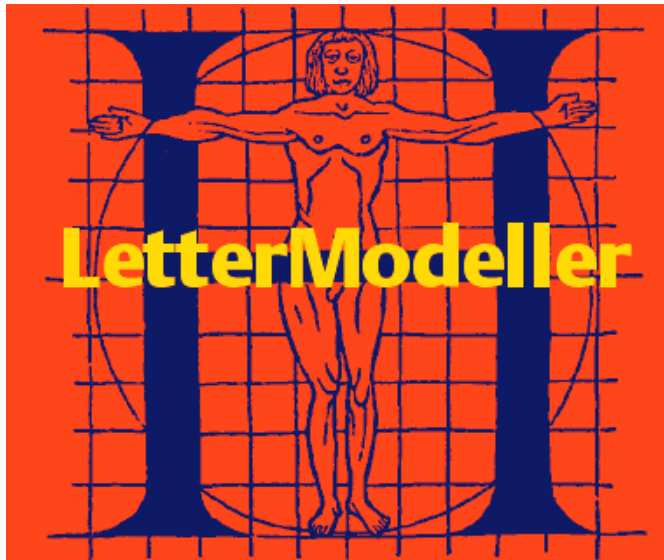


*Dutch Type Library*  
**FontMaster™**  
*Utilities*



This manual is under development still and is far from complete.  
It covers most of the editing functions, but for the generation of fonts  
the older DataMaster (part of the FM-suite) manual has been included.

Although the file structure used by LetterModeller is the same as in  
FontMaster, the interfaces of LetterModeller and DataMaster are not  
fully identical. Overall the information supplied by the DM manual  
should be sufficient though.

's-Hertogenbosch/Hamburg  
Autumn 2016



*Typography means more than bringing order to the passing on of information; it means elevating to the sublime the mould in which the process of passing on is cast.*

*Frank E. Blokland*

### **Limited user rights**

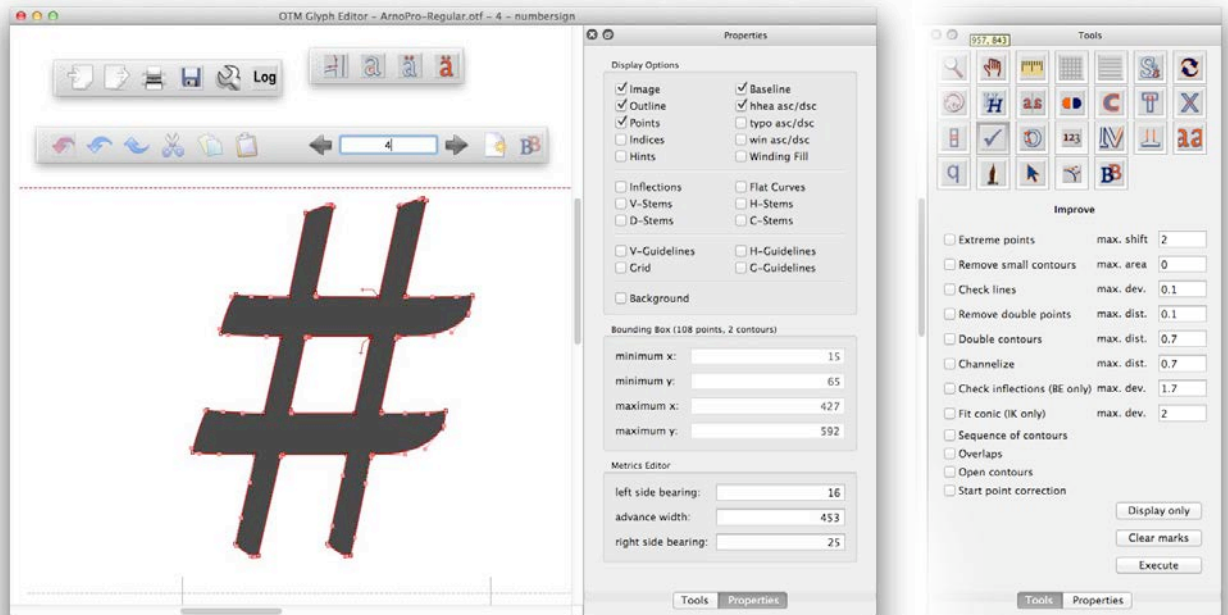
You may never use the software to edit data, including but not limited to fonts of which you do not own the rights, including but not limited to intellectual property rights, copyright and rights to trademark, unless the rightful claimant has given his written and signed consent.

The Glyph Editor	5
File	6
Import...	6
Export...	6
Print...	6
Save...	6
Character Preferences	6
Messages...	6
Clipboard...	6
Edit / Selection	6
Revert to Saved	6
Undo	7
Redo	7
Delete	7
Cut	7
Copy	7
Paste	7
Paste & Shift	7
Previous / Next	7
New	7
Select All / Deselect All	7
Select Points	7
Select Contours	7
Select Contour Groups	8
Select Character	8
Swap Background	8
View	8
Tool Bars	8
Dock Widgets	8
Display Options	8
Glyph Set	8
Reset	8
Tools	9
Zoom	9
Scroll by Hand	9
Measure	9
Grid	10
Guidelines	10
Scale	10
Rotate	11
Affine Transformation	11
Italization	11
Mirror and Fold	12
Hidden Lines	12
Contouring	13

<b>T-Disconnect</b>	13
<b>x-Disconnect</b>	13
<b>i-Disconnect</b>	14
<b>Improve</b>	14
<b>Sense of Rotation</b>	14
<b>Sequence of Points and Contours</b>	15
<b>Merge</b>	15
<b>Character Hinting</b>	16
<b>Review Changes</b>	16
<b>Quick Mode</b>	16
<b>Digitize</b>	17
<b>Shift</b>	17
<b>Shift Smooth</b>	17
<b>Background Glyph</b>	18
<b>Closing the Glyph Editor</b>	19

## The Glyph Editor

What if you find duplicate contours in a glyph of a font that needs to be shipped soon? Go back to the font editor of your choice, open the font's source data, correct the error, generate the font again, possibly repeat additional production steps with AFDKO or VOLT or at least perform some tricks to smuggle new outline data into the previous font version? The Glyph Editor helps.



The Glyph Editor's toolbars can be rearranged by simple drag & drop. They can be placed anywhere on the screen, like in the screenshot above. They can also be attached to any of the Glyph Editor window's left or right edges. The dock widget to the right shows either various display **Properties** or an overview of editing **Tools** and the options of the one currently selected.

*The Glyph Editor window consists of two areas: a main area for glyph editing and a dock widget presenting the tools.*

*There are three toolbars: **File** for importing, exporting and printing glyphs, the usual **Edit** functions including a textbox for selecting a glyph by glyph index, and various **Selection** modes. **View** modes and **Tools** are located in the dock widget.*

*Defaults can be defined in the ► Preferences.*

The Glyph Editor has three toolbars whose functions are reflected in menus too. Most of the functions can be accessed by way of shortcuts.

## FILE



This menu and toolbar allows to import, export and print glyphs:



### Import ...

Imports individual glyphs. Supported formats are:

- **EPS Encapsulated PostScript** [.eps]
- **SVG Scalable Vector Graphics** [.svg]



### Export ...

Exports individual glyphs. LetterModeller can

- export:
- **EPS Encapsulated PostScript** [.eps]
  - **SVG Scalable Vector Graphics** [.svg]
  - **Editable SVG Scalable Vector Graphics** [.svg]

The difference between the two SVG versions is that the former is an SVG font while the latter is an SVG illustration.



### Print ...

Prints the current glyph.

### Save ...

Save changes.



### Character Preferences

Opens the Preferences dialog e.g. changing colors of outlines, points and guidelines. Please see the ► *Preferences* chapter.

### Messages ...

Opens the Messages window which displays all status messages.

