

# Content

The integrated **Help** gives you support on how to use *Type-Designer 3.1*.

## Definitions

## MDI Environment

## General Parameters

## File Menu

## Character Map

## Font Parameter

## Editing Characters

## Transformations

## Kerning

## Hints

## Definitions

Font

Point Size

Character

Encoding

Rasterizer

Caching

Character Elements

Hint

Kerning

Type of Files

## Font

The term **Font** is used to mean a collection of characters which are generally similar, for example Times New Roman Italic.

Several weights make up a font family. For example, Times is made up of Regular, *Italic*, **Bold** and ***BoldItalic***.

A Type 1 or TrueType font contains, apart from descriptions of the characters which it contains, a great deal of other information, much of which can be set with Type-Designer.

## Point Size

A **Point** is a worldwide unit used to measure font sizes. In DTP a point is defined as  $\frac{1}{72}$  inch or 0.3528 mm. Accordingly, a 1 Point font requires 0.3528 mm of space. This size is the same as 1000 units in Type-Designer.

Apart from the **DTP Point** there are other typographic units, not discussed in this handbook.

Normally a font is defined so that the top of the H and the bottom of the p are 900 units apart. In the case of artistic or display fonts, this convention may not be observed.

## Character

An almost unlimited number of characters can be saved in a Font. For identification purposes, each character contains a unique **Character Name**. At present, Type-Designer can be used to store and manage up to 1024 characters in a Type 1 Font.

## Encoding

Applications do not usually call characters by their names, but by a numerical value between 0 and 255 (ASCII, ANSI etc.). Which number represents which character is determined by the **Encoding**.

Under Windows 3.1 it is only possible to address characters with the numbers between 32 and 255.

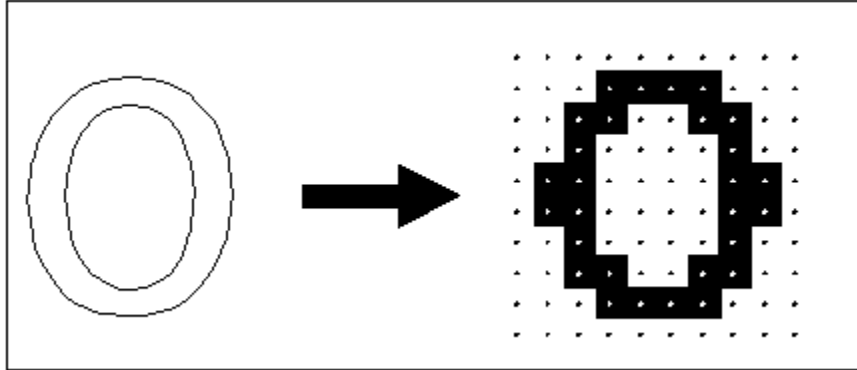
Some programs and printer drivers impose further restrictions and so it is best to use the character slots **127** and **160** only in exceptional cases.

## Rasterizer

As the characters in a font are saved as outlines it is necessary to map them to an output device. This is done by a program called a ***rasterizer***. This program may, for example, be built into a PostScript printer or be part of an operating system, like Adobe Type Manager or Windows integrated TrueType rasterizer.

The better an output device, the finer the grid onto which the character outlines are mapped.

Screens usually have a resolution of about 100 dpi (dots per inch), Image setters achieve over 3000 dpi.



## Caching

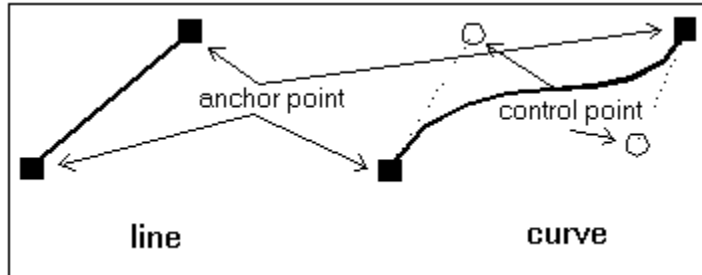
Almost all font scalers use ***caching*** to speed up the process of generating the character for the output device. Characters are saved internally or on the hard disk during rasterization. If the character is required again, then it can be called from this cache instead of having to be rasterized again.



## Elements of Characters

### Character Element

Characters in a font are generally described by an outline, which is made up of **straight lines** and **curves** (Bézier curves). By **character element**, we always mean a **straight line** or a **curve**.



### Anchor Points

An anchor point is the start or end point of a character element.

Anchor points are represented by a square. If the anchor point is open, that is not connected to another anchor point, the square is black.

### Control Points

A curve's course is determined by the position of its control points.

The curve is always drawn toward the control points.

### Segment

A segment is a series of connected character elements.

If a segment contains open anchor points, then the segment is open. Correct characters only contain closed segments.

## Hint

**Hints** enable you to considerably improve the appearance of fonts, particularly at lower resolutions.

Hints inform the rasterizer where horizontal and vertical stems are located in a character, and how the characters are to be aligned vertically and horizontally.

Comprehensive information can be found at:

[All about Hints](#)

## **Type 1**

The **Type 1** format was developed by Adobe in 1985 as part of the PostScript page description language and was kept secret for a long time. Only years later, and after considerable pressure from the leading font companies, was it made public.

The Type 1 format is the industry standard for digital typefaces.

## TrueType

With the introduction of Windows 3.1, the TrueType standard, developed by Apple, came into widespread use for the first time. TrueType also uses mathematical descriptions to store character outlines, but this appears to be the only similarity between TrueType and Type-1.

## Kerning

**Kerning** is the process of adjusting the spacing between pairs of characters.

Each character has its own Width, determined by the left and right margins.

Normally, the left margin of a character is placed exactly adjacent to the right margin of the previous character. The character widths and the position of the character between the left and right margins is usually chosen to give a uniform appearance. When looking at a page in a well-designed font from a great distance, you can see a uniform gray shape.

However, in each font an ugly white space can appear between certain characters when using default spacing.

AV Va Te

By defining kerning values for certain character pairs, the appearance of problematic character combinations can be improved.

AV Va Te

Characters can be **kerned** (spacing reduced) or **letterspaced** (spacing increased).

## File Names

**File names** for a Type 1 font usually consist of an abbreviation with two letters which describes the font family, as well as further letters which provide additional information, e.g. *I* for *Italic* or **B** for **Bold**.

Unused places are normally filled with an underscore character `_`. The extension `.PFB` should always be used.

Normally, you should use a name which allows easy identification of the font.

## File Types of Type 1 Fonts

Type 1 fonts can have up to four component files, namely:

<b>PFB file</b>	This file contains the actual Type 1 font. Apart from general font information, it contains the outline and hinting information. If you are only using Type-Designer as a kerning editor, then this file should not be generated.
<b>PFM file</b>	This file gives Windows important information about a font. Informational values, character widths and kerning values are stored here.
<b>AFM file</b>	This file contains similar information to the PFM file, but in readable form. This file can be edited with any text editor.
<b>INF file</b>	This file is only rarely required, and contains extra information about the font. If no PFM file is present, then Adobe Type Manager can generate the PFM file when a new font is added to the system, providing that the corresponding INF and AFM files are present.

Normally, PFM, AFM and INF files are stored in subdirectories of the path to the PFB file, with the corresponding directory names `PFM`, `AFM` and `INF`. When loading or saving files, Type-Designer looks for these directories. If they are not found, it assumes that the files are to be found or saved in the same directory as the PFB file.

## **MDI Environment**

Starting with Version 3.0, Type-Designer makes use of MDI, which stands for Multiple Document Interface. This makes it possible to have up to eight fonts open for editing at the same time.

After starting Type-Designer you initially see an empty window and a minimal Menu Bar.

Before you can carry out any operations, you have to create, open or import a font.

Details can be found with:

Window Types

Titel Bar

Window Management



## Wondow Types

After creating, opening or importing a font up to four Windows can be open, namely:

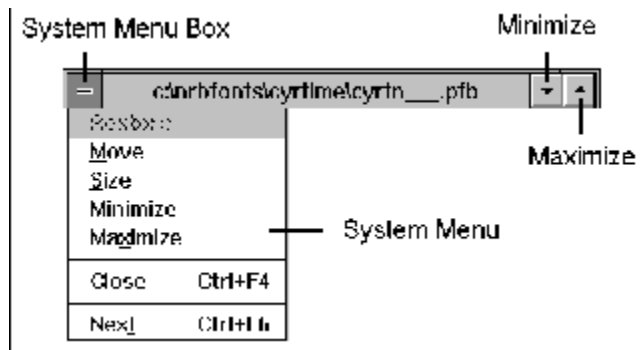
- The Character Map Window,
- The Edit Window,
- The Kerning Window and
- The Test Window.

The Character Map Window is displayed first.

In addition the Menu Bar changes automatically according to the active window. You can only select those commands which are relevant to the active window. In particular the contents of the File and Edit Menus can be very different.

## Titel Bar

Every window has its own **Titel Bar** like any Windows program.



Clicking on the system menu bar calls a menu from which you can choose the options shown in Figure 2.3.

Double-clicking on the **system menu bar** closes the window. If you close a window containing a character map, then the corresponding font will be closed, and so will all the associated windows.

By clicking on the **minimize button**, you can reduce a window to an icon. Double-clicking on an icon restores it to its original size and position.

Similarly, you can use the **maximize button** to enlarge a window to the maximum possible size.

The Window menu contains further options for managing windows.

## Window Management

The Window menu contains options for managing windows.

### Tile

If this command is chosen, all the windows are arranged so that they do not overlap each other. Windows decides the size and position of each window for itself. Usually, this option is only useful if four or less windows are open, otherwise the individual windows become too small.

### Cascade

This command forces the windows to overlap one another. The active window comes to the front, but the title bars of the inactive windows remain visible. This option is useful to get an overview of the opened windows.

### Arrange Icons

This command arranges the minimized windows icons. However, it usually makes sense to arrange the icons by hand, as this command often causes the icon descriptions to overlap.

### Save and Load Positions

You can experiment with the window arrangements as much as you like. Using the command **Save Positions** you can save a windows current position in Type-Designers .ini file, and after you have moved the window, if you dont like its new position you can restore the original position with the **Load Position** command.

<b>Note</b> that when you close a window, that position is saved to the ini file.
---

## General Settings

The menu command **Settings Program** makes it possible for you to specify various parameters for working with Type-Designer.

These settings are stored in the TYPEDSGN.INI file. When you start Type-Designer they are loaded automatically; you do not have to specify them every time you run the program.

The following parameters are possible:

Working Area

Quick Exit

Color Monitor

Context

Grid

PFB Directory

TrueType Directory

CFN Directory

EPS Directory

Soft Font Directory

## Working Area

Use the **Left** , **Right**, **Above** and **Below** settings to specify the maximum size of the working area. Make the left and right margins of the working area large enough that all of the characters can be seen completely.

If you load a font which requires a larger working area then this setting is adjusted automatically.

## Quick Exit

If this option is enabled, no confirmation is asked for when you choose to exit Type-Designer.

## **Color Monitor**

Enable this option if you are using a color monitor. If it is not enabled, then different colors are used to display the various lines in the Edit Window, optimized for a gray scale monitor.

## Context

Character strings which are displayed next to the current character in the Test and Kerning Windows can be entered in the **Left** and **Right** boxes.

This makes it much easier to judge the appearance of a font and character spacing.

You can also enter unallocated characters by entering the character name preceded by a »/«.

The strings »**Ham**« and »**urg**« have proved to be useful.



## Grid

Optionally, a grid can be displayed in the Edit Window.

You can specify the distance between the grid points by entering values in the **x-Dist** and **y-Dist** boxes.

## **PFB Directory**

Enter the directory in which you wish to store your Type 1 fonts. The default is:

`C:\PSFONTS.`

The AFM, PFM and INF files are stored or searched in corresponding subdirectories.

If the corresponding subdirectory cannot be found then the PFB Directory is used for those files.

## TrueType Directory

If you want to store new TrueType files other than in the Windows\System directory, specify another directory here.

## **CFN Directory**

Enter the directory where CFN fonts for importing can normally be found.

## **EPS Directory**

Enter the directory where Adobe Illustrator files for importing can normally be found.

The default is:

[Type-Designer Directory]\EPS.

## Soft Font Directory

Enter the directory where bitmap fonts for use as backgrounds can normally be found.

The default is:

C:\PCLFONTS.

## The Character Map

The **Character Map Window** shows all of the characters in a font.

The **name** of the character is displayed above it with a gray background.

If applicable, a red number is shown above and to the left of the character, showing the characters **encoding number**.

Characters can be marked, i.e. selected for various global translations. These characters are shown inverted.

The current, i.e. the character opened for editing, is shown with a colored background.

<b>Note:</b> The current character is always automatically selected!
--

Other informations can be found with:

[Opening Characters for Editing](#)

[Selecting Characters](#)

[Changing a Character Name](#)

[Creating a New Character](#)

[Undoing Changes](#)

[Cutting](#)

[Copying](#)

[Pasting](#)

[Deleting](#)

[Sorting Character Sets](#)

[Creating a New Character](#)

## Opening Characters for Editing

To open a character for editing, **double click** on it with the mouse pointer. An Edit Window, with the character in it, appears immediately.



## Selecting Characters

There are various ways of **selecting** or **marking** characters in the Character Map Window.

Additional information can be found with:

With the mouse

Selecting a whole font

Removing all Markers

Other selection methods

## With the mouse

If you point to a character with the mouse cursor and click with the **left mouse button**, the corresponding character is selected.

You can tell that a character is selected by its reversed appearance.

If you hold the **left mouse button** down while you move the mouse over the character map, then all the characters over which the mouse cursor is moved will be selected.

If a character is already marked, then it will be **unmarked**.

