

OpenType PS fonts (.otf) will have the above tables, but the **CFF** table replaces the **glyf** table.

When editing a font, the following tables can be included if present in the original file, or, in the case of the Gasp table, if it has been created by the user:

prep **cvt** **LTSH** **fpgm**	If 'Hinting' retained^.
EBDT **EBLC** **EBSC**	If 'Bitmaps' retained.
vhea **vmtx**	If 'Vertical metrics' retained.
hdmx* **kern** **gasp** **PCLT**** **VDMX**	Select individually.
BASE **GDEF** **GPOS** **GSUB** **JSTF**	If 'OpenType' retained***.



Font options

Font options

	Retain
Horizontal metrics	☐
Hinting	☑
GASP table	☑
Bitmaps	☐
Kerning	☑
PCL table	☑
VDMX table	☑
Vertical metrics	☐
OpenType	☑

^ These tables for TT fonts only. For PS fonts the hinting option will control glyph level hinting only.

* If you have edited any glyphs, then the **hdmx** (horizontal metrics) table may no longer be accurate – it is then recommended that you do not retain it. **VDMX** and **hdmx** tables can be re-created using a free tool from Microsoft called **CacheTT**. (**www.microsoft.com/typography/tools/tools.aspx**)

** If youhave changed the fonts description, then the **PCLT** table may no longer be accurate - it is then recommended that you do not retain it. The **PCLT** table is strongly discouraged for use with OpenType fonts.

*** Fonts saved as OpenType will also be given an empty **DSIG** table, This is to give the font the OpenType icon on Windows XP. The original digital signature (if any) will not be retained, as it is no longer valid for an edited font.

Font Basics (TrueType)

A bit of history

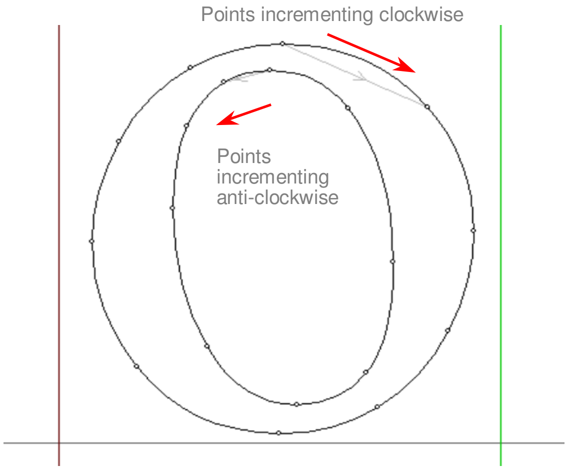
TrueType was originally developed by Apple in the late 80's, after Apple & Microsoft rejected a proposal by Adobe to use Adobe Type 1 (PostScript) fonts for their operating systems. Through a deal with Apple, TrueType was adopted by Microsoft in 1992, for their Windows 3.1 operating system.

Adobe joined forces with Microsoft in 1996 to combine their technologies and produce OpenType, which supports both TrueType and PostScript formats. Adobe finished converting its entire font library to OpenType (PostScript) fonts in 2002 with the intention that Adobe Type 1 fonts (see Appendix III) eventually be phased out.

Glyphs (TrueType)

The outlines of a TrueType glyph are defined by contours. Contours are defined by points.

Points of a contour are either on-curve (defining straight lines) or off-curve (defining a type of curve called a quadratic spline). The first point on a contour (point zero) must always be an on-curve point. See Appendix III for more details.


Points incrementing clockwise
Points incrementing anti-clockwise

The filled-in area of a TrueType glyph is always on the right-hand side of the contour. So, for the letter 'o' here, the points forming the outside contour will increment clockwise, and the points forming the inner contour will increment anti-clockwise.

(Note that technically PostScript curves should be in the opposite direction - with the outside contour counterclockwise.)

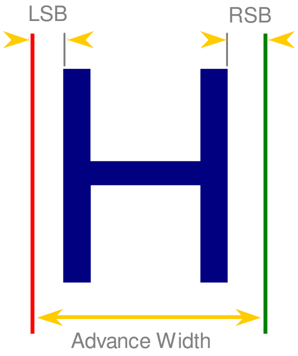
The extremes of a glyph should be defined with on-curve points.