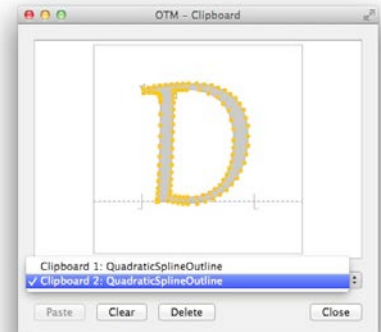
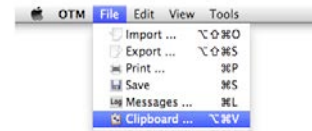
### Clipboard ...

Opens the Clipboard window. There can be multiple Clipboards now, for collecting outlines from the last couple of **Cut** or **Copy** actions.

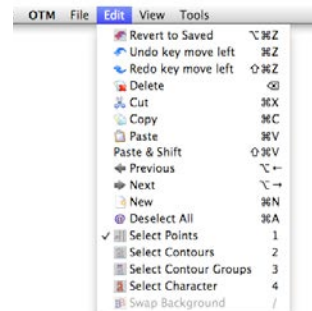
## EDIT & SELECTION



The edit toolbar holds the basic editing functions. The selection toolbar is for changing the selection mode. The edit menu holds both the editing functions and selection modes.



*The Clipboard window allows you to toggle between previously cut or copied outlines, view them, select and copy from them, **Paste** into them, **Clear** their content, or **Delete** any of these clipboards entirely.*



**Revert to Saved**

Un-does all changes since the font has been saved.

**Undo / Redo**

Un-does previous changes and re-does un-done changes, respectively.

**Delete**

Deletes the selected points or contours.

**Cut**

Cuts the selected points or contours. This will remove them from the glyph and keep them in the clipboard.

**Copy**

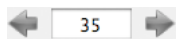
Copies selected points or contours into the clipboard without removing them from the glyph.

**Paste**

Pastes points or contours from the clipboard into the glyph, or pastes a table entry from the clipboard into the currently selected table.

**Paste & Shift**

Pastes points or contours into the glyph, but with a slight offset.

**Previous****Next**

Goes to the previous or next glyph. You may also enter the glyph index (GID) into the textbox – and do not forget to confirm with **RETURN**.

**New**

Adds a new glyph at the end of the glyph set.

**(De)Select All**

Selects all of the glyph's points or contours. If all of a glyph's points or contours are selected already, deselects all of them.

**Select Points**

Select points via mouse. Add points to (or remove them from) the selection by holding **SHIFT** and clicking on unselected (or selected) points.

**Select Contours**

Select individual contours by clicking inside of a contour. Add contours to (or remove them from) the selection by holding **SHIFT** and clicking on unselected (or selected) contours.



### Select Contour Groups

Select contour groups by clicking inside of a contour. Unlike the previous selection mode, Select Contour Groups will select not only the contour you have clicked on but also all contours inside of it. Add contour groups to (or remove them from) the selection by holding **SHIFT** and clicking on unselected (or selected) contours.



### Select Character

Select the entire character by clicking inside of any contour.

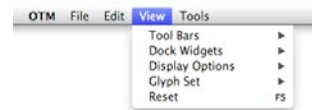
### Swap Background

Swaps foreground glyph and background glyph.

The following menus correspond to the **Properties** and **Tools** dock widgets rather than to toolbars.

### VIEW

The view menu serves to define various view options.



### Tool Bars

Check or uncheck this submenu's items to show or hide the **File**, **Edit** and **Selection** toolbars (which have been described above).

### Dock Widgets

Check or uncheck this submenu's items to define which of the two widgets, editing **Tools** and display **Properties**, are available in LetterModeller.

### Display Options

This menu's options essentially correspond to those in the **Properties** dock widget which is shown here.

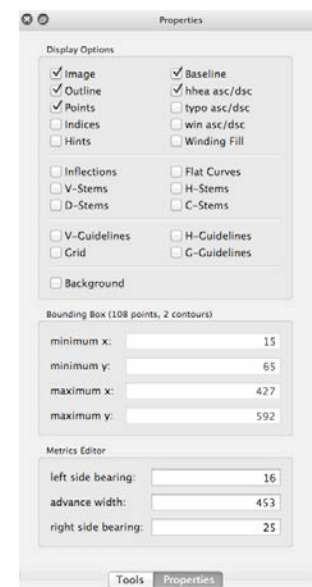
### Glyph Set

Here you may choose the glyph set which you would like to access in LetterModeller: **Glyph Index (GID)**, **Unicode 2+ semantics BMP only**, **Macintosh Roman**, or **Microsoft Unicode BMP only**.

### Reset

This will restore the default glyph size.

*Display Options as of the Properties dock widget.*

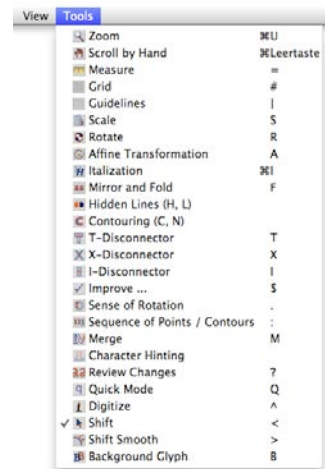
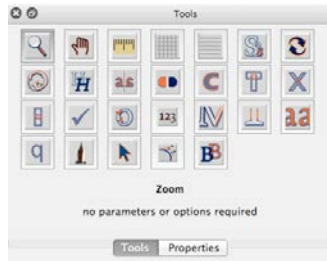


## TOOLS

And finally, the tools menu and dock widget offer functions for editing and designing glyphs.

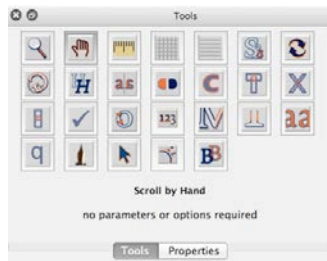
## Zoom

Click into the editing area to zoom in. Hold down the **SHIFT** key, then click into the editing area to zoom out. The position of the mouse will define the center of the new view. Alternatively, hold down the mouse button and draw a rectangle to zoom into this segment of the editing area.



## Scroll by Hand

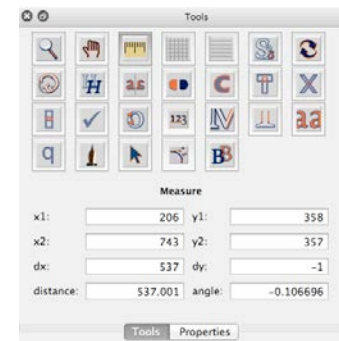
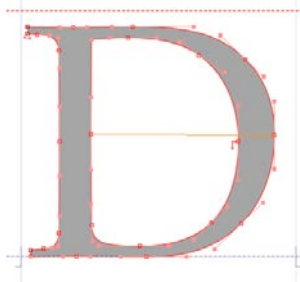
Click anywhere into the editing area, hold the mouse button down and move the mouse around to scroll the entire editing area.



## Measure

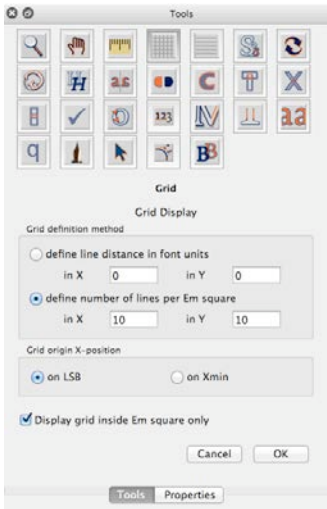
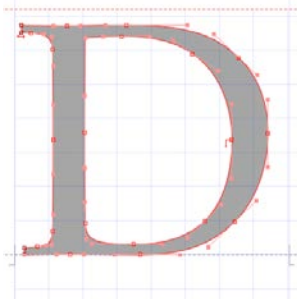
Click on a start point, move the mouse while holding down the mouse button, and release the button at the end point. The dialog will show information about the start point's **x1** and **y1**, the end point's **x2** and **y2**, the x- and y-distances between both points as **dx** and **dy**, the **distance** between both points, and the **angle**.

