

Dutch Type Library

DTL OTMaster





's-Hertogenbosch/Hamburg 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.



Introduction 7 About This Manual 8 First Steps into OTMaster 9 File Menu 9 Open 9 Close 10 Close All 10 Save 10 Save As ... 10 Save All 10 Import... 10 - URW++ Character Layout file [.cha] 10 -Adobe FDK Feature file [.fea] 10 - Unicode Variation Sequence file [.uvs] 10 - woff and woff2 [.woff/.woff2] 11 Export... 11 - URW++ Font file [.ufm] 11 - URW++ Font Metadata file [.ufm] 11 - URW++ Character Layout file [.cha] 12 - Adobe Font Metrics file from 'kern' table [.afm] 12 -Adobe FDK Feature file [.fea] 12 Export WOFF and WOFF2 ... [.woff/.woff2] 12 Print ... 13 Preferences 13 Quit 13 Messages 13 Clipboard 13 Inspecting Tables 14 Editing Tables 18 Edit Menu 18 Cut [and Delete] 18 Сору 18 Paste 18 **Grow** 18 **Sort** 18 Fixup 18 Viewing Tables 19 View Menu 19 Nested Tables 19 Text Dump 20 hex 21 dec 21 best fitting 21



OpenType Font Tables 22 Selected Notes about OpenType Font Tables 23 — AAT Tables 23 — CFF 24 — cmap 27 - COLR, CPAL 29 — DSIG 30 -gasp 30 -glyf, loca, cvt, prep, fpgm 31 — GDEF 32 — GPOS, GSUB 33 -head 36 —hhea 36 —hmtx 37 —kern 38 — maxp 39 — name 40 -0s/243— post 45 ттс and отс Fonts 46 OTMaster's Toolbox 47 **Tools Menu** 47 Font Viewer 47 **Text Viewer** 49 **Glyph Viewer** 52 Side by Side Viewer 54 Embedded Bitmap Viewer 55 Color Viewer 56 'kern' Table Viewer 60 'GPOS'/'GSUB' Viewer 62 Consistency Checker 73 -Header 73 -Name 76 -Version 76 -Statistics 77 -Unicode Ranges 77 - Codepage Ranges 78 -Languages 80 **Table Comparator** 82 Glyph Copy Tool 83 **Glyph Editor** 85 **File** 86 **Import...** 86 **Export...** 86 **Print...** 86

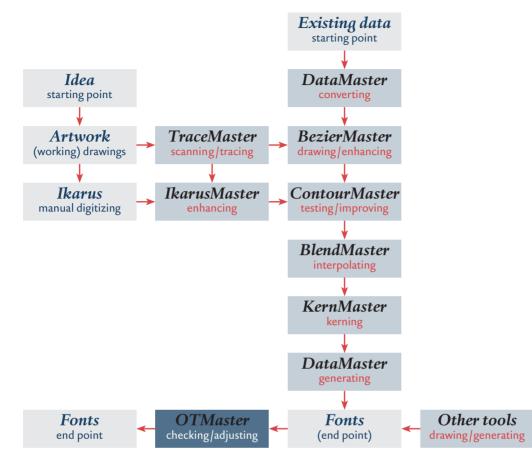


Save ... 86 **Character Preferences** 86 Messages... 86 Clipboard... 86 Edit / Selection 86 Revert to Saved 87 Undo 87 Redo 87 Delete 87 Cut 87 **Copy** 87 Paste 87 Paste & Shift 87 Previous / Next 87 **New** 87 Select All / Deselect All 87 Select Points 87 Select Contours 87 Select Contour Groups 88 Select Character 88 Swap Background 88 **View** 88 Tool Bars 88 Dock Widgets 88 **Display Options** 88 Glyph Set 88 Reset 88 Tools 89 **Zoom** 89 Scroll by Hand 89 Measure 89 Guidelines and Grid 90 Scale 90 Rotate 91 Affine Transformation 91 Italization 91 Mirror and Fold 92 Hidden Lines 92 Contouring 93 **T-Disconnector** 93 x-Disconnector 93 **I-Disconnector** 94 Improve 94 Sense of Rotation 94 Sequence of Points and Contours 95



Merge 95 Character Hinting 96 Review Changes 96 Quick Mode 96 Digitize 97 Shift 97 Shift Smooth 97 Background Glyph 98 Closing the Glyph Editor 99 Preferences 100 -File Options 100 -User Interface 100 -Foreground Character 101 -Background Character 101 -Autohinting 102 -Shortcuts 103 Shortcuts 104





Introduction

DTL OTMaster is a stand-alone application whose interface makes it easy to review and edit .otf and .ttf fonts' tables, regardless of whether these fonts have been generated with IkarusMaster, BezierMaster, DataMaster, or any other font editor.

Font editors, like the FontMaster suite, rely on their own internal data formats for type design and font production. With FontMaster, this is either an IK or a BE file, for Ikarus and Bezier outlines respectively, along with various data files for naming font and glyphs, for kerning and definition of typographic layout features. From these data, ready-to-use binary fonts are compiled as the very last step.

OTMaster is a tool whose purpose is to inspect and adjust such readyto-use binary fonts, irrespective with which font editor they have been created. Its advantage is that it allows editing of tables in a graphic user interface. Moreover, it comes with additional tools like a Glyph Editor to proof, edit and even draw glyph outlines, a 'kern' Table Viewer to proof and

DTL OTMASTER: INTRODUCTION



refine the kern table, and a 'GSUB'/'GPOS' Viewer to visually test (and in case of GPOS adjust) these OpenType layout tables.

OTMaster can open, edit and save fonts with SFNT file structure: CFF-based and glyf-based OpenType fonts, TrueType fonts, as well as TTC (TrueType Collection) fonts and OTC (OpenType Collection) fonts.

Because OTMaster allows you to edit a binary font's tables, which can be compared to open-heart surgery, it is highly recommended that you know the OpenType specification, at least as regards the tables whose entries you plan to adjust. ► *local* ► *www* (The current online version is 1.6. Please note that as of version 1.5, the OS/2 table has been updated to version 4, and a few nameIDs have been added to the name table.)

About This Manual

There are a number of links in this document. The first type of link is '**b** local' and links to the OpenType specification which Microsoft kindly provided so as to accompany this manual. As long as the folder 'OTM Manual resources' is located next to the manual's PDF file, clicking on such a link will open the according local html page in your web browser, offline. The second type of link is '**b** www' and will open the original online resource. This may be the preferred choice because these data may be more up-to-date. And finally there are internal links like '**b** Chapter', leading from one chapter to another one.



жо

ЖE

ሰ₩E

CTM File Edit View Tools Open...

Close

Close All

First Steps into OTMaster

When launching OTMaster, the main dialog is empty. Since OTMaster is meant for editing existing fonts, the first step is to open a font.



FILE MENU

Open

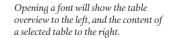
This opens an existing font using the standard Open dialog.

Alternatively, drag & drop a font file's icon onto OTMaster's main dialog. Or drag & drop one or more font files onto the OTMaster icon to launch OTMaster and open the fonts.

	Glyph Set and Code Ranges	View
199.13 Tord Root Offset Table Offset Table Table Directory * 'CFF' top dictionary 'CSUB' table 'CSUB' table 'OSUB' top	Chyph Set and Code Ranges Chyph Index (GID) $(1, 1)$ $($	Vew Vew Vew Vew Vew Vew Vew Vew
'name' header 'name' records ♥ 'post' table 'post' header		Print

onto OTMaster's main window to open it.

Tip: It is highly recommended that you only edit copies of fonts with OTMaster, to prevent that you inadvertedly overwrite a font when using Save while editing. Either, make a copy of the font file, and edit the copy in OTMaster. Or, Save As... a font immediately after opening it, by another name.





Close

Closes the currently active font. If you have made any changes which have not been saved yet, a dialog will ask whether you want to save changes now.

Close All

Closes all open fonts. If you have made any changes to a font which has not been saved yet, you will be asked whether to save recent changes or not.

Save

Saves the currently active font at its current location and thus overwrites the existing file.

Save As ...

Saves the currently active font but allows you to determine a new location in the standard Save dialog.

Save All

All open fonts will be saved at their current locations.

Import...

This dialog's options popup currently offers the following file types:

- URW++ Character Layout file [.cha]

Importing a text-based .cha file will allow OTMaster to show this file's glyph names in tables' **Comment** column. This may be helpful if you have generated a CID-keyed font from FontMaster and plan to edit it in OTMaster. Note that importing a .cha file does not have any impact on the font file or any of its tables.

OTMaster accepts .cha files that map, per glyph, its glyph index, Unicode codepoint and glyph name. Please see the column to the right.

— Adobe FDK Feature file [.fea]

This will import –i.e. compile – selected OpenType layout tables from a .fea file which conforms to Adobe's feature file syntax as described in the *OpenType Feature File Specification*. ► www

In the dialog which opens next, you may define which of the tables BASE, GDEF, GPOS, GSUB and name you would like to compile from the .fea file's data. Mind that you may define name table records in feature file syntax.

Possible import –i.e. compilation – errors are listed, and can be reviewed, in the ► *Messages* window!

- Unicode Variation Sequence file [.uvs]

Unicode Variation Sequences can be imported from a text-based .uvs file. Please see the ► *cmap* chapter.

Version 002.000
Starttable
GlyInd;UNINum;PSName
0;;.notdef
1;x0020;space
2;x0021;exclam
3;x0022;quotedbl
4;x0023;numbersign

Endtable

A .cha file as exported by OTMaster and accepted for import. The header must read 'Version 002.00'. Between tags 'Starttable' and 'Endtable', a semicolon-separated table holds the glyph information. The table's header indicates that each glyph is identified by glyph index (GlyInd) and optionally is provided with a Unicode codepoint (UNINum) and a glyph name (PSName).

The order of columns is arbitrary, but it is important that the order of data (in each row) matches the order determined in the table header.

OTMaster .cha files differ from those employed in FontMaster. OTMaster identifies glyphs by glyph index (GlyInd) while FontMaster identifies glyphs by a URW number (URWNum). To exchange .cha files between OTMaster and FontMaster, just replace the keyword 'GlyInd' by the keyword 'URWNum' (or the other way round). Please note that FontMaster .cha files may hold further information in additional columns. Keywords are described in the FontMaster manual.

Being semicolon-separated, a .cha file can easily be edited e.g. in Excel: Add the suffix '.csv' to the .cha file's file name and in Excel's Open dialog select 'Text (*.prn; *.txt; *.csv)' as file type. After adjusting the table and saving it, remove the '.csv' suffix from the file name.



-woff /woff2 [.woff/.woff2]

Upon import of a woff and woff2 font, this gets converted to a glyf table based TTF/OTF font and saved immediately. A dialog appears, asking you where to save the new font.

Convert "MyFont-Reg	ular.woff" to Open Type and open the re	sulting file.
Target Font File:	TrueType Font File (*.ttf)	
/fonts/MyFont-Regul	ar.ttf	Browse
output directory sa	me as input directory	
	Cance	ОК

Export...

Currently, these data may be exported:

OTF E	kport Dialog
Export URW++ Font File:	V URW++ BE Font File (*.be) URW++ IB Font File (*.ib) URW++ IK Font File (*.ik) Browse
Export URW++ Font Metadata File (*.ufm):	URW++ II Font File (*.ii) URW++ QQ Font File (*.qq)
/fonts/AP57X12T.ufm	URW++ IQ Font File (*.iq) Browse
Target Body Size for URW++ Font an	d Metatdata File: 15000
☑ Export URW++ Character Layout File (*.cha):	
/fonts/AP57X12T.cha	Browse
Export Adobe Font Metrics File (*.afm):	
/fonts/AP57X12T.afm	Browse
Export Unicode Variation Sequence File (*.uv	5)
/fonts/AP57X12T.uvs	Browse
🗹 Export Feature File (*.fea):	
/fonts/AP57X12T.fea	Browse
🗌 'BASE' Table 📄 'GDEF' Table 📄 'GPOS' Table	· ✔ 'GSUB' Table 🗌 'name' Table
output directory same as input directory	
	Cancel

glyphs by their glyph names. This however means that if you 'transfer' typographic layout features from one of your fonts to another one, glyphs in both fonts need to share identical names, and all glyphs referenced in the feature file must be present in the destination font. Please mind that such a 'transfer'

Please mini that such a 'transfer is not lossless because it involves interpretation: 1. Export will dump OpenType layout tables into an AFDKO-syntax feature file, and 2. Import will compile OpenType layout tables from a feature file.

Note: Feature files usually identify

Also, 'transferring' features from one font to another often is not that rewarding because most likely each font has a different glyph set with different alternates, so that different feature behavior is required.

- URW++ Font file [.be .ib .ik .ii .qq]

Depending on the input font format, one of the following Ikarus and FontMaster related formats will be generated for reuse in FontMaster: Cubic Bezier data, Ikarus data or Quadratic Bezier data, plus variants of the former two which may include instructions for intelligent scaling (commonly called 'hinting').

- URW++ Font Metadata file [.ufm]

A .ufm file contains the font metadata information which FontMaster needs for generating fonts, including names and vertical metrics. Detailed information can be found in the FontMaster manual. Note: A WOFF font's metadata will be ignored when importing it. As a consequence, a roundtrip like WOFF > TTF > WOFF will result in WOFF metadata getting lost.

DTL OTMASTER: FILE MENU



- URW++ Character Layout file [.cha]

A .cha file as exported by OTMaster maps, per glyph, its glyph index, Unicode codepoint and glyph name.

An OpenType font's .cha file is helpful when importing this font in FontMaster: if you replace 'GlyInd' by 'URWNum' in the .cha file's header, the original glyph indices will serve as URW numbers which identify glyphs in the FontMaster suite.

— Adobe Font Metrics file from 'kern' table [.afm] This will export an .afm file with the kern table's kerning – if there is a kern table in the font.

An .afm file contains font metrics and basic font metadata. For details see Adobe's *Font Metrics File Format Specification*. ► www

— Adobe FDK feature file [.fea]

This will export selected OpenType layout tables as a .fea file in Adobe's feature file syntax. As with the import function, you may choose from BASE, GDEF, GPOS, GSUB and name. Export means that the tables will be dumped which involves interpretation as described in the note on the previous page.

Please consult Adobe's *OpenType Feature File Specification* for details about the feature file keywords, syntax and examples. *www*

Export WOFF/WOFF2...

Convert to and save a font in Web Open Font Format, wOFF or WOFF2. You may choose one of them in the dialog that will open. Note: Since an .afm file's metrics are supposed to relate to an implicit UPM = 1000, OTMaster will 'scale' metrics in case if the font's UPM is not 1000.



Print ...

Because it does not make much sense to print data as represented in the user interface, it is recommended that you switch, in the View menu, to Text Dump mode and save either the text dump or XML files and print this.

Preferences

Please see the ▶ *Preferences* chapter.

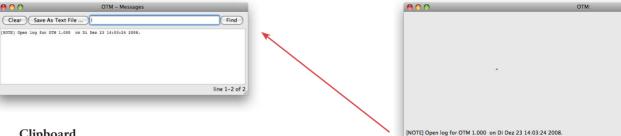
Quit

This will quit the application. If fonts have been changed but not saved yet, you will be asked whether you want to save these changes or not.

Messages

This opens the Messages window which collects all status messages as shown at the bottom of the OTMaster's main dialog. You may Clear the panel, Save As Text File ... its content, and search in it with help of Find.

A typical status message. All messages are collected in the Messages window.



Clipboard

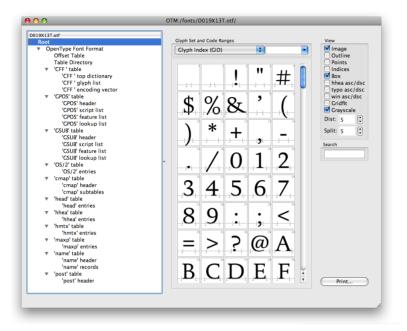
Check the content of the clipboard which you may have filled by cutting or copying anything. Also see the later > *Clipboard* section.

Note: In the Mac os version of OTMaster, both Preferences and Quit are located in the OTM menu.



Inspecting Tables

Once a font is opened, OTMaster's main dialog consists of two areas:

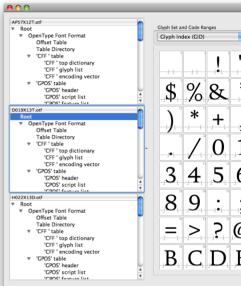


The leftside area presents a table overview with the font's file name in a header. If more than one font has been opened, this area consists of as many segments as there are open fonts. The rightside area shows the entries of the table which is currently selected in a font's table overview.

The table overview usually starts with the Root entry which represents the font as a whole. Root is the default selection in the table overview, and accordingly the rightside content area presents this font's glyph set. Strictly speaking, this glyph set overview is merely for your convenience – Root does not represent any specific table. Functionality and options for Root glyph overview and Font Viewer are identical and are described in the chapter ► Font Viewer.

Root includes OpenType Font Format which indicates that a font is an OpenType font. This in turn is populated with the OpenType font's data according to SFNT file structure: An Offset Table tells how many tables this font contains, plus information for binary search. The Table Directory points out start (offset) and length of every table. These then are followed by all tables present in the font. (A TTC font would indicate TrueType Collection Format rather than OpenType Font Format. The overall structure would differ slightly too. Please consult the chapter > TTC Fonts.)

Offset Table and Table Directory are not supposed to be adjusted manually. They are read-only and will be updated automatically as soon as tables are modified, removed or added. OTMaster's main window showing a CFF-based OpenType font. If more than one font is opened, the table overview is tiled into as many segments as there are open fonts. The currently active font's segment 'alows' in the Mac os version:



[NOTE] Extern to intern conversion of 'post' table successfully done

Tip: Use the **Search** field to quickly locate glyphs by glyph index. The viewer will jump to this glyph and highlight it.

The term OpenType Font Format refers to both CFF-based and glyfbased OpenType fonts. The former store outline data in a CFF table, the latter store them in a glyf table.



Simple tables include only [table] entries. Selecting [table] table in the table overview will exhibit the top level of this table in the content area.

19X13T.otf	6	Туре	Name	Value	Comment 'OS/2' entries			
'GSUB' lookup list		struct	Struct	+	'OS/2' entries			
 'OS/2' table 	U .							
'OS/2' entries	6							
 'cmap' table 'cmap' header 	1							

A table's top level.

The **+** sign in the **Value** column indicates that there is more information, and a click on the **+** will fold out a nested table whose content is indented:

D019X13T.otf	Type Name Value Comment		
Root OpenType Font Format	struct Struct + 'OS/2' entries		
Offset Table	Type Name	Value	Comment
Table Directory CFF ' table	USHORT version	0x0003	OS/2 table version number.
'CFF ' top dictionary	SHORT xAvgCharWidth	556	Average weighted escapement.
'CFF ' glyph list 'CFF ' encoding vector	USHORT usWeightClass	400	Indicates the visual weight.
	USHORT usWidthClass	5	Indicates a relative change from the normal aspect ratio.
'GPOS' header 'GPOS' script list	SHORT fsType	0x0004	Indicates font embedding licensing rights for the font.
'GPOS' script list 'GPOS' feature list	 SHORT ySubscriptXSize 	650	Subscript horizontal font size.
'GPOS' lookup list	SHORT ySubscriptYSize	600	Subscript vertical font size.
▼ 'GSUB' table 'GSUB' header	SHORT ySubscriptXOffset	0	Subscript x offset.
'GSUB' script list	SHORT ySubscriptYOffset	75	Subscript y offset.
'GSUB' feature list 'GSUB' lookup list	SHORT ySuperscriptXSize	650	Superscript horizontal font size.
▼ 'OS/2' table	SHORT ySuperscriptYSize	600	Superscript vertical font size.
'OS/2' entries v 'cmap' table	SHORT ySuperscriptXOffset	0	Superscript x offset.
'cmap' header	SHORT ySuperscriptYOffset	350	Superscript y offset.
'cmap' subtables	SHOPT UStrikooutSizo	20	Width of the strikeout stroke

Accessing a table's entries via its top level – by unfolding a nested table.

Since the top level of a table is not very informative, most of the time you may want to select [table] entries to access the table's entries directly, ready to be studied or adjusted:

D019X13T.otf	Type	Name	Value	Comment	16
Root OpenType Font Format	USHORT	version	0x0003	OS/2 table version number.	1
Offset Table	SHORT	xAvgCharWidth	556	Average weighted escapement.	
Table Directory CFF ' table	USHORT	usWeightClass	400	Indicates the visual weight.	
CFF ' top dictionary	USHORT	usWidthClass	5	Indicates a relative change from the normal aspect ratio.	
'CFF ' glyph list 'CFF ' encoding vector	SHORT	fsType	0×0004	Indicates font embedding licensing rights for the font.	ų
	SHORT	ySubscriptXSize	650	Subscript horizontal font size.	1
'GPOS' header 'GPOS' script list	SHORT	ySubscriptYSize	600	Subscript vertical font size.	
'GPOS' feature list	- SHORT	ySubscriptXOffset	0	Subscript x offset.	1
'GPOS' lookup list	SHORT	ySubscriptYOffset	75	Subscript y offset.	1
GSUB table	SHORT	ySuperscriptXSize	650	Superscript horizontal font size.	1
'GSUB' script list 'GSUB' feature list	SHORT	ySuperscriptYSize	600	Superscript vertical font size.	1
'GSUB' feature list 'GSUB' lookup list	SHORT	ySuperscriptXOffset	0	Superscript x offset.	1
▼ 'OS/2' table	SHORT	ySuperscriptYOffset	350	Superscript y offset.	1
'OS/2' entries v 'cmap' table	SHORT	yStrikeoutSize	20	Width of the strikeout stroke.	1
'cmap' header	SHORT	yStrikeoutPosition	274	The position of the top of the strikeout stroke.	I.
'cmap' subtables	SHORT	sFamilyClass	0x0000	Font-family class and subclass.	4

Accessing a table's entries directly.



Tables of variable length which consists of one or more subtables like cmap or kern, or multiple entries like name, then [table] table will include both [table] header and [table] entries. Like Offset Table and Table Directory, the [table] header is read-only:

D019X13T.otf	6	Type	Name	Value	Comment	
'OS/2' entries ▼ 'cmap' table		USHORT	version	0x0000	Table version number (0).	
'cmap' header	0	USHORT	NumTables	3	Number of encoding tables that follow.	
'cmap' subtables ▼ 'head' table	×.					

Header of a table which contains multiple subtables.

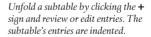
The cmap table is a good example for a table which consists of multiple subtables. This is reflected in the 'cmap' entries table content area:

D019X13T.otf	#	platformID	encodingID	format	length	languageID	GlyphMapping	Com
▼ Root	0	0	3	4	754	0	+	Unicode; 2.0+ semantics, BMP only
OpenType Font Format	0 1 2	0						
Offset Table Table Directory	1	1	0	0	262	0	+	Macintosh; Roman
▼ 'CFF ' table	2	3	1	4	762	0	+	Microsoft; Unicode BMP only
 'CFF' glyph list 'CFF' encoding vector 'CPOS' table 'CPOS' heads list 'CPOS' feature list 'CPOS' footup list 'CSUB' schipt list 'CSUB' schipt list 'CSUB' schipt list 'CSUB' schipt list 'CSUB' feature list 'CSUB' feature list 'CSC/2' feature list 'CSC/2' entries 	•							
▼ 'cmap' table 'cmap' header								
'cmap' subtables								
▼ 'head' table ▼	10) + +

A table's 'table' entries shows the available subtables in the content area, ready to be unfolded.

And again the **+** (in the **GlyphMapping** column) signals that per subtable there is some content to be unfolded. Clicking on one of them will reveal the respective subtable's mappings of Unicode codepoints (**Code**) to glyph indices (**GID**). In our example, we click the last subtable's **+** to see it's entries. (Clicking the **+** again will hide it.)

