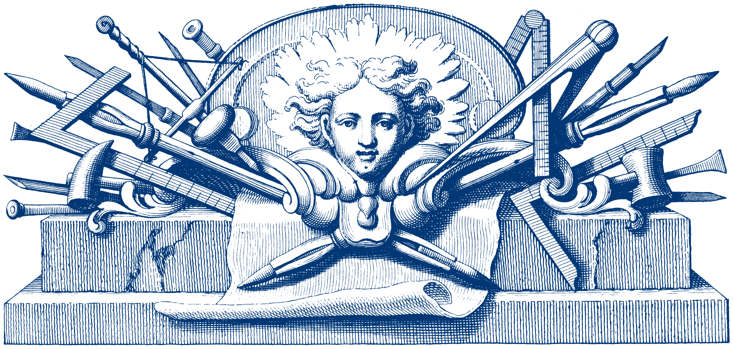
Dutch Type Library

FM 2.5.2 Supplement

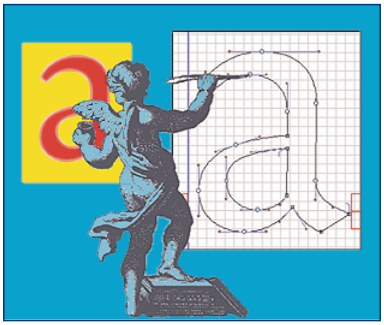


FontMaster™ *Utilities*

’s-Hertogenbosch / Hamburg
Autumn 2006

Dutch Type Library

DTL BezierMaster



’s-Hertogenbosch / Hamburg
Autumn 2006

New functionality in DTL BezierMaster

a. Direct import and export of PostScript Type 1, TrueType and OpenType formats.

b. The display of anchor and control points can be changed to small, medium and large in the 'Preferences…' dialog.

c. New function *Fit point to guideline*. This makes it possible to create, delete and shift guidelines, the latter using the pointer tool or arrow keys. The guidelines are automatically stored in the guideline array, which can be edited numerically.

There is a range of new short cuts, including the following ones for the guidelines:

–Click: Insert vertical line.
–Click + <ctrl>: Insert horizontal line.
–Click + <shift>: Adjust point to vertical line.
–Click + <ctrl> + <Shift>: Adjust point to horizontal line.
–Click + <alt>: Delete line (Windows).
–-Click + <⌘>: Delete line (Mac OS).
–Click + <alt> + <ctrl>: Select line.
–Arrow keys: Shift selected line by 1 unit.
–Arrow keys + <Shift>: Shift selected line by 10 units.
–Click + <ctrl> + <Shift>: Adjust to next guide line.
–Click + <alt> + <Shift>: Align horizontal or vertical.

d. The background BE, IK or SC image can be moved using the arrow keys now.

e. The drawing functionality has been extensively enhanced. Especially the pointer tool can be used now in combination with range of keys to insert and convert points, to fit points to guidelines, to align points horizontally and vertically, to place BCP's on the control points, et cetera.

f. Batch functionality for the improver functions (i.e. for a complete glyph database at once).

g. In the Metrics Editor an AFM file can be generated –if not existing– and changes to an AFM file can be directly saved.

h. The interface has been improved and the menu structure changed; for instance the *Config* menu has been removed and the contained 'Settings' has been moved to 'Preferences…' in the *File* or *Edit* menu (depends on active window).

i. Contouring: last menu entries are stored now.

j. 'New Character': default is 1 now, not 0 (zero is actually not allowed).

k. New icons have been made for the program, functions and menus.

l. New batch functionality in the Font Administration tool.
There is now an *Import* function in the Batch menu:

1. You can browse to select another font
2. You can select all characters or a groups of characters like 101–105 , 301–305, 501–510
3. You can specify the target number for the groups you have selected

In the example above the characters 101–110 will be copied to 4101–4110 and the characters 301–310 will be copied to 4301–4310.

4. You can decide to supersede existing characters or not.

Clicking 'OK' will import all selected characters into your font.
In case that the bodysize of the two fonts is different you will be asked to scale the characters to the target bodysize or leave them as they are.