Holding **SHIFT** while measuring ensures that measurement is strictly horizontal or vertical.



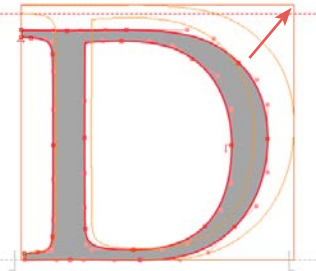
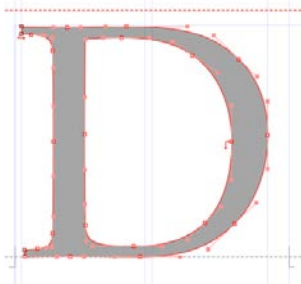
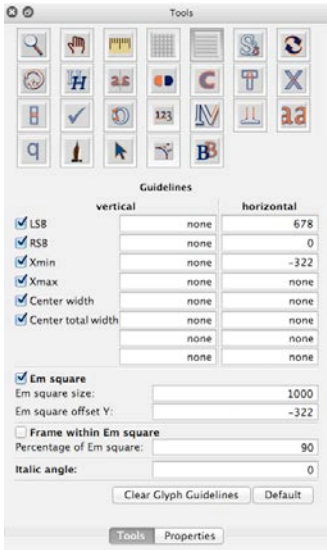
Grid

Display a grid in the background. You may define the line-to-line distance in font units or as lines per EM square, and decide at which origin point the grid is meant to start.



Guidelines

Here you may define a variety of horizontal and vertical guidelines which may be displayed in the background.

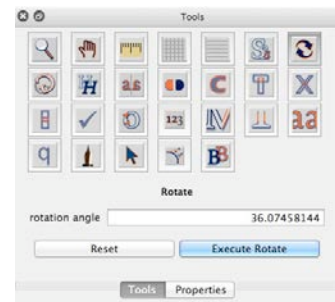
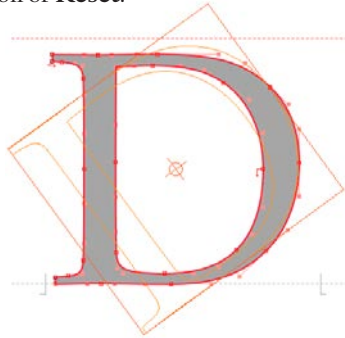


Scale

Scale the glyph by mouse or by entering precise scaling factors **fx** and **fy** and position adjustments **dx** and **dy**.

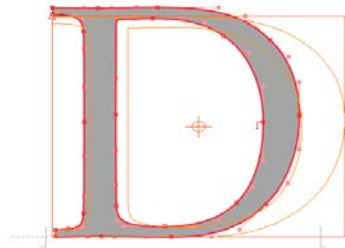
## Rotate

Use the mouse to rotate the glyph in the edit area, or enter a **rotation angle** in the editing area (a negative value is for clockwise rotation) and then **Execute Rotation** to apply the rotation or **Reset**.



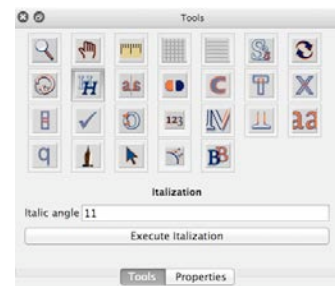
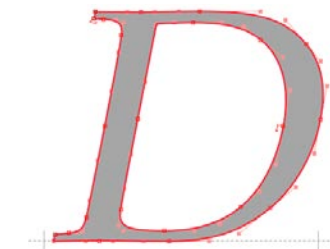
## Affine Transformation

This will transform the glyph in PostScript transformation matrix fashion. The equations are given at the bottom right corner of the editing area:



## Italization

This will italicize, or slant, the selected points or contours, either by using the mouse or by entering a value in the editing area (a positive value will slant to the right side) and confirming with **Execute Italization**.



## Mirror and Fold

Mirrors a glyph horizontally (**mirror left/right**), vertically (**mirror top/bottom**) or both ways (**mirror top left/bottom right**), and in various additional ways too.



## Hidden Lines

This function will remove contour overlaps. The four different way of doing so are illustrated below.



Before removing overlaps  
with Execute Hidden Lines.



After removing overlaps  
with **Union**.



With **1-2**: 'B' minus 'backslash'.



With **2-1**: 'backslash' minus 'B'.



With **Intersection**.

**Union** will logically add overlapping contours. **Intersection** will keep the overlapping (or intersecting) parts. **1-2** will subtract contour 1 from contour 2. **2-1** will subtract contour 2 from contour 1.

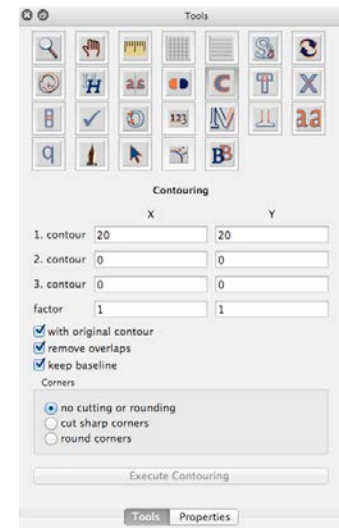
## Contouring

Use Contouring to add up to three contours around selected contours. Determine the distance of each additional **contour** from the original outline, independently for **X** and **Y** direction. Distances are given in units relative to UPM (the **head** table's **unitsPerEm**).

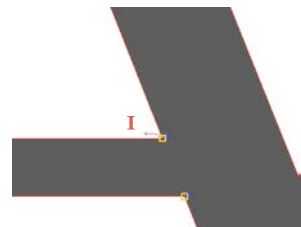


## T-Disconnecter

The T-Disconnecter will disconnect two parts of a glyph's shapes between any two points, in three simple steps:



To disconnect e.g. the bar from the rightside diagonal of an uppercase 'A',



## x-Disconnecter

This works like the T-Disconnecter.



Select points where stems, bars or diagonals cross.

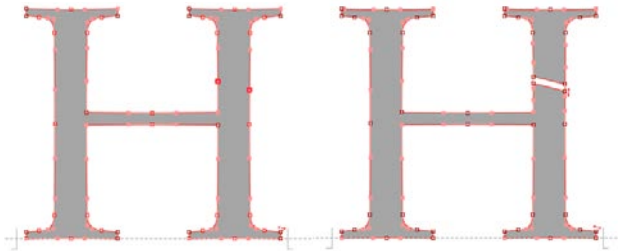
After applying x-Disconnecter, crossing stems, bars or diagonals will be disconnected.

1. select two points at which you want to 'break up' the glyph,
2. determine the **Overlap** amount in units (relative to the font's UPM) in the T-Disconnecter transformation area,
3. click **Execute T-Disconnecter** to apply disconnection:



### I-Disconnect

The I-Disconnect will disconnect a stem, bar or diagonal at any two selected points.



Select two points on a stem or bar where you would like to break it. (You may need to insert points for this purpose.)

After applying I-Disconnect, the stem or bar will be split into two.

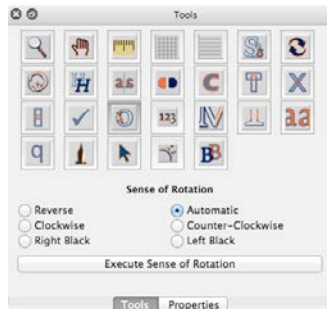


### Improve

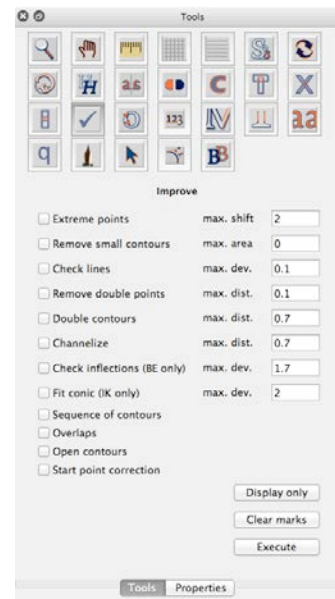
Outlines are not always flawless from a technical point of view. The image to the right shows aspects that LetterModeller can improve. You may merely review GlyphMaster's finding with **Display only** (and then **Clear marks** again) or apply suggested improvements with **Execute**.