D019X13T.otf	#	plati	formID	encodingIC	format	length	languageID	GlyphMapping	Co	r o
Root	0		0	3		754	0	+	Unicode; 2.0+ semantics, BMP on	
 OpenType Font Format Offset Table 	1		1	0		2.62	0	+	Macintosh: Roman	1
Table Directory	2		3	1		7.62	0	+	Microsoft: Unicode BMP only	II.
 'CFF ' table 'CFF ' top dictionary 	2	#	Code	GID	4	102	Comment		microsoft, oncode binr only	E.
CFF top dictionary							Comment			1
'CFF ' encoding vector		0	0x002	-	space					1
▼ 'GPOS' table		1	0x002	. 2	exclam					1
'GPOS' header 'GPOS' script list		2	0x002	2 3	quotedbl					a.
'GPOS' feature list		3	0x002	4	numbersig	,				a.
'GPOS' lookup list		4	0x002	5	dollar					I.
'GSUB' table 'GSUB' header		5	0x002	6	percent					1
'GSUB' script list		6	0x002	. 7	ampersand					1
'GSUB' feature list		7	0x002		quotesingle					1
'GSUB' lookup list ▼ 'OS/2' table		8								1
'OS/2' entries			0x002		parenleft					11
▼ 'cmap' table		9	0x002	9 10	parenright					U
'cmap' header		10	0x002;	11	asterisk					4
'cmap' subtables		11	0+0033	1.2	oluc					





However, some tables like GSUB or GPOS have a rather complex structure, which makes it hard to keep track of the hierarchy and the level to which displayed entries belong. To address this, and to help comparing tables of different fonts, OTMaster can display a table's or nested table's content in a separate window. Just double-click onto a nested table's header, like the one selected (and thus colored yellow) in the image below

D019X13T.otf	#	pla	tformID	encodingID	format	length	languageID	GlyphMapping	Cor
▼ Root	0		0	3	4	754	0	+	Unicode: 2.0+ semantics, BMP only
 OpenType Font Format Offset Table 	1		1	0	0	2.62	0	+	Macintosh; Roman
Table Directory	2		3	1		762	0	+	Microsoft: Unicode BMP only
▼ 'CFF ' table	6		Code			162	Comment		microsoft, oncode bmr only
'CFF ' top dictionary 'CFF ' glyph list		#					Comment		
'CFF ' encoding vector		0	0x002		space				
▼ 'GPOS' table		1	0x002	1 2	exclam				
'GPOS' header 'GPOS' script list		2	0x002	2 3	quotedbl				
'GPOS' feature list	-	3	0x002	3 4	numbersign	,			
'GPOS' lookup list		4	0x002	4 5	dollar				
▼ 'GSUB' table 'GSUB' header		5	0×002	5 6	percent				
'GSUB' script list		6	0x002		ampersand				
'GSUB' feature list									
'GSUB' lookup list ▼ 'OS/2' table		7	0x002		quotesingle				
VOS/2 table		8	0x002		parenleft				
▼ 'cmap' table		9	0x002	9 10	parenright				
'cmap' header	/	10	0x002	la 11	asterisk				
'cmap' subtables		11	0,000	DK 1.2	oluc				

Click onto a nested table's header ...

to create a new window. This way, you can compare multiple font's tables, and as many as you like. The only limitation is the size of your screen!

D019X13T.otf	h	#	plat	formID	encodingI	format	length	languageID	GlyphMapping	Cor
 Root OpenType Font Format 	Ш	0		0		4	754	0	+	Unicode; 2.0+ semantics, BMP only
Offset Table	ш	1		1		0 0	262	0	+	Macintosh; Roman
Table Directory CFF ' table	Ш	2	0	000	DTM - D019	x13T.otf/OT	F/cmap/List#2	/GlyphMapping	- Code to GID I	ist de BMP only
VCFF table 'CFF' top dictionary	Ш		#	Code				Comment		0
'CFF ' glyph list	Ш	- 1	0	0x00	20 1	space				-
'CFF ' encoding vector GPOS' table	ш		1	0x00	21 2	exclam				
'GPOS' header	Ш		2	0x00	22 3	quotedbl				
'GPOS' script list 'GPOS' feature list			3	0x00	23 4	numbersig	1			
GPOS' lookup list ▼ 'GSUB' table 'GSUB' header		4	0x00	24 5	dollar					
		5	0x00	25 6	percent					
'GSUB' script list			6	0x00	26 7	ampersand				
'GSUB' feature list 'GSUB' lookup list	11		7	0x00	27 104	quotesingle	2			
▼ 'OS/2' table	11		8	0x00	28 9	parenleft				
'OS/2' entries v 'cmap' table	11		9	0x00	29 10	parenright				
'cmap' header	U		10	0x00	2a 11	asterisk				¥
'cmap' subtables ▼ 'head' table	A V		11	0,000	26 12	oluc	_	_	_	2

... to open a new window for this nested table.

We have already seen some important OTMaster actions:

- I. Click a + sign to show or hide a nested table.
- 2. Change a table's entries in usual textboxes.
- 3. Double-click a nested table's header to show its content in a new window. And one addition:

4. Switch from column to column with →I (to right) and SHIFT+→I (or I←;

to left) and from line to line with \uparrow and \downarrow keys.



Editing Tables

The most important functions for removing or adding data are found in the **Edit** menu. All of these functions apply to individual tables selected in the table overview, or table entries selected in the table content area.

EDIT MENU

Cut [and Delete]

Cuts a selected table or table entry. This will remove the table from the font, or entry from the table, and keep it in the clipboard – **Cut** implies the functionality of **Delete**.

Сору

Copies a selected table or table entry into the clipboard without removing it from the font.

Paste

Pastes a table from the clipboard into the currently selected font, or pastes a table entry from the clipboard into the currently selected table.

Grow

This function does exactly what it says – help growing a table. In tables of variable length (such as name, cmap, kern) select an entry (like a name record in the name table) and use Grow to duplicate it. In other tables, Grow will create a new, empty entry.

Sort

Sorts table entries in tables of variable length. For example, this will sort name table entries by platformID-encodingID-languageID-nameID, or sort cmap table entires by Unicode codepoints.

Fixup

Use this to remove duplicate table entries from tables of variable length (like name, cmap, kern). This is helpful, for example, if you have used **Grow** to duplicate an existing name table name record but then find that you do not need an additional name record.

ОТМ	File	Edit	View	1	Tools	
	_	Cut	ş	€X	_	
		Cop	by S	₿C		
		Pas	te a	۴V		
		Gro	w a	₿G		
		Sor	t a	₩R		
		Fixu	ab 8	₿F		
				_		

Note: There is no special **Delete** function in OTMaster. However, **Cut** serves the same purpose.



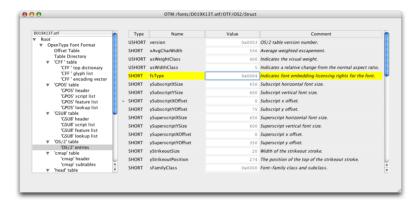
Viewing Tables

OTMaster offers two modes for viewing tables and two modes of table value representation representing table values.

VIEW MENU

Nested Tables

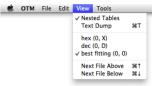
This mode presents each table's content as a typical dialog. With tables like head, hhea, OS/2 there are, per entry, its data **Type**, its **Name** and finally the **Value** in a textbox which may be edited in most cases. Where possible, there is an additional **Comment** column with a brief description. This way, OTMaster is not just a table editor but at the same time provides a built-in documentation for most of the table data. For example, OTMaster exposes glyph names in the **Comment** column which is particularly useful since tables identify glyphs by mere glyph indices.



The Nested Tables mode exposes only the top level, indicating the existence of a nested table by a **+** sign which, when clicked, will show or hide a nested table's entries.

Double-click a (nested) table's header to display its content in a separate window.

While in Nested Tables mode, a right-click into the content area will open a context menu which offers **Text Dump** and **XML Dump** modes (see the next page for details). Choosing either of them will open a new window.



The Nested Tables mode. Edit table entries in usual textboxes!



Text Dump

Text Dump will present table data as text. Please note that with complex tables –like glyf, CFF, GSUB or GPOS – this text is rather a summary:

•	OTM:/fonts/D019X13T.otf/OTF/OS2/Struct	
D019X13T.otf	xxt. Save As Text File	Find
🔻 Root	Save As Text File	Find
OpenType Font Format	True Type Table Listing	
Offset Table		
Table Directory	Date: Sun Dec 14 12:22:17 2008	
▼ 'CFF ' table		
'CFF ' top dictionary	Directory: /fonts	
'CFF ' glyph list	Filename: D019X13T.otf	
'CFF ' encoding vector		
▼ 'GPOS' table		
'GPOS' header	 '0S/2' - 0S/2 and Windows specific metrix * 	
'GPOS' script list		
'GPOS' feature list	Offset: 37400 Length: 96	
'GPOS' lookup list		
▼ 'GSUB' table	Version: 3 AvecharWidth: 556	
'GSUB' header	MeightClass: 400	
'GSUB' script list	WidthClass: 5 FaType: 4	
'GSUB' feature list	SubscriptXSize: 650	
'GSUB' lookup list	SubscriptYSize: 600 subscriptxoffset: 0	
▼ 'OS/2' table	SubscriptYOffset: 75	
'OS/2' entries	SuperscriptXSize: 650 SuperscriptYSize: 600	
▼ 'cmap' table	SuperscriptXOffset: 0	
'cmap' header	SuperscriptYOffset: 350 StrikeOutSize: 20	
'cmap' subtables	StrikeOutPosition: 274	u
▼ 'head' table	FamilyClass (hex): 0 PANOSE.bramilyType: 0	
'head' entries	PANOSE.bSerifStyle: 0	
▼ 'hhea' table	PANOSE.bWeight: 0 PANOSE.bProportion: 0	
'hhea' entries	PANOSE.bContrast: 0 PANOSE.bStrokeVariation: 0	
v 'hmtx' table	PANOSE.bArmStyle: 0	
* hmtx table	PANOSE.bLetterform: 0 PANOSE.bMidline: 0	
v maxp' table	PANOSE.bxmeight: 0	
maxp table 'maxp' entries	UnicodeRange[0]: 00000003 UnicodeRange[1]: 0000000	
maxp entries √ 'name' table	UnicodeRange[2]: D0000000	Ĭ
		Ŧ
'name' header		line 1-46 of 65
name records		11112 1-40 01 05

A table's content in Text Dump mode.

While in Text Dump mode, the **XML** checkbox at the top of the table content area allows you to switch to XML mode which will format the summary in XML style:

D019X13T.otf	VM. Save As Text File	Find
▼ Root	Jave As Text File	Fille
OpenType Font Format Offset Table Table Directory 'CFF-Eitlord dictionary 'CFF-Eitlord dictionary 'CFF-Eitlord dictionary 'CFF-Eitlord dictionary 'CFF-Eitlord dictionary 'CFOS' table 'CFOS' testare list 'CSUB' Table 'CSUB' Table 'CS	<pre>-display -display -displa</pre>	
v 'OS/2' table	<pre><pre><pre>cpanose.bNidline>0x00</pre>/panose.bNidline> <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>	
VOS/2* entries Vosapi table Vosapi subtables Vosapi subtables Vosapi subtables Vosapi subtables Vosapi subtables Vosapi subtables Vosapi table Vosapi table Vosapi table Vosapi table	<pre>distincted and apple resolutions / all acids apple resolutions and resolutions</pre>	Ĵ
'name' header		line 1-46 of 49
'name' records		line 1-40 01 49

Text dump in XML style. Please note that this is not compatible with TTX's XML-style dump!

Save As Text File ... will save a plain or XML text dump – useful for comparison in a text editor, or making printouts for proofreading. **Find** will search the (XML) text dump for a string.



hex

Hex shows values as hexadecimal numbers:

USHORT	usWidthClass	0x0005	Indicates a relative change from the normal aspect ratio.
SHORT	fsType	0x0004	Indicates font embedding licensing rights for the font.
SHORT	ySubscriptXSize	0x01f4	Subscript horizontal font size.

dec

Dec shows values as decimal numbers:

USHORT	usWidthClass	5	Indicates a relative change from the normal aspect ratio.
SHORT	fsType	4	Indicates font embedding licensing rights for the font.
SHORT	ySubscriptXSize	500	Subscript horizontal font size.

best fitting

Best fitting will choose the most appropriate representation of values, which is hexadecimal or decimal. This is the default.

USHORT	usWidthClass	5	Indicates a relative change from the normal aspect ratio.
SHORT	fsType	0×0004	Indicates font embedding licensing rights for the font.
SHORT	ySubscriptXSize	500	Subscript horizontal font size.

You will notice that with **best fitting**, which is the default, some values are decimal while others are hexadecimal – this choice is built into OTMaster.

This example is taken from the OS/2 *table entries.*



OpenType Font Tables

You are expected to know the OpenType specification, rudimentarily at least, and at least as regards the tables whose data you intend to adjust. It is not the task of this manual to reproduce or rephrase them here, though this cannot be avoided at times. ► *local* ► *www* What you will find in this chapter though is a brief summary of an OpenType font's structure, a list of common tables, with a few additional notes about things to consider: first, to make sure that data across tables is consistent, and second, to provide some OTMaster tricks for adjusting and extending tables.

As indicated in the > Inspecting Tables chapter, the structure of an OpenType font is quite simple. It starts with header information. First, an Offset Table whose sFNT version indicates whether the font is glyf-based (0x001000), CFF-based (string '0TTO') or a TrueType Collection (string 'ttcf'), and the number of tables in case of a TT/CFF-based OpenType font. Second, a Table Directory which points out where each table starts and how long it is. Header information are directly followed by the tables.

The OpenType specification provide the following list of tables:

I. Required tables

cmap	character to glyph mapping
head	font header
hhea	horizontal header
hmtx	horizontal metrics
maxp	maximum profile
name	naming table
OS/2	os/2 and Windows specific metrics
post	PostScript information

2. Tables in glyf-based OpenType fonts

cvt	control value table
fpgm	font program
glyf	glyph data
loca	index to location*
prep	CVT program

3. Tables in CFF-based OpenType fonts

CFF	PostScript font program (Compact Font Format)†
VORG	vertical origin

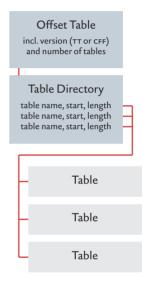
4. Tables for bitmaps

EBDT	embedded bitmap data
EBLC	embedded bitmap location data

EBSC embedded bitmap scaling data

Info: The OpenType specification. ► local ► www Especially see The OpenType Font File ► local ► www and Recommendations for OpenType Fonts ► local ► www which comes with additional information and clarifications.

An OpenType font's data structure (simplified):



* Location means: location of glyphs' data inside of the glyf table. Effectively, the loca is the glyf table's 'external' header or directory.

† The CFF table actually is a font in its own right, holding not only outline data, but also glyph- and font-level hinting information, font names and more.



5. Typographic layout tables

BASE	baseline data
GDEF	glyph definition data
GPOS	glyph positioning data
GSUB	glyph substitution data
JSTF	justification data

6. Other tables

DSIG	digital signature
gasp	grid-fitting and scan-conversion procedure
hdmx	horizontal device metrics
kern	kerning
LTSH	linear threshold data
PCLT	PCL5 data
VDMX	vertical device metrics
vhea	vertical metrics header
vmtx	vertical metrics

OpenType fonts' SFNT structure makes it easy to add tables which are not standardized by the OpenType specifications. The only requirement is that their names do not collide with standard table's names.

For example, Microsoft's VOLT – an application for defining typographic layout behavior visually– adds a table called TSIV to the font. This holds the VOLT project data. The TSIV data is removed from the font as soon as you 'ship' a font.

Selected Notes about OpenType Font Tables

Below, you will find some notes about individual OpenType font tables. These are in no way official recommendations but rather point out a few things to consider. You are strongly advised to compare with the original OpenType specification – links are provided whereever possible.

Tables not yet supported by OTMaster do appear in the table directory and table overview, but no content is shown in the table content area. When saving a font, these unsupported tables will be written back to the font in their original form.

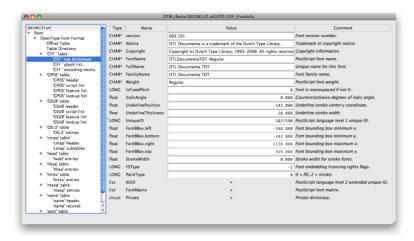
— AAT Tables

AAT (Apple Advanced Typography) tables are not supported.

Fopt Master™

— CFF

An OpenType font's Compact Font Format table is a complete font. It comes with outline data, but also with font names and (unless CID-keyed) glyph names, font- and glyph-wide hinting instructions, and more.



Adobe's The Compact Font Format Specification, #5176. ► local ► www Adobe's The Type 2 Charstring Format #5177. ► local ► www

Info: The CFF *table.* ► *local* ► *www*

Note: Adobe discourages the use of XUID and UniqueID in fonts that are not CJK fonts. **>** www

The 'CFF' top dictionary. The CFF table's Name INDEX shows up as FontName in this list.

I. 'CFF' top dictionary contains a CFF font's meta-information:

1.1 CFF and name table: The CFF table's Name INDEX – found in the 'CFF' top dictionary as **FontName** – must be identical to name table entries with nameID 6 (*PostScript* name, both Microsoft platform and Macintosh platform) and nameID 4 (*Full* name, Microsoft platform). If your font is a CFF-based OpenType font and you intend to change these names, then you need to change the 'CFF' top dictionary's **FontName** at first and only then can adjust the name table entries for *PostScript* name and *Full* name.

Other font name entries in the 'CFF' top dictionary may reflect the respective name table entries. Weight refers to weight only, not to style!

1.2 CFF and post table: Both the CFF and the post table have entries for underline position and thickness. CFF has UnderlinePosition and UnderlineThickness. post has underlinePosition and underline-Thickness. While both tables' underline thickness entries share the same value, those for underline position differ. The CFF table's Underline-Position is measured from the vertical center of the underline outline's height, or thickness. The post table' underlinePosition is measured from its highest point. Their relation is:

CFF.UnderlinePosition

= post.underlinePosition - (post.underlineThickness/2) Take care that **ItalicAngle** is identical with the post table's **italicAngle** and matches the hhea table's **caretSlopeRise** and **caretSlopeRun**. More in the ► *Consistency Checker* chapter.



1.3 Font-wide hinting information – **BlueValues**, **OtherBlues**, etc.– are found in the 'CFF' top dictionary's **Private** nested table:

Туре	Name		Value	Comment
st	BlueValues		+	Bottom alignment zone for baseline + top alignment zones.
#	Value			Comment
0	-17.000	no fur	ther information	
1	0.000	no fur	ther information	
2	458.000	no fur	ther information	
3	477.000	no fur	ther information	
4	683.000	no fur	ther information	
5	701.000	no fur	ther information	
st	OtherBlues		+	Additional bottom alignment zones.
st	FamilyBlues		+	Family standard BlueValues.
st	FamilyOtherB	ues	+	Family standard OtherBlues.
oat	BlueScale		0.000	Pixel size limit for overshoot suppression.
oat	BlueShift		7.000	Additional overshoot suppression control.
oat	BlueFuzz		1.000	Tolerance for horizontal stems within alignment zones.
oat	StdHW		48.000	Dominant width of horizontal stems.
oat	StdVW		83.000	Dominant width of vertical stems.
st	StemSnapH		+	Array of most common horizontal stem widths including StdHW.
st	StemSnapV		+	Array of most common verticall stem widths including StdVW.
ONG	ForceBold		θ	0 = false, 1 = true, i.e. interpreter may apply special emboldening techniques
ONG	LanguageGrou	q	θ	0 = Latin, Greek, 1 = Chinese ideographs and similar.
oat	ExpansionFac	tor	0.000	Size adjustment limit for counters in LanguageGroup 1.
oat	initialRandom	Seed	0.000	Random seed value (possibly used for eexec encryption?).

The **Private** dictionary which is accessed by way of the 'CFF' top dictionary – here displayed in a separate window.

2. 'CFF' glyph list holds the glyph names as found in the CFF table's CharStrings INDEX. Additional columns present, per glyph,

hAdv, the horizontal advance width,

left, the leftside bearing which is equal to xMin (smallest horizontal extension of the glyph),

bottom, the yMin (smallest vertical extension of the glyph),

right, the xMax (largest horizontal extension of the glyph),

top, the yMax (largest vertical extension of the glyph),

and **Comment** with Adobe's standard glyph names for these glyphs.

019X13T.otf	#	id	hAdv	left	bottom	right	top	Comment	1
Root • OpenType Font Format	0	.notdef	200	Θ	θ	0	Θ	.notdef, ISOAdobe or Expert or Expert Subset	
Offset Table	1	space	200	Θ	0	0	Θ	space, ISOAdobe or Expert or Expert Subset	
Table Directory ▼ 'CFF ' table	2	exclam	294	87	-10	209	693	exclam, ISOAdobe	
'CFF ' top dictionary	3	quotedbl	377	63	523	315	766	quotedbl, ISOAdobe	
'CFF ' glyph list 'CFF ' encoding vector	4	numbersign	751	54	19	693	665	numbersign, ISOAdobe	
▼ 'GPOS' table	5	dollar	597	78	-94	519	766	dollar, ISOAdobe	
'GPOS' header 'GPOS' script list	6	percent	769	40	-100	728	766	percent, ISOAdobe	
'GPOS' feature list	7	ampersand	891	65	-11	898	701	ampersand, ISOAdobe	

The 'CFF' glyph list. This is where you may change glyph names in CFFbased OpenType fonts. in glyf-based OpenType fonts, glyph names are found in the post table – unless this is a version 3 post table.



3. 'CFF' encoding vector links any of the characters covered in Adobe's Standard or Expert Encoding to glyph indices. This is read-only.

0019X13T.otf	#	id	Comment
Root OpenType Font Format	0	.notdef	Standard or Expert Encoding.
Offset Table	1	.notdef	Standard or Expert Encoding.
Table Directory CFF ' table	2	.notdef	Standard or Expert Encoding.
CFF table	3	.notdef	Standard or Expert Encoding.
'CFF ' glyph list 'CFF ' encoding vector	4	.notdef	Standard or Expert Encoding.
▼ 'GPOS' table	5	.notdef	Standard or Expert Encoding.
'GPOS' header	6	.notdef	Standard or Expert Encoding.
'GPOS' script list 'GPOS' feature list	7	.notdef	Standard or Expert Encoding.
'GPOS' lookup list) • (

The 'CFF' encoding vector.

Unlike the glyf table, the CFF table does not show outline coordinates. For editing outlines you may prefer to use the \triangleright *Glyph Editor*.

— cmap

The Character To Glyph Mapping table maps input Unicode codepoints (**Code**) to glyphs which are identified by glyph index (**GID**). It consists of one or more subtables. A standard OpenType font usually contains three of them (platformID-encodingID-format):

- o-3-4 (no platform–Unicode–format 4) I-o–[6] (Macintosh–Roman–format 6 or other)
- 3-1-4 (Microsoft–Unicode–format 4)

The last one is required for Windows. The second one is or may become increasingly obsolete.

To address supplementary Unicode codepoints beyond вмр, a font also needs a format 12 subtable (platform ID – encoding ID – format):

3-10-12 (Microsoft-Unicode UCS-4-format 12) Note that this must be present in addition to the format 4 subtable rather than instead of it. This too is required for Windows.

To add mapping information to a cmap subtable, click the + to access its entries, select one of them, choose **Edit > Grow** to duplicate the selected entry, and adjust Unicode codepoint and glyph index:

019X13T.otf	6	#	plat	ormID	encodingID	format	length	languageID	GlyphMapping	Comme
Root CopenType Font Format		0		0	3	4	754	0	+	Unicode; 2.0+ semantics, BMP only
Offset Table		1		1	Θ	Θ	262	0	+	Macintosh; Roman
Table Directory v 'CFF ' table		2		3	1	4	762	0	+	Microsoft; Unicode BMP only
'CFF ' top dictionary			#	Code	GID			Comment		
'CFF ' glyph list 'CFF ' encoding vector	U.	ĺ	0	0×08	20 1	space				
▼ 'cmap' table			1	0x08	21 2	exclam				
'cmap' header 'cmap' subtables	41		2	0×00	22 3	quotedbl				
▼ 'head' table	11		3	0x08	23 4	numbersign	,			
'head' entries ▼ 'hhea' table			4	0x08	24 5	dollar				
 'nnea' table 'hhea' entries 	¥		5	0×08	25 6	percent				
▼ 'hmtx' table	Ψ.	1								14 +

Note: You need to repeat this for every subtable if you want the according mappings to be covered by all of them.

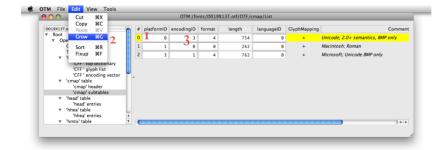
cmap table with three subtables, the last of which is unfolded.

It is possible to create a new subtables of a different format by

I. selecting an existing subtable,

2. duplicating it with Grow,

3. changing the new subtable's format and possibly platform ID and encoding ID too.

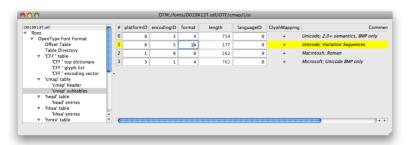


Duplicating a cmap subtable.



Info: The cmap *table.* ► *local* ► *www*

OTMaster supports Unicode Variation Sequences. With the method described above it is possible to add an according subtable. For this, platform ID – encoding ID – format need to be 0-5-14.



A Variation Sequence consists of a codepoint (**Code**) followed by a variation selector (**varSelector**). If this pair is matched in an input string, the glyph mapped to this codepoint will be replaced by another glyph referenced by **GID**. This can be compared to a typographic layout feature, yet it is the input string itself which initiates the replacement by a variant glyph. When creating such a subtable in OTMaster, glyphs' variation selectors are set to zero by default and need to be adjusted manually:

D019X13T.otf	0		plat	formID	enco	dingID	form	at	length	languageID	GlyphMapping		Comm
Root		0		Θ	_	3		4	754	9	+	Unicode; 2.0+ semantics, BMP	only
 OpenType Font Format Offset Table 		1		Θ	<u> </u>	5		14	177	9	+	Unicode; Variation Sequences	
Table Directory V CFF ' table			#	Coo	de	varSele	ctor	GID			Comment		
CFF table 'CFF top dictionary			0	0x00	9828	8×809	698	1	space, def	ault			
'CFF ' glyph list 'CFF ' encoding vector	U		1	0x00	9821	8×809	696	2	exclam, de	efault			
 creap' table 	ľ		2	0x00	9822	8×809	000	3	quotedbl,	default			
'cmap' header 'cmap' subtables			3	0x00	9823	8×869	000	4	numbersig	n, default			
 Thead' table 	ш		4	9×98	9824	8×869	000	5	dollar, def	ault			
'head' entries	ш		5	9×98	9825	8×869	000	6	percent, d	efault			
 'nnea' table 'hhea' entries 	¥.		6	9×98	9826	8×869	000	7	ampersand	l, default			
"hmtx' table	¥	1			_				-				1414

Further information about Ideographic Variation Sequences can be found in Ken Lunde's *Ivs Support* · *The Current Status and the Next Steps*. Also see the Unicode Consortium's *Ideographic Variation Database*. *www*



Note: See the cmap table specification, 'Format 14: Unicode Variation Sequence'. ► local ► www

Creating a subtable for Unicode Variation Sequences. Change the duplicate subtable's format, platform1D and encoding1D.