## Selecting a whole font

You can select a whole font by choosing the menu command **Edit Mark all** or the keyboard combination **<Ctrl><A>**.

All characters in the font will then be shown reversed and will be selected for use in transformations, etc.

**Note:** Before carrying out a global transformation (such as making an outline font, remember to select the relevant characters. Normally, this will be the whole font. Remember about the key combination **<Ctrl><A>**.

## Delete Markers

By choosing the menu command **Edit Remove Markings** or the key combination <Ctrl><U> all markings are deleted.

<b>Note:</b> The character in the <u>Edit Window</u> remains selected after carrying out this command.
--

## Other selection methods

The menu command **Edit Select character** calls a submenu of further selection methods.

<b>Upper case letters</b>	This option selects all the upper case letters. Type-Designer recognizes a character as upper case if its name begins with a capital letter. Symbol fonts, which use character names like c32 to c255 are not recognized as having upper case letters.
<b>Lower case letters</b>	This option selects all the lower case letters. Type-Designer recognizes a character as lower case if its name begins with the letters »a« to »z« and combinations of these letters with the following accents: »grave«, »acute«, »circumflex«, »tilde«, »macron«, »breve«, »caron«, »dotaccent«, »dieresis«, »ring«, »cedilla«, »hungarumlaut«, »ogonek«, »caron«, »slash« and »e«. For example, »adieresis« represents the character »ä«.
<b>Encoded characters</b>	All characters which are available in Windows using the current <u>encoding</u> are selected.
<b>Unencoded characters</b>	The characters not available to Windows are selected.
<b>Invert selection</b>	All unselected characters are selected and all selected characters are deselected.

<p><b>Tip:</b> You can use combinations of these selection methods, for example to select all the extended characters in a normal text font, select all <b>Upper case letters</b> and <b>Lower case letters</b> and then choose <b>Invert selection</b>.</p>
--

## Changing a Character Name

You can change the name of a character.

This can however only be done for the currently open character.

Before you can rename a character, you must open it by double clicking on it.

Next, change back to the Character Map Window and press <Enter>.

A dialog appears in which you can enter a new name for the character.

If you want to use a standard name, you can simply choose it from the list box.

If the name entered is already in use in that font, then a message is displayed and you are prompted to enter another name.

## Creating a New Character

You can create an empty character if you press <Ins> while being in the Character Map Window.

You will then be asked to enter a name for the character in the dialog which appears.

If you have entered a character name which is not already available, then an empty character with the new name appears before the current character in the character map.

## Undoing Changes

If you want to undo changes which you have made to individual characters or to a complete font, use the command **Edit Undo** or the key combination **<Alt><->**.

As this operation can be very complex and may require a lot of memory, it is only one level deep, i.e. only the last alteration can be undone.



## Cutting

If you choose **Edit Cut** or either of the key combinations <Ctrl><X> or <Shift><Del>, then all the marked characters will be deleted.

The deleted character is however copied at the same time to the clipboard, from where they can be pasted into another font.

## Copying

Use the command **Edit Copy** or the key combination <Ctrl><C> or <Ctrl><Ins> to copy the marked characters to the clipboard.

**Note:** Apart from the outline descriptions of the characters, the character names are also copied to the clipboard. Because of this the character names can easily be transferred to a text file or a file for working out  Kerning pairs. A further possibility is the assignment of new names to a whole font.

## Pasting

The command **Edit Paste** and the key combinations **<Ctrl><V>** and **<Shift><Ins>** make it possible to transfer character descriptions from the clipboard.

If a character with the same name exists already, it will be overwritten with the contents of the corresponding character from the clipboard.

<b>Note:</b> Existing kerning pairs are lost during <u>copying</u> and pasting operations.
--

## Deleting

If you choose the command **Edit Delete** or press the <Del> key, all the selected characters will be deleted from the font.

Characters can be restored using the command Edit Undo.

## Sorting Character Sets

The menu command **Edit Sort** calls a submenu of sorting options for the font.

- |                    |  |
|--------------------|--|
| <b>A..Z,a..z</b>   | This option sorts the characters alphabetically according to their names. Characters whose names begin with upper case letters are placed before characters whose names begin with lower case letters. |
| <b>A,a..Z,z</b>    | This option also sorts alphabetically, but upper case characters are followed directly by the corresponding lower case character.  |
| <b>As Encoding</b> | This option sorts the font in the order that the encoding makes them available to Windows. Unallocated characters are placed in alphabetical order at the end of the character map.                    |

## New Character Names

The menu command **Edit New Name** calls a submenu which allows you to allocate new names to the whole font.

- |                               |   |
|-------------------------------|---|
| <b>Standard names</b>         | This option allocates standard names to characters in the corresponding positions, e.g. the character »ydieresis« (ÿ) is usually found in position 255.   |
| <b>c32..c255</b>              | Each character is given a name which reflects its position in the encoding (»c« followed by a number). This is particularly useful if the names used in the font bear no relationship to their shapes.  |
| <b>aus der Zwischenablage</b> | <p>More complex name allocations can be carried out using the clipboard. To do this, the new character names must be stored in the clipboard. Each pair of character names must be separated by at least one space or line feed. You can use any text editor for this.</p> <p>The names will be allocated to the characters in the order in which they appear. Because of this, it is a good idea to sort the font according to its encoding.</p> |

**Note:** Windows does not at present allow the use of kerning values in TrueType fonts which do not the standard encoding. If you need a font with kerning values, then rename the characters with standard names and enable **Standard encoding** in the Encoding dialog.

**Tip:** When copying selected characters from the character map to the clipboard, the outlines and the character names (in text form) are copied to the clipboard. To copy character names from one font to another, simply copy the relevant characters to the clipboard.

## Font Parameters

The menu command **Parameters** calls a menu from which you can select options to set required and optional parameters for the Font.

The following Options are available:

Info

Encoding

Alignment Zones

Stem Widths

Kerning

## General Font Information

The menu command **Parameter Info** calls a dialog where you can view and edit important data about the font. Especially Fontnames and Attributes can be set here.

If you enter something which is obviously wrong, an error message appears. If you enter something which does not comply with the convention for that item, then a warning message appears.

If you use brackets in a text entry box, you should always make sure that each opening bracket has a corresponding closing bracket.

The following settings are possible:

Name

Compl. Name

Family

Weight

Width Class

Style

Italic Angle

ID-Number

Underline Pos

Und. Thick.

Hint Smoothing

Notice

Ascender

Descender

Line Gap

Proportional

Force Bold

Spezial Font

Protection

Direction

## **Name**

The name under which the font is to be made available to PostScript printers must be entered here.

The name used here may not be used by other fonts which are in use. **Spaces** cannot be used in the name.



## Complete Name

The full name is purely informative as far as PostScript output devices are concerned. If you are also going to generate a TrueType version of the font, some application programs use this name in the font list.

To avoid complications, you should use the fonts family name together with an extra description like `Bold` or `Oblique`.

Spaces can be used here, and should be used to improve legibility, e.g. in `Times Roman Bold Italic`.

## Family

Enter the fonts family name here. Spaces are allowed here. Note that the family name should be the same as the first part of the Compl. Name.

The name entered here is usually used in application programs font lists.

**Note:** All variations of a typeface which should be available from the same entry in the font menu **must** have the same family name. The classifications **Bold** and *Italic* are achieved with the Weight and Italic Angle parameters. However, other font attributes like Style and Proportional should be identical in all variants.

Tragen Sie hier den Familiennamen des Zeichensatzes ein. Auch hier sind Leerzeichen erlaubt. Beachten Sie bitte, daß der Familienname der erste Teil des Kompl. Namens sein soll.

## Weight

Weight describes the blackness of a font. The spectrum goes from **Ultra Thin** to **Ultra Black**.

As a numeric value, used by Windows to determine whether the font is bold, is calculated from this entry, only the predefined values from the list should be used. Using an unlisted weight always cause the font to be treated as **Regular**.

<p><b>Note:</b> Theoretically it is possible to define many variants of a font under one family name, but most windows programs cannot distinguish between more than <b>Bold</b> and Normal. Because of this you should put each weight variant into its own font family.</p>
---

## Width Class

This entry, relevant only for TrueType fonts, shows whether the font is an expanded or condensed version. The spectrum runs from **Ultra-condensed** to bis **Ultra-expanded**.

We are not aware of any Windows applications which make use of this font classification function. If you want to use your font designs with one of the usual word processors, or sell them, you should start a new font family for each condensed or expanded version.

For example, a typical family name in this case is Helvetica Narrow.

## Style

The combination of entries in the two list boxes makes it possible to assign a font to a particular style family. The general classifications are on the left, and on the right you can specify whether or not the font has serifs.

Windows allows you to select a font without knowing its exact name. These entries help ensure that there are no unpleasant surprises. As an example, it would seem strange if a sans-serif text font was needed, but a chemical symbol font was displayed instead.

<p><b>Note:</b> Some Windows programs react strangely to the settings made here. If a font which you have created does not appear in the font menus, then try the settings <b>Text</b> and <b>Any</b>.</p>
--

## Italic Angle

This entry shows the italic angle in degrees. Fonts sloping to the right have a negative angle.

The font is classed as **italic** if the angle entered here is less than -1.0 or greater than 1.0.

**Note:** This entry is purely informative. This means that the font will be displayed the same regardless of the value shown. Altering this value will not make a font italic: to do this, you must use the global transformation options.

## **ID Number**

ID numbers are issued by Adobe Inc. and assist in the effective Caching of fonts for print jobs. Within a controlled user environment, such as a departmental LAN you can use ID numbers between 4.000.000 and 4.999.999.

Otherwise, always use a value of 0, unless you have been given a number by Adobe.

## Underline Pos

Enter the Y coordinates of the upper boundary of the underscore.

Use the **underscore** character as a guide.



**Underline Thick.**

This value determines the thickness of the underline.

## Hint Smoothing

This value gives a certain amount of control over the behavior of Stem Widths and Alignment Zones.

With TrueType this value determines at what difference between current and actual stem widths smoothing will not be used. However, a large value could cause smoothing to be lost at intermediate sizes.

You can only find the optimal value by experimenting.

With Type 1 fonts, this value usually determines at what size (300 dpi output) to stop using overshoot control.

The smaller the value entered here, the sooner hint optimization determined by stroke weight and alignment zones will be discarded.

## **Notice**

This informational entry is usually used for a copyright notice.

## Ascender

Windows programs use this entry and the next two entries to work out the line gap between two lines.

The **Ascender** shows the height of the capital letters. If no value is entered here, Type-Designer works out a plausible value.

All members of a font family should have the same values for **Ascender**, Descender and Line gap. This helps to avoid differing inter-line spacing when mixing the members of a font family.

<p><b>Note:</b> Line gaps are often calculated from the Bounding Boxes (the minimum rectangles surrounding the characters) of individual characters or all characters. To get a typographically clean appearance, it is generally a good idea to use a fixed value for the line gap.</p>
--

## **Descender**

This value specifies the amount by which a normal font character should hang under the baseline.  
If a value of 0 is entered a plausible value is worked out when the font is saved.

## Line Gap

If the distance between the Ascender and Descender is too small, this value allows a supplement to the line spacing to be worked out.

However, not all application programs make use of this function.

## Proportional

In a proportional font, each character has its own **width**. With a **monospaced** or **non-proportional** font, all the characters have a uniform width.

<p><b>Note:</b> To make sure that all the characters in a monospaced font are printed with the same character widths, each character must be given the same width using the <u>Margin-Tool</u> or by using a <u>global transformation</u>.</p>
--

In einer proportionalen Schrift hat jedes Zeichen eine individuelle **Breite** oder **Dicke**. Bei **monospaced** oder **nicht-proportionalen** Schriften haben alle Zeichen die gleiche Breite.

## Force Bold

Use this option only if your font should retain its bold appearance at small sizes.

Whether or not this font is regarded as **bold** by applications, is determined independently of this setting by the weight option.



## Special Font

This entry ensures that a complicated font will be handled properly on special PostScript rasterizers.

<p><b>Note:</b> This option may <i>only</i> be used if the font is to be output on a special output device, like a Kanji imagesetter. Otherwise errors can arise during output. If normal output devices are to be used, then this option <b><i>must</i></b> be turned off. Older PostScript imagesetters are particularly likely to have problems.</p>
---

## Protection

Windows 3.1 introduced a new concept, that of allowing TrueType fonts used in a document to be saved in the document (**Embedding**). A document with embedded fonts can be transferred to any PC, even if the fonts used are not installed on the target PC.

Font vendors are not usually happy for their fonts to be copied indiscriminately without payment, and so different levels of protection can be specified.

<b>No Protection</b>	The font can be copied and installed permanently on any system.
<b>Read-Only-Protection</b>	The font can be copied to other systems with a document, but cannot be permanently installed. The font is only available to edit and print the document in which it is embedded.
<b>Full Protection</b>	The font cannot be embedded in a document and copied to another system.

## Direction

This TrueType setting specifies the direction. Neutral characters are characters which can be written equally well in either direction.

This setting is provided for future versions of Windows with support for right-to-left languages like Arabic.

## Encoding

You can use the menu command **Parameters Encoding**, you can specify which **code (key)** is to be used to call each character.

The left list box shows this information:

<b>Key</b>	The character which corresponds to the Windows ANSI character at this position is shown here.
<b>Code</b>	An identification number in the range 0 to 255.
<b>Name</b>	The character name allocated to the code.

The table on the right contains a listing of all the character names in the font.

The following options are available:

Show Names

Standard Encoding

Custom Encoding

Loading and Saving Encodings

## Show Names

If this option is **not** active, the actual characters are shown in the lists.  
Otherwise, the character names are shown instead.

## **Standard Encoding**

Use this option if you want to use standard PostScript character encoding.