### Sense of Rotation

The easiest way to correct contour directions is to select all contours and choose the option **Automatic**. However, you may also select one or more contours (the **Select Contours** mode is recommended for this) and then **Reverse** the current rotation, or define rotation to be **Clockwise** or **Counter-Clockwise**, or by categories **Right Black** or **Left Black**.



***Note:** In CFF-based OpenType fonts, the outermost contour is expected to be counter-clockwise. In glyph-based OpenType fonts, the outermost contour is expected to be clockwise. With both formats though, each contour needs to have the opposite direction than the contour in which it is located.*



### Sequence of Points and Contours

After selecting points or contours, you may change their order by selecting one of the following options:



### Merge

To merge one glyph with another one, go to the destination glyph and choose the Merge tool:

#### — Font

This is the font from which to take a glyph for merging with the destination glyph. You may open a source font with **Open Fonts ...**

#### — Glyph

This is the glyph to be merged with the destination glyph. You may select it by using the scrollbar to the right of the preview area, or enter either a **Glyph Index (GID)** or **Unicode** codepoint and confirm with **RETURN**.

#### — Adjustment

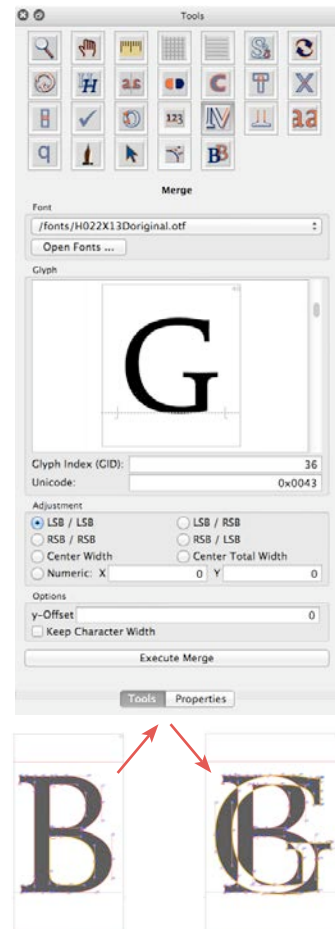
Here you may define how the added glyph shall align with the destination glyph: **LSB/LSB** will position it such that both glyphs' origin points share same coordinates, **RSB/RSB** will align glyphs at the right side, **LSB/RSB** will place the added glyph to the left of the destination glyph, **RSB/LSB** will place the added glyph to the right of the destination glyph, **Center Width** will center-align them (center being calculated from glyphs' outlines, excluding sidebearings), and **Center Total Width** will center-align them (center being calculated from glyphs' total widths, including sidebearings). Or determine the added glyph's position as **X** and **Y** adjustment from the destination glyph's origin.

#### — Options

An additional **Y-Offset**.

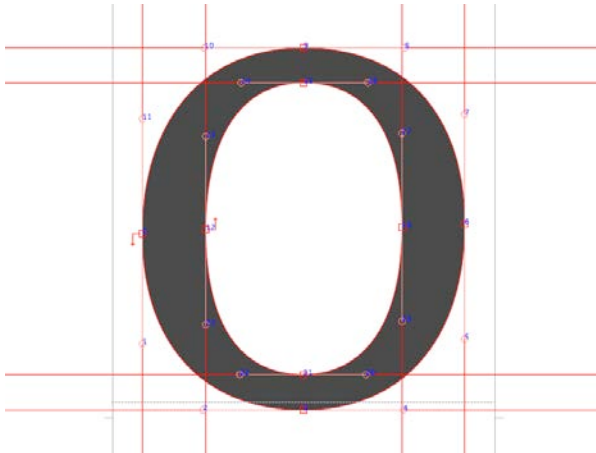
**Keep Character Width** of the destination character.

Time to remove contour overlaps with Hidden Lines!



### Character Hinting

Character Hinting expresses three choices by way of two buttons: Keep a glyph's original hinting information – this is the default. Click **Omit Original Hints** to remove existing hinting information. Or click the first button to toggle between **Autohinting On** and **Autohinting Off**.



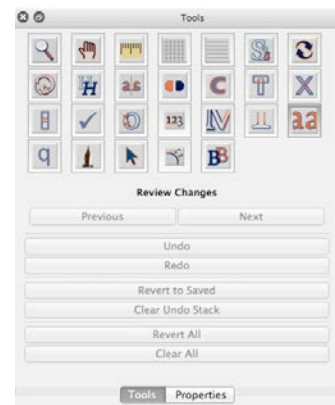
There are some more autohinting options in the Preferences – the kind of stems which the autohinter shall recognize and tolerances, and the glyphs from which alignment zones shall be derived. Please see the ► *Preferences* chapter for details.

### Review Changes

Review Changes presents a history of glyph transformations. Here you may **Undo** or **Redo** individual transformations. Or, per glyph, you either go back to the original version with **Revert to Saved** or accept your changes and **Clear Undo Stack**. Or, for all glyphs, you either go back to their original versions with **Revert All** or accept transformations and **Clear All** undo information.

### Quick Mode

Choose this mode to select and shift selected points or contours. Now with instructions right in the dock widget.



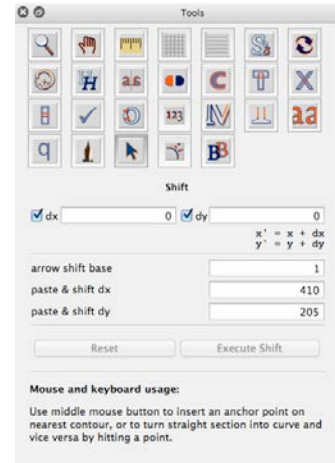
## Digitize

Time to reanimate your Aristo tablet!



## Shift

Shift by mouse, at **arrow shift base** units, or shift numerically, at **dx** and **dy** units. Additional **paste & shift dx** and **paste & shift dy** parameters allow you to define an offset at which contours are pasted, relative to their original positions. Now with instructions right in the dock widget.



## Shift Smooth

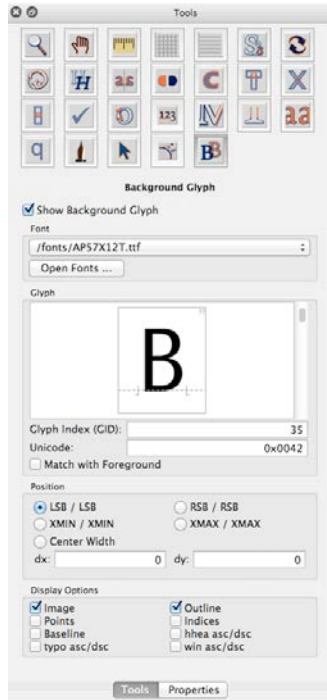
Shifting a point around will interpolate points between the selected point and neighbor extremum points, making sure that curves remain smooth.



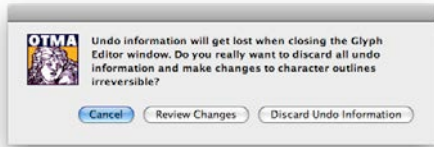
*In Shift Smooth mode, surrounding curves (up until neighboring extremum points) will remain smooth when moving a point.*

## Background Glyph

This tool allows you to select a background glyph. First, select one of the fonts opened in LetterModeller or open another font. Then, select a glyph by **Glyph Index** or **Unicode** codepoint. Finally, determine the position of the background glyph relative to the foreground glyph, and which of its information, e.g. **Points**, **Outlines**, **Indices**, you would like to see.