There is an option to execute a command file to import glyphs from different other fonts. The command file syntax is similar to the one used in Ikarus v4 and quite simple. A command file must be text file (.txt) and can look like this:

```
input D:\archive\ar0000\A057016T.be
chg 101-126
cts 3101
run
input D:\archive\ar0000\A066036T.be
chg 301-326,501-510
cts 3301,3501
run
end
```

A command allows to merge many different font files into one with a simple click. It can for example also be used to regroup and reorder characters if one uses a copy of the edited font as input.

If a command file is specified in the dialog window, all other information (such as the check box 'superseed chars', the source file, character groups, target character numbers) is ignored completely by the program. Instead the program only uses the content of the txt file, i.e. 'ssc 0/1', 'input' et cetera.

Command file syntax:

Input	specifies the input font file.
chg	specifies the characters and character groups to be copied
cha	specifies to copy all characters from the source font.
cts	specifies the target character numbers

If no cts command is present, the original character numbers will be used.
If a cts command is present, the number of characters specified in the cts

command must be equal to the number of character (group) in the chg command.
For example:

```
input c:\archive\xyz.ik
chg 101-103,301-305,501,602,
cts 1101,1301,1501,1602
```

This will copy the characters 101, 102, 103, 301, 302, 303, 304, 305, 501, 602 from the input file xyz.ik to the following numbers in the currently edited font:

```
101 -> 1101
102 -> 1102
103 -> 1103
301 -> 1301
302 -> 1302
303 -> 1303
304 -> 1304
305 -> 1305
501 -> 1501
602 -> 1602
```

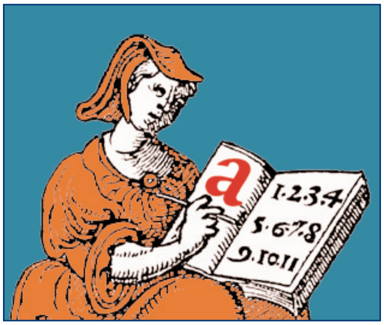
run will start the copying process
ssc takes one integer parameter, 0 or 1.
Specifies whether characters in the target font should be overwritten by characters from the input font. Default is not to overwrite existing characters.
ssc 1 will cause characters to be superseeded.
ssc 0 will prohibit superseeding of characters (default).
fbs takes one integer parameter, 0 or 1.
Specifies if characters are scaled to the bodysize of the target font, if the bodysizes of source and target font differ.
fbs 1 will cause characters to be scaled to the target bodysize (default) and fbs 0 will cause characters not to be scaled to the target bodysize.
End stops processing of the copying process
If no ‘end’ command is specified, all commands until eof will be read.
c introduces a comment line, which will be ignored by the program.

Sample file:

```
input c:\archive\abc.ik
chg 101,102,103
ssc 1
fbs 1
run
input c:\otherdir\subdir\xyz001.ik
chg 1000-2000
cts 5000
fbs 0
ssc 0
run
end
```

Dutch Type Library

DTL DataMaster



’s-Hertogenbosch / Hamburg
Autumn 2006

New functionality in DTL BezierMaster

a. Standard encoding for OpenType production.

b. Command file support for running program in full batch.

A command file can be simply generated:

– Generate a font first.

–Use ‘List’ in command mode and a listing of all settings will be shown.

– Save this file with the extension .txt or .cmd

This command file can be altered with a normal text editor. To make it executable some entries have to be modified:
‘Command Cancel’ must be disabled. This can be done for instance by removing the line or writing a number sign (#) at the beginning which will mark this line as comment.

After editing you can use the run button to select the command file and to produce the font(s). You can include other files, for example to create a TrueType font, a PostScript Type 1 font and an OpenType font at the same time or use the ‘CommandOpen’ command for activating other files.