You should always use this option if you are creating a Latin character set which is not just for your private use. Remember that in this case, you must use the standard PostScript character names.

If you are using a font with standard encoding under Adobe Type Manager, you will get the Windows ANSI encoding, not the standard PostScript encoding.

## Custom Encoding

It is only possible to use a custom encoding if the Standard encoding option is turned off.

Proceed as follows:

=> Position the black bar in the left table on the code which you want to allocate

=> In the right table, position the bar on the name which you want to allocate to the code.

=> Click on the arrow in the middle.

You can see the effect of the change at once.

Both bars then move down one position, so it is easy to produce a series of consecutive allocations just by repeatedly clicking on the bar.

## Loading and Saving Encodings

You can save an individual encoding in a file or carry out encoding by loading a corresponding file.

Clicking on the **Load** or **Save** button calls a standard filing dialog.

Some encoding files are included with Type-Designer. These include **Windows ANSI**, **Macintosh** and **isoLatin-1**.

<b>Note:</b> If you load an encoding, a character name will only be allocated a code, if the corresponding character name is used in that font.
---

<b>Tip:</b> You can avoid having to generate a fresh encoding table for italic, bold, and bold-italic variants by saving the normal weights encoding and loading it into the other weights.
---

## Alignment Zone

An **alignment zone** is a vertical area: each anchor point in such an area, if it correlates with a vertical hint, is placed on the baseline of the area. To define such an area, enter a y-coordinate and a positive height.

Note that the baseline may be the upper or lower margin of the alignment zone. This depends on whether a character can extend *above* the ascender height, e.g. the tips of the letter »**N**« or the top of the letter »**O**«, or *below*, like the bottom of the letter »**O**« or the lower point of the letter »**V**«.

In the first case the corresponding parts of the letter should be *nudged down*, which means that the baseline of the corresponding alignment zone must be below it. In the second case the letter should be *nudged upwards*, so that the base line of the alignment zone lies above it.



## Defining Alignment Zones

You can use the menu command **Parameter Alignment Zones** to define **guidelines** to be used when creating characters and to improve the output quality at small sizes.

The vertical height of characters which are rounded or have fine points should, to appear consistent to the human eye, be between 1.5% and 3.0% larger than characters with flat tops or bottoms. At very small sizes, this can make the font look untidy because of rounding effects. At very small sizes, it is desirable to nudge these characters so that they appear to be exactly the same size (*overshoot control*, see Chap. 8.2.3).

You can use this dialog to define guidelines which, with the help of suitably defined Hints, make **overshoot control** possible.

<b>Note:</b> Areas which have a <b>Height</b> greater than 30, should be avoided, as they may cause <i>unexpected results</i> on some output devices.
---

<b>Tip:</b> You can turn display of the Guidelines in the <u>Edit Window</u> on and off with the <b>Guideline</b> option.
---

The following areas can be defined:

Baseline

Upper Zones

Lower Zones

## Baseline

The first entry describes the **baseline alignment**, i.e. the alignment zone for characters which extend slightly below the baseline (y coordinate 0).

This must be a negative value. Height must be the same value as the **y-Coor**, except positive, as **y-Coor** plus **Height** must equal 0.

<b>Tip:</b> You can usually use the y coordinates of the capital »O«s lowest anchor point as the <b>y-Coordinate</b> .
--

## Upper Zones

You can define alignment zones here for characters which extend upwards above other characters and which should therefore be nudged downwards, e.g. »**O**«, »**o**«, or the upper bowl of the »**g**«.

<b>Tip:</b> If anchor points which lie in the upper zone have a vertical hint applied to them, they always point downwards
--

## Lower Zones

You can define alignment zones here for characters which extend downwards below other characters and should therefore be nudged upwards (such as the lower bowl of the »g« or the curve of the »9«).

<b>Tip:</b> If vertical hints are applied to an anchor point which lies in a lower alignment zone, then these hints always point upwards.
---

## Stem Width

Using the menu command **Parameters Stem widths** you can define the most important stem widths. These numeric values, often referred to as **global hints**, ensure optimal output of Type 1 and TrueType fonts at the smallest sizes.

Irregularities caused by rounding effects or small errors in the digitization, can be reliably excluded using these values.

Additionally, you can use this feature, for example, to ensure that the stroke weights of upper and lower case characters are the same at small sizes. This regularization can be influenced by the **Hint Smoothing** setting of the Parameter Info dialog. This value determines the size at which hint regularization should not be used any more.

**Tip:** In the Edit Window, you can click the xy tool on two points to see their coordinates. This enables you to see the exact distance between two points.

The following areas can be defined:

Main Stem Width

Other Stem Width

## **Main Stem Width**

This entry defines the stroke weight used most often.

A value derived from the low case »o« is often capable of giving good results.

## **Other Stem Widths**

Enter other important stroke weights here.

Note that it is only worth entering those stroke weights which differ greatly from others. Stroke weights less than 10 units different from others are not usually worth entering.

## Editing Characters

Double-clicking on a character in the Character Map opens the **Edit Window** with that character open.

Various information, selection and setting options are shown on the left. In detail:

<u><b>Tool palette</b></u>	Used to select a tool for editing the character
<u><b>Speed-Buttons</b></u>	Can be used to call frequently-used commands.
<u><b>Status Information</b></u>	Displays the cursor position, distances and angles.
<u><b>Control Panel</b></u>	Used to turn viewing and editing options on and off.

Also, other Editing functions can be accessed through **hot keys** or **menu options**.



## Speed Buttons

The **Speed Buttons** can be used to access often used functions easily.

The following buttons are available:



Get Character



Save Character



Selecting Characters



Undo



Redo



Autohinting



Testing a Character

## Get Character

The **Get** button loads the most recent version of a character.

If you have made several test modifications to a character and want to discard them, you can use this function. All changes since the last save are lost.

### Combine Characters

The **Get** button has another important function. It can be used to combine characters to make a new character. This is useful, for example, to place an **accent** on a normal character.

Follow these steps.

=> Load the basic character into the Edit Window. This should be the character which determines the width of the combined character. If you have to adjust the width at all, e.g. making the character **œ** as a combination of **o** and **e**, then it does not matter which character you use as a basic character.

=> In the Character Map, mark the character which you wish to combine with the basic character.

=> Hold down the **<Shift>** key.

Click on the **Get** button.

All of the characters which are marked will be added to the basic character.

### See also

[Save in other Position](#)

## Save Character

If you have finished editing a character, or want to save an intermediate stage, then click on the **Save** button.

The character will be saved, and the changes will appear in the Character Map.

**Note:** The character may not contain open segments. If there are open segments, a warning message is displayed. Close the open segments and try again.

Note that the Test function always uses the last-saved version. If you have made changes, you must save the character before you can view the character in the Test Window.

### Save in other positions

You can use the **Save** button to save the character in the Edit Window to a different character position.

Follow these steps:

=> Mark the position in the Character Map to which you wish to save the current character.

=> Hold down the <**Shift**> key.

=> Click on the **Save** button.

All marked characters are overwritten by the current character.

### See also

[Creating a new Character](#)

## Selecting Characters

You can use the



buttons to select characters without changing back to the [Character Map](#).

**Note:** Clicking one of these buttons automatically saves the current character. You do not need to click on the **Save** button.

This is what the buttons do:



Puts the first character of the font in the Edit Window.



Puts the previous character of the font into the Edit Window.



Puts the next character of the font into the Edit Window.



puts the last character of the font in the Edit Window.

## Undo

You can use the



button or the key combination **<Alt><->** to successively delete the last ten changes made in the Edit Window.

Alternatively, you can use the menu command **Edit Undo**.

## Redo

You can use the



button or the key combination <Ctrl><Y> to redo changes undone with the Undo button. Alternatively, you can use the menu command **Edit Redo**.

## Autohint Button

If you click on the



button, then hints are automatically inserted into the characters outline description.

Exactly how these hints are calculated, depends on the Autohinting settings defined with the menu command Settings Hinting.

**Note:** Hints are only shown in the Edit Window if at least one of the tools for hint editing (bottom row of tools) is active.

### See also

Hints

## Testing a Character

Clicking on the



button in the Edit Window, calls the **Test Window**, where you can see the current character at various sizes.

<b>Note:</b> Save the character before calling the Test Window so that you can see the changes.
---

Optionally a short piece of text may be displayed on either side of the character. You can define this text using the menu command Settings Program.

You can use the scroll bars to check the quality of the character at various sizes. You can easily check the Hinting and **alignment**, i.e. the width and position of the characters relative to one another.



## Status Information

The Status Information shows the position of the cursors in the Edit Window as well as the spacing and angles.

When moving the mouse cursor in the Edit Window, distances and angles are shown relative to the thumbtack icon:



On the other hand, when drawing the reference point is the start point. This makes it very easy to draw, for example a 45° gradient.

These values are shown:



The **absolute** cursor position.



The **horizontal** and **vertical** distance to the thumbtack.



The **length** and **angle** of the straight line between the cursor and the position of the thumbtack.

## Control Panel

Through activation of the various options in the control panel you can get additional information shown in the Edit Window or set special editing options for some tools.

This is in detail:

Margins

Guidelines

Grid

Background

Snapping

rounded

relativ

Movement Options horizontal / vertical

Curve Type smooth / elliptical

## Margins

The **margins**, or character boundaries are very important to the positioning of characters.

### See also

[Border Tool](#)

## Guidelines

**Guidelines** or alignment zones are very important to ensure that all characters look right at very small sizes.

## Grid

The **grid** spacing, which can be optionally shown by activating this option, can be defined by choosing Settings Program.

<b>Note:</b> Edit functions which move an anchor point or control point snap to the nearest grid point if the grid is turned on.
--

## Background

You can use this option to turn a **bitmap background** on or off.

This is only possible if a background bitmap font has been loaded, or if a bitmap has been pasted in from the clipboard.

## Snapping

If the **Snapping** option is **on**, open segments made when drawing lines or curves will be joined to other open segments if they touch them. The same applies to moving anchor points.

Two anchor points touch if the black boxes surrounding them overlap.

If the **Snapping** option is **off**, open anchor points are only joined if they are at exactly the same coordinates.

## Relative

This option is only relevant to the rotation and distortion tool.

If it is turned **off**, then the **character origin** is used as the reference point for rotation and distortion. This is very useful when altering a whole character.

If it is turned **on**, then the **center** of all marked character elements is used as the reference point. This is very useful when altering individual character elements.



## **Rounded**

If this option is on, then the position of the mouse cursor in the Edit Window and all altered coordinates are rounded to whole numbers.

This option should always be turned on unless you require very small structures in a character.

Note that unrounded coordinates need about three times more space in the font file.

All functions which recalculate anchor or control points (such as global changes) make use of this setting.

## Movement Options

The **Horizontal** and **Vertical** buttons determine the movement directions for various editing operations.

Only one of the two options can be active at once.

Deactivate an option by clicking on it again.

Alternatively, use the <H> and <V> keys to turn the options on or off.

The movement options affect these tools:

Line Tool

Move Tool

Reflection Tool

Distortion Tool

Rotation Tool

Hinting Tool

## Curve Type

The **smooth Curve** and **ellipt. Curve** determine how curves are drawn.

Only one option can be active. To turn an option off, click on it again.

You can use the <**S**> and <**E**> keys instead of the mouse to turn these options on or off.

### Smooth Curve

If this option is on, then the transitions from curve to curve or line to curve are automatically smoothed when drawing Bézier curves.

### Elliptical Curve

If this option is on, then quarter-ellipses can be drawn in an anticlockwise direction.

## Editing Tools

This section covers all the **tools** available for character editing.



Select a tool by clicking on it with the mouse.

The active tool is shown in reverse.

If you move the mouse cursor into the character editing area, it changes to represent the selected tool.

To get **help** on a tool, click on it with the **right mouse button**.

The following tools are available:

[Line](#)

[Curve](#)

[Mark](#)

[Magnifier](#)

[Character Boundaries](#)

[Cut and Join](#)

[Coordinate Control](#)

[Move](#)

[Reflect](#)

[Rotate](#)

[Distort](#)

[Thumbtack](#)

[Segment Order](#)

[Starting Point](#)

[Hint](#)

## Line Tool



Use the line tool to draw lines and rectangles.

### Drawing Lines

=> Move the center of the mouse cursor to the position where the line should start.

=> Press the **left mouse button** and hold it down.

=> Move the mouse to the position where the line should end.

=> Release the mouse button.

According to the movement direction settings, you can draw horizontal, vertical or arbitrary lines.

If **Snapping** is **on**, open segments are closed as soon as the line is pulled over an open segment point.

If **Snapping** is **off**, segment points are only joined if they are on exactly the same coordinates.

### Drawing Rectangles

=> Move the center of the mouse cursor to the position where the upper left corner of the rectangle should be.

=> Press the **right mouse button** and hold it down.

=> Move the mouse down and to the right, until the rectangle is the correct size.

=> Release the mouse button.

A rectangle of a corresponding size is inserted.

<b>Tip:</b> If you hold down the <Ctrl> key when drawing a rectangle, a <b>square</b> is drawn.
---

## Curve Tool



Use the **curve tool** to draw **curves** and **ellipses** and **circles**.

### Drawing Curves

According to the curve type setting, you can draw arbitrary, smooth or elliptic curves. The procedure is slightly different for each type of curve.

**Note:** When setting the end points, a connection with an open segment may be made automatically (Adjacent if **Snapping** is **on** or identical placement if it is **off**). If such a join is made, then that end point can no longer be moved. Drawing continues with the next control point, or curve drawing is finished.