Closing the Glyph Editor window will produce the following dialog:



**Cancel** will get you back to the Glyph Editor so that you may continue drawing.

**Review Changes** will get you back to the Glyph Editor too and present Undo and Redo functions – allowing you to reconsider whether or not you want to apply individual transformations. See ► *Review Changes*.

**Discard Undo Information** will delete all undo information and apply all transformations to the font.

### **DTL LetterModeller Manual**

Edition 0.1 /2016; supports DTL LetterModeller  
version 5 © 2016 Dutch Type Library

*Text,*

*Design* Karsten Lücke

*Typeset* in DTL Argo, DTL Haarlemmer and DTL Haarlemmer Sans

### **Dutch Type Library**

Zwaenenstede 49  
5221 KC 's-Hertogenbosch  
The Netherlands

*phone* +31 (0)73 614 95 36

*fax* +31 (0)73 613 98 23

*e-mail* info@dutchtypelibrary.com

*website* www.dtl.nl and www.fonttools.org

The third party product names used in the DTL LetterModeller manual are for identification purposes only. All trademarks and registered trademarks are the property of their respective owners. The following trademarks may or may not be marked in this manual:

OpenType is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

PostScript is a registered trademark of Adobe Systems Incorporated.

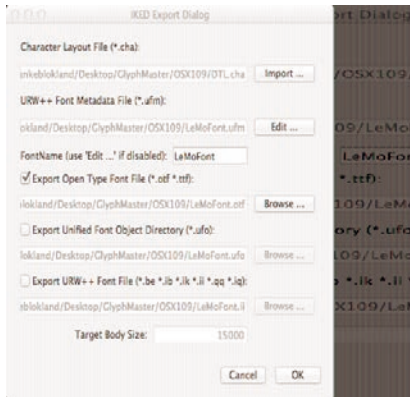
TrueType is a trademark of Apple Computer, Incorporated.

Adobe is a registered trademark of Adobe Systems Incorporated.

Apple and Macintosh are registered trademarks of Apple Computer, Incorporated.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Other company, product, and service names occasionally or incidentally mentioned in this manual may be trademarks or service marks of others.



The Export Dialog shows the following options:

– Character Layout File (\*.cha)

This is a file that connects the locations (‘slots’) of glyphs in the BE or IK databases with an encoding, Unicode code points, and glyph names. In the directory in which LetterModeller is placed by the installers for Mac OS and Windows, or the unzipped folder for Linux, the ‘DTL.cha’ file can be found. This file is optional, as is the encoding info in it; the end-user can define a proprietary .cha file also, as described in the attached DTL DataMaster (DM) manual. Please note that for exporting font and files a .cha file has to be selected first.

– URW++ Font Metadata File (\*.ufm)

Based on the BE or IK filename, the entries in the .ufm file will be automatically generated. LetterModeller will apply some rules, but it is far from guaranteed that the generated names are fully correct and probably the outcome will differ from personal preferences. In case of the latter, the ‘Edit...’ button will open the dialog for this (see DM manual).

– Export OpenType Font File (\*.otf \*.ttf)

Here the export path for font formats can be selected. Depending on the entered suffix, either .otf or .ttf fonts will be exported.

– Export Unified Font Object Directory (\*.ufo)

Here the export path for the .ufo can be selected. Please note that the currently exported .ufo files don’t not show up correctly in most tools that support this format (to be investigated).

– URW++ Font File (\*.be \*.ib \*.ik \*.ii \*.qq \*.iq)

This option can be used for exporting the different file formats that can be used in an IKARUS-based font production workflow. The ‘Target Body Size’ option is only applicable for the IKARUS-based file formats.

*Dutch Type Library*  
**FontMaster™**  
*Utilities*

# DTL DataMaster

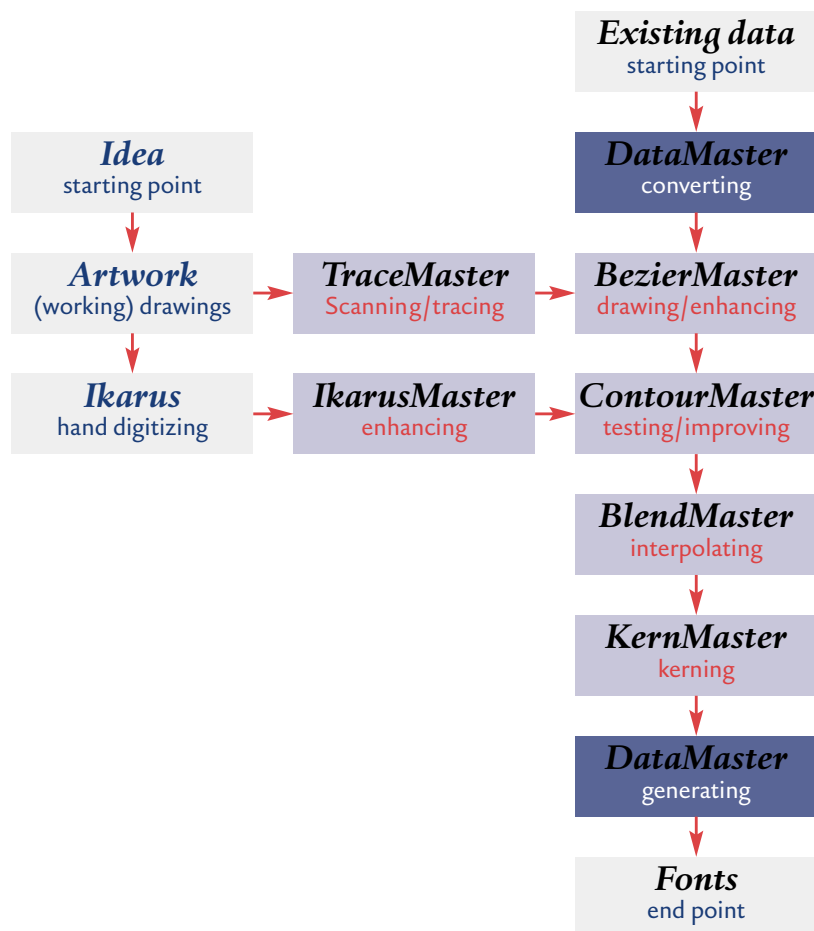


's-Hertogenbosch / Hamburg  
Autumn 2016

## INTRODUCTION

## FontMaster™ Utilities

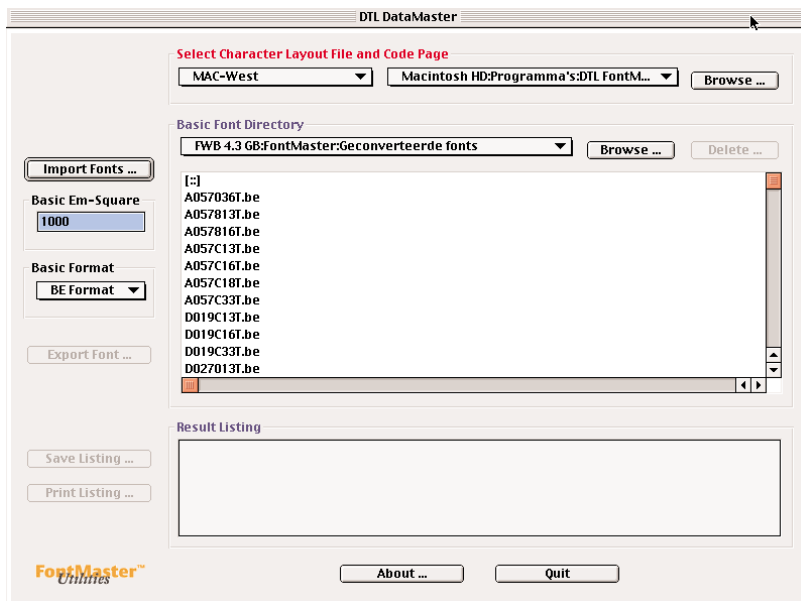
The diagram shows a possible production scheme based on the modules of DTL FontMaster.