Glyph metrics

Certain metrics define the horizontal dimensions of a glyph. These can be set using **glyph metrics** from the **glyph menu**, or by dragging the top of the red and green vertical markers.

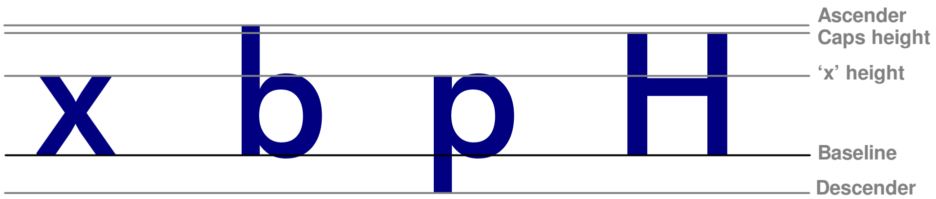
The Advance Width (AW), Left Side Bearing (LSB) and Right Side Bearing (RSB) determine the spacing between characters.

AW = LSB + character width + RSB



Font Metrics

Certain metrics define the vertical dimensions of characters in an OpenType font:



ASCENDER:	Height of ascender (usually height of 'b')
DESCENDER:	Height of descender (usually depth of 'p')
CAPS HEIGHT:	Height of uppercase characters (defn: height of 'H')
'x' HEIGHT:	Height of lowercase characters (defn: height of 'x')
BASELINE:	At y=0

Do not confuse the ascender and descender with ascent and descent (called WinAscent and WinDescent in **metrics** from the **font menu**). Ascent and descent define the upper and lower limits of all glyphs – anything outside these limits will be clipped.

The **EM square** determines the size of the font when it is displayed. For example, when the font is displayed at 12 points, the EM square will be 12 points high (1 point = 1/72 inch).

The **EM square** was traditionally (from type-setting days) defined as the size of an uppercase 'M', but typically the **EM square** encompasses the ascenders and descenders with some extra (internal) leading as well.

The size of the **EM square** is usually set at 2048 units for a TrueType or OpenType TT font, and usually set at 1000 units for OpenType PS fonts.

Standard glyphs

It is standard for a OpenType font to contain mapping to the Macintosh Roman character set, even if the font is only to be used for Microsoft Windows. Also, to meet Apple specifications, the first four glyphs of a OpenType Font should be these:

GLYPH #0	Used for **undefined** characters - normally a box shape
GLYPH #1	Special glyph with no contours, and zero width
GLYPH #2	**CR** character - no contours, but with a defined width (mapped to 0009 and 000D)
GLYPH #3	The **space** character - no contours, but with a defined width (mapped to 0020)

Visit the typography links here for more information about OpenType and Unicode:
www.cr8software.net/links.html

PostScript, TrueType and OpenType

Curves and Outlines

Outlines (the curves that form characters) are stored as a series of points, using one of two methods to mathematically describe their shape. We can say that fonts either have **PostScript outlines** or **TrueType outlines**. This appendix briefly describes the difference between the two, and the implications for font creation and editing.

Types of outline fonts

Isabella.ttf

TrueType
extension: **.ttf**
TrueType outlines

Pilgrim.pfb

Type 1 (not supported by Type light 3.0)
extension: **.pfb** & **.pfm**
PostScript outlines.

arial.ttf

OpenType (TT)
extension: **.ttf**
TrueType outlines. Actually identical to TrueType fonts,
but may contain additional OpenType information.

Balthasar.otf

OpenType (PS)
extension: **.otf**
PostScript outlines. Essentially Type 1 fonts wrapped in a
TrueType file structure.

Abode's intention is that Type 1 fonts be eventually phased out and replaced by OpenType PS fonts. Adobe Type 1 fonts are not supported by Type light 3.0.

TrueType and PostScript outlines

Outlines in a font are described by a series of points. To describe a straight line, you only need to specify the co-ordinates of the two end points, but to specify a curve, you need some extra points in between. A mathematical equation called a bezier curve is used.

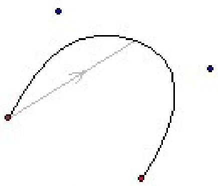


Fig 1a

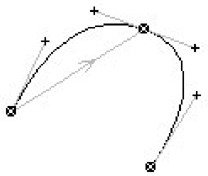


Fig 1b

Fig. 1a shows a curve described by two end points and two off-curve points. This is a quadratic bezier, or TrueType outline.