Additional Information can be found with:

[Arbitrary Curve](#)

[Smooth Curve](#)

[Elliptical Curve](#)

[Ellipses and circles](#)

## Arbitrary Curve

- => To draw an arbitrary curve you can proceed as follows:
- => Move the center of the mouse cursor to the position at which the curve should start.
- => Press the **left mouse button** and hold it down.
- => Move the mouse to the position at which the curve should end.
- => Release the mouse button (or make a connection with an open segment).
- => Move the control point to the required position.
- => Click the **left mouse button**.
- => Move the second control point to the required position.
- => Click the **left mouse button**.

<b>Tip:</b> Press the <Ctrl> key at any time when setting control points to get smooth transitions to adjoining curves and lines.
---

## Smooth Curve

To draw a smooth curve, proceed as described with the arbitrary curve, but activate the **smooth Curve** option.

Movement when setting control points is restricted (control point on the connecting straight line).

This is only the case if setting the end point would create a connection to an open segment; otherwise, the control point can be placed anywhere.



## Elliptical Curve

When drawing elliptical curves, control points are set automatically so that a quarter ellipse emerges.

The ellipse is drawn anti-clockwise.

=> Turn **on** the **ellipt. Curve** option.

=> Set the start point by clicking the **left mouse button**.

=> Hold it down and move the mouse pointer to the end point of the ellipse.

=> Release the mouse button.

## Ellipses and Circles

Instead of drawing four quarter ellipses and combining them to a full ellipse with the reflection tool, its possible to draw an ellipse in a similar way to a rectangle.

=> Move the center of the mouse cursor to the position, at which the upper left corner of the rectangle surrounding the ellipse should be.

=> Press the **right mouse button** and hold it down.

=> Move the mouse down and to the right until the required size is reached.

=> Release the mouse button.

An ellipse of the corresponding size is inserted.

<b>Tip:</b> If you hold down the <Ctrl> key while drawing an ellipse, a circle is drawn.
--

## Marking Tool



Use the marking tool to **mark** character elements or **move** points or character elements.

Character elements have to be marked before:

[Copying to the clipboard](#)

[Deleting](#)

[Cutting](#)

[Duplicating](#)

[Moving character elements](#)

[Reflection](#)

[Distorting](#)

[Rotating](#)

The following options are available with the **Marking Tool**:

[Marking individual Character Elements](#)

[Marking Segments](#)

[Using a Marquee](#)

[Deleting Markings](#)

[Moving Anchor and Control Points](#)

[Moving Character Elements](#)

### See also

[Mark All](#)

[Delete All Markers](#)

## Marking individual Character Elements

=> Move the mouse cursor close to the character element to be marked.

=> Click the **left mouse button**.

The character element is now marked and is shown in a different color.

## Marking Segments

=> Move the mouse cursor close to the segment to be marked.

=> Double-click the **left mouse button**.

The segment is now marked and is shown in a different color.

## Using a Marquee

=> Move the mouse cursor to a point above and to the left of the character elements to be marked.

=> Press the **left mouse button** and hold it down.

=> Move the mouse down and to the right, until the rectangle encloses all the elements to be marked.

=> Release the mouse button.

All character elements whose **start** and **end** points lie within the rectangle are immediately marked.

## Deleting Markings

If you carry out the marking operations on an already-marked element or segment removes the marking. Holding down the <Shift> key when dragging out a marking rectangle, removes all markings within the rectangle.

## Moving Anchor and Control Points

It is possible to move anchor and control points using the marking tool.

=> Move the point of the mouse cursor onto an anchor or control point.

=> Press the **left mouse button** and hold it down.

=> Move the mouse to the required position.

=> Release the mouse button.

Note that moving curve anchor points always moves the adjoining control points. This can be useful, for example, when moving an elliptical arc.

When moving control points, the gradient of straight lines between adjoining control and anchor points remains the same.

<b>Note:</b> It is also possible to carry out other movement operations with the <u>movement tool</u> .
---



## Moving Character Elements

This functions in the same way as moving character elements with the [movement tool](#).

<b>Tip:</b> When moving points and character elements, use the <b>horizontal</b> and <b>vertical</b> options to restrict the movement direction.
--

## Magnifier Tool (Zooming)



Use the magnifier to view details of a character or to switch back to a previous detail view.

Details are available with:

[Viewing a Detail](#)

[Revert to previous view](#)

[Viewing the whole character \(Totale\)](#)

## Viewing a Detail

=> Move the center of the mouse cursor to the upper left corner of the area you wish to zoom in on.

=> Press the **left mouse button** and hold it down.

=> Move the mouse down and to the right until a rectangle encloses the area you want to zoom in.

=> Release the mouse button.

The selected section now appears in the Edit Window.

## **Revert to previous view**

To revert to the previous view,

=> move the mouse cursor into the character editing area

=> and click the **right mouse button**.

## Viewing the whole character

To switch from detail view back to **whole** character view

=> move the mouse cursor into the right of the window and

=> click the **left mouse button**.

The whole view of the character appears in the Edit Window.

## Character Boundaries Tool



Use the **character boundaries tool** to set character boundaries.

These are: the **left** and **right** margins, the **baseline** and the **height** of a character (for vertical fonts).

Selecting this tool displays the character boundaries even if the Margins setting is off.

A window appears at the upper left of the Edit Window which displays the settings numerically.

### Using the Mouse

You can change the margins by moving the lines on screen:

=> Move the top of the arrow onto the bounding line which you wish to move.

=> Press the **left mouse button** and hold it down.

=> Move the mouse to the required position. The change is also shown numerically.

=> Release the mouse button.

### Direct Input

You can change the character boundaries by changing the corresponding numeric value.

The lines on screen are redrawn to reflect any changes made.

<p><b>Note:</b> If you are using Adobe Type Manager or generating TrueType fonts, then you should set the <b>baseline</b> and the <b>height</b> of all characters to 0.</p>
---

## Seperation Tool



Use the separation tool to insert new anchor points into existing character elements or to join two character elements with one another.

### Simple Separation

=> Move the tip of the mouse cursor near to the character element into which the new anchor point is to be inserted.

=> Press the **left mouse button** and hold it down.

=> Move the mouse so that the line from the tip of the knife to the start point crosses the character element to be separated.

=> Release the mouse button.

An anchor point is inserted.

### Multiple Separation

If the line from the knife crosses several character elements, then a new anchor point is inserted wherever it crosses a character element.

### Joining Character Elements

You can use the separation tool to make a single curve from two character elements.

=> Move the tip of the mouse cursor on to the anchor point which lies between the two character elements to be joined.

=> Click the **right mouse button**.

The two character elements are merged into a single curve.

<p><b>Note:</b> In theory it is possible to join arbitrary curves and straight lines, but this may not make sense as the appearance of two character elements often cannot be simulated by a single curve. The use of this function should be confined to merging character elements up to the size of a <math>\frac{1}{4}</math>-Ellipse.</p>
--

<p><b>Tip:</b> If you want to merge a long and a short curve, in general you will get better results if you first split the long curve in the middle, and then merge the short and cut curves.</p>
--

## Coordinate Control Tool



Use the coordinate control tool if you want to directly alter the coordinates of anchor or control points.

### Viewing the Coordinates

=> Move the tip of the mouse cursor to the point whose coordinates you wish to change.

=> Click the left mouse button.

A small box appears underneath the point clicked on, where the points coordinates are shown.

### Changing the Coordinates

=> Click on one of the arrows to the right of the coordinate to change it. Clicking on the **up arrow** increases the value, the **down arrow** decreases it.

<b>Note:</b> If the <u>rounded option</u> is not activated, the coordinates change in <b>0.1</b> unit steps.
--

<b>Tip:</b> The new coordinates can also be entered directly.
---

### Marked Character Elements

If the select point is in a marked character element, all the marked character elements are moved by the same amount.

### Multiple Boxes

Up to four coordinate boxes can be displayed at once. If you try to display a fifth box, the first opened box is closed.

<b>Tip:</b> You can use coordinate control to check the exact positions of points for <u>alignment</u> purposes.
--



## Movement Tool



Use the movement tool to move **anchor** points, **control** points or **character elements** to another position.

More information is shown with:

[Moving Control and Anchor Points](#)

[Moving Smoothly](#)

[Moving Character Elements](#)

[Separating Anchor Points](#)

## Moving Control and Anchor Points

=> Position the top of the mouse cursor on the point you wish to move.

=> Press the **left mouse button**.

=> Move the mouse cursor to the new position.

=> Release the mouse button.

If the movement direction has been set to **vertical** or **horizontal**, the selected point can only be moved in this direction.

## Moving Smoothly

Use the **right mouse button** to move anchor or control points with smooth transitions.

Apart from this, use the procedure described with Moving Control and Anchor Points.

The **horizontal** and **vertical** options have no effect in this case.

When moving points smoothly, they are moved independently of the movement option settings.

## Moving Character Elements

Before you can move a whole character, it must be marked using the marking tool, the Edit Mark all command or <Ctrl><A>.

=> Move the top of the arrow to an anchor point between two marked character elements or on a marked curves control point.

=> Press the **left mouse button**.

=> Move the mouse to the required position.

=> Release the mouse button.

Note that this takes the movement direction settings into account and that the connections from marked to unmarked character elements are retained.

## Separating Anchor Points

If you hold down the <Shift> key when moving anchor points, connected anchor points are separated. This can be used to open closed segments.

As long as you hold the <Shift> key down, a connection to the next open anchor point cannot be made.

## Reflection Tool



Use the mirror tool if you want to reflect character elements.

Note that the movement direction setting applies to reflection operations.

Only those character elements which have been marked with the marking tool or the menu command Edit Mark all are reflected.

### Random Reflection

Random reflection allows you to draw a straight line between two points, about which the marked character elements are reflected.

You can often use anchor points for this. This is easier if **Snapping** is on. The start and end points of the line will then stick to the anchor points.

=> Move the center of the mouse cursor to the point where the reflection axis should start.

=> Press the **left mouse button** and hold it down.

=> Move the mouse to draw a suitable line.

=> Release the mouse button.

The marked character elements are reflected about the line which you have defined.

### Horizontal or vertical Reflection

You can use the **Snapping** option when using reflection about a horizontal or vertical line to align the line with anchor points.

=> Press the **left mouse button** and hold it down.

=> Move the mouse so that a suitable line is drawn.

=> Release the mouse button.

## Rotate Tool



Use the rotate tool to **rotate** or **italicize** characters or character elements.

Note that rotation uses the movement direction settings. For example, by restricting the movement to **horizontal** you can italicize characters.

Only those character elements which have been marked with the marking tool or the menu command Edit Mark all are rotated.

In addition, this command uses the **relative** movement option.

If it is **off**, then the character is rotated about its **origin**, otherwise it is rotated about the **mid-point** of all marked character elements.

After selecting the rotate tool, a dialog appears in which the rotation angle is shown.

### Using the Mouse

=> Move the tip of the arrow onto an anchor point between two marked character elements, or onto a marked curves control point.

=> Press the **left mouse button**.

=> Move the cursor until the required rotation angle is achieved.

=> Release the mouse button.

### Direct Input

You can enter the required angle directly into the dialog.

## Distortion Tool



Use the distortion tool to **enlarge** or **reduce** character elements.

Note that rotation uses the movement direction settings.

Only those character elements which have been marked with the marking tool or the menu command Edit Mark all are distorted.

In addition, this command uses the **relative** movement option.

If it is **off**, then the character is distorted about its **origin**, otherwise it is distorted about the **mid-point** of all marked character elements.

Selecting the distort tool makes a small dialog, showing the distortion and stretching factors of the marked character elements, appear in the Edit Window.

### Using the Mouse

=> Move the tip of the arrow onto an anchor point between two marked character elements, or onto a marked curves control point.

=> Press the **left mouse button**.

=> Move the cursor until you reach the required size. The distortion factors appear in the upper left of the Edit Window.

=> Release the mouse button.

### Direct Input

You can enter the distortion factors or stretch values directly in the dialog.



## Thumbtacks Tool



Thumbtacks are used as position markers in the Edit Window. Angle and spacing values in the status display are shown relative to the position of the thumbtacks.

Thumbtacks are shown in the Edit Window by a



and can be set by this procedure:

=> Move the mouse cursor to the position where you wish to place the thumbtack.

=> Click the **left mouse button**.

The thumbtack will be placed at the new position.

## Segment Order Tool

1,2

Use the segment order tool, if you want to order the segments in a character.

This tool is only usefull if hint replacements are necessary.

### Specifying the order

If you have selected the segment order tool, then the segment number appears to the right of the segments start point.

Note that you can only alter the segment number of a closed segment.

=> Move the point of the mouse cursor to the segment which you wish to make the first section.

=> Click the **left mouse button**.

=> A 1 appears next to the segment start point.

=> Repeat these steps for the following segments.

## Starting Point Tool



Use the start point tool to specify the **start point** of a segment or the point where a **hint replacement** should be made.

This tool and specifying the Segment Order are only needed if extensive hint replacements has to be made in a character.

### Specifying the Start Point

If you have selected the start point tool, the number of the segment is shown next to its start point.

Note that you can only change the start points of closed segments.

To convert an anchor point to a new start point,

=> move the mouse cursor onto the point which should become the start point.

=> Click the **left mouse button**.

The segment number will now be shown next to the new start point.

### Specifying Hint Replacement locations

Hint replacement points are shown by an »E« to the left of the anchor point.

Note the following point:

As an outline is a closed shape, the start point is also the end point and appears twice in the outline description. In complex characters, the start point and end point may need different hint replacements.

To set a hint replacement point, use the same procedure as for setting a start point, but use the **right mouse button**.

If you click several times on a segments start point, then up to two Es can appear next to the anchor point. The lower, lighter, »E« signifies the hint substitution for the start point, the *normal* »E«, that for the end point.