DTL DataMaster is the batch module for managing, converting and generating database and font formats. DTL DataMaster converts the PostScript Type 1 and TrueType fonts for Mac OS and Windows to the BE and IK formats that are internally used by the DTL FontMaster utilities. It is possible to generate PostScript Type 1 and TrueType fonts for Mac OS and Windows. In addition, DTL DataMaster creates CFF- and TrueType-flavored OpenType fonts according to the specs.

The OpenType support in DTL DataMaster is based on Adobes OpenType SDK and therefor all existing features from Adobes PRO fonts are covered. DTL DataMaster provides a simple way to generate OTF's also for nonexperts; simply selecting different layouts and/or feature files is sufficient to create fonts with different character and feature sets.

The hints that DTL DataMaster generates for the TrueType fonts can be imported into Microsoft's VTT and used as a basis for delta hinting. The data associated with naming the fonts for the various formats for Mac OS and Windows is saved in platform-independent files.



### Starting DTL DataMaster

The first dialog has two major functions:

- Import fonts*
- Export fonts*

Furthermore it makes it possible to select a Character Layout File (\*.cha) and to change the resolution of the imported font. DTL DataMaster accepts PostScript Type 1, TrueType, BE and Ikarus format for input. The BE format can be converted to Ikarus format and vice versa. From the BE and Ikarus format it is possible to generate PostScript Type 1, TrueType and OpenType fonts.

#### I. Import Fonts ...

Before fonts are imported a series of selections have to be made, like the selection of the Character Layout File, the Basic Format and, of course, the Basic Font Directory.

##### I.I Character Layout File and Code Page

The code page defines the layout of the font. A code page consists of 256 character slots, of which a part is used for system functions. There are different code pages for many scripts, such as West-European, East-European, Greek and Cyrillic. Between the layout of the fonts for Mac OS and Windows are differences; the code pages for West-European are not equal on both platforms. The selection of the code page only has influence on an exported font and is of no importance when importing a font into DTL DataMaster.

The choice of the Character Layout File (\*.cha) is very important when a font is imported. A BE or Ikarus database does not contain any information concerning the PostScript name of a character (like adieresis for the character ä) or Unicode number (00E4 for the same character). Based on the original character names and Unicode numbers of the original PostScript Type 1 or TrueType fonts, each character gets a decimal number in the BE or Ikarus database. This process is reversed when a font is generated; each character gets based on its decimal number a PostScript name and in case of TrueType and OpenType fonts an Unicode number also.

The default Character Layout File is beeditor.cha. For producing an OpenType font the urwotf.cha file has to be selected. An OpenType font produced with the beeditor.cha will not support all possible features.

### **1.2 Basic Em-Square**

The resolution of the description of a font can vary, although there are standards for the different font and database formats. The resolution is defined in the *Em-square* and its units have no absolute size nor any relation to the resolution of any output device. The standard for PostScript fonts is 1000 x 1000 units and for TrueType 2048 x 2048 units. This does not mean that for these font formats no other Em-squares can be used but when the behaviour of such fonts can be unpredictable if the standards are not followed.

The default Em-square for the Ikarus format is 15.000 x 15.000 units. For the BE format an Em square of 1000 x 1000 units is recommended. The smallest Em-square supported by DTL DataMaster is 256 x 256 units.

### **1.3 Basic Format**

There are two options: BE Format and Ikarus format. The selection normally depends on the editor that will be used: DTL BezierMaster or DTL IkarusMaster. In case DTL DataMaster is used only for converting PostScript Type 1 or TrueType fonts into other formats the default BE Format is normally the best choice.

### **1.4 Basic Font Directory**

The Basic Font Directory is the place where the converted fonts are stored. So, although named 'Basic' this is the *target* directory.

## KEYWORDS

**Format** None  
**Structure** None  
**Element** None  
**Comment** Specifies the vertical offset between the accents for lowercase and uppercase characters

**AccentOffset**

**Format** AFM  
**Structure** Global font information  
**Element** Ascender  
**Comment** Top of lower case d

**Ascender**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmLowerCaseAscent  
**Comment** Distance that the ascender of lowercase letters extend above the baseline

**Format** TTF  
**Structure** 'hhea' table  
**Element** Ascender  
**Comment** Distance from baseline of highest ascender, typographic Ascent (Apple)

**Ascender HHEA**

**Format** All  
**Structure** None  
**Element** None  
**Comment** Used to scale the UFM entries to the targed grid

**Bodysize**

**Format** AFM  
**Structure** global font information  
**Element** CapHeight  
**Comment** Top of upper case H

**CapHeight**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmCapHeight  
**Comment** The height of uppercase characters

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfAscent, CapHeight, dfInternalLeading  
**Comment** Specifies the distance from the top of a character definition cell to the baseline of the typographical font. It is usefull for aligning the baseline of fonts of different height

## KEYWORDS

**Format** TI  
**Structure** Private dictionary  
**Element** BlueValues  
**Comment** Cap-height alignment

**Format** AFM  
**Structure**  
**Element** Comment  
**Comment** Arbitrary text, may be present in an AFM file

**Comment**

**Format** TI  
**Structure**  
**Element** %  
**Comment** Header

**Format** AFM  
**Structure**  
**Element** Comment  
**Comment** Text regarding the copyright

**Copyright**

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfCopyright(60)  
**Comment** as AFM

**Format** TI  
**Structure** FontInfo dictionary  
**Element** Copyright  
**Comment** as AFM

**Format** TTF  
**Structure** 'name' table  
**Element** Name ID 0  
**Comment** as AFM

**Format** AFM  
**Structure** Global font information  
**Element** Decender  
**Comment** Bottom of lower case p

**Decender**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmLowerCaseDescent  
**Comment** Distance that the descender of lower case letters extends below the baseline

## KEYWORDS

**Format** TTF  
**Structure** 'hhea' table  
**Element** Descender  
**Comment** Distance from baseline of lowest descender, typographic descent (Apple)

**Descender HHEA**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmDoubleLowerUnderlineOffset  
**Comment** Offset downward from the baseline where the top of the lower double underline should appear

**DoubleLowerUnderline-Offset**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmDoubleLowerUnderlineWidth  
**Comment** Thickness of the lower double underline bar

**DoubleLowerUnderline-Width**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmDoubleUpperUnderlineOffset  
**Comment** Offset downward from the baseline where the top of the upper double underline should appear

**DoubleUpperUnderline-Offset**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmDoubleUpperUnderlineWidth  
**Comment** Thickness of the upper double underline bar

**DoubleUpperUnderline-Width**

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfExternalLeading  
**Comment** Amount of extra leading that the designer requests the application add between rows

**ExternalLeading**

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfFace(pointer)  
**Comment** Microsoft Windows font Name. Up to four fonts may build up a family having the same FaceName. These fonts must make up two pairs with the same dfWeight, but different to the entry of the other pair. The two fonts having the same dfWeight must differ in the dfItalic byte

**FaceName**

## KEYWORDS

**Format** FOND  
**Structure** Macintosh resource  
**Element**  
**Comment** Name of the FOND seen in the menu if the resource does contain a family of several fonts

**Format** TTF  
**Structure** 'os2' table  
**Element** sFamilyClass  
**Comment** First number in UFM describes ibm Font-Family class, the second one the IBM Subfamily class. This parameter is intended for use in selecting an alternate font when the present one is not available

**FamilyClass**

**Format** AFM  
**Structure** Global font information  
**Element** FamilyName  
**Comment** Name of the 'font family' to which the font belongs