Adjust the glyph's variation selector (varSelector).

Tip: Use \uparrow and \downarrow keys to flip through all mapping entries quickly, and $\rightarrow i$ (to right) and SHIFT+ $\rightarrow i$ (or $i \leftarrow$; to left) to jump from column to column.



- COLR

- CPAL

The COLR table maps (usually encoded) base glyphs to color-specific glyphs and color definitions. The CPAL table holds these color definitions.

Info: The COLR table. ► local ► www The CPAL table. ► local ► www

seguiemj.ttf	8	264	+ uni21A	A : GID265
Root OpenType Font Format	9	266	+ uni231	A : GID267, GID268, GID2090, GID2146
Offset Table	#	GID	paletteIndex	Comment
Table Directory	0	267	1 (GID267, r=132 g=132 b=132 a=255
 COLR' table 'COLR' header 	1	268	2 (GID268, r=127 g=85 b=38 a=255
'COLR' base glyph records	2	2090	6 (GID2090, r=255 g=255 b=255 a=255
▼ 'CPAL' table	. 3	2146	0 0	GID2146, r=63 g=67 b=63 a=255
'CPAL' entries 'DSIG' table	10	269	+ uni231	B : GID270, GID271
V GDEF table	11	272	+ uni23E	9 : uni23E9
'GDEF' header	12	273	+ uni23E	A : uni23EA
'GDEF' glyph classes	13	274	+ uni23E	B : uni23EB
'GDEF' attachment points 'GDEF' ligature caret positions	14	275	+ uni23E	C : uni23EC
'GDEF' mark attachment classes	15	276	+ uni23E	D : uni23ED
'GDEF' mark set definitions				

The **COLR** *table with one nested* **GID** to **paletteIndex** *sublist unfolded*.

The COLR table maps base glyphs, identified by **GID**, to color-specific glyphs, also identified by **GID**, and references to color definitions, by **paletteIndex**.

eguiemj.ttf	Ту	pe	Nam	ne Va	alue		Comment
▼ Root	USHO	ORT	versio	n	Θ	Table ve	rsion number (0).
 OpenType Font Format Offset Table 	list		palett	es	+ 3	Set of co	olor palettes.
Table Directory	#	со	lorReco	ords			Comment
▼ 'COLR' table	0		+	arr	ay of 4	4 color	5
'COLR' header 'COLR' base glyph records		#	red	green	blue	alpha	
▼ 'CPAL' table		0	63	67	63	255	RGBA color values
'CPAL' entries	<u>^</u>	1	132	132	132	255	RGBA color values
'DSIG' table						1	
▼ 'GDEF' table		2	127	85	38	255	RGBA color values
'GDEF' header		3	168	168	168	255	RGBA color values
'GDEF' glyph classes 'GDEF' attachment points		4	255	253	194	255	RGBA color values
'GDEF' ligature caret positions		5	Θ	Θ	Θ	255	RGBA color values
'GDEF' mark attachment classes 'GDEF' mark set definitions		6	255	255	255	255	RGBA color values

The CPAL table in turn provides the **colorRecords**. A record's **#** column corresponds to the **paletteIndex** that the COLR table refers to. There may be multiple color **palettes** in the CPAL table, each of which contains a full set of **colorRecords** in a nested list. A color is defined by **red**, **green**, **blue** and **alpha** values.

A special **paletteIndex** value of oxFFFF indicates that the user-defined foreground color is to be used.

Both tables can be edited the usual way, as described in the previous section about the cmap table.

The CPAL table with a nested sublist containing the first palette's **colorRecords**.

Note: Other color tables are -- CBDT -- CBLC as supported by Google, -- sbix as supported by Apple, and -- sVG as supported by Adobe.

— DSIG

If the Digital Signature table exists in a font which you are about to edit in OTMaster you may want to delete it with **Edit>Cut**. Editing the font would invalidate this table anyway and indicate that this font is not in the state in which it was originally delivered.

—gasp

The Grid-Fitting And Scan-Conversion Procedure table determines which kind of rasterization is preferred for specific ppem size ranges.

Times New Roman.ttf	6	#	rangeMaxPPEM	rangeGaspBehaviour	Comment
 'gasp' table 'gasp' header 		0	8	0×0002	grayscale, ppem<=8
'gasp' ranges	0	- 1	17	0×0001	gridfit, ppem<=17
 glyf' table 'glyf' entries 	Ţ	2	65535	0×0003	gridfit, grayscale, ppem<=65535
'hdmx' table	Ŧ	C) (

The upper limit of a range is its maximum ppem size (**rangeGaspPPEM**), the lower limit is 0 or the previous range's maximum ppem size +1. The last record must have a **rangeGaspPPEM** of 0xFFFF or 65 535. Four kinds of preferred rasterization (**rangeGaspBehavior**) have been defined:

```
grayscale (0x0002)
gridfit (0x0001)
grayscale & gridfit (0x0003)
none of them (0x0000)
```

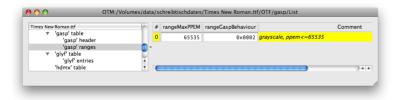
There are three additional values which address ClearType and are considered to be independent of the above ones:

```
symmetric gritfit (0x0004)
symmetric smoothing (0x0008)
```

symmetric smoothing & symmetric gritfit (0x000C)

Please note that in the head table, bit 13 of the **flags** entry indicates whether or not a font is optimized for ClearType.

For example, to adjust the above gasp table such that it will encourage grayscaling for all ppem sizes, select the first two records one by one and **Edit > Cut** them. Finally, change **rangeGaspBehavior** to **'0x0002'**.





Info: The DSIG *table.* ► *local* ► *www*

Note: It is assumed that you are either editing your own font or have permission to do so!

Info: The gasp *table.* ► *local* ► *www*

Selecting the last record of the gasp table.

The gasp table after removing the first two records and adjusting **rangeGaspBehavior**.



—glyf —loca

The glyf table contains TT outline information as well as glyph-level instructions, i.e. hinting information. This table is read-only in OTMaster, so we do not go into details here. Try the \blacktriangleright *Glyph Editor* for editing glyph outlines.

Info: The glyf table. \triangleright local \triangleright www The loca table. \triangleright local \triangleright www

CALIBRI.T		- A	# n	umberOfContours	xMin	yMi	in xMax	yMax	glyphData		Cor
	'OS/2' table 'OS/2' entries		0	5	6	-	397 978	1294	+	GIDO	
	'cmap' table		1	8	G		0 0	Θ	-	GID1	
	'cmap' header 'cmap' subtables		2	θ	6		0 0	0	-	GID2	
	'cvt ' table		3	θ	6		0 0	Θ	-	uni0020	
*	'cvt ' value list 'fpom' table		4	2	35		-6 1150	1300	+	uni0041	
*	'fpgm' instructions	110	Туре	Name		Value			Comm	ent	
	'gasp' table		USHOR	T numberOfCont	ours	2	Number of con	ntours. If r	egative, this	s is a composite glyph.	
	'gasp' header 'gasp' ranges	n ^	USHOR	T numberOfPoint	s	39	Number of poi	nts.			
Ψ.	'glyf' table		USHOR	T numberOfInstru	uctions	157	Number of ins	truction o	ocodes.		
	'glyf' entries 'head' table		list	Contours		+	Array of last p	oints of ea	ch contour.		
*	'head' entries		list	Points		+				to 0/0, next relative to	last one.
	'hhea' table		list	Instructions		+	List of instruct				
	'hhea' entries 'hmtx' table										
*	'hmtx' entries		5	-1	35		-6 1150	1689	+	uni00C0	
	'kern' table	¥.	6	-1	35		-6 1150	1689	+	uni00C1	
	'kern' header	Ŧ	7	-1	35		-6 1150	1689	+	uni00C2	
		•) 4 +

The loca table (not shown here) can be considered as the glyf table's directory or header. Per every glyph index, the loca table points out (by way of an 'offset') where this glyph's data starts in the glyf table.

Just like tables' header information are read-only, the loca table is readonly and will be updated automatically by OTMaster.

> — cvt — prep — fpgm

With glyf-based OpenType fonts (quadratic Bezier curve description in a glyf table), the Control Value Table, Control Value Program and Font Program hold font-level instructions. OTMaster indicates their meaning in the **Comment** column.

For details about the above tables and TrueType instructions please see Apple's TrueType specification. ► *www*



— GDEF

The Glyph Definition table belongs to the typographic layout tables and provides additional information to which GSUB and GDEF may refer.

Curently, OTMaster only supports glyph class definitions which are located in 'GDEF' glyph classes. Each glyph, identified by glyph index (GID), is associated with no or one of these classes, identified by **ClassValue**:

- I. base glyphs
- 2. ligature glyphs
- 3. (combining) mark glyphs
- 4. glyph components

iptoeKLTF-Bd.otf	#	GID	ClassValue	
Root	0	368	1	a, base glyph
 OpenType Font Format Offset Table 	1	381	1	b, base glyph
Table Directory	2	382	1	c, base glyph
'CFF ' top dictionary	3	508	2	f_f, ligature glyph
'CFF ' glyph list 'CFF ' encoding vector	4	509	2	f_t, ligature glyph
▼ 'GDEF' table	5	510	2	f_f_t, ligature glyph
'GDEF' header	6	557	3	uni0307, mark glyph
'GDEF' glyph classes 'GDEF' attachment points	7	558	3	uni0308, mark glyph
'GDEF' ligature caret positions	6			





— GPOS

— GSUB

Glyph Positioning and the Glyph Substitution tables start with three lists: script list, language list, and feature list.

D019X13		#	S	criptTag De	faultLangSys	LangSysCount	Lan	gSysList			Comment
▼ Root		0		'DFLT'	+	0		+	Defau	lt	
₹ 0	penType Font Format Offset Table	1	h	'cvrl'	+	0		+	Cvrilli	c	
	Table Directory	2		'grek'	+	0			Greek		
	'CFF ' table 'CFF ' top dictionary	3		'latn'	+	2			Latin		
	CFF top dictionary	3								-	
	'CFF ' encoding vector		#	LangSysTag		r ReqFeaturel		Feature	Count	FeatureIndex	
	'GPOS' table		0	'AZE '	-	65	535		1	+	Azeri
	'GPOS' header 'GPOS' script list		1	'CRT '	:	6.5	535		1	+	Crimean Tatar
	'GPOS' feature list		2	'DEU '	-	65	535		1	+	German (Standard)
	'GPOS' lookup list	-	3	'MOL '		65	535		1	+	Moldavian
	'GSUB' table 'GSUB' header		4	'ROM '	-	65			1	+	Romanian
	'GSUB' script list		5	'SRB '		61			1	+	Serbian
	'GSUB' feature list	J	6	'TRK '					1	+	Turkish
	'GSUB' lookup list 'OS/2' table		0		-	0.5			1	+	Turkish
	'OS/2' entries			# Index			Cor	nment			
	'cmap' table			0 10	'kern', Kerning						
	'cmap' header										
	'cmap' subtables 'head' table										
	'head' entries	1									
	'hhea' table	10	-)+

Info: The GPOS table. ► local ► www The GSUB table. ► local ► www Also see the OpenType Layout Common Table Formats document. ► local ► www

The script list, with unfolded language list for the 'latn' script.

The script list sums up all scripts explicitly addressed by the layout table. These are scripts – or writing systems – like 'latn' for Latin or 'cyrl' for Cyrillic.

Per each script as found in the **ScriptTag** column, there is a list of languages explicitly addressed by the layout table, plus a 'default' language for which there is a special place in the data structure. (In case that a layout application cannot find a match for the selected language – e.g. by way of the spelling dictionary in Adobe InDesign – in the font's layout table, it would fall back to this 'default' language.) The former is accessed by clicking the **+** in the **LangSysList** column, the latter by clicking the **+** in the **DefaultLangSys** column. In the example above, the 'latn' script's languages are folded out.

Per each language there is a list of features associated with it. This is unfolded by clicking the **+** in the according language's **FeatureIndex** column. In the example above, there is one 'kern' feature associated with the Turkish language or 'TRK' – to know what the actual feature behavior of 'kern' is for Turkish, we memorize feature **Index** 10 and switch to the feature list.

DTL OTMASTER: OT TABLES

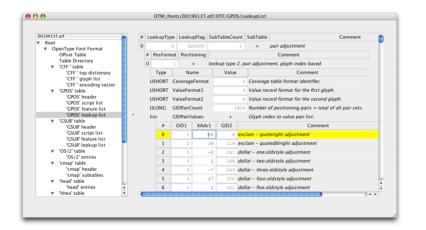


D019X13T.otf	#	FeatureTag	FeatureParams	LookupCount	LookupIndex		Comment
▼ Root	0	'kern'		1	+	Kernina	
 OpenType Font Format Offset Table 	1	'kern'	-	1	+	Kerning	
Table Directory	2	'kern'		1	+	Kerning	
 'CFF ' table 'CFF ' top dictionary 	3	'kern'	-	1	+	Kerning	
CFF top dictionary				1			
'CFF ' encoding vector	4	'kern'		1	+	Kerning	
▼ 'GPOS' table 'GPOS' header	5	'kern'	-	1	+	Kerning	
'GPOS' script list	6	'kern'		1	+	Kerning	
'GPOS' feature list	7	'kern'	-	1	+	Kerning	
'GPOS' lookup list v 'GSUB' table	^ 8	'kern'	-	1	+	Kerning	
GSUB table	9	'kern'		1	+	Kerning	
'GSUB' script list	10	'kern'		1	+	Kerning	
'GSUB' feature list 'GSUB' lookup list	#	Index		Co	omment		
▼ 'OS/2' table	0	0.4	ookup type 2, pa	ir adjustment			
'OS/2' entries			ionap type 2, pa	r uujustinent			
▼ 'cmap' table 'cmap' header 'cmap' subtables							
▼ 'head' table 'head' entries							
▼ 'hhea' table ▼	-) 4

The Feature list.

The Lookup list.

The feature list, for each feature to which a script/language combination refers, points to one or more lookups via **LookupIndex**. These lookups define the actual substitution and positioning behavior for this feature. Going back to our example, we remember feature index number 10 (the **#** column) and in this feature's **LookupIndex** column click the **+** which opens a nested table pointing to the relevant lookups' indices.



The lookup list, finally, holds a list of all lookups. These are presented as nested tables, and you can start unfolding nested tables which strictly represent the lookups' structure. In our example above, there is but a single lookup for kerning, of **LookupType** 2. We click the **+** in the **SubTable** column and get to know about the single subtable's format which is **PosFormat** 1. Clicking the **+** in the **Positioning** column unfolds the single subtable's header information. Clicking the **+** in the **GIDPairValues** row finally presents the kerning pairs and values.

Note: The structure of nested tables depends on lookup types and subtable formats – this example really is just that: an example.



With OTMaster you may view and edit user interface names associated with Stylistic Set I–20 ('sso1'–'ss20') and Character Variant I–99 ('cv01'–'cv99') features.

To examine the name of the 'sso9' feature of a given font, choose the 'GSUB' feature list from the table overview. Scroll down to 'sso9' in the content area – there are multiple entries for 'sso9', one per each script/ language combination. If there is a + in the **FeatureParams** column, then this feature is associated with an interface name. Click the + to reveal **Version** and **UINAMEID**. The latter is nothing else but a pointer to a nameID associated with one or more name table records.

GrotextKLTF-Rg.otf	#	FeatureTag	FeatureParams	LookupCount	LookupIndex		Comm
 Root OpenType Font Format 	114	'ss08'	+	1	+	Stylistic Set 8	
Offset Table	115	'ss08'	+	1	+	Stylistic Set 8	
Table Directory CFF ' table	116	'ss08'	+	1	+	Stylistic Set 8	
► 'GPOS' table	117	'ss09'	+	1	+	Stylistic Set 9	
 'GSUB' table 'GSUB' header 		Type Na	ime Value		Comment		
'GSUB' neader 'GSUB' script list	US	HORT Versi	on 0	Minor version n	umber (set to ()).	
'GSUB' feature list	US	HORT UINa	melD 256	'name' table nai	me ID for user-	interface label.	
'GSUB' lookup list ▶ 'OS/2' table	118	'ss09'	+	1	+	Stylistic Set 9	C C
▶ 'cmap' table	119	'ss09'	+	1	+	Stylistic Set 9	
 'head' table 'hhea' table 	120	'ss09'	+	1	+	Stylistic Set 9	
▶ 'hmtx' table	121	'ss09'	+	1	+	Stylistic Set 9	
 'maxp' table 'name' table 	122	'ss09'	+	1	+	Stylistic Set 9	
'name' header	123	'ss09'	+	1	+	Stylistic Set 9	
'name' records ▶ 'post' table	124	'ss10'	+	1	+	Stylistic Set 10	1
post table) 4 +

So keep this name ID in mind and choose 'name' records from the table overview. The content area shows, for name ID 256, that this Stylistic Set feature is called 'Arrows 2'. Please note that there may be multiple name records with name ID 256. As usual, there may be different strings for different platforms, encodings, languages.

0					
0	1	0	0	256	Arrows 2
1	1	Θ	θ	257	Arrows 1
2	1	θ	θ	258	Centered Comma
3	1	Θ	θ	259	Small Caps Numerals
4	1	Θ	θ	260	Short J
5	1	0	Θ	261	Schoolbook a d g l
6	1	0	Θ	262	Broken a g h m n u
7	1	0	0	263	Umlaut With E
8	1	0	Θ	264	Punctuation Spacing
9	3	1	1033	0	Copyright (c) Karsten Luecke, 2001, 2010. All rig
10	3	1	1033	1	Grotext KLTF Rg
11	3	1	1033	2	Regular
12	3	1	1033	3	1.800;KLTF;GrotextKLTF-Rg
13	3	1	1033		GrotextKLTF-Rg
	2 3 4 5 6 7 8 9 10 11	2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 3 10 3 11 3	2 1 0 3 1 0 4 1 0 5 1 0 6 1 0 7 1 0 8 1 0 9 3 11 10 3 11	2 1 0 0 0 3 1 0 0 0 0 4 1 0 0 0 0 5 1 0 0 0 0 6 1 0 0 0 0 7 1 0 0 0 0 9 3 1 1033 1 1033 10 3 1 1033 1 1033	2 1 0 0 258 3 1 0 0 259 4 1 0 0 260 5 1 0 0 260 6 1 0 0 262 7 1 0 0 263 8 1 0 0 263 9 3 1 1033 0 10 3 1 1033 1 11 3 1 1033 2

Locating **UIName ID** in the 'GSUB' feature list.

The string associated with this feature found in the name table.



-head

The Font Header table holds the most important value – the font's **unitsPerEm** or UPM, i.e. the number of units into which the em-square is divided. All of the fonts metrics relate to it.

	Type	Name	Value	Comment
'GPOS' header 'GPOS' script list	Fixed	version	0×08010808	0x00010000 for version 1.0.
'GPOS' feature list	Fixed	fontRevision	0x080219db	Set by font manufacturer.
'GPOS' lookup list CSUB' table	ULONG	checkSumAdjustment	0xf8430859	0xB1B0AFBA - (sum of entire font as ULONG).
'GSUB' header	ULONG	magicNumber	0x5f0f3cf5	Set to 0x5F0F3CF5.
'GSUB' script list 'GSUB' feature list	USHORT	flags	0×0003	Font flags (Bit 0 – 15).
'GSUB' lookup list	USHORT	unitsPerEm	1000	A power of 2 for TrueType outlines, 16 – 16384.
 'OS/2' table 'OS/2' entries 	LONGDATETIME	created	2008-Dez-09 11:36:16	Creation date/time, YYYY-MMM-DD hh:mm:ss.
v 'cmap' table	LONGDATETIME	modified	2008-Dez-09 16:02:58	Modification data/time, YYYY-MMM-DD hh:mm:ss.
'cmap' header	FWORD	xMin	-260	For all glyph bounding boxes.
'cmap' subtables ▼ 'head' table	FWORD	yMin	- 242	For all glyph bounding boxes.
'head' entries	FWORD	хМах	1130	For all glyph bounding boxes.
 'hhea' table 'hhea' entries 	FWORD	yMax	929	For all glyph bounding boxes.
▼ 'hmtx' table	USHORT	macStyle	0×0000	Macintosh style bits.
'hmtx' entries ▼ 'maxp' table	USHORT	lowestRecPPEM	3	Smallest readable size in pixels.
'maxp' entries	SHORT	fontDirectionHint	2	Left to right / right to left / mixed mode
▼ 'name' table 'name' header	SHORT	indexToLocFormat	Θ	0 for short offsets, 1 for long.
'name' records	SHORT	glyphDataFormat	0	0 for current format.
▼ 'post' table 'post' header				

macStyle relates to the OS/2 table's fsSelection (and usWeightClass and usWidthClass) and name table records with name1D 2.

— hhea

Many Horizontal Header table entries are not meant to be edited manually.

019X13T.otf	Type	Name	Value	Comment
'GPOS' header 'GPOS' script list		version	0×00010000	0x00010000 for version 1.0.
'GPOS' feature list	FWORD	ascender	766	Distance from baseline of highest ascender.
'GPOS' lookup list CSUB' table	FWORD	descender	-234	Distance from baseline of lowest descender.
GSUB table	FWORD	lineGap	0	Typographic line gap.
'GSUB' script list	UFWORD	advanceWidthMax	1164	Maximum advance width value in 'hmtx' table.
'GSUB' feature list 'GSUB' lookup list	FWORD	minLeftSideBearing	- 260	Minimum left sidebearing value in 'hmtx' table.
▼ 'OS/2' table	FWORD	minRightSideBearing	-261	Min(aw - Isb - (xMax - xMin)).
'OS/2' entries ▼ 'cmap' table FWORD	WORD xMaxExtent		Max(lsb + (xMax - xMin)),	
'cmap' header	SHORT	caretSlopeRise	1	Used to calculate the slope of the cursor (rise/run): 1 for vertical.
'cmap' subtables v 'head' table	SHORT	caretSlopeRun	6	0 for vertical.
'head' entries	FWORD	maxInterCharExpansion	8	unknown
"hhea' table 'hhea' entries	FWORD	maxInterCharCompression		unknown
▼ 'hmtx' table	FWORD	spaceCharMax		unknown
'hmtx' entries	FWORD	spaceCharMin		unknown
 'maxp' table 'maxp' entries 	USHORT	slopRatio		unknown
	SHORT	metricDataFormat	-	0 for current format.
'name' header 'name' records	USHORT	numberLongMetrics		Number of hMetric entries in 'hmtx' table.
▼ 'post' table	USHORT	numberLongMetrics	257	Number of nimetric entries in nintx table.
'post' header	Ŧ.			

caretSlopeRise and **caretSlopeRun** should be in tune with the post table's **italicAngle** value and, if this is a CFF-based OpenType font, also with the 'CFF' top dictionary's **ItalicAngle**.

ascender, descender, lineGap relate to OS/2 table's sTypoAscender, sTypoDescender, sTypoLineGap, usWinAscent, usWinDescent.

Info: The head table. ► local ► www

head table entries.

Info: The hhea *table.* ► *local* ► *www*

hhea table entries.

Note: For both italic/slant information and vertical font metrics see the > Consistency Checker chapter.



Info: The hmtx *table*. ► *local*

▶ www

-hmtx

The Horizontal Metrics table contains, per glyph, **advanceWidth** and **leftSideBearing**.

0019X13T.otf	#	advanceWidth	leftSideBearing	Comment
▼ 'GPOS' table 'CPOS' header	0	200	0	.notdef, ISOAdobe or Expert or Expert Subset
'GPOS' script list	1	200	Θ	space, ISOAdobe or Expert or Expert Subset
'GPOS' feature list 'GPOS' lookup list	2	294	87	exclam, ISOAdobe
▼ 'GSUB' table	3	377	63	quotedbl, ISOAdobe
'GSUB' header	4	751	54	numbersign, ISOAdobe
'GSUB' script list 'GSUB' feature list	5	597	78	dollar, ISOAdobe
'GSUB' lookup list	6	769	40	percent, ISOAdobe
 'OS/2' table 'OS/2' entries 	7	891		ampersand, ISOAdobe
▼ 'cmap' table	8	257		quoteright, ISOAdobe
'cmap' header	^ 0	350		parenleft, ISOAdobe
'cmap' subtables ▼ 'head' table	10	350		parenright, ISOAdobe
'head' entries	10			asterisk, ISOAdobe
'hhea' table 'hhea' entries		520		
▼ 'hmtx' table	12	549		plus, ISOAdobe
'hmtx' entries	13	257		comma, ISOAdobe
 'maxp' table 'maxp' entries 	14	405	66	hyphen, ISOAdobe
▼ 'name' table	15	228	52	period, ISOAdobe
'name' header 🛛 🍟	16	577	0	slash, ISOAdobe
'name' records v 'post' table	17	597	45	zero, ISOAdobe

The basic glyph metrics, advance width and left sidebearing, listed in the hmtx table.