An example of a command file for OpenType production:

```
# DTL DataMaster 2.5.0 Command File
# Select Character Layout File and Code Page.
CharacterLayoutFile D:\FM\Release-2.5.1-Full\urwotf.cha
CharacterLayoutCodePage PostScript-Number
# CharacterLayoutUseStandard false
# Set Basic Em-Square.
BasicEmSquare 15000
# Select Basic Format.
BasicFormat BE Format
# Select Basic Font Directory.
BasicFontDirectory D:\archive\ar0008
# Select Basic Font(s).
# SelectBasicFont all
SelectBasicFont none
SelectBasicFont D027013T.be
# Set up Target Directories for all Export Formats.
ExportFontFormat TrueType Format
ExportTargetDirectory D:\archive\ar0008
ExportFontFormat PostScript Type 1 Format
ExportTargetDirectory D:\archive\ar0008
ExportFontFormat OpenType (CFF)
ExportTargetDirectory D:\archive\ar0008
ExportFontFormat OpenType (TT outlines)
ExportTargetDirectory D:\archive\ar0008
```

```
# Select Export Format.
# NOTE : This will also select the related target directory.
ExportFontFormat OpenType (CFF)
# Select Export Target Directory.
# ExportTargetDirectory D:\archive\ar0008
# Cancel Command Mode.
# NOTE : Remove the following line to continue execution.
# CommandCancel
# Avoid Export Font Dialog.
ExportSkipDialog true
# Avoid existing file overwrite confirmation dialog.
OverwriteExistingFiles true
# Optionally include other command files, e.g.:
# CommandOpen Setup1.cmd
# CommandOpen BESelection1.cmd
# Start Font Export.
StartFontExport
```

Dutch Type Library

DTL IkarusMaster

’s-Hertogenbosch / Hamburg
Autumn 2006

New functionality in DTL IkarusMaster

a. Direct import and export of PostScript Type 1, TrueType and OpenType formats.

b. The display of anchor and control points can be changed to small, medium and large in the 'Preferences…' dialog.

c. New function *Fit point to guideline*. This makes it possible to create, delete and shift guidelines, the latter using the pointer tool or arrow keys. The guidelines are automatically stored in the guideline array, which can be edited numerically.

There is a range of new short cuts, including the following ones for the guidelines:

–Click: Insert vertical line.
–Click + <ctrl>: Insert horizontal line.
–Click + <shift>: Adjust point to vertical line.
–Click + <ctrl> + <Shift>: Adjust point to horizontal line.
–Click + <alt>: Delete line (Windows).
–-Click + <⌘>: Delete line (Mac OS).
–Click + <alt> + <ctrl>: Select line.
–Arrow keys: Shift selected line by 1 unit.
–Arrow keys + <Shift>: Shift selected line by 10 units.
–Click + <ctrl> + <Shift>: Adjust to next guide line.
–Click + <alt> + <Shift>: Align horizontal or vertical.

d. The background BE, IK or SC image can be moved using the arrow keys now.

e. The drawing functionality has been extensively enhanced. Especially the pointer tool can be used now in combination with range of keys to insert and convert points, to fit points to guidelines, to align points horizontally and vertically, to place BCP's on the control points, et cetera.

f. Batch functionality for the improver functions (i.e. for a complete glyph database at once).

g. In the Metrics Editor an AFM file can be generated –if not existing– and changes to an AFM file can be directly saved.

h. The interface has been improved and the menu structure changed; for instance the *Config* menu has been removed and the contained 'Settings' has been moved to 'Preferences…' in the *File* or *Edit* menu (depends on active window).

i. Contouring: last menu entries are stored now.

j. 'New Character': default is 1 now, not 0 (zero is actually not allowed).

k. New icons have been made for the program, functions and menus.

l. New batch functionality in the Font Administration tool.
There is now an *Import* function in the Batch menu:
1. You can browse to select another font
2. You can select all characters or a groups of characters like 101–105 , 301–305, 501–510
3. You can specify the target number for the groups you have selected

In the example above the characters 101–110 will be copied to 4101–4110 and the characters 301–310 will be copied to 4301–4310.