**FamilyName**

**Format** TI  
**Structure** FontInfo dictionary  
**Element** FamilyName  
**Comment** Human-readable name for a group of fonts that are stylistic variants of the same design. All fonts that are members of such a group should have exactly the same FamilyName. It should be suitable for use in a font selection menu

**Format** TI  
**Structure** Private dictionary  
**Element** BlueValues  
**Comment** figure-size alignment

**FigureSize**

**Format** TTF  
**Structure** 'os2' table  
**Element** usFirstCharIndex  
**Comment** Minimum Unicode index (char.code) in this font according to the cmap subtable for platform ID 3 and encoding ID 0 or 1. Should be 0x0020 for most fonts supporting Win-ANSI or other char.sets

**FirstCharIndex**

**Format** FOND  
**Structure** Macintosh resource  
**Element**  
**Comment** Makes up the baseline-to-baseline distance together with FondDescender and FondLeading on a Mac

**FondAscender**

## KEYWORDS

**Format** FOND  
**Structure** Macintosh resource  
**Element**  
**Comment** Makes up the baseline-to-baseline distance together with FondAscender and FondLeading on a Mac

**FondDescender**

**Format** FOND  
**Structure** Macintosh resource  
**Element**  
**Comment** FOND Family ID

**FontID**

**Format** NFNT  
**Structure** Macintosh resource  
**Element**  
**Comment** Together with the UniqueID a NFNT Resource ID is calculated

**Format** SFNT  
**Structure** Macintosh resource  
**Element**  
**Comment** SFNT Resource ID

**Format** FOND  
**Structure** Macintosh resource  
**Element**  
**Comment** Makes up the baseline-to-baseline distance together with FondAscender and FondDescender on a Mac. Some applications assume this value to be equal to 0

**FontLeading**

**Format** FOND  
**Structure** Macintosh resource  
**Element**  
**Comment** Name of the FOND seen in the menu if the resource contains only one font

**FondName**

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 1  
**Comment** The name the user sees in the font-menu

**FontFamilyName**

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 4  
**Comment** First part of FullFontName

## KEYWORDS

**Format** AFM  
**Structure** Global font information  
**Element** NameID 4  
**Comment** Name of the font program as presented to the PostScript language *findfont* operator

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfDriverInfo (pointer)  
**Comment** PostScript font name

**Format** TI  
**Structure** FontInfo dictionary  
**Element** FontName  
**Comment** Font's name, past to the PostScript *define* font operator by program TO. Should be unique. Can be a condensation of the FullName by removing spaces. It is customary to limit it's length to less than 40 characters.

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 6  
**Comment** PostScript name for the font separated to base- and suffix names. Each suffix starts with an uppercase letter after an hyphen has occurred in the Name

**Format** AFM  
**Structure** Global font information  
**Element** FullName  
**Comment** Full text name of the font

**Format** TI  
**Structure** FontInfo dictionary  
**Element** FullName  
**Comment** Unique, human-readable name for an individual font. Typically, it begins with the FamilyName and continues with various style descriptors separated by spaces

**Format** TTF  
**Structure** 'os2' table  
**Element** fsType  
**Comment** Indicates font embedding licensing rights for the font. Makes temporary loading of a font possible by an embedding-aware application. This licensing rights are granted by the vendor of the font

FontName

FullName

FsType

## KEYWORDS

**Format** TI  
**Structure** file header  
**Element** %!  
**Comment** The beginning of the first line in a Type 1 font program. The default is PS-AdobeFont-1.0

## Identifier

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfinternalLeading  
**Comment** Amount of leading inside the bounds set by the dfPixHeight member. Accent marks may occur in this area

## Internal Leading

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfAscent, CapHeight, dfInternalLeading  
**Comment** see **CapHeight**

**Format** AFM  
**Structure** Writing direction metrics  
**Element** IsFixedPitch  
**Comment** If true, this indicates that the font program is a monospaced font. A value false indicates a proportionally spaced font

## IsFixedPitch

**Format** TI  
**Structure** FontInfor dictionary  
**Element** IsFixedPitch  
**Comment** As AFM

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfPitchandFamily  
**Comment** As AFM. If true, the low order 4 bits are set to 0x00, else 0x01

**Format** TTF  
**Structure** 'post' table  
**Element** IsFixedPitch  
**Comment** As AFM. False is interpreted as 0, true is interpreted as 1

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 0  
**Comment** Japanese text regarding the copyright

## JPNCopyright

## KEYWORDS

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 1  
**Comment** Japanese name the user sees in the font-menu

JPNFontFamilyName

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 4  
**Comment** First part of japanese FullFontName

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 2  
**Comment** Japanes style and weight second part of Japanes FullFontName

JPNSubfamilyName

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 3  
**Comment** Unique Japanese identifier that applications can store to identify the font being used

JPNTrueTypeID

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 5  
**Comment** Japanese release and version information from the font vendor

JPNVersion

**Format** TTF  
**Structure** 'hhea' table  
**Element** LineGap  
**Comment** Typographic line gap (Apple)

LineGap<sub>HHEA</sub>

**Format** TTF  
**Structure** 'head' table  
**Element** LowestRecPPEM  
**Comment** Smallest readable size in pixels

LowestRecPpem

**Format** NFNT  
**Structure** Mac file system  
**Element** MacFileName.scr  
**Comment** Filename seen in the folder if FOND contains NFNT

MacFileName

## KEYWORDS

**Format** SFNT  
**Structure**  
**Element** MacFileName.ttf  
**Comment** Filename seen in the folder if FOND contains SFNT

**Format** FOND  
**Structure** Macintosh resource  
**Element**  
**Comment** Determines the style under which the font appears in the menu. It is:  
0 Regular or single font  
1 Bold in the family  
2 Italic in the family  
3 Bold Italic in the family

## MacStyle

**Format** PFM  
**Structure** PFMHEADER  
**Element** dflitalic  
**Comment** If MacStyle equals 2 or 3, dflitalic is set to 1 so that the font is recognized as Italic

**Format** TTF  
**Structure** 'head' table  
**Element** macStyle  
**Comment** as FOND

**Format** TTF  
**Structure** 'os2' table  
**Element** fsSelection  
**Comment** MacStyle 0 -> fsSelection 64  
MacStyle 1 -> fsSelection 32  
MacStyle 2 -> fsSelection 1  
MacStyle 3 -> fsSelection 33

**Format** AFM  
**Structure** Global metric information  
**Element** Notice  
**Comment** Font name trademark or copyright notice

## Notice

**Format** TI  
**Structure** FiontInfo dictionary  
**Element** AS AFM  
**Comment** AS AFM

## KEYWORDS

**Format** TTF  
**Structure** 'OS2' table  
**Element** Panose  
**Comment** 10 numbers to describe the visual characteristics of a font

**Panose**

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfWeight  
**Comment** Weight of the characters on a scale from 1–1000. The value of the UFM is taken times 100. It means:  
1 thin  
2 extra light  
3 light  
4 normal  
5 medium  
6 semi bold  
7 bold  
8 extra bold  
9 heavy

**PCWeight**

**Format** TTF  
**Structure** 'OS2' table  
**Element** usWeightClass  
**Comment** As PFM

**Format** PFM  
**Structure** PFMHEADER  
**Element** dfPitchAndFamily  
**Comment** Indicates, in a general way, the look of a font. There are:  
–Dontcare  
–Roman  
–Swiss  
–Modern  
–Script  
–Decorative

**PitchAndFamily**

**Format** UFM  
**Structure** typeface header  
**Element** Slant (= –Italic angle)  
**Comment** Italic angle in tenth of degrees counter-clockwise from the vertical

**Slant**

**Format** AFM  
**Structure** Writing direction metrics

## KEYWORDS

**Element** ItalicAngle  
**Comment** Angle in degrees counter-clockwise from the vertical of dominant vertical strokes of the font