In CFF-based OpenType fonts, hmtx table entries are read-only and cannot be edited.

DTL OTMASTER: OT TABLES

-kern

Theoretically speaking, the Kerning table is a remainder of the (pre-OpenType) TrueType format. Practically speaking, it is still required because many applications cannot read OpenType font's layout tables.

With OpenType fonts, kerning information better be provided by way of a kern feature in the GPOS table. This can deal with 'normal' left-to-right as well as right-to-left and vertical kerning. The specification even says – albeit strangely worded – that CFF-based OpenType fonts are not supposed to have a kern table at all.

Since some applications, even most popular ones, do not support typographic layout features (GSUB, GPOS etc.), unfortunately there is no way around adding a kern table to provide them with kerning information.

However, adding a kern table brings some inconveniences with it. Fonts are getting bigger and in turn require more kerning pairs. For a kern table to be recognized by Windows, it must contain a single subtable of format 0. Since the format 0 subtable's value which indicates the number of kerning pairs can be 65 535 at maximum (not to mention that the value which indicates the subtable's length is similarly restricted),* the number of kerning pairs that may go into such a subtable is limited – not enough to cover all pairs which would result from expanding class-based kerning as found in an OpenType font's GPOS 'kern' feature.

More about kern table editing in the > 'kern' Table Viewer chapter.



Info: The kern *table.* ► *local* ► *www*

* A detailed description has been posted to the OpenType List in 2008 by Ascender Corp.'s Joshua Hadley.



Info: The maxp *table*. ► *local*

▶ www

— maxp

With CFF-based OpenType fonts (cubic Bezier curve description in a CFF table), a Maximum Profile table of version 0.5 merely tells about the number of glyphs (**numGlyphs**) in a font.

00	OTM	:/fonts/D019	K13T.otf/OTF/r	naxp/Struct	
D019X13T.otf	Type	Name	Value	Comment	
 'maxp' table 'maxp' entries 	Fixed	version	0×00005000	0x00010000 for version 1.0.	
'name' header 🛛 🛣	- USHORT	numGlyphs	257	The number of glyphs in the font.	
'name' records					

The version 0.5 maxp table and its two entries.

With glyf-based OpenType fonts (quadratic Bezier curve description in a glyf table), a Maximum Profile table of version 1.0 holds additional information about glyf, cvt, prep and fpgm tables.

tc	12	Туре	Name	Value	Comment
a.c.	'cmap' header				0x00010000 for version 1.0.
	'cmap' subtable	Fixed	version		
۳	'cvt ' table 'cvt ' value list	USHORT	numGlyphs	22213	The number of glyphs in the font.
÷	'fpgm' table	USHORT	maxPoints	488	Maximum points in a non-composite glyph.
	'fpgm' instruction	USHORT	maxContours	77	Maximum contours in a non-composite glyph.
Ŧ	'gasp' table 'gasp' header	USHORT	maxCompositePoints	θ	Maximum points in a composite glyph.
	'gasp' ranges	USHORT	maxCompositeContours	0	Maximum contours in a composite glyph.
Ŧ	'glyf' table 'glyf' entries	USHORT	maxZones	1	1 if instructions do not use the twilight zone (Z0), else 2.
Ŧ	'head' table	USHORT	maxTwilightPoints	0	Maximum points used in Z0.
_	'head' entries 'hhea' table	USHORT	maxStorage	0	Number of Storage Area locations.
*	'hhea' entries	USHORT	maxFunctionDefs	32	Number of FDEFs.
Ŧ	'hmtx' table 'hmtx' entries	USHORT	maxInstructionDefs	θ	Number of IDEFs.
Ŧ	'loca' table	USHORT	maxStackElements	1024	Maximum stack depth.
	'loca' offsets	USHORT	maxSizeOfInstructions	303	Maximum byte count for glyph instructions.
Ŧ	'maxp' table 'maxp' entries	USHORT	maxComponentElements	θ	Maximum number of components referenced at "top level" for any composite glyph.
v	'name' table	USHORT	maxComponentDepth	0	Maximum levels of recursion; 1 for simple components.
	'name' header 🍒				
	'name' records 🔻				
		1			

More details about these information and to which tables they relate can be found in Apple's TrueType specification. ► www

The version 1.0 maxp table and its additional entries.



— name

The Naming table is essential for a font to be identified.

Info: The name *table*. ► *local* ► *www*

019X13T.otf	n.	#	platformID	encodingID	languageID	namelD	nameString	Comment
 'GPOS' table 'GPOS' header 		0	1	8	0	Θ	Copyright (c) Dutch Type Library, 1993-2008. All rights reserved	Macintosh; Roman; English; Copyright notice
'GPOS' script list		1	1	θ	0	1	DTL Documenta TOT	Macintosh; Roman; English; Font Family name
'GPOS' feature list 'GPOS' lookup list		2	1	8	0	2	Regular	Macintosh; Roman; English; Font Subfamily name
▼ 'GSUB' table		3	1	θ	Θ	3	D019X13T	Macintosh; Roman; English; Unique font identifier
'CSUB' header 'CSUB' script list		4	1	θ	0	4	DTL Documenta TOT	Macintosh; Roman; English; Full font name
'GSUB' feature list		5	1	8	0	5	Version 002.101	Macintosh; Roman; English; Version string
'GSUB' lookup list ▼ 'OS/2' table		6	1	8	Θ	6	DTLDocumentaTOT-Regular	Macintosh; Roman; English; Postscript name
'OS/2' entries		7	1	θ	Θ	7	DTL Documenta is a trademark of the Dutch Type Library.	Macintosh; Roman; English; Trademark
 'cmap' table 'cmap' header 		8	1	8	0	8	Dutch Type Library	Macintosh; Roman; English; Manufacturer Name
'cmap' subtables		9	1	8	0	9	Frank E. Blokland	Macintosh; Roman; English; Designer
'head' table 'head' entries		10	1	8	Θ	11	http://www.dutchtypelibrary.com	Macintosh; Roman; English; URL Vendor
▼ 'hhea' table		11	1	θ	0	13	By downloading, unpacking and/or installing DTL Font Software	Macintosh; Roman; English; License Description
'hhea' entries ▼ 'hmtx' table		12	1	8	0	14	http://www.dutchtypelibrary.nl/PDF/Licenses/DTL_FS_License.j	Macintosh; Roman; English; License Info URL
hmtx' entries		13	1	0	0	18	DTL Documenta TOT	Macintosh; Roman; English; Compatible Full
 'maxp' table 'maxp' entries 		14	1	8	0	19	The quick brown fox jumps over the lazy dog.	Macintosh; Roman; English; Sample text
▼ 'name' table		15	3	1	1033	Θ	Copyright (c) Dutch Type Library, 1993-2008. All rights reserved	Microsoft; Unicode BMP only; English (United States); Copyright notice
'name' header		16	3	1	1033	1	DTL Documenta TOT	Microsoft; Unicode BMP only; English (United States); Font Family name
'name' records ▼ 'post' table	Y	17	3	1	1033	2	Regular	Microsoft; Unicode BMP only; English (United States); Font Subfamily nar
'post' header	Ŧ	18	3	1	1033	3	D019X13T	Microsoft; Unicode BMP only; English (United States); Unique font identii

You may remove, edit or add new name entries. To add a name entry, select an existing one and click **Edit>Grow** to duplicate it. Adjust the platform ID, encoding ID, language ID, name ID and the name tag. Once you have finished adding name entries, click **Edit>Sort** which will reorder all name entries.

Pre-defined nameIDs are, according to the name table specification:

o. Copyright notice.

Family name. Each family, i.e. a collection of fonts which share the same *Family* name, is supposed to consist of no more than these four styles: 'Regular', 'Italic', 'Bold', 'Bold Italic', defined in the *Subfamily* name (2).
 Subfamily (or *Style*) name. For each family – defined by *Family* name (1) – this may be one of 'Regular', 'Italic', 'Bold', 'Bold Italic'. The OS/2 table's **fsSelection ►** *local* ► *www* and the head table's **macStyle** ► *local* ► *www* need to reflect the style defined by *Subfamily* name.

3. Unique Font Identifier. This could be a combination of Designer (9) or Manufacturer (8) name, Postscript name (6) and year, separated by ';'. 4. Full name. A combination of Preferred Family name (16) and Preferred Subfamily name (17) if present or – according to the specification – Family name (1) and Subfamily name (2). If the Microsoft platform Subfamily name (2) is 'Regular', the Full name should match this platform's Family name (1). 5. Version string. This should begin with 'Version [major].[minor]' whereby [major] and [minor] may be any number smaller than 65 535. A Version string could look like 'Version 1.500'. Additional information may be added and should be separated by ';'. The name table entries. Thanks to the **Comment** column, the numeric platformID, encodingID, languageID and nameID are self-explaining.

Tip: For details see the name table specification as well as the ► Consistency Checker. Additional information can be found in the MakeOTF User Guide which is part of the AFDKO download, especially the chapter New OS/2 Bits. ► local ► www Additional name table entries and the version 4 OS/2 table's fsSelection bits 7–9 relate to WFF's font selection model. See Microsoft's paper WFF Font Selection Model ► www for detailed information about wwsconformant families.

Note: The PostScript name (6) description mentions an additional condition which the Full name (4) needs to meet in CFF-based Open-Type fonts.



6. *PostScript* name. This is required for a PostScript interpreter to identify a PostScript language font corresponding to this OpenType font. There must be a record with nameID 6 for both (platformID-encodingID-languageID):

I-O-O (Macintosh) 3-I-I033 (Microsoft)

Both strings must be identical when translated to ASCII and not longer than 63 characters. Characters are restricted to ASCII codes 33–126 excluding any of '[](){<>/%'. Usually built like the *Full* name (4) – but without any spaces and with with a '-' between family and subfamily part of the string.

There is an additional requirement for CFF-based OpenType fonts: The *PostScript* name (6) for both Microsoft platform and Macintosh platform, the *Full* name (4) for the Microsoft platform as well as the CFF Name INDEX must be identical. OTMaster displays the CFF Name INDEX as **FontName** inside of the 'CFF' top dictionary, thereby deviating a bit from the CFF table's structure.

7. Trademark notice.

8. Manufacturer name.

9. Designer name.

10. Description of the typeface.

II. *Vendor URL*. This should include 'http://', 'ftp://' etc.

12. Designer URL. This should include 'http://', 'ftp://' etc.

13. *License* description. This is expected to be a summary of the terms rather than, as the specification puts it, 'legalese'.

14. *License Info URL*. A link to the full license text. This should include 'http://', 'ftp://' etc.

15. Reserved, set to zero.

16. *Preferred Family* name. If a family consists of more, or other, styles than 'Regular', 'Italic', 'Bold', 'Bold Italic', you may define a *Preferred Family* name for which there are no restrictions as to the number of styles.

17. Preferred Subfamily (or Style) name. This relates to the Preferred Family name (16). If the OS/2 table's version is 4 and if a family's Preferred styles are wws-conformant (i.e. are distinguished by the categories weight, width and slope alone), you may set the OS/2 table's **fsSelection** bit 8 to 1 which will signal wws-conformancy. If a family's Preferred styles are not wws-conformant, set bit 8 to 0 and provide wws-conformant *wws Family* name (21) and *wws Subfamily* name (22).

18. Compatible Full name. Macintosh platform only. If the Full name (4) turns out to be too long, i.e. longer than 27 (or more generously: 31) characters, you may either abbreviate the Macintosh platform Full name (4) or otherwise provide an additional abbreviated Compatible Full name. If the Compatible Full name is present and the Subfamily name (2) is 'Regular', the Compatible Full name should match the Family name (1).
19. Sample text.

20. PostScript CID findfont name. Please see the name table specification.

Note: If you set the OS/2 table's version to 4, then all the fsSelection bits 7–9 need to be set to 0 or 1 consciously because starting with this table version, these bit settings bear a special meaning.

DTL OTMASTER: OT TABLES

21. *wws Family* name. A wws-conformant family's styles must address no other categories than weight, width and slope. If this name is provided, the OS/2 table's **fsSelection** bit 8 should be set to 0.

22. wws Subfamily (or Style) name. If this name is provided, the OS/2 table's **fsSelection** bit 8 should be set to 0. The OS/2 table's **usWeightClass**, **usWidthClass** as well as **fsSelection** bits 0 (italic) and 9 (oblique) should reflect the wws Subfamily name.

The name table specification \blacktriangleright *local* \triangleright *www* comes with an example string for each nameID, so that you are encouraged to read this documents carefully.

Some OpenType layout features like Stylistic Set and Character Variant features can be associated with strings that applications may display in their user interfaces. These strings are located in the name table and referenced from the GSUB table. Please see the \triangleright *GPOS/GSUB* section for an example.



Note: A wws-conformant font may have both Italic and Oblique styles! See Microsoft's paper wPF Font Selection Model. ► www

Note: If you set the OS/2 table's version to 4, then all the fsSelection bits 7–9 need to be set to 0 or 1 consciously because starting with this table-version, these bits bear a special meaning.



-os/2

Most font-wide information are located in the OS/2 table. The content area's **Comment** column provides a description for every entry.

019X13T.otf	Type	Name	Value	Comment
Root ▼ OpenType Font Format	USHORT	version	0×0003	OS/2 table version number.
Offset Table	SHORT	xAvgCharWidth	556	Average weighted escapement.
Table Directory	USHORT	usWeightClass		Indicates the visual weight.
 'CFF ' table 'CFF ' top dictionary 		usWidthClass		Indicates a relative change from the normal aspect ratio
'CFF ' glyph list	SHORT	fsType		Indicates font embedding licensing rights for the font.
'CFF ' encoding vector ▼ 'GPOS' table	SHORT	ySubscriptXSize		Subscript horizontal font size.
'GPOS' header	SHORT	ySubscriptYSize		Subscript vertical font size.
'GPOS' script list	SHORT	ySubscriptXOffset		Subscript vertical font size.
'GPOS' feature list 'GPOS' lookup list		ySubscriptYOffset		
▼ 'GSUB' table	SHORT			Subscript y offset.
'GSUB' header 'GSUB' script list		ySuperscriptXSize		Superscript horizontal font size.
'GSUB' feature list	SHORT	ySuperscriptYSize		Superscript vertical font size.
'GSUB' lookup list VOS/2' table	SHORT	ySuperscriptXOffset		Superscript x offset.
 'OS/2' table 'OS/2' entries 	SHORT	ySuperscriptYOffset		Superscript y offset.
▼ 'cmap' table	SHORT	yStrikeoutSize		Width of the strikeout stroke.
'cmap' header 'cmap' subtables	SHORT	yStrikeoutPosition		The position of the top of the strikeout stroke.
▼ 'head' table	SHORT	sFamilyClass	8×0800	Font-family class and subclass.
'head' entries	CHAR	panose.bFamilyType	0×00	Panose family type.
'hhea' table 'hhea' entries	CHAR	panose.bSerifStyle	0×00	Panose serif style.
▼ 'hmtx' table	CHAR	panose.bWeight	0×00	Panose weight.
'hmtx' entries v 'maxp' table	CHAR	panose.bProportion	0×00	Panose proportion.
'maxp' entries	CHAR	panose.bContrast	0×00	Panose contrast.
 'name' table 'name' header 	CHAR	panose.bStrokeVariation	0×80	Panose stroke variation.
'name' records	CHAR	panose.bArmStyle	0×00	Panose arm style
▼ 'post' table	CHAR	panose.bLetterform	0×80	Panose letterform.
'post' header	CHAR	panose.bMidline		Panose midline.
	CHAR	panose.bXHeight		Panose x-height.
	ULONG	ulUnicodeRange1		Unicode Character Range (Bits 0 - 31).
	ULONG	ulUnicodeRange2		Unicode Character Range (Bits 32 – 63).
	ULONG	ulUnicodeRange3		Unicode Character Range (Bits 64 – 95).
	ULONG	ulUnicodeRange4		Unicode Character Range (Bits 96 – 127).
		achVendID		Font Vendor Identification.
		fsSelection		
		usFirstCharIndex		Font selection flags. Minimum Unicode index.
		usLastCharIndex		Maximum Unicode index.
	SHORT	sTypoAscender		Typographic ascender.
	SHORT	sTypoDescender		Typographic descender.
	SHORT	sTypoLineGap		Typographic line gap.
		usWinAscent		Ascender metric for Windows.
	USHORT	usWinDescent	250	Descender metric for Windows.
	ULONG	ulCodePageRange1	0×20000001	Code Page Character Range (Bits 0 - 31).
	ULONG	ulCodePageRange2	0×00800880	Code Page Character Range (Bits 32 -63).
	SHORT	sxHeight	458	Height of lowercase letters.
	SHORT	sCapHeight	683	Height of uppercase letters.
	USHORT	usDefaultChar	8×0000	Unicode of default character for Windows.
	USHORT	usBreakChar	0×0020	Unicode of word separating character for Windows.
	USHORT	usMaxContext	3	Maximum length of target glyph context for features.
	_			

usWeightClass, **usWidthClass** and **fsSelection** relate to the name table records with nameIDs I/2, I6/17, 2I/22, while **fsSelection** also relates to the head table's **macStyle**. Please see the **>** *name* table section.

fsType defines embedding licensing rights for this font. May it be embedded into a document? What, then, is allowed with such a document?

sTypoAscender, **sTypoDescender**, **sTypoLineGap**, **usWinAscent**, **usWinDescent** − and the hhea table's **ascender**, **descender**, **lineGap** − are dealt with in the ► *Consistency Checker* chapter. *Info: The* OS/2 *table.* ► *local* ► *www*



If you set the OS/2 table's version to 4, you need to set **fsSelection** bits 7-9 to 0 or 1 consciously because starting with this table version, each of these bits carries a special meaning:

7. To tell applications that the default line-to-line distance should be calculated from **sTypoAscender/sTypoDescender/sTypoLineGap**, set this bit to 1. Otherwise, set this bit to 0. See the ► *Consistency Checker* chapter for more information about setting vertical font metrics.

8. If this family is wws-conformant (the name table *Preferred Subfamily* names (17) address no other categories than weight, width and slope), set this bit to 1. If this is not so (e.g. if *Preferred Subfamily* names (17) refer to optical size or even something like 'Outline'), set this bit to 0 but add wws-conformant *wws Family* name (21) and *wws Subfamily* name (22).

9. If this font's slope can be described as 'Oblique' rather than 'Italic'–wpF distinguishes between 'Oblique' and 'Italic'–, set this bit to 1. Otherwise, set this bit to 0. (If the font is 'Italic', set bit 0 to 1 as usual.)

If you set the OS/2 table's version to 5, you will notice two additional entries. These allow to create optical size specific fonts and define the intended size range for which a font was designed. Please note that these entries interact with name table records. Microsoft, who came up with this, has not published a specification about the details yet.

USHORT version	0×0005	OS/2 table version number.
USHORT usLowerPointSize	θ	Lower value of size range for multiple optical styles.
USHORT usUpperPointSize	65535	Upper value of size range for multiple optical styles.

usLowerPointSize is the lower value of the size range at which the font is to be used while **usUpperPointSize** is the upper value of the size range at which the font is to be used. One font's **usLowerPointSize** of one optical size is expected to be identical to another font's **usUpperPointSize** since the latter is meant to be exclusive (up to, but not including, the stated point size).

One unit corresponds to 0.05 pt, or 1 twip.

For font families that do not feature special designs for separate point size ranges, **usLowerPointSize** should be o and **usUpperPointSize** should be 65535 which is an infinitely large size.

Tip: See Microsoft's paper WPF Font Selection Model, *especially pp. 4–6 and the 'Guidelines for font manufacturers' on pp. 17–18.* ► www

Info: Preliminary information on these new values by John Hudson.

Note: usLowerPointSize nad usUpperPointSize unit is a 'twip' which equals 0.05 pt.



-post

The post table holds information for printing fonts on PostScript printers. glyf-based OpenType fonts store glyph names in a post table of version 2.0. CFF-based OpenType fonts, unless they are CID-keyed, store glyph names in the CFF table. In this case, the post table has format 3.0 which signifies an abbreviated version of this table that omits glyph name information. It is possible for glyf-based OpenType fonts, too, to make use of a post table of version 3.0. In this case, the font does not contain glyph names at all.

Changing the **formatType**, i.e. the table's version number, from 2 to 3 will remove glyph names from the table. Changing it from 3 to 2 will add glyph names (placeholders, actually).

D019X13T.otf	6	Type	Name	Value	Comment
'cmap' subtables table	- 11	Fixed	formatType	0x08030000	0x00010000 for version 1.0.
'head' entries	- 11	Fixed	italicAngle	0.000	Italic angle in counter-clockwise degrees from the vertical.
 'hhea' table 'hhea' entries 	- 11	FWORD	underlinePosition	-133	Distance top of underline from baseline (negative = below baseline).
▼ 'hmtx' table		FWORD	underlineThickness	20	Suggested value for the underline thickness.
'hmtx' entries vimaxp' table		[^] ULONG	isFixedPitch	θ	Set to 0 if the font is proportionally spaced.
'maxp' entries		ULONG	minMemType42	0	Minimum memory usage when downloaded.
Iname' table 'name' header		ULONG	maxMemType42	0	Maximum memory usage when downloaded.
'name' records		ULONG	minMemType1	0	Minimum memory usage when downloaded as a Type 1.
▼ 'post' table	1	ULONG	maxMemTvpe1	A	Maximum memory usage when downloaded as a Type 1.

The **italicAngle** value should be consistent with the **ItalicAngle** in 'CFF' top dictionary (if this is a CFF-based OpenType font) and with the hhea table's **caretSlopeRise** and **caretSlopeRun**. More in the > *Consistency Checker* chapter.

If the font is a CFF-based OpenType font, make sure **underlinePosition** and **underlineThickness** are consistent with the 'CFF' top dictionary's **UnderlinePosition** and **UnderlineThickness**. The underline position is calculated differently in both tables, please see the **>** CFF section for details.

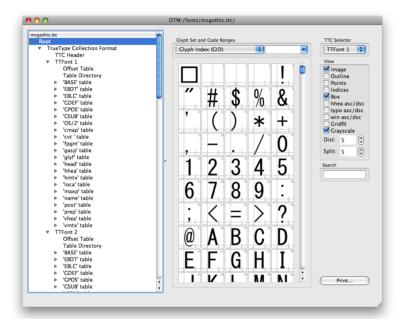
Info: The post table. ► *local* ► *www*

A version 3.0 post table.



ттс and отс Fonts

A TTC (TrueType Collection) font contains more than one sub-font. A TTC font's TTC Header points to each sub-font's header information consisting of Offset Table and Table Directory which points to tables associated with this sub-font. (Version 2.0 of the TTC Header will also reference a common DSIG table.) These then are followed by all tables. This allows a TTC's sub-fonts to 'share' tables that are identical, and –which is the example given in the OpenType specification – even use a single glyf table holding all glyphs of all sub-fonts, yet each sub-font's loca table only refers to glyphs relevant for this sub-font.

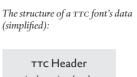


With TTC fonts, OTMaster's table overview starts with the Root entry. Root holds TrueType Collection Format which in turn holds TTFont 1, TTFont 2, TTFont 3, etc., as many as there are sub-fonts in the TTC font. Each TTFont (like the OpenType Font Format) holds an Offset Table and Table Directory, followed by all tables found in the sub-font.

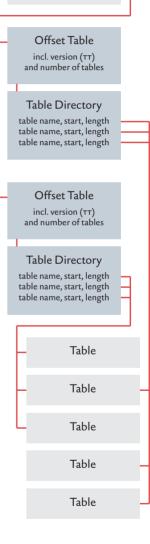
OTMaster lists, for each sub-font, all referenced tables – even if these are 'shared' by multiple sub-fonts. Once a 'shared' table is adjusted for a single font, this table will be duplicated.

OTMaster supports OTC (OpenType Collection) fonts too. An OTC is essentially the same as a TTC except that it is CFF table based rather than glyf table based.

Also see The OpenType Font File. ► local ► www



incl. version (ттс), number of sub-fonts and sub-fonts' start points





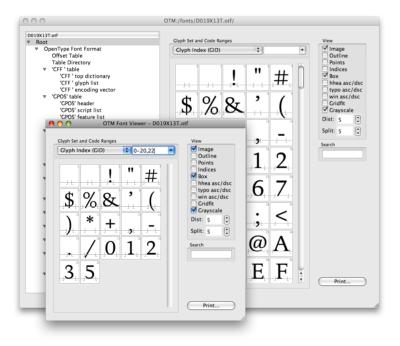
OTMaster's Toolbox

OTMaster is not only good for reviewing and editing a font's 'raw' tables but comes with tools for higher-level – and thus more user-friendly– access to some font data. Some of these tools do not show a specific table's content but represent data in a thematic/categorized way: The Font Viewer shows all, or a selection, of a font's glyphs. The Glyph Viewer shows all data related to an individual glyph, thereby combining data from various tables like glyf or CFF for outlines, post or CFF for glyph names, cmap for Unicode codepoints, hmtx and/or vmtx for metrics, etc.

TOOLS MENU

Font Viewer

The table overview's Root and the Font Viewer alike give a visual summary of a font's glyph set:

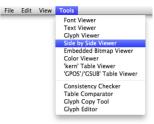


- Glyph Set and Code Ranges

These define which part of the glyph set to show, and how:

The leftside popup menu offers categories for defining glyph sub-sets by **Glyph Index (GID)**, **Unicode**, or any cmap subtable present in the font. The category also determines how glyphs are sorted in the overview.