### Deleting Hint Replacements

To delete a hint replacement point, simply click on the anchor point with the **right mouse button** until there is no »E« visible.

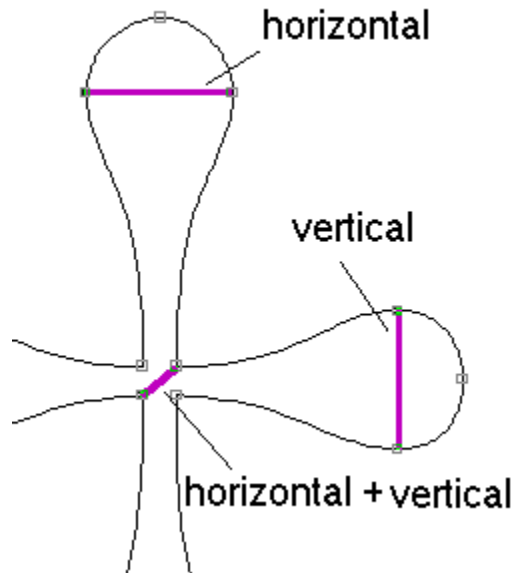
## Hint Tool



You can use this tool to insert Hints into a character's outline description.

Hints ensure that characters look as near as possible the same at all sizes and resolutions.

### Representation of Hints



Hints in a character are shown as thick, colored lines, and the start point of a hint is shown as a black circle.

More details are given with:

[Setting a Hint](#)

[Deleting a Hint](#)

[Extra Hint for the Start Point](#)

## Setting a Hint

Hints can only start at an anchor point.

Generally, Hints should end at the x or y coordinates opposite an anchor point, but they do not have to. If hints do not run from one anchor point to another, then they can still affect the overshoot control.

Horizontal hints, which do not end at an opposite anchor point are therefore mostly ineffective.

**Note:** Only horizontal and vertical hints are required. You can construct a diagonal hint (if there are no movement direction restrictions) but it consists of one horizontal and one vertical hint.

To ensure that the hint ends exactly at the x or y coordinate of an anchor point, turn **Snapping on**.

=> Move the center of the cross onto the anchor point from which the hint should start.

=> Press the **left mouse button** and hold it down.

=> Move the mouse to the required position.

=> Release the mouse button.

## Deleting a Hint

To delete a hint,

=> Move the center of the cross to the anchor point where the hint begins.

=> Click the **left mouse button**.

## Extra Hint for the Start Point

As the start point of a segment is also the end of a segment, it can be that they require different hints. The hints for the end point are described with [Setting a Hint](#).

To set the hints for the **start** point use the **right mouse button** instead of the left mouse button.

For display purposes, the start point hint is shown in a different color to other hints.

## Other Edit Functions

Other editing functions for editing characters can be chosen from the **Edit menu**.

In Detail:

Cut

Copy

Paste

Delete

Duplicate

Mark All

Delete All Markers

Insert Extrema

Insert Intersections

Import AI EPS

Background Images



## Cutting

Use the menu command **Edit Cut** or the key combination **<Ctrl><X>**, to delete marked character elements.

However, unlike the command Edit Delete this command stores the deleted segment in the clipboard: it can be pasted into another character using the command Edit Paste.

## Copying

Using the menu command **Edit Copy** or the key combination <Ctrl><C> or <Ctrl><Ins> you can copy the marked character elements to the clipboard.

## Pasting

The command **Edit Paste** or the key combinations <Ctrl><V> and <Shift><Ins> paste the character elements from the clipboard into the current character.

<b>Note:</b> You can use this command to paste bitmaps from the clipboard into the <u>editing background</u> .
--

## Deleting

You can delete all marked character elements using the menu command **Edit Delete** or the <Del> key.

Unlike using the Cut command, the deleted characters are not stored on the clipboard.

Accidentally deleted character elements can be restored with the command Edit Undo.

## Duplicating

You can use the command **Edit Duplicate** or the key combination <Ctrl><D> to duplicate character elements.

Only marked character elements can be duplicated.

After duplicating, you can use the move tool to reposition the duplicated character element.

## Inserting Extrema

The menu command **Edit Insert extrema** or the key combination <Ctrl><E> helps improve the quality of fonts.

Extrema are points which are touched by lines parallel to the x and y axis.

This Outlines are much more well suited for the insertion of Hints.

Anchor points which are no longer needed after applying this command can be reduced using the separation tool.

## Inserting Intersections

Correct characters should not contain overlapping areas.

The command **Edit Insert intersections** or the key combination **<Ctrl><P>** helps ensure this.

After choosing this command, an extra anchor point is inserted wherever two parts of a character **overlap**.

If the intersections are in different segments, then the character parts which lie inside the character outline are marked automatically.

Overlaps can then be easily removed using the command **Edit Delete** or the **<Del>** key.

<b>Note:</b> Very complex overlaps can lead to confusion about which parts are inside and can be deleted. Check the markings, and if need be change them using the marking tool.
--

<b>Tip:</b> If two character parts lie <i>exactly</i> on top of one another, one of the character parts is marked when this command is called. This makes it easy to find and remove duplicate character parts.
---

## Import AI EPS

The menu command **File Import AI EPS** makes it possible to import characters created in a graphics program, provided that the character is saved in **Adobe Illustrator format**.

If your graphics program cannot save in Adobe Illustrator format, it may be possible to achieve similar results by printing to a PostScript printer connected to a file. Files created like this are encapsulated PostScript files, which sometimes contain the same line and curve commands for curves and lines as an Adobe Illustrator file.

Choosing **File Import AI EPS** calls a standard filing dialog.

When you have chosen an AI file from this dialog, the file is read in. During this process, the program works out how much space the imported graphic needs.

If you choose **automatic Scaling**, then the program ensures that the imported graphic is not more than 1000 units high and 2000 units wide.

However, if you are importing a lot of characters which have already been sized relative to one another in the graphics program, we recommend that you use manual scaling.

The height and width of the imported graphic is shown immediately after changing the scaling factor.

<p><b>Note:</b> If you import Adobe Illustrator format graphics, then all the segments are not necessarily closed, and there may be various overlaps. It is usually necessary to tidy up the character with Type-Designers editing tools.</p>
---



## Background Pictures

Type-Designer allows you to use **bitmap pictures** and **bitmap fonts** as background items.

More details can be found with:

[Background display](#)

[From the Clipboard](#)

[Open Bitmap Font](#)

[Select Bitmap Character](#)

[Background Size](#)

## Background Display

Backgrounds are only displayed if the **Background** option is on.

**Note:** The hint editing tools turn background display **off**. You must select another tool to see the background again.

To improve system performance, it is a good idea to turn background display off during some operations.

**Note:** Displaying background bitmaps can be a problem for some graphics card and drivers at certain resolutions and color depths. At very high zoom levels, the background graphic may not be displayed any more. If you have difficulty displaying background graphics at normal zoom levels, we recommend that you use standard Windows drivers or obtain an updated driver from the manufacturer.

## Background Pictures from the Clipboard

Bitmap graphics can be pasted from the clipboard with the command Edit Paste or the key combinations <Ct**rl**><V> or <S**hift**><I**ns**>.

After pasting in the graphic, the **Background** display option in the Control Panel becomes available.

After turning this option on the graphic is displayed in the Edit Window.

Generally, it is a good idea to match the size of the bitmap to your requirements using the menu command Background Size.

## Bitmap Font as Background

You can use the command **Background Bitmap Font** to import bitmap fonts to use as a basis for developing your own Type 1 and TrueType fonts.

### Loading a Bitmap font

After selecting a bitmap font from a dialog box, it is loaded. If the font contains an error or cannot be read fully then an error message will be displayed.

It is possible to use very large or very small bitmap fonts as backgrounds, but you should note these points:

- Sizes up to 24 points are usually too crude to see the full structure.
- Sizes over 64 points can require a lot of memory space and can be very slow during redraws when resized.

The following formats are supported:

<b>HP soft font</b>	Uncompressed soft fonts for the most widespread laser printer standard (HP-Laserjet II).
<b>WITEX Laser Font</b>	Compressed and uncompressed laser printer fonts for the scientific word processor WITEX (Versions 3.x to 4.x).

Soft fonts are scaled for use as background masks according to the size information saved in the font. Unfortunately this is not always correct (particularly with Public Domain fonts).

Because of this, you can use the Background Size command to alter the background scaling factor.

## Selecting Characters

If you have loaded a bitmap font, you can use the command **Background Character** to choose a character to use as a background mask.

Simply select the required character with the mouse.

Note that the character will only be displayed if the **Background** option of the Control Panel is on.

## Background Size

Use this command to change the scaling factor of a background picture.

The larger the **x** and **y Factor** you choose, the bigger the bitmap displayed on screen.

As a guide, a 40 point bitmap font requires factors of about 60.

You can also change the size of a character after it has been created, using the [distortion tool](#).

## Mark All Character Parts

You can mark the whole character using the menu command **Edit Mark all** or the key combination **<Ctrl><A>**.

### See also

[Marking Tool](#)

[Delete All Markers](#)

## Delete All Markers

You can remove all the markings from a character using the menu command **Edit Remove Markers** or **<Ctrl><U>**.

### See also

[Marker Tool](#)

[Mark All](#)



## Global Transformations

The menu command **Global Transformations** calls a dialog where you can choose various transformations.

**Note:** Only those characters which are marked in the Character Map Window are affected by global transformation operations. To apply a transformation to the whole font, you must first select all the characters using the command Edit Mark all or <Ctrl><A>.

When you have chosen one or more of the possible transformations, click on the **OK** button to start the transformation for the marked characters.

During transformation, the transformation being carried out, the character being worked on, and the progress (in percent) are shown in the lower part of the dialog.

You may choose from the following transformations:

Stretching

Moving

Rotating

Italicize

Character Widths

Monospaced

Stroke Weight

Outline

## Global Stretching

You can use the menu command Global Trnasformations to **stretch (resize, distort)** a font globally.

Enter the percentage by which the character should be stretched horizontally and vertically into the **X** and **Y** fields.

Each character is stretched with reference to its origin, i.e. the intersection of its left margin and the baseline.

The character width is adjusted automatically.

<b>Tip:</b> By entering <b>-100%</b> you can generate a <b>mirrored</b> font.
---

## Global Moving

You can use the menu command Global Trnasformations to move the coordinates of all the marked characters in a font.

Enter the **X** and **Y** values by which the character should be moved.

**Tip:** Stretching a character by less than **100%** and then moving it, can be used to create **subscript** and **superscript** characters.

## Global Rotating

You can use the menu command Global Trnasformations to **rotate** all marked characters in a font.

By entering an **angle**, you can rotate the character clockwise.

Usually, a character which has been rotated will have to be moved to put it in a position enabling further editing.

<b>Note:</b> After rotation, it is usually necessary to adjust the <u>character boundaries</u> .
--

### See also

Adjusting Width

Monospaced

## Global Italicizing

You can use the menu command Global Trnasformations to **italicize** all marked characters in a font. Enter the **angle** by which the font should be inclined. A positive angle indicates inclination to the right.

<b>Note:</b> Angles larger than <b>20°</b> are not usually used.
--

## Adjusting Character Widths

You can use the menu command Global Trnasformations to **adjust the width** of all marked characters in a font.

Enter the percentage by which you wish to **stretch** or **compress** each individual character width.

This is comparable with stretching a font, but without changing the characters outline descriptions.

## Monospaced

You can use the menu command Global Trnasformations to generate a **monospaced** font.

If you enter a value larger than **-1**, the width of all marked characters is set to this value.

The left and right margins of all affected characters are adjusted symmetrically to the new width.

You can use this function to make a monospaced version of a proportional font, but the main use is to check that all the characters in a font are the same width.

## Stroke Weight Variations

You can use the menu command Global Trnasformations to vary the **stroke weight** of the marked characters.

Enter the value to increase the individual characters stoke weights. Both the **x** and **x** coordinates are altered. Positive values **thicken** and negative values **thin** the characters.

With these transformation, which work by reworking the outline descriptions, which normaly leads to bigger or smaller outline descriptions, optionally the altered characters can be returned to their original sizes.

If you want to do this, you can choose separately for the x and y coordinates whether or not **to adjust the size**.

**Note:** Rounding effects can create uneven stroke weights in different characters when adjusting sizes. These can only be adjusted by manual editing.

**Tip:** If the modification is minor, then Hints and the definition of stem widths can be used to achieve a uniform appearance. As the original hints loose their references to the outline description during transformation, use the Autohinting function to guarantee the new fonts quality

According to the complexity of the character and degree of modification, **overlaps** may be created during the transformation. For this reason, complex characters should be checked and, if need be, modified in the Edit Window.

**Tip:** The command Edit Insert intersections can be used to remove overlaps.



## Outline Font

You can use the menu command Global Trnasformations to generate an **outline** font.

The outline algorithm works on the same principle as **stroke weight modification**. However, the original outlines are kept, unlike in stroke weight modification where the original outlines are discarded.

Enter the **stroke weights** for the outline font which you wish to generate. A positive value places the new outlines outside the character and a negative value fits them inside the character.

Not all fonts are suitable as a basis for outline fonts. Bold sans serif fonts are better than thin fonts with serifs.

<b>Tip:</b> Best results can be achieved by placing an outline inside a bold font and not adjusting the size.
---

## All about Kerning

Kerning is the process of adjusting the spacing between pairs of characters.

Type 1 fonts save kerning information in the PFM and AFM files. TrueType fonts contain the kerning pairs in the TTF file itself.

**Note:** Well defined fonts do not usually need many kerning pairs. Before defining kerning pairs, check the normal spacing carefully, using the test function, as well as by printing sample texts.

Optimization of normal character spacing is important for other reasons.