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmSlant = -Slant  
**Comment** Angle in in tenth of degrees clockwise from the upright version of the font

**Format** TI  
**Structure** FontInfo dictionary  
**Element** AS AFM  
**Comment** AS AFM

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmStrikeOutOffset = StrikeOutOffset + StrikeOutWidth  
**Comment** Offset upward from the baseline where the top of a strike-out bar should appear

### StrikeOutOffset

**Format** TTF  
**Structure** 'os2' table  
**Element** yStrikeoutPosition  
**Comment** Position of the bottom of the strike-out relative to the baseline

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmStrikeOutWidth  
**Comment** Thickness of the strike-out bar

### StrikeOutWidth

**Format** TTF  
**Structure** 'os2' table  
**Element** yStrikeoutSize  
**Comment** as PFM

### SubfamilyName

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 2  
**Comment** Address only style and weight

**Format** TTF  
**Structure** 'name' table

## KEYWORDS

**Element** NameID 4  
**Comment** If not equal to Regular, second part of FullFontName

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmSubScript  
**Comment** Recommend vertical offset of subscript characters from baseline

### SubScript

**Format** TTF  
**Structure** 'os2' table  
**Element** ySubscriptYOffset  
**Comment** AS PFM

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmSubScriptSize  
**Comment** Recommend vertical size of subscript characters

### SubScriptSize

**Format** TTF  
**Structure** 'os2' table  
**Element** ySubscriptYSize  
**Comment** AS PFM

**Format** TTF  
**Structure** 'os2' table  
**Element** ySubscriptXOffset  
**Comment** Recommend horizontal size of subscript characters

### SubscriptXSize

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmSuperScript  
**Comment** Recommend vertical offset of superscript characters from the baseline

**Format** TTF  
**Structure** 'os2' table  
**Element** ySuperscripttYoffset  
**Comment** AS PFM

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmSuperScriptSize  
**Comment** Recommend vertical size of superscript characters

### SuperScriptSize

## KEYWORDS

**Format** TTF  
**Structure** 'os2' table  
**Element** ySuperscripttYSize  
**Comment** AS PFM

**Format** TTF  
**Structure** 'os2' table  
**Element** ySuperscripttXSize  
**Comment** Recommended horizontal size of superscript characters

**Format** TTF  
**Structure** 'name' table  
**Element** NameID 3  
**Comment** Unique identifier that applications can store to identify the font beeing used

**Format** TTF  
**Structure** 'os2' table  
**Element** sTypoAscender  
**Comment** New typographic ascender. One good source is the Ascender value from an AFM file

**Format** TTF  
**Structure** 'os2' table  
**Element** sTypoDescender  
**Comment** New typographic descender. One good source is the Descender value from an AFM file

**Format** TTF  
**Structure** 'os2' table  
**Element** sTypoLineGap  
**Comment** New typographic line gap. Typical values average 7–10% of units per em

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmUnderlineOffset  
**Comment** Offset downward from the baseline where the top of a single bar should appear

**Format** AFM  
**Structure** writing direction metrics  
**Element** UnderlinePosition  
**Comment** Recommend distance from the baseline for centering under-

**SuperScriptXSize**

**TrueTypeID**

**TypoAscender**

**TypoDescender**

**TypoLineGap**

**UnderlineOffset**

**UnderlinePosition**

lining strokes. This is the y coordinate of the center of the stroke

**Format** TI  
**Structure** FontInfo dictionary  
**Element** As AFM  
**Comment** As AFM

**Format** TTF  
**Structure** 'post' table  
**Element** underline Position  
**Comment** As AFM

**Format** AFM  
**Structure** writing direction metrics  
**Element** UnderlineThickness  
**Comment** Recommend stroke width for underlining

## UnderlineThickness

**Format** TI  
**Structure** FontInfo dictionary  
**Element** As AFM  
**Comment** As AFM

**Format** TTF  
**Structure** 'post' table  
**Element** UnderlineThickness  
**Comment** As AFM

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmUnderlineWidth  
**Comment** Thickness of the underline bar

## UnderlineWidth

**Format** TI  
**Structure** Private dictionary  
**Element** UniqueID  
**Comment** Integer in the range from 0 to 16777215 that uniquely identifies the font. The numbers from 4000000 to 4999999 form an open range and may be used in a controlled environment. To distribute a font widely a UniqueID should be obtained from Adobe Systems Inc.

## UniqueID

**Format** NFNT  
**Structure**

## KEYWORDS

### *Element*

**Comment** Together with the FondID a nfnt Resource ID is calculated

**Format** TTF

**Structure** 'os2' table

**Element** achVendID(4)

**Comment** Four character identifier for vendor of given font

**VendID**

**Format** AFM

**Structure** Global font information

**Element** Version

**Comment** Font program version identifier

**Version**

**Format** TI

**Structure** FontInfo dictionary

**Element** version

**Comment** As AFM

**Format** TTF

**Structure** 'head' table

**Element** fontRevision

**Comment** Version number

**Format** TTF

**Structure** 'name' table

**Element** NameID 5

**Comment** Release and version information from the font vendor

**Format** AFM

**Structure** Global font information

**Element** Weight

**Comment** Weight of the font. E.g. Bold

**Weight**

**Format** TI

**Structure** FontInfo dictionary

**Element** Weight

**Comment** Human readable name for the weight or 'boldness' attribute of a font

**Format** TTF

**Structure** 'os2' table

**Element** usWidthClass

**Comment** Relative change from the normal width to height ratio by a fontdesigner for the glyphs in a font. It means:

**WidthClass**

## KEYWORDS

- 1 Ultra-Condensed
- 2 Extra-Condensed
- 3 Condensed
- 4 Semi-Condensed
- 5 Medium (normal)
- 6 Semi-Expanded
- 7 Expanded
- 8 Extra-Expanded
- 9 Ultra-Expanded

**Format** TTF  
**Structure** 'os2' table  
**Element** usWinAscent  
**Comment** Ascender metric for windows, yMax for all characters in the Windows ANSI set. For platform 3 encoding o fonts same as yMax

**WinAscent**

**Format** TTF  
**Structure** 'os2' table  
**Element** usWinDescent  
**Comment** Descender metric for Windows, -yMin for all characters in the Windows ANSI set. For platform 3 encoding o fonts same as -yMin

**WinDescent**

**Format** TTF  
**Structure** 'os2' table  
**Element** xAvgCharWidth  
**Comment** Arithmetic average of the width of all of the 26 lowercase letters a through z. If any of the 26 lowercase letters are not present, this parameter should equal the weighted average of all glyphs. For non-UGL (platform 3, encoding o) fonts, use the unweighted average.

**XavrCharWidth**

**Format** AFM  
**Structure** Global font information  
**Element** XHeight  
**Comment** Top of lower case x

**XHeight**

**Format** PFM  
**Structure** EXTTEXTMETRIC  
**Element** etmXHeight  
**Comment** height of lower case letters in the font

## KEYWORDS

FontMaster™  
*Utilities*

<b><i>Format</i></b>	Tl
<b><i>Structure</i></b>	Private dictionary
<b><i>Element</i></b>	BlueValues
<b><i>Comment</i></b>	x-height alignment

## ACKNOWLEDGEMENTS

FontMaster™  
*Utilities*

DTL LetterModeller | DataMaster manual

*Edition* 0.01/2016

ISBN

©2016 Dutch Type Library

*Design* Frank E. Blokland

*Typeset* in DTL Argo and DTL Haarlemmer

*Dutch Type Library*

Zwaenenstede 49

5221 KC 's-Hertogenbosch

Daliënwaerd 71

5221 KE 's-Hertogenbosch

The Netherlands

*phone* + 31 (0)73 614 9536

*fax* + 31 (0)73 613 9823

*e-mail* [info@dutchtypelibrary.com](mailto:info@dutchtypelibrary.com)

*website* <http://www.dtl.nl>