The rightside textbox allows you to restrict the glyph overview to selected glyphs or glyph ranges. These are defined by GID or Unicode codepoint, depending on the category selected in the popup to the left.



When selecting Root, the content area shows the same glyph overview as does the Font Viewer.

Tip: Use the **Search** field to quickly locate glyphs by glyph index. The viewer will jump to this glyph and highlight it.

Glyph indices are expected to be integer numbers (0, 1, 123). Unicode codepoints are expected to be hexadecimal numbers, without leading zeros but preceded by '0x' (0x31, 0xB5).

You can provide a single identifier (0x20), a comma-separated list of identifiers (0x20, 0x61), a range delimited by first and last identifier (0x20-0x61), or a combination thereof (0x20, 0x40-0x61, 0xB5).

Confirm with RETURN.

The text input box will remember previous glyph sub-sets, and you may get back to them by using the boxes' popup functionality – by clicking on the popup menu's arrow button or by placing the cursor inside the box and using \uparrow or \downarrow to jump to any previous glyph sub-set.



-View

Image shows filled shapes.
Outline adds a colored outline.
Points shows on-curve points.
Indices shows points' index numbers.
Box draws the bounding box around each glyph.
hhea asc/dsc are the hhea table's ascender/descender heights.
typo asc/dsc are the OS/2 table's sTypoAscender/sTypoDescender height.
win asc/dsc are the OS/2 table's usWinAscent/usWinDescent heights.
Gritfit will align points, at current ppem size, to the pixel grid.
Grayscale draws grayscaled outlines.
Dist. is the distance between bounding boxes. With a value of 0, bounding boxes will touch.
Split determines the number of glyphs per line.

A glyph index is shown in the top left corner of each glyph's bounding box.

A double-click on a glyph in the Font Viewer glyph overview, or in the Root content area, will +

SHIFT+ double-click on a glyph will open it in the ► *Glyph Editor*.

Note: Since OTMaster uses the FreeType rasterizer, Grayscale may produce results that differ from 'real world' output, especially at small ppem sizes.



Text Viewer

The Harf Buzz-based Text Viewer allows testing a font's OpenType Layout features. The result is displayed in the main field.

Гір:	You may open more than one
Text	Viewer at the same time.

Script:	'latn', Latin		\$	
anguage:	<default></default>		\$	
Feature	State	Description	Comment	
'c2sc'	🔵 off 💿 default 🔵 on	Small Capitals From Capitals	off by default	
'calt'	🔵 off 💿 default 🔵 on	Contextual Alternates	on by default	
'case'	🔵 off 💿 default 🔵 on	Case-Sensitive Forms	off by default	
'ccmp'	🔵 off 💿 default 🔵 on	Glyph Composition / Decomposition	on by default	
'cpsp'	🔾 off 💿 default 🔾 on	Capital Spacing	off by default	
'dnom' 'falt' 'frac' 'kern'	Please s	oth Text Viewer - CALIF		
'falt' 'frac' 'kern' 'liga' 'lnum'		elect a prope		
'falt' 'frac' 'kern' 'liga' 'lnum' 'numr'	Please so	elect a prope		
'falt' 'frac' 'kern' 'liga' 'lnum' 'numr' 'salt'	Please so			
'falt' 'frac' 'kern' 'liga' 'lnum' 'numr'	Please so	elect a prope	er sam	ple tex
'falt' 'frac' 'kern' 'liga' 'lnum' 'numr' 'salt' 'sinf'	Please so Font: Open Featu Text: Open Edit.	elect a prope	e Reload	iple tex
'falt' 'frac' 'kern' 'liga' 'lnum' 'numr' 'salt' 'sinf' 'smcp'	Please so Font: Open Featu Text: Open Edit	elect a prope	e Reload	iple tex
'falt' 'frac' 'kern' 'liga' 'lnum' 'numr' 'salt' 'sinf' 'smcp' 'sso1'	Please so Font: Open Featu Text: Open Edit	elect a prope	e Reload	iple tex

The Text Viewer (foreground window) along with the Feature Selector (background window).

Tip: Drag & drop characters, i.e. encoded glyphs, from OTMaster's main dialog or from the Font Viewer into the Text Viewer to append these characters at the end of the sample text!

-Font

By default, the font currently selected in the Font Viewer will be displayed. **Open ...** a font not yet loaded in OTMaster. Activate or deactivate individual OpenType Layout features in the Feature Selector. Open this selector via **Features ...** Or select one of the fonts already loaded in OTMaster.

-Text

Open ... a sample text file.

Edit or enter sample text in the Text Editor. Open this editor via Edit ...

Print ... the content of the Text Viewer's main field.

Close the sample text file that you have opened before.

Reload the sample text file that you have opened, e.g. after having edited (but not saved) the sample text in the Text Editor.

Markings shows glyph boundaries. You may adjust the Size [ppem] as well as Line Spacing [%], and choose from writing directions left to right or right to left.

Winding Fill chooses the non-zero winding rule over the (default) even-odd rule for determining glyphs' fill.

Note: Close and Reload have an effect only if a sample text file has been opened before.



Click on a character to open its glyph in the Character Metrics dialog. (See below.)

SHIFT+ click on a character to open its glyph in the > *Glyph Editor*.

-Feature Selector

Script allows you to select a script whose OpenType Layout features you would like to test.

Language offers all languages available for the selected script. The subsequent table lists all features available for the selected script and language, each with the usual **Feature** tag, the **State** of selection (**off**, **default**, **on**), a **Description** of the feature's intended behavior, and a **Comment** that reports this feature's default state.

Finally, you can **Reset** script, language and features' states to **default**.

Please select a proper sample text file.	Open
	Reload
	Load
	Detach
	Attach
	Save
	Save As
	Close
	Apply to All
	Apply to
	Apply
	Cancel

Note: Each Feature Selector's settings affect only the Text Viewer from within which the selector has been opened.

Note: Please keep in mind that the content of the Text Editor reflects the content of a file that you have Open...ed before and will eventually Close again. It is this file's content that you are editing. That is, unless you Detach the content from the file, in which case you will need to Save the content in a new file, or unless you have never opened a file in the first place.

- Text Editor

This little but full-fledged Text Editor can be used

I. to enter some ad hoc sample text which remains independent of any file. Apply or Apply to All so that changes become effective in the respective Text Viewer or in all Text Viewers.

2. to edit a sample text file that you have opened via the Text Viewer.

3. to **Open ...**, edit, and then **Save** or **Save As ...** any arbitrary text file, irrespective of whether you will use it as sample text in the Text Viewer.



- Character Metrics

The Character Metrics dialog offers a glyph's **Glyph Index (GID)**, or **Unicode** codepoint as well as **left side bearing**, **advance width**, and **right side bearing**.



Since OTM 6, you may also select a glyph right in the Text Viewer to adjust the LSB (via the white arrow), the outline's position (via the black arrow) while keeping the glyph's width intact, or the RSB (via the red arrow). Alternatively, you may adjust **LSB**, **AW** [advance width], and **RSB** in the numeric fields that appear as soon as you select a glyph:

Please select a proper sample text
Glyph Index (GID): 286 Unicode: 0x0065 LSB 93 AW 1019 RSB 98 Font: Open Features /fonts/CALIBRI.TTF ‡
Text: Open Edit Clear Print Close Reload
Size [ppem]: 52 2 Line Spacing [K]: 30 2 left to right + Markings Winding Fill



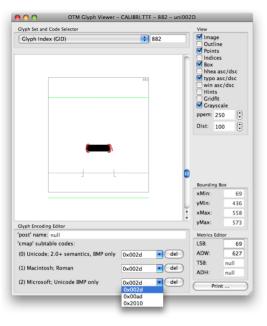
Glyph Viewer

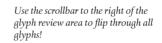
Per glyph, the Glyph Viewer interface gives access to all glyph-related data.

- Glyph Set and Code Selector

In the left popup menu choose whether to identify a glyph by **Glyph Index** (GID) or **Unicode** codepoint. In the right text input box, define a glyph's glyph index or codepoint.

Glyph indices are expected to be integer numbers (0, 1, 123). Unicode codepoints are expected to be hexadecimal numbers, without leading zeros but preceded by '**0**x' (0x31, 0xB5).





-View

Image shows filled shapes.

Outline adds a colored outline.

Points shows on-curve points.

Indices shows points' index numbers.

Box draws the bounding box around each glyph.

hhea asc/dsc are the hhea table's ascender/descender heights.

typo asc/dsc are the OS/2 table's sTypoAscender/sTypoDescender height.

win asc/dsc are the OS/2 table's usWinAscent/usWinDescent heights.

Hints shows hints if existing.

Gritfit will align points, at current ppem size, to the pixel grid.

Grayscale draws grayscaled outlines.

ppem is the ppem size at which the glyph is rasterized.

Zoom allows you to scale the rasterized image up or down which makes it possible to see the actual rasterization in detail.

Note: Since OTMaster uses the FreeType rasterizer, Grayscale may produce results that differ from 'real world' output, especially at small ppem sizes.

Tip: If a font's UPM (units per em) is 1000 and if you choose 160 ppem, then the 1000 units will be scaled such that they fit into 160 pixels upon glyph rasterization. The effect of this can be studied by increasing the Zoom factor.

DTL OTMASTER: TOOLS MENU



- Glyph Encoding Editor

The first entry is the glyph name, which is called **I. 'CFF' id** with CFF-based OpenType fonts ('id' because in CID-keyed fonts this would not be a name but a mere index value), and

2. 'post' name with glyf-based OpenType fonts.

'cmap' subtable codes shows as many entries as there are subtables in the cmap table. For each subtable there is a popup text box which may contain none, one or more Unicode codepoints. Press the **DEL** key to delete the respective subtable's codepoints.

Changes in this area will be reflected in 'cmap' table and vice versa.

-Bounding Box

xMin is the smallest horizontal extension of the glyph.yMin is the smallest vertical extension of the glyph.xMax is the largest horizontal extension of the glyph.yMax is the largest vertical extension of the glyph.

-Metrics Editor

LSB is the left sidebearing which is equal to xMin.
ADW is the advance width.
TSB is the top bearing.
ADH is the advance height.
The right sidebearing is implicit and can be calculated as

RSB = ADW - xMax

TSB and **ADH** relate to vertically oriented glyphs whose origin point, in CFF-based OpenType fonts, is at the top center of a glyph's bounding box. An illustrations for this can be found in Adobe's AFM specifications, p. II, fig. I. ► *www*

Tip: Make sure that you provide or adjust codepoints for all cmap subtables, not just one ...

Note: In CFF-based OpenType fonts, these values relate to smallest or largest extensions of a glyph regardless if there are on-curve extremum points or not. In glyfbased OpenType fonts, these values relate to coordinates of outermost on-curve or off-curve points.



Side by Side Viewer

000 OTM Side by Side Viewer - 0x0054 - T Move Left Move Right Show Hidden Right Rese ACaslonPro-Regular.otf ACaslonPro-Italic.otf ACaslonPro-Bold.otf Code Ranges: 0x0000 codes mapped in any font Code Set: ۵ Display Options: 🗹 Image 🗹 Outline 🗌 Winding Fill

glyphs from all fonts that are currently opened in OTMaster.

The Side by Side Viewer shows, for a given character, the corresponding

Use the scrollbar to the right of the review area to navigate through the available or selected range of characters.

As with other viewers too, use the scrollbar to navigate through the character set.

Code Ranges allows you to select a range of characters offered for review by stating single, or ranges of, Unicode codepoints.

Code Set helps you narrow down the selection of characters offered for a visual comparison: codes mapped in any font shows characters even if they do not exist in one or more of the fonts, codes mapped in all fonts shows characters only if present in all fonts, codes mapped in [...] but unmapped in other fonts shows all characters present in the according font even if they do not exist in one or more of the other fonts. Display Options are: Show a filled glyph Image. Show a glyph's Outline. Determine whether fill, i.e. what is black and white in the glyph image, is calculated using the non-zero Winding Fill rule. ▶ www If the latter option is deactivated, fill is calculated using the even-odd rule. ▶ www

A right-click into a specific glyph's box brings up a context menu whose functions help rearranging the displayed glyph. It can be moved to the left or right or can be hidden, and shown again. The manual arrangement can be reset to restore the default arrangement.

A double-click into a glyph's box will open it in the > *Glyph Viewer*.

A **SHIFT**+click into a glyph's box will open it in the ► *Glyph Editor*.

Unicode codepoints are expected to be hexa-decimal numbers, without leading zeros but preceded by '**0x**' (0x31, 0xB5).

You can provide a single identifier (0x20), a comma-separated list of identifiers (0x20, 0x61), a range delimited by first and last identifier (0x20-0x61), or a combination thereof (0x20, 0x40-0x61, 0xB5). Confirm with **RETURN**.



Embedded Bitmap Viewer

The Embedded Bitmap Viewer visualizes bitmaps from EBLC and EBDT or other bitmap tables.

		OTM Bitma	ap View	ver – msgothic.tt	c		
					0	TTC Selector	
			43 (688	048)	×	TTFont 1	\$
						View	
						Tables:	
						EBLC/EBDT	A T
						Sizes:	
						12 x 12	+
						Glyphs:	-
						Giypiis.	
						43	
							:
) <u>4</u> ¥	43	:
Glyph Metrics			Ŀ			43 Zoom:	
Glyph Metrics width	6	height	12			43 Zoom: 20	
	6	height horiBearingY	12 10	horiAdvance		43 Zoom: 20	

The Embedded Bitmap Viewer. Use the scrollbar to the right of the glyph review area to flip through all glyphs!

—View

Tables allows you to select which tables' bitmap information to show.

Sizes are the bitmap sizes covered by the selected tables.

Glyphs allows choosing a glyph by its glyph index.

Zoom will zoom in or out.

Outline displays the outline as well.

- Glyph Metrics

width is the number of columns of data, i.e. the number of pixels in horizontal direction.

height is the number of rows of data, i.e. the number of pixels in vertical direction.

horiBearingX is the number of pixels from the origin to the left edge of the bitmap (horizontal layout).

horiBearingY is the number of pixels from the origin to the top edge of the bitmap (horizontal layout).

horiAdvance is the bitmap's advance width (horizontal layout).

vertBearingX is the number of pixels from the origin to the left edge of the bitmap (vertical layout).

vertBearingY is the number of pixels from the origin to the top edge of the bitmap (vertical layout).

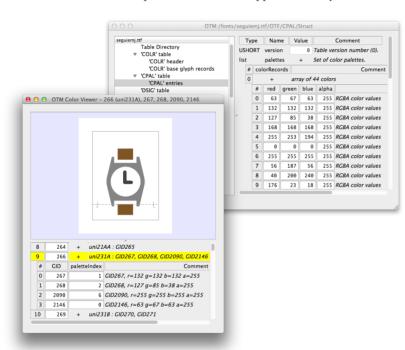
vertAdvance is the bitmap's advance height (vertical layout).

All these values are read-only.

FoptMaster™

Color Viewer

In summer 2013, Microsoft presented their solution for color fonts and introduced two new tables, COLR and CPAL. The solution is elegantly simple. Starting point is a base glyph which usually is encoded and serves as a fallback for applications that do not know the new tables. The COLR table maps this to-be-colored base glyph to one or more glyphs, each of which holds outlines for a specific color plus references to the respective color definitions. The CPAL table in turn provides the color definitions, stored as RGBA values. There may be multiple sets of color definitions, called 'palettes', from which applications may choose.





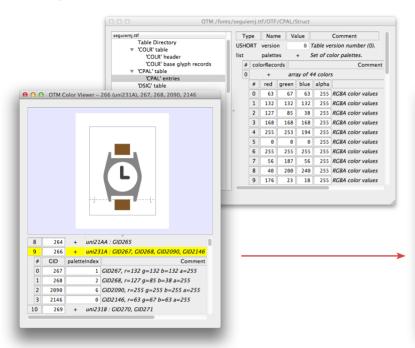
Top: Base glyph 266. COLR will map it to color-specific glyphs. Middle: Color-specific glyphs. Glyph 267 for grey outlines, glyph 2090 for white outlines, glyph 2146 for dark grey outlines, glyph 268 for brown outlines. Bottom: Colored glyphs combined.

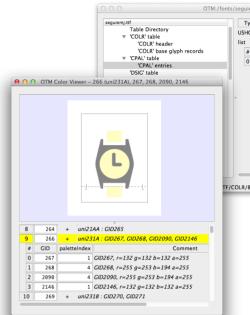
Open the Color Viewer and 'CPAL' entries side by side to review both a color glyph's composition and the colors it makes use of.

This structure is reflected in the Color Viewer which essentially visualizes the content of the COLR table. The top level is a list of base glyphs, identified by GID. Clicking the + sign next to one of them will reveal a nested list of pairs each consisting of one color-specific glyph, again by GID, and a reference to a color definition, by **paletteIndex**. The latter in turn refers to a **colorRecord** stored in the CPAL table; see the **#** column for the **paletteIndex**. Please note that there may be multiple color **palettes** in the CPAL table, each of which has its own definition for a given **colorRecord** by way of **red**, **green**, **blue** and **alpha** values. A special **paletteIndex** value of oxFFFF indicates that the user-defined foreground color is to be used.

FoptMaster™

There is more than one way to change a color: Either adjust a **paletteIndex** in the Color Viewer, thereby selecting another predefined color. Or open the CPAL table in OTMaster's main dialog, choose a palette, then redefine the **red**, **green**, **blue** and **alpha** values of a given **colorRecord**. Or add a new **colorRecord** by duplicating an existing one, adjusting its values, and referring to it from the Color Viewer.

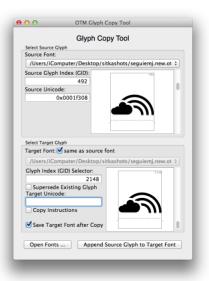




Creating a color glyph from a normal glyph takes only a few steps:

I. Open the Glyph Copy Tool. Choose a base glyph, either by entering a numeric Source Glyph Index (GID) or by using the scrollbar next to the glyph image. For every color, make a copy of this glyph by clicking the Append Source Glyph to Target Font button. Target Font should be same as source font. Please note that glyph copies will be added after the (target) Glyph Index (GID) Selector.

To add new glyphs at the end of the font, enter any number higher than the number of glyphs in the font. OTMaster will automatically make it the last GID in the font.





In our example, we appended new glyphs at the end of the font:

Root	Glyph Set and Code Ranges	View
		M Image
OpenType Fo	Glyph Index (GID) 🗘	Outline
Offset Tab		Points
Table Dire	2142 2143 2144 2145 2146 2147	Indices
▼ 'COLR' tab		Box
'COLR'		hhea asc/dsc
'COLR'	2148 2149 2150 2151	typo asc/dsc
▼ 'CPAL' tab		win asc/dsc
'CPAL'		Gridfit
'DSIG' tabl		Grayscale
		Dist: 5
'GDEF' tab		
▼ 'GDEF' tab 'GDEF'		Split: 7

2. Open each of the new glyphs in the Glyph Editor, choose the **Shift** tool, select and delete all contours not needed in this glyph. The result:

Root	Glyph Set and Code Ranges	View
OpenType Fo Offset Tab		Image Outline
Table Dire COLR' tab 'COLR' tab 'COLR' 'COR' 'CPAL' tab 'CPAL' 'DSIG' tabl 'GDEF' tab 'GDEF' 'GDEF'		Points Indices Box hhea asc/dsc Vin asc/dsc Gridfit Grayscale Dist: 5 Split: 7



Deleting contours not needed in a specific color glyph via the **Shift** tool.

3. Go to the COLR table's content, select any entry and use the Edit menu's **Grow** function. This will duplicate the selected entry which serves as a template for our new color glyph definition. Below, we chose entry number 825 to create the new entry number 826. (If there are no COLR and CPAL tables in our font, you may **Cut** and **Paste** them over from another font.)

eguiemj.new.otf	825	2141	+ uni27	10 : GID2142, GID2143, GID2144
 Root OpenType Fo 	826	2141	+ uni27	10 : GID2142, GID2143, GID2144
OpenType Fo Offset Tab	#	GID	paletteIndex	Comment
Table Dire	0	2142	14	GID2142, r=255 g=103 b=178 a=255
▼ 'COLR' tab 'COLR'	1	2143	12	GID2143, r=255 g=211 b=55 a=255
'COLR'	2	2144	0	GID2144, r=63 g=67 b=63 a=255
▼ 'CPAL' tab				

After opening the nest by clicking the **+** sign, you'll see entries for specific color glyphs. If there are too many, select and **Cut** those not needed. If there are not enough, select and **Grow** any of them. In our example, we need four entries – one for the cloud, three for the rainbow elements. Now we adjust GID numbers. The top-level GID will reflect the base glyph's 492, the nested color glyph GIDs will reflect the new glyphs' 2148–2151.



4. Open the Color Viewer. Scroll down to the **COLR** table entry created and adjusted in the last step. Change colors by adjusting the **paletteIndex** for each specific color glyph's **GID**. Below, we assigned color definitions 1–4.

eguiemj.new.otf	81	13	2100	● ● C	OTM C	olor View	er - 4	492 (uni0001F308), 2148, 2149, 2150, 2151
 Root OpenType Fo 	81	14	2103					
OpenType Fo Offset Tab	81	15	2107					
Table Dire	81	16	2111					
▼ 'COLR' tab	81	17	2115					
'COLR' 'COLR'	81	18	2118					
▼ 'CPAL' tab	81	19	2121					
'CPAL'	82	20	2124					\frown
'DSIG' tabl ▼ 'GDEF' tab	82	21	2128					
'GDEF'	82	22	2131					
'GDEF'	6 82	23	2134					
'GDEF' 'GDEF'	82	24	2137					
'GDEF'	82	25	2141					
'GDEF'	82	26	492					
▼ 'GPOS' tab 'GPOS'		#	GID					
'GPOS'		0	2148	825	2141	+ 'u	ini27	710 : GID2142, GID2143, GID2144
'GPOS'		1	2149	826	492			001F308 : GID2148, GID2149, GID2150, GID.
		2	2150	#	GID	palettelr		Comment
'GPOS'		3	2151	0	2148			GID2148, r=132 g=132 b=132 a=255
				1	2149	1		GID2149, r=127 q=85 b=38 a=255
'GPOS' ▼ 'GSUB' tab				2	2150	1		GID2150, r=168 q=168 b=168 a=255
'GPOS' ▼ 'GSUB' tab 'GSUB'								
'GPOS' ▼ 'GSUB' tab 'GSUB'	tf" ha	ıs be	en chang	3	2151		4	GID2151, r=255 g=253 b=194 a=255



'kern' Table Viewer

This is a most useful tool to visually check and adjust the kern table. The currently selected pair is presented in main part of the window.

					ppem	
					160	•
		Δ	d	350		
			1.1			
					Subtable Subtable 0	*) *
20	4	82		uni0041 uni014E		;
21	4	83	-23	uni0041 uni0150		:
21 22			-23 -23	uni0041 uni0150 uni0041 uni0152		:
21 22 23	4	83	-23 -23 -23	uni0041 uni0150 uni0041 uni0152 uni0041 uni0051		;
21 22 23 24	4	83 86	-23 -23 -23	uni0041 uni0150 uni0041 uni0152		;
21 22 23	4 4 4	83 86 89	-23 -23 -23 -160	uni0041 uni0150 uni0041 uni0152 uni0041 uni0051		;
21 22 23 24	4 4 4 4	83 86 89 100	-23 -23 -23 -160 -160	uni0041 uni0150 uni0041 uni0152 uni0041 uni0051 uni0041 uni0054		;

The 'kern' Table Viewer dialog.

— ppem

This single option serves to enlarge or reduce the size of the kerning pair, by choosing another ppem size.

- Subtable

Here you may select which subtable's kerning you want to review or edit.

The list of kerning pairs which follows consists of three editable columns: **left** glyph of the pair by glyph index,

right glyph of the pair by glyph index,

and the kerning **value** which is relative to the font's UPM value.

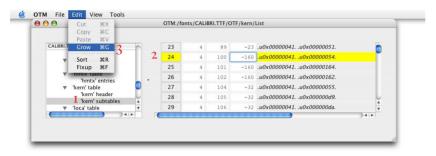
In addition, a **Comment** column translates left and right glyphs' indices into glyph names as found in either CFF or post table.

Like in **Nested Tables** viewing mode, use ↑ and ↓ keys to flip through all kerning pairs quickly, and →I (to right) and SHIFT+→I (or I←; to left) to jump from column to column.

Note: A kern table may contain more than one subtable, but please be aware that Windows and os/2 accept a kern table only if it holds a single subtable of format 0. Also see the kern table specification. ► local ► www



Existing kerning pairs –**left** and **right** glyphs' indices as well as the kerning **value**– can be reviewed and adjusted in the 'kern' Table Viewer, but it is not possible to remove or add kerning pairs. With a little trick, though, you may do so:



Remove or add a kerning pair in three steps.

I. Go to the main window and select 'kern' subtables (inside of 'kern' table) from the table overview. Click the + sign next to the subtable which you intend to edit, this will fold out the subtable as a nested table. Possibly double-click the subtable's header to show its entries in a new window.

Now either remove a kerning pair:

- 2. Select the pair which you intend to remove.
- 3. Choose Edit > Cut.

Or add a new kerning pair:

- 2. Select any kerning pair, preferrably the last one.
- 3. Choose **Edit > Grow** to duplicate this selected pair.