- Only a few programs make use of kerning pairs.
- Those programs which do allow the use of kerning pairs may restrict the number of kerning pairs which can be used or
- allow them to be used only with Type 1 or TrueType fonts.

Type-Designer supports:

Manual Kerning

Automatic Kerning

## Manual Kerning

Start the **Kerning Editor** with the menu command **Parameters Kerning**.

On the left and right are list boxes from which the **left** and **right** characters of a kerning pair can be chosen, as well as additional elements which make it easier to select and set values.

Either the character names or the characters themselves can be displayed in the list boxes. Use the **Show names** button to change this setting.

The current kerning pair is displayed in the middle of the window.

The **Context** option of Settings Program allows character strings, which are displayed to the right and left of the selected character pair, to be defined.

If no character names appear in the box on the left when the kerning editor is called, then the font does not contain any kerning pairs (when it first opens, only predefined kerning pairs are listed).

To define new kerning pairs, activate the **All Characters** option.

If you choose one character from each list box, then the corresponding characters appear in the central window. In addition there is one horizontal and three vertical lines.

The horizontal line shows the baseline. The vertical lines show the character boundaries. The filled line shows the right boundary of the left hand character, and the two dashed lines show the boundaries of the right hand character.

You can find additional information with:

With the mouse

Direct Input

Zoom

UNDO

## Setting the Kerning Value with the Mouse

In the Kerning Window you can directly set the Kerning Value using the mouse:

=> Move the mouse cursor onto the dashed left boundary of the right hand character.

=> Press the **left mouse button** and hold it down.

=> Move the mouse to the position where you want to have the left boundary. The two dashed lines move with the mouse.

=> Release the .

The change in kerning is now shown.

## Direct Input of the Kerning Value

You can enter a value directly into the small edit box at the upper right. The change is shown immediately in the central area.

The two arrow buttons next to the small edit box can be used to make **fine adjustments**, either up or down.

Click on the **Reset** button to restore the original values.

## **Zoom the Kerning Window**

Resizing the kerning window affects the displayed character size.  
The larger the window, the larger the characters.

## UNDO in the Kerning Window

The command **Edit Undo** or the key combination <Alt><-> can be used to undo the last change to the selected character pair.

If you want to undo all the changes to the kerning values, make the Character Map window active and choose **Edit Undo** or the key combination <Alt><->.

## Automatic Kerning

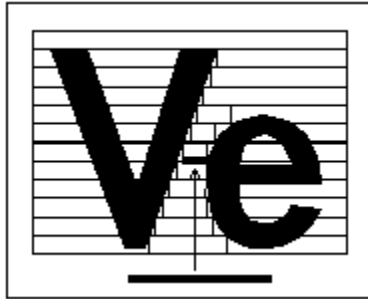
Type-Designer can work out kerning pairs automatically.

To call the Autokerning dialog, choose the command **Global Autokerning**.

As well as the parameters specified in the dialog, a text file containing the kerning pairs to be calculated is required. This is explained in the next section.

**Note:** Only the characters marked in the Character Map are affected by autokerning.

The autokerning algorithm uses the following principle:



Each character is divided into up to **64** bands.

**Maximum** and **minimum** x values are determined for each of these bands.

The maximum values from left hand bands are combined with the minimum values from right hand bands to obtain a kerning value with minimal spacing between the characters.

These settings control autokerning:

Min. Spacing

Min. Kerning

Max. Kerning

Number of Bands

Max. Number

Positive allowed

Overwrite existing

Select Kerning File



## Minimum Spacing

Enter the **minimum spacing** to be used at the *narrowest* point.

A value of 0 would mean that all of the characters touched.

## Minimum Kerning

Very small kerning values do not usually affect the fonts overall appearance.

The **minimum kerning** value to be used can be entered here.

We recommend values between 10 and 15.

## **Maximum Kerning**

This value determines the maximum distance apart two characters can be separated by.

The value entered here is used if there are no direct points of contact between the left hand characters bands and the right hand characters bands.

## **Number of Bands**

This value determines how many bands the character should be split into.

Values between **8** and **64** are allowed.

The larger this value, the more precisely the character contours can be fitted together.

More accuracy is however slower.

## Maximum Number

This value specifies the **maximum number of kerning pairs** to be calculated.

If this number is exceeded, then pairs with small kerning values are **deleted** automatically.

## Positive allowed

If this option is on, then **character blocking** is allowed.

It can make sense to use different parameters, especially for minimum spacing, for different character groups and blockings.

In particular, serif fonts which use kerning between uppercase letters often require a small amount of blocking.

## Overwrite Existing

If this option is on, then any kerning pairs which already exist in the font are overwritten with the new ones.

Kerning pairs can also be deleted, if the **Maximum Number** is exceeded.

## Select Kerning File

Click on the button under **Kerning File** to call a filing dialog from which you can choose a kerning file.

The kerning file lies down exactly which kerning pairs should be calculated, and which character combinations should be regarded as equivalent.

These kerning files are included with Type-Designer:

<b>ANSI.KRN</b>	Contains a complete description for the full ANSI character set. Because of the large number of possible kerning pairs, thousands of kerning pairs may be calculated.
<b>GERMAN.KRN</b>	Contains only the special characters required in German. This file is sufficient if you only use English and German.
<b>AEQUI.KRN</b>	Only equivalent characters from the ANSI character set are listed here. No completely new pairs are calculated, but existing ones are copied for equivalent characters.

## Kerning File

The **kerning file** is a text file which uses keywords and a simple syntactical structure to store information about the kerning pairs to be calculated.

They can be generated using any text editor. Apart from key words, only the character names are used.

Each character name and each keyword must be separated from other character names and keywords by a **space** or **line end**.

**Tip:** Instead of typing all the character names, they can be pasted in via the clipboard. Simply mark the characters whose names you wish to use in the Type-Designer Character Map and copy the characters into the clipboard. Then the paste names into the text editor.

The following keywords can be used:

SIMILARSLEFT

SIMILARSRIGHT

ALLCHARS

SMALLCHARS

PAIRS



## SIMILARSLEFT

The keyword **SIMILARSLEFT** starts an area in which equivalent left hand characters can be defined.

E.g. not just the pairs **Av**, **Aw** etc. should be calculated, but also. **Äv**, **Äw** etc. In general, the kerning values for **A** are identical with those for **Ä** and so the value for **Ä** can easily be taken from those for **A**.

Letter equivalents can be defined as followond this keyword, according to:

=> The first character name determines the basis character. Kerning values are calculated for this character.

=> Following this, enter all the characters which are equivalent to the first character.

=> End each definition with the word **END**.

**Example:**       SIMILARSLEFT  
                  A Adieresis Aogonek END

## SIMILARSRIGHT

Character equivalents can be defined for the right hand side of the kerning pair, using the same rules as for the left hand equivalents.

**Example:**       SIMILARSRIGHT  
                  c ccedilla END

## ALLCHARS

You can enter all the character names here which are used in the right side of kerning pairs.

Enter all the names and end the definition with the keyword **END**.

Later, you can simply use the placeholder **ALL** for all these character names. This also saves some typing.

**Example:**

```
ALLCHARS
quoteright parenleft parenright ...
A B C D ... x y z
Lslash lslash ... ydieresis zcaron
END
```

## PAIRS

All character pairs to be included in the calculation are defined after this keyword.

Each left hand character must have its own definition if it is not already defined by an entry in the SIMILARSLEFT section.

After the character names for the first character, there is a list of all the characters which should be combined with the first character.

In addition the placeholders **ALL** and **SMALL** can be used here.

Each definition must end with the keyword **END**.

**Example:**

```
PAIRS
period quotedblright quoteright END
parenleft ALL END
A ALL END
B ALL END
a SMALL END
b SMALL END
dotlessi SMALL END
```

## SMALLCHARS

Usually character combinations made up of a small letter followed by a capital, e.g. **vA** do not usually appear in the kerning table. Small letters therefore should not be combined with all characters (ALLCHARS).

Characters can be defined after the keyword **SMALLCHARS** which are usually only combined with small letters.

The placeholder **SMALL** can be used later for all these characters.

**Example:**

```
SMALLCHARS
quoteright parenleft parenright ...
a b c ... x y z
lslash ... ydieresis zcaron
END
```

## All about Hints

All font technologies which have high quality standards use **Hints**, to improve output quality above all at small sizes and low resolutions.

The following problems can arise at small sizes:

Different Stem Width

Appearance with no Overshoot Control

Information on how you can use Type-Designers hinting functions to overcome this problems can be found with:

Setting Hints

Starting Point and Hint Replacement

Segment Order

Elementary Stems

Overlapping Hints

Overshoot Control using Hints

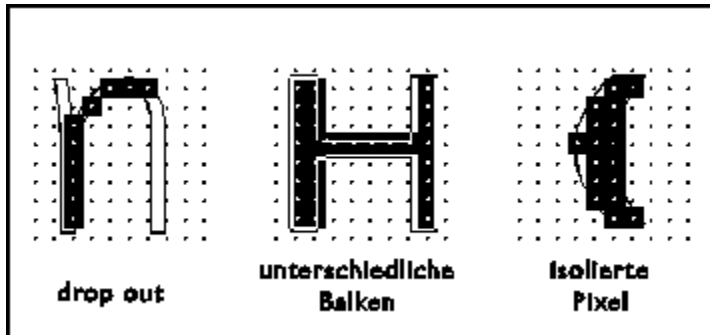
More interesting details and examples can be found in the printed manual and with

Autohinting feature of Type-Designer.

## Different Stem Width

If you map a mathematical character outline description onto an output grid, problems inevitably arise with the character representation.

Note that the term **stem** is also used within roundings. If you look at, for example, the right hand rounding of the character »O«, the **stem width** of this arc is defined as the horizontal distance from the outermost points.



The first case shows an effect known as **drop out**. Although there is an outline, it does not fall on any pixel in the grid. This is possible if an outline runs between adjacent points in the output grid.

The second typical problem is that stems which have the same width in the outline are rasterized with different numbers of pixels. Letters like »H« then appear unsymmetrically.

Finally, with curves the problem of individual pixels not appearing or being surplus to requirements can occur in roundings. This effect can be avoided by laying the end points of the curve on the center of output grid points, a process called **grid fitting**.

## Appearance without Overshoot Control

With the description of the Alignment Zones we mentioned the necessity of allowing rounded characters to extend a little further than characters with flat tops or bottoms.

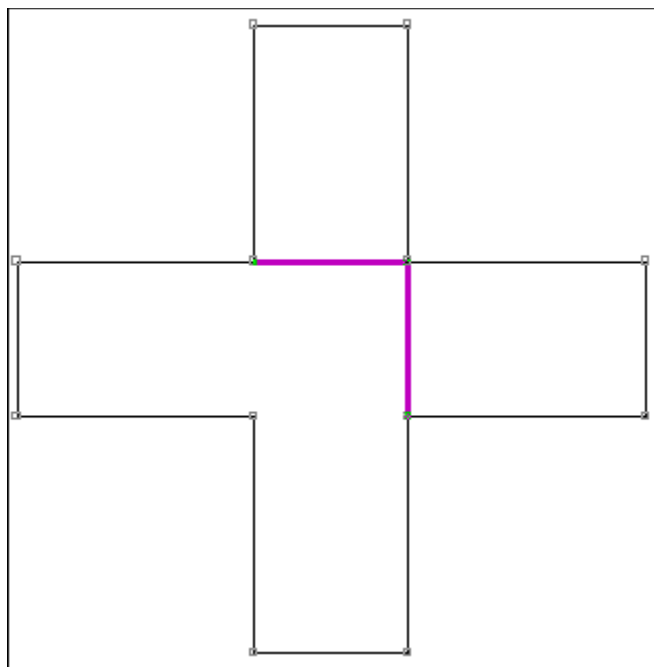
If you allow this when there are only a few pixels available to represent the character, the text can easily appear untidy as the height of characters in the same font can vary.





## Elementary Stems

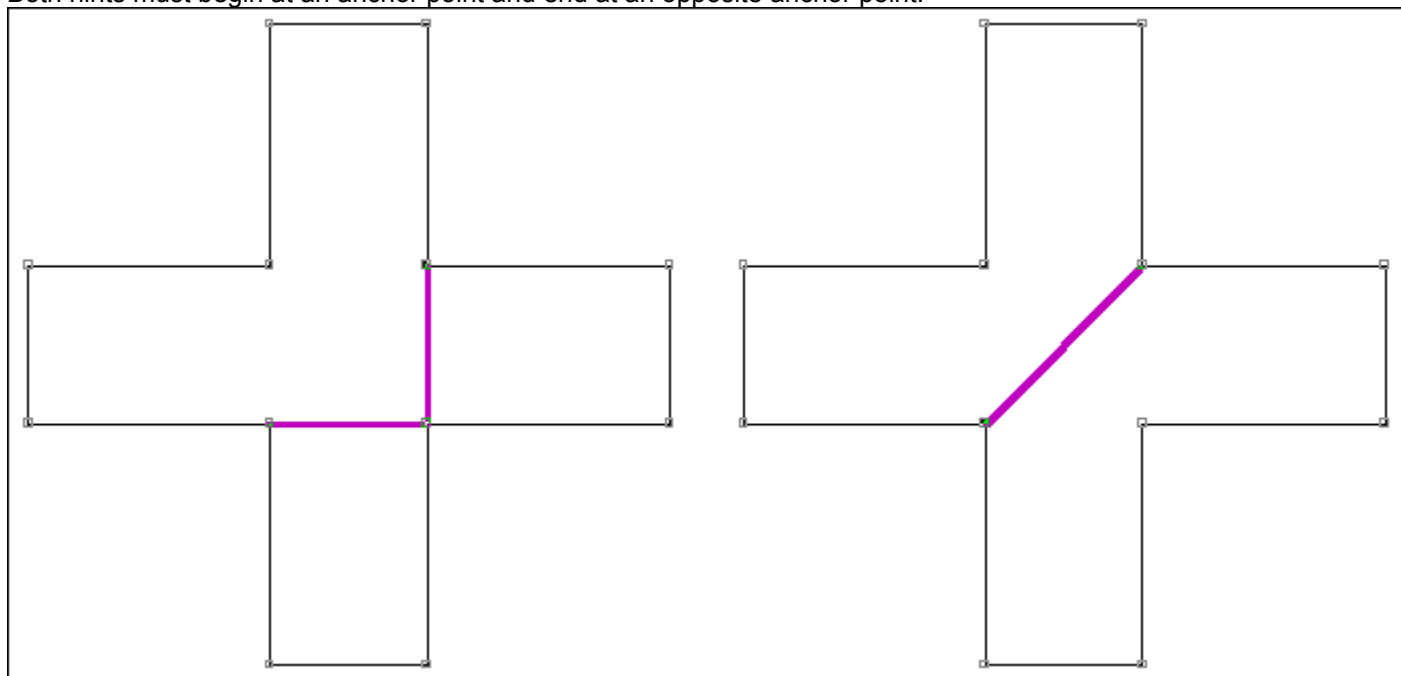
Horizontal and vertical stems in a character should always have stems applied to them.