Fig. 1b shows the same curve with three nodes and their control points. This is a cubic bezier, or PostScript outline.

In fact, quadratic beziers are a subset of cubic beziers, so any TrueType curve can be converted exactly to a PostScript one (like in Figure 1). Conversion in the other direction is not so simple, and it may require several quadratic curves to approximate a particular cubic bezier.

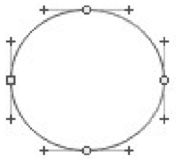


Fig 2a

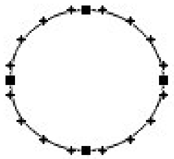


Fig 2b

Fig. 2a shows a circular curve represented by a PostScript outline.

Fig. 2b, the equivalent TrueType outline, requires many more points and is only a near approximation to the original curve.

So conversion of TrueType fonts to PostScript ones is an exact science (aside from other factors such as scale and hinting), but conversion of PostScript fonts to TrueType is only a close approximation. Most commercial TrueType fonts are designed as PostScript curves, and then converted to TrueType.

Because PostScript curves are a superset of TrueType, a greater range of curves can be drawn with fewer points, making design much easier. So the cubic bezier, with its nodes and control points, is the design industry's drawing standard.

Based on an article: www.cr8software.net/article004.html

Keyboard Shortcuts

CTRL + O	Open glyph data
CTRL + S	Save glyph data
ARROW KEYS	Move glyph in edit window
+ and -	Magnify and reduce glyph in edit window
CTRL + T	Toggle view mode
CTRL + Y	Toggle mapping view
CTRL + X	Copy whole glyph
CTRL + C	Copy selected points
CTRL + V	Paste
CTRL + B	Paste metrics only
CTRL + Z	Undo
CTRL + U	Glyph information
CTRL + A	Select all
CTRL + D	Select none
ESC	Deselect point
CTRL + N	Create a new glyph
CTRL + M	Clear current glyph
CTRL + P	Point/node properties
CTRL + E	Select contour
8	Make smooth (control points free)
9	Make smooth (control points linked)
0	Make symmetric (control points linked)
CTRL + R	Reverse contour
CTRL + DEL	Delete contour
DEL	Delete selected points
CTRL + F	Set size of grid
CTRL + G	Show-hide grid
F1 - F6	Select tools (1-6)
SHIFT +F1- F4	Select tools (7-10)

Copyright Notice

Type light 3.0 is Copyright ©2006-2011 Allan Murray, CR8 Software Solutions ("Software Publisher"). All rights reserved.

This End User License Agreement accompanies the Type light 3.0 ("Software") and applies to any upgrades, documentation, modified versions or updates of the Software.

1. USE OF THE SOFTWARE.

You may use the Software product for the production of fonts for personal or limited commercial use (you can sell fonts produced by the software if you are not a business, and selling fonts is not your primary income). The Software may not be used in any other commericial capacity (including the production of fonts for sale by a foundry or editing or altering of existing fonts for printing, design or other industrial or commercial use).

2. COPYRIGHT.

The Software is copyright of the Software Publisher. You may not copy, sell or redistribute the Software. This Agreement does not grant you any intellectual property rights in The Software.

3. RESTRICTIONS.

You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software. You may not alter or modify the installer program or create a new installer for the Software.

4. LIMITED WARRANTY

In no event will the Software Publisher be liable for indirect, special, incidental, tort, economic, cover or consequential damages arising out of the use of or inability to use the Software, including, without limitation, damages or costs relating to the loss of profits, business, goodwill, data or computer programs, even if advised of the possibility of such damages. In no case shall the Software Publisher be liable for money damages exceed the amount paid by you for the Software out of which such claim arose. The Software Publisher limits liability, according to the terms of this Agreement, to the extent permissible at law.

This End User License Agreement does not apply to certain third party fonts bundled with this package. These fonts, and their respective licenses, are contained in the /enc subdirectory of the install directory.

YOUR ACCEPTANCE OF THE FOREGOING AGREEMENT WAS INDICATED DURING INSTALLATION.

CR8 Software Solutions
www.cr8software.net

040611011