Change **left** glyph index, **right** glyph index and/or kerning **value**, and you have added a new kerning pair! Go back to the 'kern' Table Viewer and visually adjust the kerning **value**. If you added new pair(s) at the end of the kern table, you know where to find them ...

Finally choose **Edit > Fixup** to reorder kerning pairs by glyph indices.



'GPOS'/'GSUB' Viewer

The 'GPOS'/'GSUB' Viewer is a tool for reviewing the content of the two most important OpenType layout tables, GSUB for glyph substitution and GPOS for glyph positioning. In case of the GPOS table, positioning values can be adjust. The viewer presents these tables' data visually, which makes it possible to check a font's layout behavior in a comfortable way. Data is, structurally, presented in the same way as found in these tables.

Both GSUB and GPOS tables consist of three lists: Script list, Feature list, and Lookup list.

The Script list sums up all scripts explicitly addressed by the layout table (these are scripts – or writing systems – like 'latn' for Latin or 'cyrl' for Cyrillic).* Per each script, there is a list of languages explicitly addressed by the layout table as well as a 'default' language for which there is a special place in the data structure. In case that a layout application cannot find a match for the selected language (e.g. by way of the spelling dictionary in Adobe InDesign) in the font's layout table, it would fall back to this 'default' language. And per each language there is a list of features associated with it.

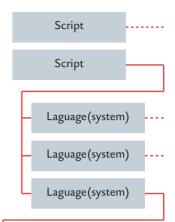
The Feature list, for each feature to which a script/language combination refers, points to one or more lookups in which the actual substitution and positioning behavior is defined.

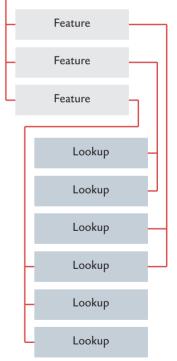
Finally, the Lookup list points to individual lookups, each of which defines portions of layout behavior.

Users of VOLT are familiar with this structure because VOLT presents layout data in a way which resembles layout tables' data structure. Users of AFDKO and Adobe's feature file syntax may need some time to get accustomed to it because the higher-level nature of Adobe's feature file syntax hides the complexity of the data structure of compiled layout tables. So again our recommendation that you study especially the documents related to OpenType layout tables, in particular to GSUB and GPOS.

* The default script, or 'DFLT' (all-caps!), was introduced rather late by Adobe. While there is a special place for the default language, or 'dflt' (lowercase!), in the Language list, there is no such thing for the 'DFLT' script – hence it is a script like any other, included in the regular Script list, and identified by its tag.

Info: The GPOS table. ► local ► www The GSUB table. ► local ► www Also see the OpenType Layout Common Table Formats document. ► local ► www OpenType layout data is organized by script, language system, typographic feature and lookup:







The 'GPOS'/'GSUB' Viewer's top popup boxes reflect the layout tables' internal structure:

OTM GPOS/GSUB Vie 000 Layout Table Script Language <defaults • 'GSUB' table 'arab', Arabi 1 Feature Writing Dire . (4) 'liga', Standard Ligatures right to left (4) ligature substitution Subtable View Data Report Image (4) Ligature Substitution (0,4) uniFEF6 <- uniFEE0, uniFE82 ppem: 160

left to right	
right to left	1
top to bottom	

Writing Direction options.

— Layout Table

At first you need to choose from 'GSUB' table or 'GPOS' table – 'GSUB' table cares for glyph substitution, 'GPOS' table cares for glyph positioning.

- Script

Select the script whose lookups you plan to review.

-Language

Select the language whose lookups you plan to review. The choise of languages depends on which **Script** you have selected previously.

— Feature

Select a feature whose lookups you plan to review. The choice features depends on which **Language** you have selected previously.

—Lookup

The selection of lookups shown in this popup box is determined by your previous choices of **Script**, **Language** and **Feature**.

-Writing Direction

Options **left to right, right to left** and **top to bottom** make sure that glyph strings are presented in correct order.

Tip: To see more lookups, and independently of previous choices of Script, Language and/or Feature, select the option <any> in Script, Language and/or Feature.



- Subtable View

The content of the lookup can be viewed in three modes.

I. Data. This is exactly what you see if you inspect 'GSUB' lookups or 'GPOS' lookups in the table content area:

ayo	out Table			Script					Language	
'GI	POS' tab	e	•	'latn',	Latin		•		<default></default>	
Feat	ture					Looku	þ			
(1)) 'cpsp',	Capita	al Spi	acing	•	(0) s	ingle adjustment			
Subt	table Viev	,								
						Dat	a Report Imag	e	e	
#	PosFa	rmat	Pos	itioning			(~ ~	Comment	
C		1	105	+	look	un tyne			t, all glyphs the same	
	Туре	-	N	lame		/alue	,		Comment	
				geForma		_	Coverage table form	at		
	USHOR	T Va	lueFo	ormat		5				
	USHOR	T GI	OValı	ueCount		137	Number of adjustme	n	nt values = number of glyphs covere	
	list	GI	OValı	ues		+	Glyph index to value	21	list.	
	#	G	D	XPIc		XAdv			Comment	
	0	1	34	10	00	200	A			
	1		35	10	00	200	B			
	2		36	10	00	200	С			
	3		37	10	00	200	D			
	4		38	16	00	200	Ε			
	5		39	16	00	200	F			
	6		40	10	90	200				
	7		41	10	90	200	Н			
	8	_	42	16	-	200	_			
	9	_	43	10	-	200				
	10		44	16	IA .	286	K) 4 +	

2. **Report**. Provides you with a text report of the selected lookup's content:

Layout Table	Script		Language	
'GPOS' table	e 📫 🛙 🕻 Iatn	', Latin 🔹	<default></default>	\$
Feature		Lookup		
(1) 'cpsp', C	apital Spacing	(0) single adjustment		\$
Subtable View				
		Data Report Image	2	
• lan • fea • lool refe 'cps	kup: (index=0,t) red to by featur p'. itioning table:		3) 'cpsp', (4) 'cpsp', (5) 'cpsp', (6)	
#	glyph	positioning		
0	(34) A	XPIc=100, XAdv=200		
1	(35) B	XPlc=100, XAdv=200		
2	(36) C	XPlc=100, XAdv=200		
3	(37) D	XPIc=100, XAdv=200		
4	(38) E	XPIc=100, XAdv=200		
5	(39) F	XPIc=100, XAdv=200		
6	(40) G (41) H	XPlc=100, XAdv=200 XPlc=100, XAdv=200		
8	(41) H	XPIc=100, XAdv=200 XPIc=100, XAdv=200		
9	(42) 1	XPIc=100, XAdv=200 XPIc=100, XAdv=200		
-	(43) J (44) K	XPIc=100, XAdv=200 XPIc=100, XAdv=200		
	(44) K (45) L	XPIc=100, XAdv=200		
	(45) L (46) M	XPIc=100, XAdv=200		
	(47) N	XPIc=100, XAdv=200		×.



3. **Image**. This is the default viewing mode. Substitution or re-positioning is visualized, so that you may evaluate a lookup's layout behavior quickly at a glance, and even evaluate positioning adjustments:

ayout Table	Script		Language		
'GPOS' table 📫	'latn', Latin	\$	<default></default>		\$
eature	Lookup				
(1) 'cpsp', Capital Spa	cing 🛟 (0) single ac	ljustment			\$
ubtable View			_		
	Data Re	port Imag			
	(0) Single Adjustr	nent		View	
SubTable#0, Po	osFormat 1 (one value for al		Values#0.		60
			U	write horiz	
	34			 write vertic 	ally
	34				
				Glyph 1	
				A Glyph Index	\$
				Clyph Index XPlacement:	34 100

Use the scrollbar to the right of the preview area to flip through all substitutions or positioning adjustments.

The main area is headed by this lookup's index (in parentheses) and type, followed by additional information like the index of the current subtable or the substitution or positioning format. The original glyph is colored RED, the replaced or repositioned glyph is colored GREEN.

Like in other OTMaster dialogs, use the scrollbar next to the review image to switch from one substitution or positioning entry to another!

Which information are shown in **Image** viewing mode depends on whether you are inspecting 'GSUB' table or 'GPOS' table and also depends on the lookup type. A few examples are given below.



I. GSUB:

I.I. With non-contextual substitution, there are no further options. You may use the scrollbar to flip through glyph substitutions:

	OTM GPOS/GSUB Viewer - AP57	X12T.otf	
Layout Table Script		Language	
'GSUB' table 🛟 🤇 'latn', Lat	in ;	<pre><default></default></pre>	÷
eature	Lookup		
(7) 'liga', Standard Ligatures 🛟	(18) ligature substitution		\$
ubtable View			
	Data Report Image		
	(18) Ligature Substitut	tion	
(0,0) ffi <- f, f, i	(10) Ligataro Sabolita	ppem: 126	
	ffi	÷	

ppem is the ppem size at which the review image is rasterized.

Since version 2.0 of OTMaster, the 'GPOS'/'GSUB' Viewer offers an additional option **Writing Direction**. to change the display order.

Layout Table	Script		Language		
'GSUB' table 🛟	'arab', Arabic	•	<default></default>		¢
Feature		Lookup		Writing Direction	
(4) 'liga', Standard I	igatures 🛟	(4) ligature substitution	\$	right to left	\$
Subtable View		1			
		Data Report Image			
	(4) Ligature Substituti	on		
(0,4) uniFEF6 <-	- uniFEE0, uniFE82			pem: 160 🕄	
		6498 6592		100	
		0130			
		\sim -			
		6614			
					t
				U	
				4	
				4	



1.2. With contextual substitution, there are a few more pieces of information. Besides a RED original glyph and a GREEN adjusted glyph, there is a BLACK context glyph in the preview area:

Layout Table	Script	Language	
'GSUB' table	latn', Latin	<default></default>	\$
Feature	Lookup		
(15) 'calt', Conte:	xtual Altern 🛟 🛛 (42) chaining contextual	substitution	
Subtable View			
	Data Report Imag	le	
	(42) Chaining Contextual Substitutio		
	[42] Ghaining Contextual Substitutio Fable#1, SubstFormat 3 (glyph coverage base	P (7)	
		ed).	0
	185 484		
	388 464		
		U	
		Back	ktrack
		Inpu	d.
		0	AE
		U	ML I
			kahead

— ppem

The size at which the review image is rasterized.

- -Backtrack
- -Input
- -Lookahead

Input is always shown, this is the glyph or glyphs will be substituted, together with **Backtrack** and/or **Lookahead**. Contextual substitution may involve more than just one of each!

If (in the order of appearance) **Backtrack** or **Input** or **Lookahead** involves but a single glyph, the respective popup is greyed out and shows the single glyph's name. If **Backtrack** or **Input** or **Lookahead** involves a glyph class, then the popup shows the first glyph by default. You may use the popup to select any other glyph: just click on the popup and move the mouse up or down the list, and the review will instantly show the glyph touched by the mouse – no need to click.



2. GPOS:

2.1. With single positioning, the affected glyph as well as the positioning adjustment value are shown:

OTM GPOS/GSUB Viewer - TiptoeBasicKLTF-Bold.otf 0 0 Layout Table Script Language 'GPOS' table • • latn', Latin <default> Feature (1) 'cpsp', Capital Spacing (0) single adjustme • Subtable View Data Report Image (0) Single Adjustment SubTable#0, PosFormat 1 (one value for all glyphs), GIDValues#0. 160 🗘 nnem • write horizontally write vertically Glyph : Glyph Index 34 XPlacement 100 XAdvance 200



Writing Direction options are available in the GPOS too, of course.

-View

ppem is the ppem size at which the review image is rasterized. In addition, two radio buttons allow to choose from **write horizontally** and **write vertically** which will arrange the glyphs involved either side by side or one above the other.

— Glyph 1

For each glyph there is a popup with the glyph name. **Glyph Index** is the selected glyph's index. Depending on which kind of adjustments have been made, values for **XPositioning**, **XAdvance**, **YPositioning** and/ or **YAdvance** adjustment are shown.



2.2. With pair positioning, there are a few additions:

Layout Table	Script	Language			
'GPOS' table	latn', Latin	<defaul< th=""><th>t></th><th></th><th></th></defaul<>	t>		
Feature	Lookup				
(0) 'kern', Kerning	(0) pair adjustme	ent			
Subtable View					
	Data Repor	t Image			
	(0) Pair Adjustment			View	
SubTable#0, P	osFormat 1 (glyph index based)		6	ppem:	160 🕻
				💽 write hori	zontally
				O write vert	ically
	14 53				
	34 53				
	34 53				
	34 53				
	34 33				
				_Glyph 1	
	- T			Glyph 1 hyphen	\$
				hyphen	14
				hyphen Glyph Index	14
				hyphen Glyph Index XAdvance:	¢ 14 -29

-View

ppem is the ppem size at which the review image is rasterized. Again, two radio buttons allow to choose from **write horizontally** and **write vertically** which will arrange the glyphs involved either side by side or one above the other.

— Glyph 1 — Glyph 2

For each of them there is a popup with the glyph name. If it is but a single glyph, the popup will be greyed out. If there is more than one, you may click on the popup and move the mouse up or down the list to see any of the other glyphs in the review – no need for clicking. **Glyph Index** is the glyph index of the glyph currently shown. Depending on which kind of adjustments have been made, values for **XPositioning**, **XAdvance**, **YPositioning** and/or **YAdvance** adjustment are shown. The latter values can be adjusted. However, the general structure of GSUB and GPOS tables cannot be changed: it is not possible to add anything that has not been in these tables before, nor to remove anything.



2.3. With contextual positioning, BLACK context glyphs join RED original glyph and GREEN adjusted glyph:

Layout Table	Script		Language	
'GPOS' table 🛟	('latn', Latin	•	<pre><default></default></pre>	\$
Feature	Lookup			
(1) 'kern', Kerning	(2) chaining	g contextual adj	ustment	\$
Subtable View			_	
	Data R	eport Image]	
(2) (Chaining Contextual P	ositioning	ppem	
	#13, PosFormat 3 (glyph co		160	(
	114 1	6.d		
			0	
	114 1	\$4		
			Backtrack	
			Input	
			0	D
			Lookahead	
			0 space	\$
			1 A	
			Y	

-ppem

The size at which the review image is rasterized.

- -Backtrack
- -Input
- -Lookahead

Input is always shown, this is the glyph or glyphs to be substituted, together with **Backtrack** and/or **Lookahead**.

No values are shown, but the effect is visible in the preview. In the example above, the 'D's right sidebearing is increased when followed by space and a class which contains an 'A'.

Note: The 'GSUB'/'GPOS' Viewer allows you to review individual lookups. If a feature refers to more than one lookups, these lookups' behavior will be additive. This has different effects with GSUB and GPOS.

GSUB: All lookups associated with a feature will be applied. However, if a previous lookup has substituted an input glyph already, additional lookups will not find a match for the same input glyph any more because this has been substituted already and does not exist any more in the input string. In so far, lookup order matters, as does the order of substitutions inside of a subtable. GPOS: And again, all lookups associated with a feature will be applied. If more than one lookup adjust positioning and/or advance width of a specific glyph, then all lookups' adjustments will add up. It is important to keep this in mind, because the 'GSUB'/'GPOS' Viewer visualizes only individual lookups' adjustments, i.e. does not show the result of a feature's total positioning adjustments!

DTL OTMASTER: TOOLS MENU

2.4. Mark-to-base positioning serves to define where mark glyphs (accents as well as vowels and dots in Arabic-script fonts) attach to base glyphs (letters). Each mark glyph carries an anchor point and is associated with a mark class, and each base glyph carries an anchor point per mark class. Thus anchor points determine where on a base glyph a mark glyph sits.

	Script	Language	
'GPOS' table 📫	'DFLT', Default	default>	
Feature	Lookup		Writing Direction
(0) 'mark', Mark Pos	itioning 🛟 🛛	(0) mark to base attachment	right to left
Subtable View			
	Data	Report Image	
		View	
		70 size [ppem]	296 🗘
		spacing [%]	0 🗘
		all glyphs on mark glyphs	
	Λ	base glyphs	
		all glyphs all	
	+	Filter	
		Classes	-
		Mark Array #	
		Base Array #	•
	6	Mark Glyph (#3,GI	D=759,Class=0)
		uni0302	•
		x	0
		Y	440
		Base Glyph (#2,Gl	0=70)
		e	•
			220
		×	220

An anchor point is visualized by a red cross. Its position, in relation to the (black) base glyph, can be adjusted via drag/drop of the red cross. Its position, in relation to the mark glyph, can be adjusted via drag/drop of the (blue) mark glyph. Numerical adjustment is possible too.

-View

size [ppem] is the ppem size at which the review image is rasterized.
spacing [%] is the distance between the glyphs' boxes, in percent of ppem.
There are four display options – all glyphs one by one (a single mark-to-base pair), mark glyphs all at once (a row of all mark-to-base pairs for the selected base glyph), base glyphs all at once (a row of all mark-to-base pairs for the selected mark glyph), all glyphs all at once (a table of all mark-to-base pairs with a column per base glyph and a row per mark glyph).

- Mark Glyph

-Base Glyph

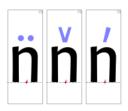
For each mark and base there is a popup from which a glyph can be chosen by name. Which glyphs are shown in each popup depends on the selected **Classes** in the **Filter** section (usually there are separate classes e.g. for top and bottom mark glyphs). **X** and **Y** are the anchor's coordinates.



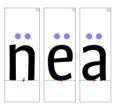
Below are examples for the four *display options*.



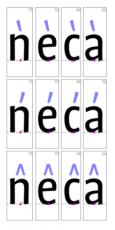
all glyphs one by one



mark glyphs all at once



base glyphs all at once



all glyphs all at once

Note: Adjusting a mark glyph's anchor position will reposition this mark for all base glyphs. And adjusting a base glyph's anchor position will reposition all marks associated with this anchor.



2.5. Mark-to-mark positioning serves to define where mark glyphs attach to other mark glyphs. Each mark glyph (attaching to another mark glyph) carries an anchor point and is associated with a mark class, and each mark glyph (allowing another mark to attach to it) carries an anchor point per mark class.

ayout Table Script	Language	
'GPOS' table 🗧 🛛 ('DFLT', Default	default>	
eature	Lookup	Writing Direction
(3) 'mkmk', Mark to Mark Positionir	(1) mark to mark attachment 🛟	right to left
ubtable View		
	a Report Image	
	View	
	756 size [ppem]	296 🗘
	spacing [%]	5 🗘
	 all glyphs one 	
	 mark glyphs al base glyphs all 	
$\bullet \bullet$	O all glyphs all a	
	Filter	
	Classes	-
	Mark Array #	-
	Base Array #	-
	Mark1 Glyph (#0,GID	=756,Class=0)
	uni0308	•
	x	0
	Y	480
	Mark2 Glyph (#0,GID	=756)
	uni0308	•
	x	0
	Y	610
		010

The interfaces for mark-to-mark positioning and for mark-to-base positioning are identical.

This resembles mark-to-base positioning. As a consequence, the dialog and editing behavior is identical.



Consistency Checker

This tool helps finding inconsistencies across OpenType fonts' tables. Most Consistency Checker dialogs follow the same pattern. They present table entries which are expected to be consistent. **Check** compares entries' values and suggests corrections. **Uncheck** reverts to original values. And **Apply** finally applies automatically or manually refined values to the font.

-Header

The Header section of the Consistency Checker gives an overview, in one dialog, of different table's entries which need to be consistent:

Header Name Ver		tistics	ency Chec		Language
neader Name vers	sion sta	usues	Onicode	Code Page	Language
Weight and Style					
💿 use macStyle 🔵 use fsSelec	tion 🔵 us	e usW	eightClass		
'head' table : macStyle	0x0000	Regu	lar		
OS/2' table : fsSelection	0x00c0	REGU	JLAR USE_TY	PO_METRICS	
OS/2' table : usWeightClass	400	Norn	nal (Regular)		
All values	match.			Check Unche	eck Apply
Font Metrics					
💿 use hhea entries 🔵 use typ	o entries (use	win entries	🗹 use font box fo	r win entries
'hhea' table : ascender	(578	'OS/2' table	: sTypoAscender	678
'hhea' table : descender	-3	322	'OS/2' table	: sTypoDescender	-322
'hhea' table : lineGap		200	'OS/2' table	: sTypoLineGap	200
OS/2' table : usWinAscent	8	371	'OS/2' table	: usWinDescent	376
All values	match.			Check Unche	eck Apply
Italic Angle					
💿 use 'post' angle 🔵 use 'CFF	' angle				
post' table : italicAngle	0.0	000	'CFF ' table :	ItalicAngle	(
hhea' table : caretSlopeRise		1	'vhea' table	caretSlopeRise	none
'hhea' table : caretSlopeRun		0	'vhea' table	caretSlopeRun	none
'head' table : macStyle	0x0	000	Regular		
OS/2' table : fsSelection	0×0	0c0	REGULAR USE	_TYPO_METRICS	
All values	match.			Check Unche	eck Apply

I. Weight and Style compares the OS/2 table's fsSelection,

usWeightClass and the head table's **macStyle**. If a font is a bold style, this should be reflected in all three entries, if a font is an italic style, this should be reflected in **fsSelection** and **macStyle**.

Please see the OS/2 table spec \triangleright *local* \triangleright *www* for **fsSelection** and **usWeightClass** with the head table spec \triangleright *local* \triangleright *www* for **macStyle**.

2. Font Metrics compares the hhea table's ascender/descender (the hhea entries) with the OS/2 table's sTypoAscender/sTypoDescender and usWinAscent/usWinDescent (the typo entries and win entries). Via the top row radio buttons you may choose against which of these entries the other ones will be checked. The use font box for win entries option derives the OS/2 table's usWinAscent/usWinDescent values from the largest glyph dimensions found in the font.

Tip: In addition, consider using font checking tools like Microsoft's Font Validator ▶ www as well as the command-line tool CompareFamily ▶ www which is included in Adobe's AFDKO.

DTL OTMASTER: TOOLS MENU

Please review **Check**'s adjustments – you may not agree with them because it sets:

ascender = sTypoAscender = usWinAscent descender = sTypoDescender = usWinDescent Both UPM and the three sets of vertical metrics

a. head unitsPerEm

b. hhea ascender/descender/lineGap

c. OS/2 sTypoAscender/sTypoDescender/sTypoLineGap d. OS/2 usWinAscent/usWinDescent

need to be defined carefully because different applications pick different entries to determine the default line-to-line distance. To achieve consistent default line-to-line distance in most – though not in all – applications, you need to take care that the 'sum' of each of the first three sets is the same. Microsoft's and Adobe's current practice for determining (vertical) font metrics – which may be interpreted as a recommendation – is this:

2.1 The pair **sTypoAscender/sTypoDescender** indicates how much of UPM (the head table's **unitsPerEm**) is reserved for ascenders and descenders. This can be expressed as:

sTypoAscender – sTypoDescender = UPM It is expected that your typeface is designed and UPM is defined such that normal ascenders and descenders as of 'A'–'Z' and 'a'–'z' remain within the UPM's boundaries, or sTypoAscender and sTypoDescender.

2.2 **sTypoAscender/sTypoDescender/sTypoLineGap** (set c) defines a font's ideal line-to-line distance. It depends on the typeface's design, but usually **sTypoLineGap** is about 20% of:

2.2.1 sTypoAscender – sTypoDescender

2.2.2 UPM

(According to 2.1, 2.2.1 and 2.2.2 are equal.) For example, UPM = 1000 and sTypoLineGap = 200 result in a default line-to-line distance of 120% of UPM which would translate into 10/12 pt.

2.3 Since **sTypoAscender/sTypoDescender/sTypoLineGap** (c) were defined such that the 'sum' results in an ideal default line-to-line distance, you may set the OS/2 table's version to 4 and **fsSelection** bit 7 to 1. This indicates that **sTypo**-values should be used for calculating default line-to-line distance rather than **usWin**-values. Once the OS/2 table's version is 4 however, you need to set **fsSelection** bits 8 and 9 consciously!

2.5 sTypoAscender/sTypoDescender/sTypoLineGap (c) and **ascender/descender/lineGap** (b) share same values, so you can simply reuse the first set's values for the latter set:

ascender = sTypoAscender descender = sTypoDescender lineGap = sTypoLineGap

2.6 usWinAscent/usWinDescent (d) provide information about a font's clipping zones – the largest extensions that you do not want to see clipped, i.e. cut off. These values reflect the tallest glyphs in the font.



An example that meets all conditions: head = 1000 OS/2 hhea sTypoAscender = ascender = 800 sTypoDescender = descender = -200 sTypoLineGap = lineGap = 200 usWinAscent = 850 usWinDescent = 350

Tip: Please consult John Hudson's Setting Cross-Platform Vertical Metrics, especially the latest 'Update'. ▶ www Rather for the illustration than the method itself which is obsolete now, see Karsten Lücke's Font Metrics. ▶ www

Note: This is a recommendation, not a requirement! Also see \triangleright os/2.

Note: The hhea table's descender & the OS/2 table's sTypoDescender are given as negative values – but usWinDescent must be given as a positive value! Therefore, the term 'sum' ist not exact in a strict sense.

DTL OTMASTER: TOOLS MENU



With a typeface of normally sized ascenders and descenders, the 'sum' of the **usWin** set will be smaller than the 'sum' of the **sTypo** set. Increase **usWin**-values a bit so that the 'sum' of each of the three sets is identical:

sTypoAscender - sTypoDescender + sTypoLineGap (c)

= usWinAscent + usWinDescent (d)

= ascender - descender + lineGap (b)

In case that a font contains excessively tall glyphs it may be impossible to achieve this equation.