The **plus**” symbol consists of just one horizontal and one vertical stem.

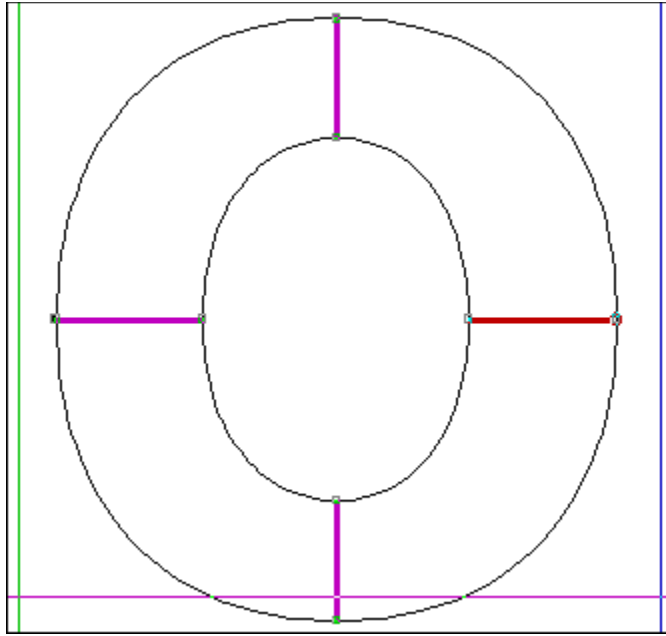
To optimize the vertical stem, a horizontal hint must be inserted, and to optimize the horizontal stem a vertical hint must be inserted.

Both hints must begin at an anchor point and end at an opposite anchor point.



It does not matter which anchor points are used as the start points, as long as only the stems defined by the opposite points are marked by a hint.

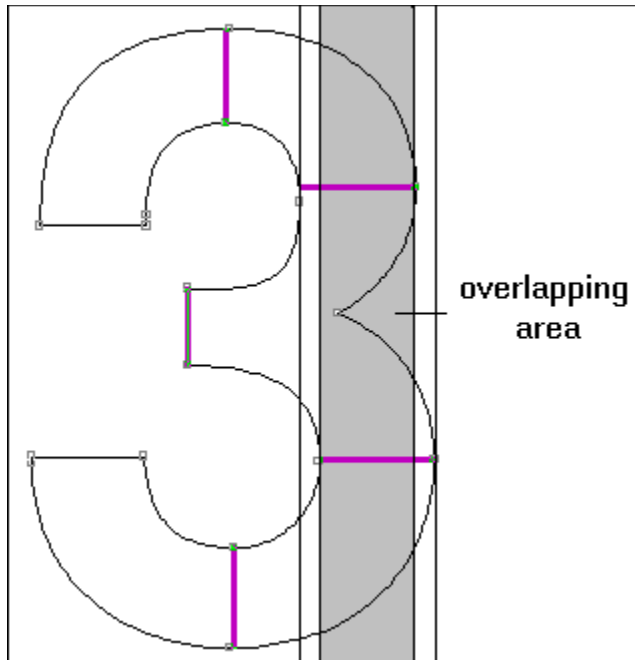
As already mentioned, stems can also be used in roundings. In this case it is important that the end of curves lie at extrema, as tangents parallel to the x and y axes touch these points.



You can use the command Edit Insert extrema to ensure that there are actually anchor points at the extrema. If this causes two anchor points to fall very close to one another, it is a good idea (at least for autohinting) to remove the anchor point which does not lie at the extreme. Use the separate tool to do this. Note the tip given there.

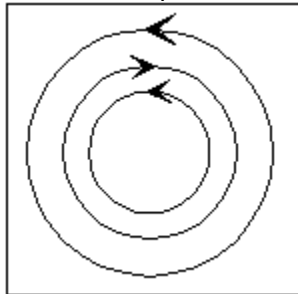
## Overlapping Hints

One particular feature of hints in Type 1 format can make the hinting process very complex.

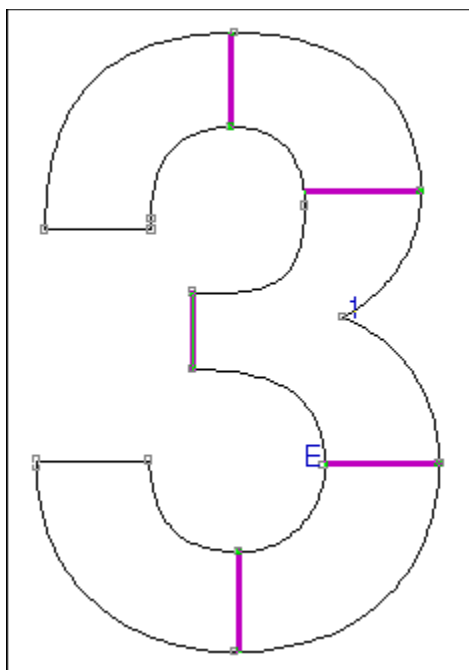


Whenever two stems which can use hints overlap, i.e. both partly cover the same coordinates, then the rasterizer has to be told, using a hint replacement which hint to use for which part of the character.

To set hints in such a case, the working method of the rasterizer must be followed. The segment order and the start point are very important, as is the orientation of each segment.



As shown in the figure above, segments which lie within each other always use opposing orientations. The segments are built up on the output grid one after another according to their orientation, beginning with the relevant start point.



If a hint covering an area already covered by a previous hint is found, a hint replacement has to be made at this point.

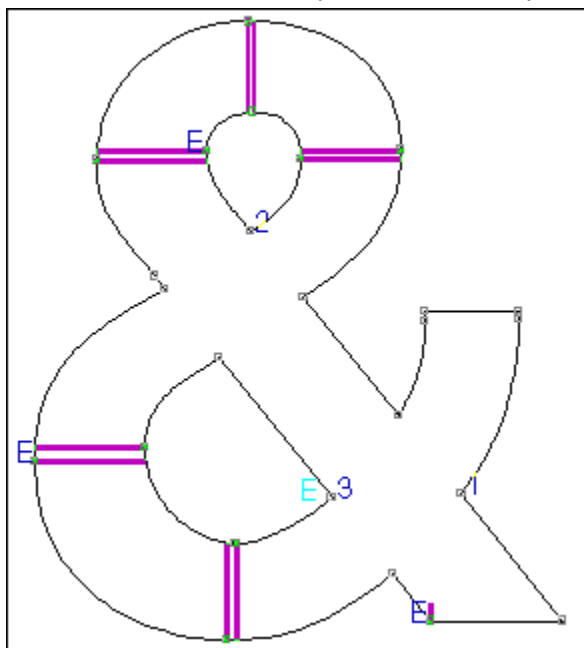
Study the outline of the character »3« in an anticlockwise direction from the start point.

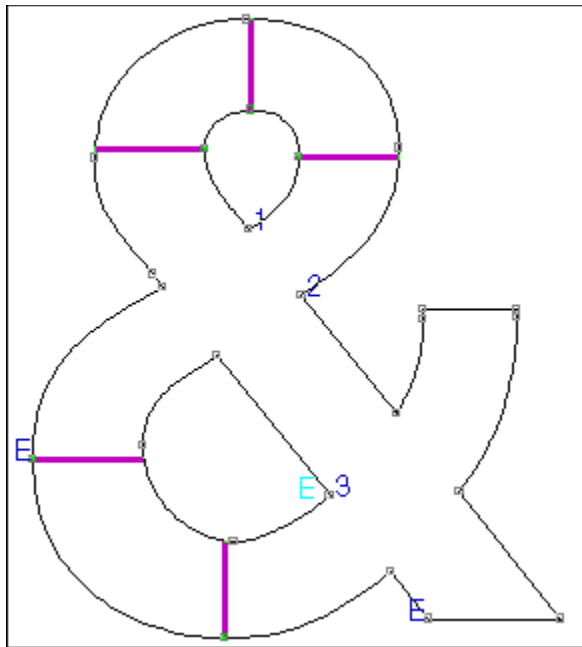
As long as no two horizontal or vertical hints overlap, then the rasterizer will not have any problems in handling the character properly. However, the hint beginning at point »E« partly covers the same area as the upper right hint.

This point must therefore be marked as a hint replacement point.

From this point on, the hints previously used are no longer valid.

The following figure shows, how new hints have to be set in a complex character, even if hints already existed at the coordinates prior to the hint replacements.





The choice of segment order and the position of the start point can be used to minimize the number of hint replacement points and the number of repeating hints.

## Overshoot Control using Hints

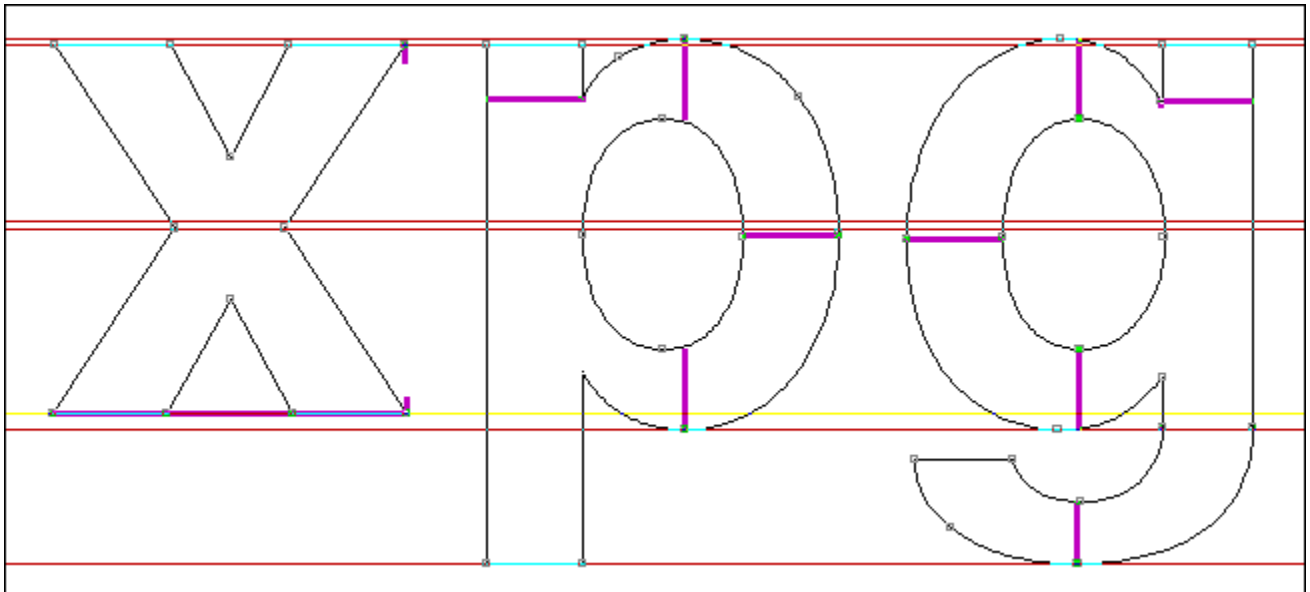
Using Alignment Zones and hints, you can carry out overshoot control.

At small sizes, all anchor points which lie in an alignment area and coincide with the start or end point of a vertical hint, are aligned in the same way.

It is necessary to distinguish between alignment *to the top* and *to the bottom*. This following figure shows overshoot areas for the **baseline**, **ascenders**, **x-height**, and **descenders**.



Anchor points which extend below the baseline or the descenders and lie between the corresponding horizontal lines are aligned upwards, while anchor points which extend above the x-height are moved downwards.



The vertical hints used with »x« is only needed for overshoot control. These hints are referred to as **ghost stems** and are only usefull if they lie inside an alignment zone.

## **Autohinting**

Type-Designer has powerful autohinting functions which make it easy to generate high quality Hints with minimal effort.

To ensure optimal performance, the

### **Autohinting Parameters**

must be matched to the current requirements.

Hints for an individual character can be added automatically by clicking on the Edit Windows Hint button.

The

### **Autohinting Dialog**

can be used to place hints in a whole font (or selected characters).

Additionally Extrema can be inserted and Stem Widths and Alignment Zones can be calculated.

## Autohinting Parameters

Choose the menu command **Settings Hinting** to call a dialog where you can specify autohinting parameters.

Separate values must be entered for the **x** and **y** coordinates (i.e. for **horizontal** and **vertical** hints).

The following Parameters can be set:

Minimum Hint

Minimum Structure

Overlap Distance

Gradient

Negative



## Minimum Hint

Enter the **minimum length** which a hint may have here.