4. You can decide to supersede existing characters or not.

Clicking 'OK' will import all selected characters into your font.
In case that the bodysize of the two fonts is different you will be asked to scale the characters to the target bodysize or leave them as they are.

There is an option to execute a command file to import glyphs from different other fonts. The command file syntax is similar to the one used in Ikarus v4 and quite simple. A command file must be text file (.txt) and can look like this:

```
input D:\archive\ar0000\A057016T.be
chg 101-126
cts 3101
run
input D:\archive\ar0000\A066036T.be
chg 301-326,501-510
cts 3301,3501
run
end
```

A command allows to merge many different font files into one with a simple click. It can for example also be used to regroup and reorder characters if one uses a copy of the edited font as input.

If a command file is specified in the dialog window, all other information (such as the check box 'superseed chars', the source file, character groups, target character numbers) is ignored completely by the program. Instead the program only uses the content of the txt file, i.e. 'ssc 0/1', 'input' et cetera.

Command file syntax:

Input	specifies the input font file.
chg	specifies the characters and character groups to be copied
cha	specifies to copy all characters from the source font.
cts	specifies the target character numbers

If no cts command is present, the original character numbers will be used.
If a cts command is present, the number of characters specified in the cts

command must be equal to the number of character (group) in the chg command.
For example:

```
input c:\archive\xyz.ik
chg 101-103,301-305,501,602,
cts 1101,1301,1501,1602
```

This will copy the characters 101, 102, 103, 301, 302, 303, 304, 305, 501, 602 from the input file xyz.ik to the following numbers in the currently edited font:

```
101 -> 1101
102 -> 1102
103 -> 1103
301 -> 1301
302 -> 1302
303 -> 1303
304 -> 1304
305 -> 1305
501 -> 1501
602 -> 1602
```

run will start the copying process

ssc takes one integer parameter, 0 or 1.
Specifies whether characters in the target font should be overwritten by characters from the input font. Default is not to overwrite existing characters.
ssc 1 will cause characters to be superseeded.
ssc 0 will prohibit superseeding of characters (default).

fbs takes one integer parameter, 0 or 1.
Specifies if characters are scaled to the bodysize of the target font, if the bodysizes of source and target font differ.
fbs 1 will cause characters to be scaled to the target bodysize (default) and fbs 0 will cause characters not to be scaled to the tar get bodysize.

End stops processing of the copying process
If no ‘end’ command is specified, all commands until eof will be read.

c introduces a comment line, which will be ignored by the program.

Sample file:

```
input c:\archive\abc.ik
chg 101,102,103
ssc 1
fbs 1
run
input c:\otherdir\subdir\xyz001.ik
chg 1000-2000
cts 5000
fbs 0
ssc 0
run
end
```

m. Support for manual digitizing via a Wacom tablet with lens cursor.

1. It is advisable to work with a 15000 grid. The digitizer coordinates will be measured in 1/100 mm and directly converted into IK format, so 15 cm size of the artwork will be tranlated into 15000 IK units.
   You can of course work in for instance A3 with large artwork and later scale it to the requested bodysize.
2. To digitize a new glyph open a font first –existing or new– and create a new empty character. It is also possible to digitize and add data to an already existing glyph.
3. Align the working drawing.
4. To start digitizing select *Digitize* from the menu and the digitizer will ask for an origin. This should be the crossing of the baseline and the left sidebearing.
5. You can now start digitizing:
   Use button 1 to digitize a corner point, button 2 for a tangent point and button 3 for a curve point.
   The first point will automatically be a start point.

If the a point is close to the startpoint (within a small tolerance) the contour will be closed automatically.

6. You can force the closing of a contour with button 4. This automatically closes the contour.
7. After closing a contour it will be saved and stored permanently in the glyph data.
8. You can continue digitizing and the next point after a closed contour will be a start point again.
9. On screen the points will be preliminary connected by straight lines until a contour is closed.

10. If you change to a different function during digitizing with the mouse all current data (except for already closed contours) will be lost.
11. Zoom up and down (using keyboard shortcuts) is possible.