It is strongly recommended that you define vertical metrics such that they are identical across all fonts that belong to a family. This to make sure that if a user relies on default line-to-line distance, this would not vary if one paragraph is set in Regular, the other in Italic or Bold style.

3. Italic Angle allows the comparison of the post table's italicAngle with the OS/2 table's ItalicAngle and the hhea table's caretSlopeRise and caretSlopeRun. The relation between these two is

tan -italicAngle = slopeRun / slopeRise

The vhea table's **caretSlopeRise** and **caretSlopeRun** can be checked too if this table is present.

For the sake of consistency, the OS/2 table's **fsSelection** and **macStyle** are shown again, since their values need to be in tune with italic angle and slope rise and run.

Tip: If your font is a CFF-based OpenType font, also compare with the 'CFF' top dictionary **ItalicAngle**, and possibly with slanting as results from the FontMatrix which is located in 'CFF' top dictionary as a nested table.



-Name

The Name section of the Consistency Checker helps inspecting individual name table entries and serves as a built-in documentation of the name table nameIDs, pointing out their correlation with other table's entries. So you may flip through your name table's records and compare your entries with the recommendations.

Header	Name Version Statistics Unicode Code Page Language
#4 [1;0;0;4] Macir	itosh; Roman; English; Full font name
[p;e;l;id]	Name String
0;0;0;0]	© 2007 Adobe Systems Incorporated. All Rights Reserved.
1 ;0;0;1]	Arno Pro
2 ;0;0;2]	Regular
3 ;0;0;3]	1.011;ADBE;ArnoPro-Regular
4 ;0;0;4]	Arno Pro
5 .;0;0;5]	Version 1.011;PS 1.000;hotconv 1.0.50;makeotf.lib2.0.16025
6 ;0;0;6]	ArnoPro-Regular
7 .;0;0;7]	Arno is a trademark of Adobe Systems Incorporated in the United State
8 ;0;0;9]	Robert Slimbach
9 ;0;0;11]	http://www.adobe.com/type
10 ;0;0;14]	http://www.adobe.com/type/legal.html
11 ;0;0;256]	Regular
12 ;1;1033;0]	© 2007 Adobe Systems Incorporated. All Rights Reserved.
13 1:1:1033:11	Arno Pro
'Regular" as indic Exception 2: for M must be identical Font Family name	e should be a combination of strings 1 and 2 Exception 1: if the front is ated in string 2, then use only the family name contained in string 1 licrosoft platform strings for CFF OpenType fonts the Full font name string to the PostScript FontName in the CFF Name INDEX. (01) 1 is "Arno Pro", Font Subfamily name (ID 2) is "Regular". he string seems to fit the OTF guidelines.
Selec	t a name string for further information. Reset Apply

Reviewing the name record with NameID 2 (the Subfamily name, Macintosh platform).

—Version

The Version section displays, side by side, various version information as found in an OpenType font.

			sistency Che			
Header	Name V	ersion Stat	istics Unico	de Co	ode Page	Language
ead' table : fontR	evision	1.01	1 'CFF ' table :	CIDFon	tRevision	
FF ' table : versio	n 1.00	0	'CFF ' table :	CIDFon	tVersion	
'name' table nameID	5 Version stri	ngs				
#5 [1;0;0;5] Mad	intosh; Rom	an; English; Ve	ersion string			
Version 1.011;P	S 1.000;hoto	onv 1.0.50;m	akeotf.lib2.0.1	5025		
#17 [3;1;1033;5] Microsoft;	Unicode BMP o	only; English, U	nited Sta	ates; Versio	on string
Version 1.011;F	S 1.000;hoto	onv 1.0.50;m	akeotf.lib2.0.1	5025		

A font's version information side by side.



- Statistics

Upon clicking the **Check** button, the Statistics section of the Consistency Checker will gather minimum and maximum metrics. It also shows the glyphs to which these metrics relate (identified by **GID**). This makes it easy to track down possibly problematic glyphs. Values which Consistency Checker considers to be wrong are colored in red.

Header	Name Versi	on St	tatistics Unicode Coo	le Page	
'head' table		GID	'CFF ' table		GID
xMin	-260.	99	FontBBox.left	-260	99
yMin	-242	72	FontBBox.bottom	-242	72
xMax	1130	56	FontBBox.right	1130	56
yMax	929	171	FontBBox.top	929	171
'hhea' table			'vhea' table		
advanceWidthMax	1164	122	advanceHeightMax	none	0
minLeftSideBearing	-260	99	minTopSideBearing	none	171
minRightSideBearing	-261	99	minBottomSideBearing	none	72
xMaxExtent	1130	56	yMaxExtent	none	72
'maxp' table					
maxPoints	none	152	maxCompositePoints	none	none
maxContours	none	122	maxCompositeContours	none	none
maxZones	none	none	maxTwilightPoints	none	none
maxStorage	none	none			
maxFunctionDefs	none	none	maxInstructionDefs	none	none
maxStackElements	none	none	maxSizeOfInstructions	none	none
maxComponentElements	none	none	maxComponentDepth	none	none
'OS/2' table					
usWinAscent	929	171	usWinDescent	242	72

Various metrics and the glyphs to which they relate. The value 'none' indicates that either the according table does not exist or that the table version does not include these data.

- Unicode Ranges

The Unicode Ranges section of the Consistency Checker shows the OS/2 table **ulUnicodeRange1-ulUnicodeRange4** values and presents individual bits and their meaning in a list. Editing behavior is described in context of the **>** *Codepage Ranges* section.

'OS/2' Table Character I	Ranges						
ulUnicodeRange1	-	0xa00002af	Unico	ode Character Range (Bits 0 - 31).			
ulUnicodeRange2		0x500078fb	Unicode Character Range (Bits 32 - 63				
ulUnicodeRange3		0x00000000		ode Character Range (Bits 64 - 95).			
ulUnicodeRange4		0x00000000	Unico	ode Character Range (Bits 96 – 127			
Bit	Range	Status		Description			
🗹 Unicode Bit 0	0x0000-0x007f	not available		Basic Latin			
🗹 Unicode Bit 1	0x0080-0x00ff	not available		Latin-1 Supplement			
🗹 Unicode Bit 2	0x0100-0x017f	not available		Latin Extended-A			
🗹 Unicode Bit 3	0x0180-0x024f	not available		Latin Extended-B			
Unicode Bit 3 Inicode Statistics: no			nt)	Latin Extended-B			

ulUnicodeRange bytes represented in hexadecimal form and as individual bits.



- Codepage Ranges

The Codepage Ranges section of the Consistency Checker shows the OS/2 table **ulCodePageRange1** and **ulCodePageRange2** values and presents individual bits and their meaning in a list.

'OS/2' Table Character Ri	2220			de Code Page			
ulCodePageRange1	0x40000093 Code Page Character Range (Bits						
ulCodePageRange2		ge Character Range (Bits 32	-63				
Bit	Code Page	Status		Description			
Code Page Bit 0	1252	not available		Latin 1			
Code Page Bit 1	1250	not available		Latin 2: Eastern Europe			
Code Page Bit 2	1251	not available		Cyrillic			
Code Page Bit 3	1253	not available		Greek			

ulCodePageRange bytes in hexadecimal form and as individual bits.

Clicking a checkbox will change the state of the according bit and highlight the adjustment in red – including in the hexadecimal representation:

Head		Version Statistics	Unico	ode Code Page
'OS/2' Table Character Ra ulCodePageRange1 ulCodePageRange2	inges	0x40000091 0x00000000		ge Character Range (Bits 0 – 31) ge Character Range (Bits 32 –63
Bit	Code Page	Status		Description
🗹 Code Page Bit 0	1252	not available		Latin 1
📄 Code Page Bit 1	1250	not available		Latin 2: Eastern Europe
Code Page Bit 2	1251	not available		Cyrillic
Code Page Bit 3	1253	not available		Greek
Click on values y			int)	Check Uncheck Apply

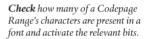
Accept adjustments by clicking the **Apply** button.

Your adjustments highlighted in red.



Upon clicking the **Check** button, the Codepage Ranges section (like the Unicode Ranges section) will show, in the **Status** column, how many of the characters referenced by each Codepage Range (or Unicode Range) are covered in the font's cmap table:

'OS/2' Table Character R		Version Statistics	Unicode Code Page					
ulCodePageRange1 ulCodePageRange2	0x00000001 Code Page Character Range (Bits 0 - 31							
Bit	Code Page	Status	Descrit	ation				
Code Page Bit 0	1252	205 of 206 codes mapp	ed Latin 1					
Code Page Bit 1	1250	216 of 217 codes mapp	ed Latin 2: Eastern	Europe				
Code Page Bit 2	1251	126 of 221 codes mapp	ed Cyrillic					
Code Page Bit 3	1253	138 of 206 codes mapp	ed Greek					



This also deactivates all bits except those that are relevant based on the actual character coverage. Again click **Apply** to accept the changes.



-Languages

The Languages section of the Consistency Checker compares the font's coverage of Unicode codepoints against the ICU database. It reports which scripts and languages are completely covered (green), incomplete (yellow) or are entirely missing (red). Scripts are first in hierarchy, languages are second and are shown after clicking the triangle in front of a script's name.

Head	OTF Consistency C er Name Version Statistics Uni	code Code Page Language
Head	er Name Version Statistics Oni	code Code Page Language
205		Charle Care Miner of Mining do DMD and
	es, 144 complete, 18 faulty, 43 missing	Giyph Set: Microsoft; Unicode BMP oni
hide faulty		0x0020:1 0x0021:2 0x0022:3 0x0023:4 0x0024:5
Script / Langua	0	! " # \$
▼ Latin	97 of 105 languages complete	
Afar	42 of 42 unicodes mapped (100.0%)	$0/2 \ 8_{\tau} \ () \ ()$
Afrik		+ZUL XL +++ (+)
Alban.		0x002x110x002b12 0x002c13 0x002d14 0x002e15
Azer	68 of 69 unicodes mapped (98.6%)	* +
Basque	56 of 56 unicodes mapped (100.0%)	··J·{··d···d···d/2-··d/2-···d·{·
Belar	64 of 64 unicodes mapped (100.0%)	0x002218 0x003017 0x003118 0x003219 0x003320
Bislam	a 52 of 52 unicodes mapped (100.0%)	/ 0 1 2 3
Bosnia	n 63 of 63 unicodes mapped (100.0%)	-3-F 1-2-F 1-2-F 1-2-F 1-2-F.
Breton		
	n 79 of 79 unicodes mapped (100.0%)	4 3 0 / 8
	56 of 56 unicodes mapped (100.0%)	0x0039.24(0x003x.27(0x003b.28(0x003c.29(0x003d.30
	56 of 56 unicodes mapped (100.0%)	
Com		
Czech		
Danish		
Dutch		
English		
Esper	. 64 of 64 unicodes mapped (100.0%)	

Tip: Don't forget to click *Check* so you see any results at all!

A character is missing for Azerbaijani.

To avoid getting lost in irrelevant information, you may **hide missing** scripts, hide missing languages, but also hide incomplete or hide complete entires.

If you select an incompletely covered language, you are shown a list of missing characters in the bottom right area:

Heade		Consisten Statistics	Unicode	Code Page	Language
Heade	r Name Version	statistics	Unicode	Code Page	Language
	, 144 complete, 18 fau		ng Unicode	e Glyph Mappi	ng: Latin/Azerb
hide missin hide faulty	g scripts 🗹 hide missin		0x004134	0x0042:35 0x0043:36	0x0044:37 0x0045:38
		ete		$\mathbf{R} \subset$	DF
ript / Languag	§ Status 97 of 105 languages of		XX		
Afar	42 of 42 unicodes ma		0x0046:39	0x0047:40 0x0048:41	0x0049:42 0x004a:43
Afar Afrik			F	G[H]	TT
	56 of 56 unicodes ma			╷╲┹ _╢ ╷┹╶┻╷	 ↓
Alban			0x004b:44	0x004c:45 0x004d:46	0x004e:47 0x004f:48
Basque				IM	NO
	64 of 64 unicodes ma			TA TAT	
	52 of 52 unicodes ma		0x0050:49	0x0051.50 0x0052.51	0x0053:52 0x0054:53
	63 of 63 unicodes ma		D	OR	C T
Breton				VΓ	
Catalan	79 of 79 unicodes ma		0x0055.54	0x0056.55 0x0057.56	0x0058.57 0x0059.58
	56 of 56 unicodes ma		TT	X7 XA7	VV
Chich	56 of 56 unicodes ma	pped (10			
Com	46 of 46 unicodes ma	pped (10	(fonte /byn	oPro-Regular.ot	
Czech	88 of 88 unicodes ma	pped (10	Unicode Gl Missing un	yph Mapping: La	tin/Azerbaijani
Danish	106 of 106 unicodes	mapped (0x02bc	(MODIFIER LETT	ER APOSTROPHE)
Dutch	64 of 64 unicodes ma				
English					
Esper	64 of 64 unicodes ma	pped (10			
Pri	int Status Print G	lyphs	Print Missin		eck Uncheck
	Fine d			g	Oncheck

This character is missing!



There are three printing options. **Print Status** ... reflects the window's leftside area, the script/language overview:

OTM Co	onsistency Checker	r - ArnoPro-			
Regular	otf - Language				mapped (100.0%)
	Script / Language	Status	Latin	Faroese	80 of 80 unicodes mapped (100.0%)
Latin	action / cangange	97 of 105 languages	Latio	Filan	45 of 45 unicodes
		complete	Later	-gas	mapped (100.0%)
Latin	Alar	42 of 42 unicodes	Latin	Filiping (Tagalog	54 of 54 unicodes
		mapped (200.0%)		and a second second	mapped (100.0%)
Latin	Afrikaans	80 of 80 unicodes	Latin	Finnish	72 of 72 unicodes
		mapped (100.0%)			mapped (100.0%)
Latin	Albanian	S6 of S6 unicodes	Latin	Flemish	64 of 64 unicodes
		mapped (100.0%)			mapped (100.0%)
Ladin	Azerbaijani	E8 of 69 unicodes thapped (98,6%)	Latin	French	\$5 of \$8 unicodes
Latin	Ranque	S6 of S6 unicodes			mapped (100.0%)
Lasn	an of the	mapped (200.0%)	Latin	Gaelic (Irish)	74 of 74 unicodes mapped (100.0%)
Latin	Selecution.	64 of 64 unicodes	Latio	Gaelic (Mata)	S& of S& unicodes
		mapped (100.0%)	Later	Caseoc (Matx)	mapped (100.0%)
Latin	Bislama	52 of 52 unicodes	Latin	Gaelic (Scottish)	72 of 72 unicodes
		mapped (100.0%)		Canal (accesses)	mapped (100.0%)
Latin	Bossian	63 of 63 unicodes	Latin	Geomet	66 of 66 unicodes
		mapped (100.0%)			mapped (100.0%)
Latin	Breton	67 of 67 unicodes mapped (100.0%)	Latin	German	67 of 67 unicodes
	Catalan	79 of 79 unicodes			mapped (100.0%)
Ladin	Caratan	mapped (100.0%)	Latin	Gikaya	41 of 41 unicodes mapped (100.0%)
Latin	Chamorro	S6 of S6 unicodes		Gibertese/Kinbati	24 of 24 unicodes
	Chandro	mapped (100.0%)	Latin	Cilbertete/Kindati	mapped (100.0%)
Latin	Chicheva	S6 of S6 unicodes	Latin	Greenlandic	#5 of #5 unicodes
		mapped (100.0%)			mapped (100.0%)
Latin	Comorian	46 of 46 unicodes	Litin	Guarani	62 of 62 unicodes
		mapped (100.0%)			mapped (100.0%)
Latin	Cauch	\$8 of \$8 unicodes	Litin	Haltian Creole	48 of 48 unicodes
		mapped (100.0%)			mapped (100.0%)
Latin	Danish	106 of 106 unicodes mapped (100.0%)	Latin	Hawalian	62 of 62 unicodes
Latin	Detrib	64 of 64 unicodes			mapped (100.0%)
		mapped (100.0%)	Latin	Hangarian	72 of 72 unicodes mapped (100.0%)
Latin	English	75 of 75 unicodes	Latin	kelandis	#3 of #3 unicodes
		mapped (100.0%)			mapped (100.0%)
Latin	Esperanto	64 of 64 unicodes	Latin	kao/kabo	54 of 56 unicodes
		mapped (100.0%)			mapped (96.4%)
	Extonian				

Print Glyphs... reflects the window's rightside area, the glyph overview:

	1		#	\$	%	&	1	Ţ,)	¥	1	S		C	a	«	-	-	Ľ
*	, + ,	,			1	0	,1,	2	3	-	0	±	2	3		μ	g		Ì,
4	5	6	,7,	8	9	÷	;	,<	E	1	0	»	1/4	1/2	3/4	ż	À	Á	į
>	?	@	A	B	C	D	E	F	G	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì,
H,	I	J	K	Ĭ.	М	N	0	Р	Q	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ì
R,	S	Ţ,	U	V	W	X	Y	Z	Ţ	x	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	Ì,
Y].	^	-	<u>,</u>	a	b	c	d	e	á	â	ã	ä	å	æ	ç	è	é	Ì,
f	g	h	i	j,	k	1	m	n	0	ë	ì	í	î	ï	ð	ñ	ò	ó	ļ
p	q	r	s	t	,u	v	w	x	y	õ	ö	÷	ø	ù	ú	û	ü	ý	Ì
- 7.	{	Ļ	}	~		÷	¢	£.	,¤	ÿ	Ā	ā	Ă	ă	Ą	ą	Ć	ć	(

Print Missing ... is a list of missing glyphs as of the bottom right area:

/forts/Ac Linicole Missing u Gx02bc	noPro-Regular.od Jyph Mapping: Latin/Azerbaijani iscoles: (MCDIFIER LETTER APOSTROPHE)			
		1		



Table Comparator

This tool displays the content of (mostly) fixed-length tables side by side. Currently the tables head, hhea, vhea, OS/2, maxp, post, CFF, PCLT and name are supported. All fonts' **Table Lengths** can be compared too. The leftmost column is reserved for the font which is selected in OTMaster's main dialog. It serves as reference font. Other fonts are compared against it. Identical values are marked in green, differences are marked in red.

Table Length	ns 'head' entries	'hhea' entries v	hea' entries OS/	2' entries 'maxp' entries 'post' header 'CFF ' top dictionary 'PCLT' entries Name Strings				
	ACasionPro-Regular.otf	ACaslonPro-Italic.otf	ACasionPro-Bold.otf	Description				
formatType	0x00030000	0x00030000	0x00030000	0x00010000 for version 1.0.				
talicAngle	0.000	-22.000	0.000	Italic angle in counter-clockwise degrees from the vertical.				
underlinePosition	-75	-75	-75	Distance top of underline from baseline (negative = below baseline).				
underlineThickness	50	50	50	Suggested value for the underline thickness.				
isFixedPitch	0	0	0	Set to 0 if the font is proportionally spaced.				
minMemType42	0	0	0	Minimum memory usage when downloaded.				
maxMemType42	0	0	0	Maximum memory usage when downloaded.				
minMemType1	0	0	0	Minimum memory usage when downloaded as a Type 1.				
maxMemType1	0	0	0	Maximum memory usage when downloaded as a Type 1.				

Table values can be edited right in the Table Comparator.



Glyph Copy Tool

This tool makes it easy to copy a glyph from one font to another, or to duplicate a glyph within the same font e.g. to add a same-looking but differently named and reencoded character.

	Glyph Copy Tool	
Select Source Glyp	h	
Source Font:		
/fonts/D019X	13T.otf	\$
Source Glyph In	dex (GID):	1
7		u u
Source Unicode		
0x0026		
Select Target Glyp	h	
Target Font: 🗌 :	same as source font	
/fonts/AP57X	12T.otf	\$
Glyph Index (GI	D) Selector:	7
7		u u
Supersede Ex Target Unicode	kisting Glyph	
0x0026		
🗹 Copy Instruc	tions	X
-	Font after Copy	

The Glyph Copy Tool's dialog.

Tip: Use the Glyph Copy Tool to copy outlines from CFF-based to glyf-based OpenType fonts and vice versa. It will automatically convert outlines into the required format and scale them according to the destination font's UPM as defined in the head table.

- Select Source Glyph

Source Font offers all fonts opened in OTMaster in a popup menu. To copy a glyph from a font not opened yet, use **Open Fonts**... to add it to popup list.

Source Glyph Index (GID) is the index of the glyph you want to copy. **Source Unicode** is the Unicode codepoint of the glyph you want to copy. So you may determine the source glyph either by its index, its Unicode codepoint, or by using the scrollbar next to the glyph image to flip through all glyphs and choose by appearance.

- Select Target Glyph

Target Font offers all fonts opened in OTMaster in a popup menu. In case that you intend to merely duplicate a glyph within the same font, activate the checkbox **same as source font**.

Glyph Index (GID) Selector does not have a function as long as you do not intend to supersede (i.e. replace) an existing glyph. Glyphs always are appended as the last glyph of the font.

Supersede Existing Glyph – rather than appending the new glyph to the glyph set, activating this option will replace an existing glyph which is defined by the **Target Glyph Index (GID)** right above the checkbox. **Target Unicode** is the Unicode codepoint for the new glyph.

Copy Instructions will copy hinting information with CFF-based OpenType fonts.

Note: When copying a glyph, the newly appended glyph's name is derived from its destination glyph index. To give it a friendlier name, you may want to open the Glyph Viewer and change the 'CFF' id or 'post' name.

Note: Hinting instructions are ignored with glyf-based fonts.

FoptMaster™

Save Target Font after Copy will save the target font immediately after appending or replacing a glyph. This is necessary for OTMaster to be able to rasterize the newly appended e.g. in the Font Viewer or Glyph Viewer.

The above image's settings are for appending a font's glyph to another font. As another scenario, the settings in the image below are such that a source glyph will supersede (i.e. replace) the 'same' glyph of another font – here we replace Argo's ampersand by Documenta's:

Tip: For this reason, it is highly recommended that you only edit copies of fonts with OTMaster, to prevent that you inadvertedly overwrite a font when using Save while editing. Either, make a copy of the font file, and edit the copy in OTMaster. Or, Save As... a font immediately after opening it, by another name.

Gly	yph Copy Tool
Select Source Glyph	
Source Font:	
/fonts/D019X13T.otf	:
Source Glyph Index (GID):	
7	
Source Unicode:	
0x0026	XT
	t t
Select Target Glyph	
Target Font: Same as se	ource font
/fonts/AP57X12T.otf	*
Target Glyph Index (GID):	
7	
Supersede Existing Gly	/ph
Target Unicode:	
0x0026	\mathbf{N}
Copy Instructions	
Save Target Font after	Copy
Open Fonts) Su	persede Target Glyph by Source Glyph



Glyph Editor

OTMaster is not intended to be a full font editor (font editor in the sense in which FontMaster or its components BezierMaster or IkarusMaster are). But we did not want to miss some basic glyph editing functionality even in a 'mere' table editor as OTMaster. What if you find duplicate contours in a glyph of a font ready to be distributed? 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? Glyph Editor helps.

OTM Glyph Editor - ArnoPro-Regular.otf - 4 - numbersign		
	S O Properties	3 O 957, 843 Tools
1) 2 = 1 2 i a	Display Options ✓ Image ✓ Baseline	
i 🖉 🛷 👟 🔏 🕥 🖆 🛛 🔶 🛋 🔿 诸 🖻	✓ Outline ✓ hhea asc/dsc ✓ Points Typo asc/dsc Indices Win asc/dsc Hints Winding Fill	₩ 44 •• C T X B ✓ 0 33 W L 13 q
11	Inflections Flat Curves V-Stems H-Stems D-Stems C-Stems	L M B
	V-Guidelines H-Guidelines Grid G-Guidelines	Extreme points max. shift 2 Remove small contours max. area 0
	Background	Check lines max. dev. 0.1 Remove double points max. dist. 0.1
	Bounding Box (108 points, 2 contours)	Double contours max. dist. 0.7
	minimum x: 15	Channelize max. dist. 0.7
	minimum y: 65	Check inflections (BE only) max. dev. 1.7
	maximum x: 427	Fit conic (IK only) max. dev. 2
	maximum y: 592	Sequence of contours
	Metrics Editor	Overlaps Open contours
	left side bearing: 16	Start point correction
	advance width: 453	Display only
	right side bearing: 25	Clear marks Execute
	Tools Properties	Tools Properties

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 which is new in OTMaster 3.6. 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.



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. OTMaster 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 ...

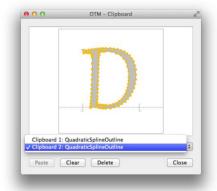
Opens the Clipboard window. There can be multiple Clipboards now, for collecting outlines from the last couple of **Cut** or **Copy** actions.

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.

🐔 O	TM File	Edit	View	Tools	
	ę.	Import		て 企業 0	_
		Export	'	℃ ଫ ജ S	
		Print		жP	
	6	Save		жs	
	La	Messag	jes	жL	
	6	Clipboa	ard	٧ж٦	



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.



DTL OTMASTER: TOOLS MENU



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.

ն Сору

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

35

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

Previous



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.

DTL OTMASTER: TOOLS MENU



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 the Glyph Editor.

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 the Glyph Editor: Glyph Index (GID), Unicode 2+ semantics BMP only, Macintosh Roman, or Microsoft Unicode BMP only.