Bold fonts usually require larger values than thin ones.

This setting ensures that the smallest structures (which do not affect appearance at small sizes) are not included in the hinting process.

## Minimum Structure

The **Minimum Structure** specifies how long areas have to be regarded as stems by the autohinting process.

If too high a value is specified, useful hints may not be set.

If too small a value is set, too many hints may be generated.

## Overlapping Distance

When using autohinting, certain character forms, in particular slight roundings, can cause a lot of hint and hint replacements to be generated at almost identical coordinates.

This value determines the spacing between anchor points before a new hint is generated.

## Gradient

In slightly italic characters it can make sense to treat gentle slopes exactly like horizontal or vertical lines.

This entry specifies the **gradient** up to which this should happen. For **horizontal** hints the slope **s** of a line is defined by the formula:

$$s = \text{ABS}((y_2 - y_1) / (x_2 - x_1))$$

For **vertical** hints the enumerator and denominator change places.

## Negative Characters

**Negative characters** are those which appear white on a black background.

In a font, these are usually those characters placed on a rectangle, ellipse or similar.

Hinting such characters often results in a distortion rather than an improvement in appearance.

There are also characters in which both cases apply. It cannot be a coincidence that the Yin-Yang symbol (amongst other things a sign of balance and harmony) is one of these characters.

## Global Autohinting

Choosing the menu command **Global Autohinting** calls a dialog where you can choose to calculate character related hints and hinting-related parameters.

<b>Note:</b> Autohinting only affects those characters <u>marked</u> in the Character Map.
--

The following calculations can be done for you:

Extrema

Hints

Alignment Zones

Stem Width

## Extrema

**Extrema** are inserted into all the marked characters, exactly as extrema are inserted into a character in the Edit Window by choosing Edit Insert extrema.

This option is particularly useful after italicizing a font, or when working with public domain fonts.

## Hints

If this option is active, hints are inserted into every marked character using the den Autohinting Parameters.

This is the same as clicking the Hint button in the Edit Window.



## Alignment Zones

The alignment zones required for overshoot control, which is important at small sizes, are calculated if this option is selected.

It is calculated from a statistical distribution of plausible values for anchor points with vertical hints:

- To make a sensible choice a suitable number of characters must be available.
- On the one hand the hints available in the characters are used to calculate the guidelines. On the other hand alignment zones affect autohinting.

<p><b>Tip:</b> Enable both the <b>Hints</b> and <b>Alignment Zones</b> options. Check the guidelines which are generated visually in the Edit Window using various character pairs, e.g. <b>g</b> and <b>p</b>, <b>x</b> and <b>o</b>, as well as <b>X</b> and <b>O</b>. If need be, make corrections by hand and then carry out the hint generation again.</p>
---

## **Stem Widths**

A statistical distribution of frequently occurring hint lengths is also used to calculate the stem widths.

This ensures a consistent appearance for the font at small sizes.

Stem widths should also be controlled manually.

## File Menu

You can use entries in the **File Menu** to create new fonts, load existing fonts or import TrueType fonts.

After carrying out one of these commands, further entries appear in the **File Menu**, allowing you to close and save fonts, import kerning tables, export TrueType fonts and print the current font.

In addition, this menu contains a command for exiting Type-Designer.

You will also see that Type-Designer adds the last eight fonts opened to the end of the file menu so that you can open one of those fonts by choosing its menu entry.

You can choose from the following menu entries:

New

Open

Close

Save

Save as

Build TrueType

Import TrueType

Import CFN

Import Kerning

Printer Setup

Print

Exit

## File New

Use the menu command **File New**, if you want to create a new font.

Initially, the active directory is always the one specified as the PFB Directory using Settings Program. By using the **Directory** list box you can select any other directory.

Next, enter a new filename, which is not already in use.

Choose the **Character Set**, to be used by your new font. Choose from:

<b>Latin Characters</b>	Contains all the roman character names supported by original Adobe fonts.
<b>Symbols</b>	Contains all the character names from the Adobe symbol set.
<b>Other</b>	Contains only the <code>.notdef</code> and <code>space</code> characters which each font must contain. All further characters have to be inserted into the font later.

After this, you go automatically to the FontInfo dialog, where you can specify necessary and informative font information.

## File Open

Use this menu command to load a font for editing.

Simply select the file to be edited from the list in the dialog, as usual in Windows.

The **font directory** can be specified in the dialog called with the Settings Program command. This directory is then used automatically when loading fonts.

The following options are available with the Open dialog:

Font Names

Font Sample

Delete

**Font Names**

The actual font names, and not the file names, will be displayed if you enable the **Font Names** option.

### **Font Sample**

In addition to the font names, by enabling the **Font sample** check box you can display a sample of the font.

After setting the check box, you must click on an entry in the left hand list box.

Previews are only possible if Adobe Type Manager is installed and active and Type-Designer can find the relevant PFM file.

**Delete**

The **Delete** button physically removes all files relating to the selected font: the PFB, PFM, AFM and INF files. In addition all related entries in the ATM.INI and WIN.INI files are removed.

This option is very useful if, for example you have altered a font step by step and saved each version in a separate file, to remove all the extra files

Use this option with care as physically removed fonts may only be recoverable with difficulty, or not at all.



## **Font Close**

Use this menu option if you do not need a Font any longer. All windows descendant from that font are closed

If you haven't saved the font, you now will get a message box asking you if the font should be saved.

## Font Save

Use this command to save a font.

Only save imported TrueType outlines as a Type 1 font, and use that font as the basis for all subsequent modifications.

You can choose which files Type-Designer should create. We recommend that you always create the PFB, AFM and PFM files.

If you are only using Type-Designer as a kerning editor then do not create the PFB file.

INF files are only rarely used, and need not be created.

The following Options are available with the Save Dialog:

Install

Save without Hints

## Back Up Copy

Type-Designer automatically makes a file called

`<Filename>~FB`

which contains the previous version of the font.

## Adobe Type Manager

Unfortunately, it is not possible to modify fonts which are being used by Adobe Type Manager. If a font which is being used by Adobe Type Manager is modified then in some circumstances a system crash might occur. After modifying a font which is being used by Adobe Type Manager, Windows should always be restarted.

If need be, a dialog is displayed which allows you to restart Windows quickly.

## **File Save As**

Use this menu command if you want to save a font under another file name.

It calls the same dialog as File New.

Enter the file name for the new font, just as for File New.

After this, you have the same options as for File Save.

If a file with this name already exists, the program asks whether or not it should be overwritten.

## TrueType Erzeugen

This command enables you to build TrueType fonts.

The following options are available:

Automatic Install

Other Filename

Clicking on OK starts the build process.

To show how far the procedure has got, the character being worked on and the progress, in percent, are shown at the bottom of the dialog.

If a TrueType font with the same name already exists, you will be asked whether or not you wish to overwrite the file.

If an error occurs during the build, a warning message appears.

If the process is completed without errors, then the new TrueType font is immediately available to all Windows applications which support scaleable fonts.

## Automatic Install

If you want to install the newly created TrueType font into Windows, then leave the **Install Font in System** option active. This automatically creates a `.FOT` file, which gives Windows important information about the TrueType font. In addition, the `WIN.INI` file is updated automatically.

This option must also be active if the corresponding TrueType font already exists and you have simply modified it.

## Other Filename

Normally the TrueType font is given the same file name as the Type 1 font, but with the extension `.TTF`.

It is normally stored in the Windows\System directory. You can choose a different file name or directory by clicking on the **Other Filename** button. This calls the same dialog as the menu command File Save as.

The directory where you want to store TrueType fonts can also be specified by choosing Settings  
Program.

## Import TrueType

The menu command **Import TrueType** makes it possible to use outlines from installed TrueType fonts in your own characters.

This is done via the standard Windows fonts common dialog. Bear in mind that only installed fonts are shown.

If you wish to use an outline from an uninstalled TrueType font, you must install it first.

You can change the size of the sample displayed in the **Sample window** from the **Size list**, but this will have no effect on the imported outline.

When you have selected a font and clicked on OK, all the character outlines belonging to that font are loaded into Type-Designer.

The font information and character names visible in Type-Designer are based on values extracted from Windows system functions.

**Note:** Unfortunately, it is not possible to extract high-quality character descriptions from TrueType fonts. In particular, all hinting information is lost during the import process. The complexity of this problem can be compared to trying to extract source code from an EXE file. Therefore we recommend that you always use Type 1 fonts as the source for your fonts to modify. You can build high-quality TrueType fonts from such outlines. In complexity, this is like compiling a Pascal program. If you still have problems importing TrueType fonts, then carry out the process again. Do not import, export and import again. Instead, save the imported outline as a Type 1 font and use that font as the basis for all modifications, even if you do not intend to use Type 1 fonts directly.

**Tip:** Using Type-Designers Autohinting function, it is possible to insert Hints into imported TrueType outlines.

## Import CFN-Font

The menu command **Import TrueType** makes it possible to use outlines from fonts in the Calamus CFN format in your own characters.

You will see a dialog for selecting files.

Das standard path for CFN fonts can be defined through the menu item Settings Program.

Simply select the font you want to import and click the **OK** button.

**Hint:** It is not possible to extract high-quality character descriptions from CFN fonts, because they have no hinting information at all. Also you should always check the Font Informations and correct them if needed. Another problem may arise from the encoding of CFN fonts. After the import they will always have StandardEncoding (Windows-ANSI) assigned. Since there are other characters and positions used in CFN format, this may result in uncomplete fonts. You can however load a the encoding file `CFN.ENC` through the Encoding Dialog to assign the original encoding. This is especially useful with **symbol fonts**.

**Tip:** Using Type-Designers Autohinting function, it is possible to insert Hints into imported CFN outlines.



## Import Kerning

Use this command to import the kerning information from an existing font.

You can choose to import the kerning information from an AFM or PFM file. PFM files only hold kerning data for encoded characters, but AFM files can hold kerning data for all the characters in a font. It only makes sense to import kerning information from an AFM file if the same character names are used both in the current font and the AFM file.

**Note:** It only makes sense to import kerning pairs if the fonts have similar characteristics. Even then, it is almost always necessary to check the imported kerning pairs by hand.

**See also:**

[File Types](#)

## Printer Setup

If you have more than one printer, you should make sure that it is correctly set up using this command before making test prints of a font.

Simply choose the printer which you want to use for the test prints. If you do not select a specific printer, then the default printer will be used.

## Printer Configuration

Clicking on the **Setup** button calls the Printer Setup dialog which allows you to change the settings for the selected printer. For example, you can choose whether to print in portrait or landscape orientation. The options for different printers can be very different: if you have problems, refer to your Window manual or the printers handbook.

<b>Note:</b> Type-Designer does not change the default printer and its configuration. If you want to change the default printer, use the Printers option in the Windows Control Panel.
--

## **Print**

This command allows you to make various test prints of fonts.

The following options are available:

[Character Map](#)

[Single Characters](#)

[Kerning Pairs](#)

[Encoding](#)

[Sample Text](#)

## Character Map

This option prints an overview of all the characters in the current font. Up to 256 characters are printed on each page.

<b>Tip:</b> If printing in portrait orientation leads to adjacent characters overlapping, then switch to landscape orientation.
---

## Single Character

Using this option, it is possible to print enlargements of single characters. Because of the higher resolution of printers, it is possible to judge details of designs better on a print out than on screen. To save paper, up to four characters are printed on a page.

<b>Note:</b> The character to be printed must first be <u>selected</u> in the Character Map Window.
---

## **Kerning Pairs**

This option allows you to print out all the kerning pairs in a font.

In portrait orientation, you can print up to 128 kerning pairs per page, and in landscape orientation up to 256 per page.

## Encoding

This option is useful if you want to get an overview of the encoding, i.e. the allocation of characters to keys and their ANSI positions.

## Sample Text

To judge the appearance of a font, it can be helpful to print out a piece of text using it. You can choose to print any text file: click on the TEST.TXT button to call a dialog where you can choose a text file to print. You can create this file using any text editor. Type-Designer formats the text according to the following settings.

**Tip:** You can include unassigned characters in your sample text. To do this, insert the name of the character preceded by a »/«. Put a space or line end after the character name. You can insert the »/« character by entering »//«.

Example: /questiondown /fi

The following table gives the effects of the various options on the printed sample.

<b>Text Size</b>	You can enter the text size (in points) here.
<b>Line Sp.</b>	Enter the distance between lines (in points) here.
<b>Justified</b>	The text will normally be left aligned. By choosing <i>Justify</i> the spacing is adjusted so that each line is the same length.
<b>With Kerning</b>	This option must be enabled if the sample is to be printed using the fonts kerning pairs.

**Note:** Sample printing is restricted to one page. At small sizes, the length of each line is reduced to make it easier to evaluate the text.



## **Exit Type-Designer**

With this menu option it is possible to exit from Type-Designer.

If some of the open fonts were not saved, you now will get an message box remembering you to save the fonts.

With Setting Program you can set an option if Type-Designer should be exited immediatly or if a security check should apply.

## Install font

Type-Designer can automatically install a Type 1 font into your Windows system.

You can install in two ways:

- |                  |   |
|------------------|---|
| <b>Permanent</b> | Choose this option if the font is also to be available in future Windows sessions. The WIN.INI and ATM.INI files are altered accordingly.                   |
| <b>Temporary</b> | Choose this option if the font is to be installed but not available in future Windows sessions. In this case the WIN.INI and ATM.INI files are not altered. |

## **Save Without Hints**

Under certain circumstances, problems can arise with very complex fonts and PostScript image setters. These problems are not usually evident with Adobe Type Manager or PostScript laser printers. If your font causes problems on an image setter, save it under a different name without hints.