Reset

This will restore the default glyph size.

 OTM
 File
 Edit
 View
 Tools

 Tool Bars
 ►
 Dock Widgets
 ►

 Display Options
 ►
 C(lynh Set
 ►

 Reset
 F5
 ►
 ►

Display Options as of the **Properties** dock widget.

9	Properties
Display Options	
✓ Image	✓ Baseline
✓ Outline	✓ hhea asc/dsc
Points	typo asc/dsc
Indices	win asc/dsc
Hints	Winding Fill
Inflections	Flat Curves
V-Stems	H–Stems
D-Stems	C-Stems
V-Guidelines	H-Guidelines
Grid	G-Guidelines
Background	
Bounding Box (108 point	s, 2 contours)
minimum x:	15
minimum y:	65
maximum x:	427
maximum y:	592
Metrics Editor	
left side bearing:	16
advance width:	453
right side bearing:	25



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.

00		Tools						
9	3	hulml		S	3	\odot		
H	as		C	T	X	B		
\checkmark		123	\mathbb{N}	Ц	aa	q		
1	K	Ý	B					
			Zoom					
	no p	oaramete	ers or opt	tions req	uired			
	Tools Properties							

View Tools Con Zoon жu Scroll by Hand ₩Leertaste 💷 Measure Grid Guidelines Scale S Rotate R Affine Transformation Α # Italization £۱ as Mirror and Fold F Hidden Lines (H, L) Contouring (C, N) T-Disconnector т X-Disconnector х I-Disconnector Improve ... S Sense of Rotation 3 Sequence of Points / Contours Merge м L Character Hinting aa Review Changes 7 q Quick Mode Q / Digitize ٨ 🗸 📐 Shift < Shift Smooth B Background Glyph В

Scroll by Hand

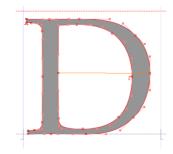
Click anywhere into the editing area, hold the mouse button down and move the mouse around to scroll the entire editing area.



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.

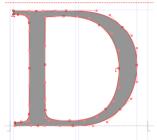




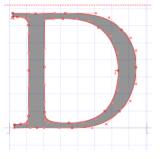
Fopt Master™

Guidelines and Grid

Define a variety of horizontal and vertical **Guidelines**, either at glyphspecific positions, taken from LSB, RSB, Xmin, Xmax, Center [of outline] width, Center [of] total width, or at manually defined positions. Furthermore, guidelines may indicate an Em square at desired **Size** and **Offset**, plus a extra **Frame within** [the] **Em square**. Finally, guidelines may visualize the **Italic angle**.

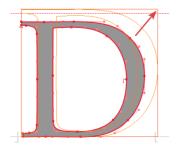


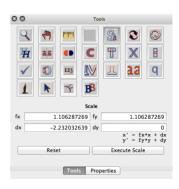
Display a **Grid** in the background. You may define the line-to-line distance in font units or as lines per EM square, decide at which origin point the grid is meant to start, and request that the grid be displayed only within the Em square.



Scale

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



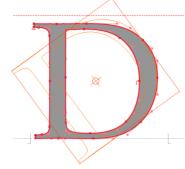




Fopt Master™

Rotate

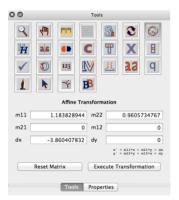
Use the mouse to rotate the glyph in the edit area, or enter a **rotation angle** in the editing area (a negative value is for clockwise rotation) and then **Execute Rotation** to apply the rotation or **Reset**.

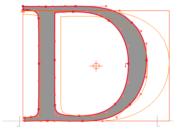


80	Tools						
	Rotate						
	rotation angle	36.07458144					
xc	0 ус	0					
rotation cen	er:						
	Em square selection rectangle						
		🗌 reset after execute					
	Reset	Execute Rotate					
	Tools Pro	perties					

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 italizise, 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**.







2

aa

q

₸ 🗶 🗄

W

Mirror and Fold

Execute Mirror / Fold

Tools Properties

00

als

123

mirror left / right mirror top / bottom mirror top left / bottom right fold bottom to top fold left to right fold up to bottom fold right to left fold quadrant l (right top) fold quadrant II (left bottom) fold quadrant IV (right bottom)

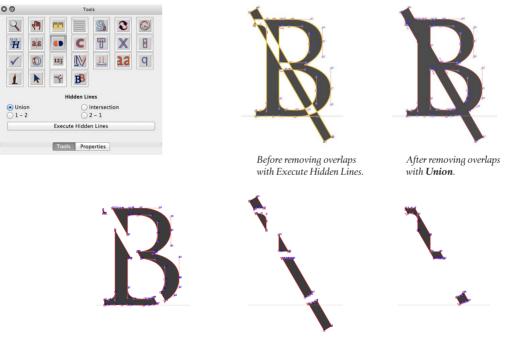
Mirror and Fold

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.



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.

DTL OTMASTER: TOOLS MENU



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).



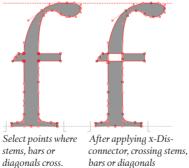
T-Disconnector

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

00		Tools					
9	3	Indial		S	2		
H	as		C	T	X	B	
\checkmark		123	\mathbb{N}	Ц	aa	q	
1		Ý	B				
		т-	Disconne	ctor		_	
Overlap	20					2	
		Execut	e T-Disc	onnector	2		
					3		
		Tool	s Pro	perties			

x-Disconnector This works like the **T**-Disconnector.

Q	۹D	Inimi	Тос	Q		0
	57			28	-	
H	as		C	T	X	H
\checkmark		123	\mathbb{N}	Ц	aa	q
1	K	Ý	B			
Overlap		x -	Disconne	ctor		
		Execut	e X-Disc	onnector		



will be disconnected.



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



I. 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-Disconnector transformation area, 3. click Execute T-Disconnector to apply disconnection:





IN

I-Disconnecto

Execute I-Disconnector

XB

aa q

00

al

123

Overlap 20

I-Disconnector

The I-Disconnector 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 1-Disconnector, the stem or bar will be split into two.

Improve

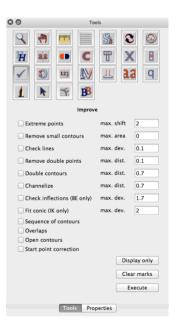
Outlines are not always flawless from a technical point of view. The image to the right shows aspects that OTMaster can improve. You may merely review OTMaster's finding with **Display only** (and then **Clear marks** again) or apply suggested improvements with **Execute**.

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 glyf-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:

80	Tools						
	123 **	■		2 X 22	© 8 9		
Se	quence o	of Points	/ Contou	irs			
Move to End Move to Start Automatic Move by Mouse Click							
Move Contours							
Tools Properties							

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.

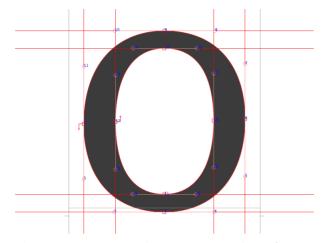
Time to remove contour overlaps with Hidden Lines!

Image: Second	0 U 100IS	
Fort /forts/H022X13Doriginal.of Open Fonts Clyph Clyph Clyph Clyph Index (CID): 36 Unicode: 0x0043 Adjustment SIS8 / LS8 Center Width Center Total Width Numeric: X 0 Y 0 Options y-Offset 0 Execute Merge	₩ as • C ¶ ✓ 0 ¤3 ₩ ⊥	XE
//fonts/H022X13Doriginal.otf ÷ Open Fonts Glyph Glyph Glyph Glyph Index (GID): 36 Unicode: 0x0043 Adjustment 0 SB / RSB OSB / RSB CBB / RSB Center Width Center Total Width Options 0 Y-Offset 0 Execute Merge 0	Merge	
Open Fonts Clyph Clyph Clyph Clyph Clyph Index (CID): 36 Unicode: 0x0043 Adjustment OxS8 / K58 Center Width Center Total Width Numeric: 0 V 0 Options Y-Offset 0 Execute Merge		
Clyph Clyph Glyph Glyph Glyph Index (CID): 36 Unicode: 0x0043 Adjustment State		÷
Clyph Index (CID): 36 Unicode: 0x0043 Adjustment © LSB / LSB © Cherer Width Numeric: X 0 Y 0 Options y-Offset 0 Keep Character Width		
Glyph Index (GID): 36 Unicode: 0x0043 Adjustment © LSB / LSB Caster Vident © Center Width Center Total Width Numeric: 0 V 0 Options Y-Offset 0 Rece Character Width Execute Merge	Glyph	48
Unicode: 0x0043 Adjustment SIS8 / LS8 / RS8 Center Width Center Total Width Options y-Offset Execute Merge	G	
Adjustment • JS8 / LS8 CSB / RS8 RS8 / LS8 Center Width Center Width Center Total Width Options y-Offsel 0 Keep Character Width	Glyph Index (GID):	36
LSB / LSB / LSB / LSB / RSB / LSB / RSB / R	Unicode:	0x0043
y-Offset 0 Keep Character Width Execute Merge	LSB / LSB LSB / LSB LSB / LSB / LSB / I RSB / RSB / RSB / RSB / I Center Width Center	SB Total Width
Keep Character Width Execute Merge	Options	
		0
		R



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 \blacktriangleright *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.



Fopt Master™

Digitize

Time to reanimate your Aristo tablet!

80	Tools					
	■ ■ Sa C G • C T X I ■ M I 22 C	2 1 1				
	Digitize					
Mouse button u	sage:					
#1 Left: Create	and drag contour start point.					
	#2 Middle: Append and drag anchor point, close contour by hitting the start point.					
#3 Right: Append and drag control point, close contour by hitting the start point.						
#4 Left+%: Not in use.						
#5 Middle+% : Force closing contour whithout hitting the start point.						
#6 Right+%:No	t in use.					

Shift

Shift by mouse, at **arrow shift base** units, or shift numerically, at **dx** and **dy** units. Additional **paste & shift dx** and **past & 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.

0			То	ols		
< ₩ ✓	 ■ ■	123	■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■		2 X 23	e e e e e e e e e e e e e e e e e e e
		s	hift Smo	oth		
🗹 dx	-32	2.92592	593 💌	dy	16.46	296296
					× y	x + dx y + dy
arrow sh	hift base	2				1
paste &	shift dx					200
paste &	shift dy					100
	Res	et		Exe	cute Shift	t
		Tool	s Pro	perties		

Shift Smooth

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

0		Too	ls		
	123 **	C 1000 1000 1000 1000 1000		2 X 23	l P
		Shift			
🗹 dx		0 🗹	dy		0
				x' = y' =	x + dx y + dy
arrow shift base					1
paste & shift dx					410
paste & shift dy					205
Rese	t		Exec	ute Shift	

Mouse and keyboard usage:

Use middle mouse button to insert an anchor point on nearest contour, or to turn straight section into curve and vice versa by hitting a point.



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 OTMaster 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.

0		Tools	
	Backgro	ound Glyph	
Show Backgro	ound Glyph		
Font			
/fonts/AP57X	12T.ttf		\$
Open Fonts .			
Glyph			
Glyph Index (G	ID):		35
Unicode:			0x0042
Position	oreground		
LSB / LSB XMIN / XM Center Wid		C RSB / RSB XMAX / XMAX	:
dx:	0	dy:	0
Display Options			
✓ Image Points Baseline typo asc/ds	c	Outline Indices hhea asc/dsc win asc/dsc	
	Tools	Properties	

DTL OTMASTER: TOOLS MENU



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 \triangleright *Review Changes*.

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



Preferences

The first option relates to CFF-based OpenType fonts:

-File Options

Outline Conversion Options determine how cubic beziers are converted into quadratic beziers, i.e. these options affect only the convertion from CFF-based to glyf-based OpenType fonts. (The other way round is lossless anyway.)

erences		Options	
File Options Glyph Editor Options	Outline Conversion Options		
Shortcuts	Outline Tolerance:	15	
	Double Point Tolerance:	1	
	Extreme Point Tolerance:	30	
	OTF Options		
	Subroutinize charstrings when write	ting 'CFF ' table.	
		ing 'CFF ' table.	

Outline Tolerance restricts how much converted outlines may deviate from original outlines. **Double Point Tolerance** defines how close two points may get. If they get closer than that, one of them will be deleted. **Extreme Point Tolerance** relates to missing extrema. If an extremum point is missing, it will be inserted. If another point is too close to this new extremum point, i.e. within the specified distance, then this other point will be deleted.

The single **OTF Option** named **Subroutinize charstrings when writing** '**CFF' table** serves to compress the CFF table by finding common contour segments in all glyphs' outline descriptions, creating a dictionary of these, and referencing them.

The following set of options relates to the Glyph Editor:

-User Interface

The option **Point Snap Distance** defines how far away from an outline or point the mouse may be while still being able to select them.

Zoom Rate Up/Down defines how much to zoom with every zoom step.

eferences		User Interface	
File Options Glyph Editor Options User Interface	lcon Size [pixel]:	32 🗘	
Foreground Chara Background Char Autohinting Shortcuts	Point Snap Distance [pixel]: Zoom Rate Up/Down [%]:	5 200	
Reset Restore Defa	ults Apply		Cancel

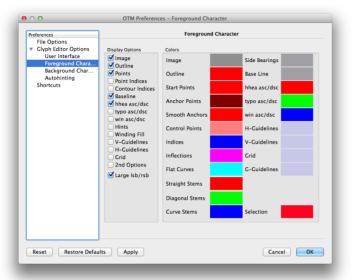
Tip: You can adjust the Icon Size since OTMaster 6!



-Foreground Character

-Background Character

There are two sets of options for **Foreground Character** and for **Background Character** the latter of which has a few options less:



Checkboxes in the Display Options area define which information the

Glyph Editor will visualize by default:

Image shows filled shapes.

Outline adds a colored outline.

Points shows on-curve points.

Point Indices shows points' index numbers.

Contour Indices shows contours' index numbers.

Baseline shows the baseline.

hhea asc/dsc are the hhea table's ascender/descender heights.

typo asc/dsc are the OS/2 table's sTypoAscender/sTypoDescender height.

win asc/dsc are the OS/2 table's usWinAscent/usWinDescent heights.

Hints shows a glyph's hints.

Winding Fill chooses the non-zero winding rule over the (default) even-odd rule for determining glyphs' fill.

v-Guidelines and H-Guidelines display the respective guidelines.

Grid shows the grid.

2nd Options helps toggle between two sets of Colors.

Large lsb/rsb displays sidebearings by full-size vertical lines to the left

and right of a glyph. Normally, and in best Ikarus tradition, sidebearings are indicated by small stubs sitting on the baseline.

In the **Colors** area you may adjust the colors of these pieces of information.



-Autohinting

These options determine how autohinting will work:

Preferences		Autohinting		
File Options Glyph Editor Options User Interface	Recognition of Stems	Key Characters		
Foreground Chara Background Char	Straight Stems Curve Stems	Key	Unicode	GID
Autohinting Shortcuts	Diagonal Stems Tolerance for Recognition of Straight Stems [units]: 0.6666666667 Minimum Length of Straight Stems [units] 3.33333333 Base Size [unitsPerEm]:	baseline overshoot:	x0000004f	none
		ascender:	none	none
		xheight: of xheight overshoot: 333 capheight:	none	none
			(0000078	none
			x0000006f	none
			x00000048	none
	1000	figureheight:	(0000035	none
		figureheight overshoot:	(0000030	none
		descender:	(0000070	none
		Ignore Baselines		
	Autohint all unhinted glyp	bhs when saving font.		

In the **Recognition of Stems** area you may choose whether or not to recognize **Straight Stems**, **Curve Stems** and **Diagonal Stems**.

Tolerance for Recognition of Straight Stems allows recognizing stems even if they are not exactly horizontal or vertical.

Minimum Length of Straight Stems defines how long straight segments need to be to be considered as being part of stems.

Base Size is the font's UPM size – the former two values relate to it.

The **Key Characters** area offers an easy way to define alignment zones by referring to glyphs from which these values are to be derived. Glyphs are identified either by **Unicode** codepoint or by **GID**. The **baseline** value usually is zero and does not need to be extracted from any glyph.

By default, OTMaster requires that you autohint glyphs individually in the Glyph Editor. This is to make sure that existing hinting will not get lost. However, you may choose to **Autohint all unhinted glyphs when saving font**.



And finally there are the ...

- Shortcuts

Here you may check, or redefine, which shortcuts are associated with which of OTMaster's actions:

Preferences File Options		Shortcuts			
Glyph Editor Options	#	Action	Portable	Native	
User Interface Foreground Chara	1	Revert to Saved	Ctrl+Alt+Z	∖₩Z	
Background Char	2	Undo	Ctrl+Z	жz	
Autohinting Shortcuts	3	Redo	Ctrl+Shift+Z	ሰ₩Z	
	4	Delete	Backspace; Del	<2; ≥	
	5	Cut	Ctrl+X	жx	
	6	Сору	Ctrl+C	жc	
	7	Paste	Ctrl+V	₩V	
	8	Paste & Shift	Ctrl+Shift+V	企業V	
	9	Grow	Ctrl+G	ЖG	
	10	Previous	Alt+Left		
	11	Next	Alt+Right	$\Sigma \rightarrow$	
	12	Next File Above	Ctrl+Up	96 1	
	13	Next File Below	Ctrl+Down	96 1	
	14	Select All	Ctrl+A	96A	
	15	Zoom In	U; +	U; +	

Each action has a number, a name, a transcription of the shortcut (**Portable**) and a visualization of it (**Native**).

To adjust an action's shortcut, click into a cell in the **Portable** column and edit the cell's text. An action may be associated with more than one shortcut. In this case, a semicolon serves as a separator, as can be seen with action 15, 'Zoom In', which has two shortcuts enumerated as 'U; +'.



Function	Mac osx	Windows	Linux
Main Dialog			
File Menu			
Open	CMD+O	CTRL+O	CTRL+O
Close	CMD+E	ctrl+E	ctrl+E
Close All	CMD+SHIFT+E	ctrl+shift+E	ctrl+shift+E
Save	CMD+S	CTRL+S	CTRL+S
Save As	CMD+SHIFT+S	CTRL+SHIFT+S	CTRL+SHIFT+S
Save All	CMD+ALT+S	CTRL+ALT+S	CTRL+ALT+S
Preferences	CMD+,	CTRL+,	CTRL+,
Messages	CMD+L	ctrl+L	ctrl+L
Clipboard	CMD+ALT+V	CTRL+ALT+V	CTRL+ALT+V
Quit	CMD+Q		ctrl+Q
Edit Menu			
Cut	CMD+X	CTRL+X	CTRL+X
Сору	CMD+C	CTRL+C	CTRL+C
Paste	CMD+V	CTRL+V	CTRL+V
Grow	CMD+G	ctrl+G	ctrl+G
View Menu			
Number as Decimal	Ø>D	Ø>D	Ø>D
Number as Hexadecimal	Ø>D Ø>X	Ø > X	Ø>D Ø>X
	Ø>A Ø>C	Ø>X Ø>C	Ø>X Ø>C
Number as Composite	0>0 0>0	0>C 0>0	Ø>0 Ø>Ø
Number as Appropriate Data as Text	очо смр+Т	020 CTRL+T	020 CTRL+T
Data as Text	CMD+1	CTRL+ I	CTRL+1
Table Overview			
File Above	CMD+	CTRL+	CTRL+
File Below	СМD+↓	CTRL+↓	CTRL+↓
Application-wide			
11			

Close Window

CMD+W

Info: If there is a + inbetween two keys, hold the first key and press the second key. If there is a > inbetween two keys, press the first key and then press the second key shortly thereafter.



Function	Mac osx	Windows	Linux
Glyph Editor			
File			
Import	cmd+alt+shift+O	CTRL+ALT+SHIFT+O	CTRL+ALT+SHIFT+O
Export	$\mathbf{CMD} + \mathbf{ALT} + \mathbf{SHIFT} + \mathbf{S}$	CTRL+ALT+SHIFT+S	CTRL+ALT+SHIFT+S
Print	CMD+P	ctrl+P	CTRL+P
Edit / Selection			
Revert to Saved	CMD+ALT+Z	CTRL+ALT+Z	CTRL+ALT+Z
Undo	CMD+Z	CTRL+Z	CTRL+Z
Redo	cmd+shift+Z	CTRL+Y	CTRL+SHIFT+Z
Delete	BACKSPACE or DEL	BACKSPACE or DEL	BACKSPACE or DEL
Cut	CMD+X	CTRL+X	CTRL+X
Сору	CMD+C	ctrl+C	ctrl+C
Paste	CMD+V	CTRL+V	CTRL+V
Paste & Shift	$\mathbf{CMD} + \mathbf{SHIFT} + \mathbf{V}$	$\mathbf{CTRL} + \mathbf{SHIFT} + \mathbf{V}$	\mathbf{CTRL} + SHIFT + V
Previous	ALT+←	CTRL+←	CTRL+←
Next	$ALT + \rightarrow$	$CTRL+\rightarrow$	$CTRL+\rightarrow$
New Character	CMD+N	CTRL+N	ctrl+N
Duplicate Character	CMD + SHIFT + C	$\mathbf{CTRL} + \mathbf{SHIFT} + \mathbf{C}$	$\mathbf{CTRL} + \mathbf{SHIFT} + \mathbf{C}$
Swap Background			
Character	1	1	1
Select All / Deselect All	CMD+A	CTRL+A	CTRL+A
Selection Levels			
Select Points	I	I	I
Select Contours	2	2	2
Select Groups	3	3	3
Select Character	4	4	4
			•
View			
Zoom In	U	U	U
Zoom Out	D	D	D
Reset (to default size)	F5 or CMD+R	F5 or CTRL+R	F5 or CTRL+R

Info: If there is a + inbetween two keys, hold the first key and press the second key. If there is a > inbetween two keys, press the first key and then press the second key shortly thereafter.



Function	Mac osx	Windows	Linux
Tools			
Zoom	CMD+U or CMD++	CTRL+U or CTRL++	CTRL+U or CTRL++
Scroll By Hand	CMD+SPACE	CTRL+SPACE	CTRL+SPACE
Measure	= or ENTER	= or ENTER	= or ENTER
Grid	#	#	#
Guidelines	1	1	
Adjust to Guidelines	G	G	G
Scale	S	S	S
Scale Intelligent	shift + S	shift+S	shift+S
Rotate	R	R	R
Affine Transformation	Α	Α	Α
Italization	CMD+I	ctrl+I	ctrl+I
Mirror and Fold	F	F	F
Hidden Lines	H>L	H>L	H>L
Contouring	C>N	C>N	C>N
T-Disconnector	Т	Т	Т
X-Disconnector	Х	Х	Х
I-Disconnector	Ι	Ι	Ι
Improve	\$	\$	\$
Rounding	0	0	0
Stem Adjustment	CMD+B	CTRL+B	CTRL+B
Sense of Rotation			
Sequence of Points			
and Contours	:	:	:
Merge	Μ	Μ	Μ
Interpolation	%	%	%
Autohint	!	!	!
Review Changes	?	?	?
Contruction	K	K	K
Quick Mode	Q	Q	Q
Digitize	^	٨	۸
Shift	<	<	<
Shift Smooth	>	>	>
Background Glyph	В	В	В
Channelize	C>H	C>H	C>H
Crop	C>R	C>R	C>R

Info: If there is a + inbetween two keys, hold the first key and press the second key. If there is a > inbetween two keys, press the first key and then press the second key shortly thereafter.



Function	Mac osx	Windows	Linux
Display Options			
Image	,>I	,>I	,>I
Outline	,>0	,>0	,>O
Points	,>P	,>P	,>P
Point Indices	,>N	,>N	,>N
Contour Indices	,>M	,>M	,>M
Baseline	,>Ø	,>Ø	,>Ø
hhea asc/dsc	,>I	,>I	,>I
type asc/dsc	,>2	,>2	,>2
win asc/dsc	,>3	,>3	,>3
Inflections	,>S	,>S	,>S
Flat Curves	,>L	,>L	,>L
H-Stems	,>H	,>H	,>H
V-Stems	,>V	,>V	,>V
D-Stems	,>D	,>D	,>D
C-Stems	,>C	,>C	,>C
Hints	,>E	,>E	,>E
H-Guidelines	,>X	,>X	,>X
V-Guidelines	,>Y	,>Y	,>Y
G-Guidelines	,>Z	,>Z	,>Z
Grid	,>G	,>G	,>G
Target Image	,>T	,>T	,>T
Target Outline	,>U	,>U	,>U
Background	,>B	,>B	,>B
Winding Fill	,>F	,>F	,>F
2nd Option	+	+	+

Info: If there is a + inbetween two keys, hold the first key and press the second key. If there is a > inbetween two keys, press the first key and then press the second key shortly thereafter.

ACKNOWLEDGEMENTS

DTL OTMaster Manual

Edition 6.0.0/2016; reflects DTL OTMaster version 6.3 © 2016 Dutch Type Library and URW++ Design & Development

Text,

Design Karsten Lücke Typeset in DTL Argo, DTL Haarlemmer and DTL Haarlemmer Sans

Thanks to Microsoft for allowing the inclusion of the OpenType specs

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

URW++ Design & Development GmbH

Poppenbütteler Bogen 36 22399 Hamburg Germany phone +49 (0)40 60 60 52 28 fax +49 (0)40 60 60 51 11 e-mail info@urwpp.de website www.urwpp.de



The third party product names used in the DTL OTMaster 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.