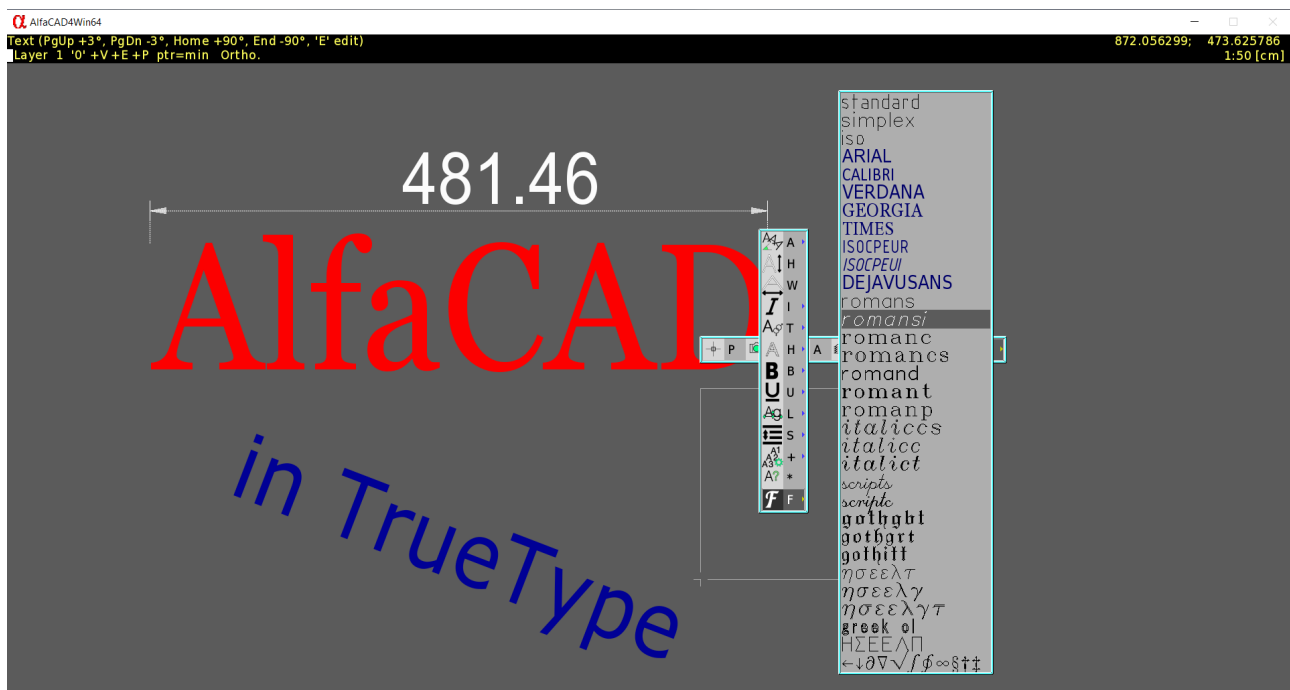


AlfaCAD4Win in TrueType

The purpose of this short message is to provide the basic information about the latest changes and enhancements, and how to use them to improve your work and increase overall effectiveness of using AlfaCAD.

TrueType fonts.

This is good news for anyone who cares about the high quality of drawings, especially their printouts. TrueType fonts have been implemented in AlfaCAD both in the graphical interface (in menus and dialog boxes) and in drawings. TrueType fonts are now available on a par with vector fonts, implemented in the previous version of the program, and recently also optimized to speed up graphics operations.



The AlfaCAD package includes vector fonts in AFF format and legacy format CHR fonts (low quality, implemented long time ago, left only for the compatibility with archival drawings made in AlfCAD-Pro and Aster-CAD programs).

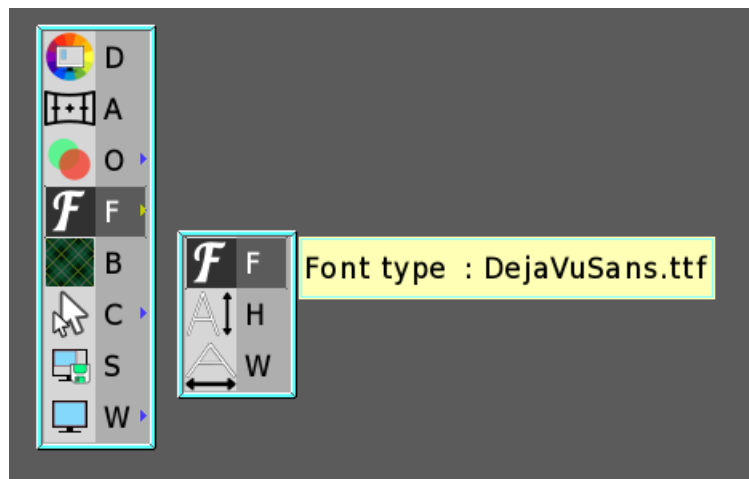
Next to them all fonts accessible in your Windows system (usually in folder \Windows\Fonts) can be used, both as a user interface basic font, and in drawings, freely scalable, identically to the vector fonts. Moreover, any TTF fonts, for example downloaded from such pages like <https://www.101freefonts.com> or www.ultimatefontdownload.com and many other, can be used.

In fact, on technical drawings we should avoid such cacophony of fonts, rather using one or at most few different font styles, especially those looking more "technical" than "fancy". That's why in primary AlfaCAD fonts setup (in ALFACAD.INI file) just few TTF fonts are selected, mostly useful for technical drawings. However, at any time any new font can be added freely and once is used in any drawing, re-edition of this drawing will trigger loading of this font again, even if is not included in the primary setup.

In further part of this message there will be more about TrueType fonts use. It's just worth to mention now that next to the high quality TrueType fonts implemented on the screen, also even much higher quality of TrueType fonts were implemented on printouts, both printed directly on the printer, and saved in output image files in several formats as well as in PDF documents.

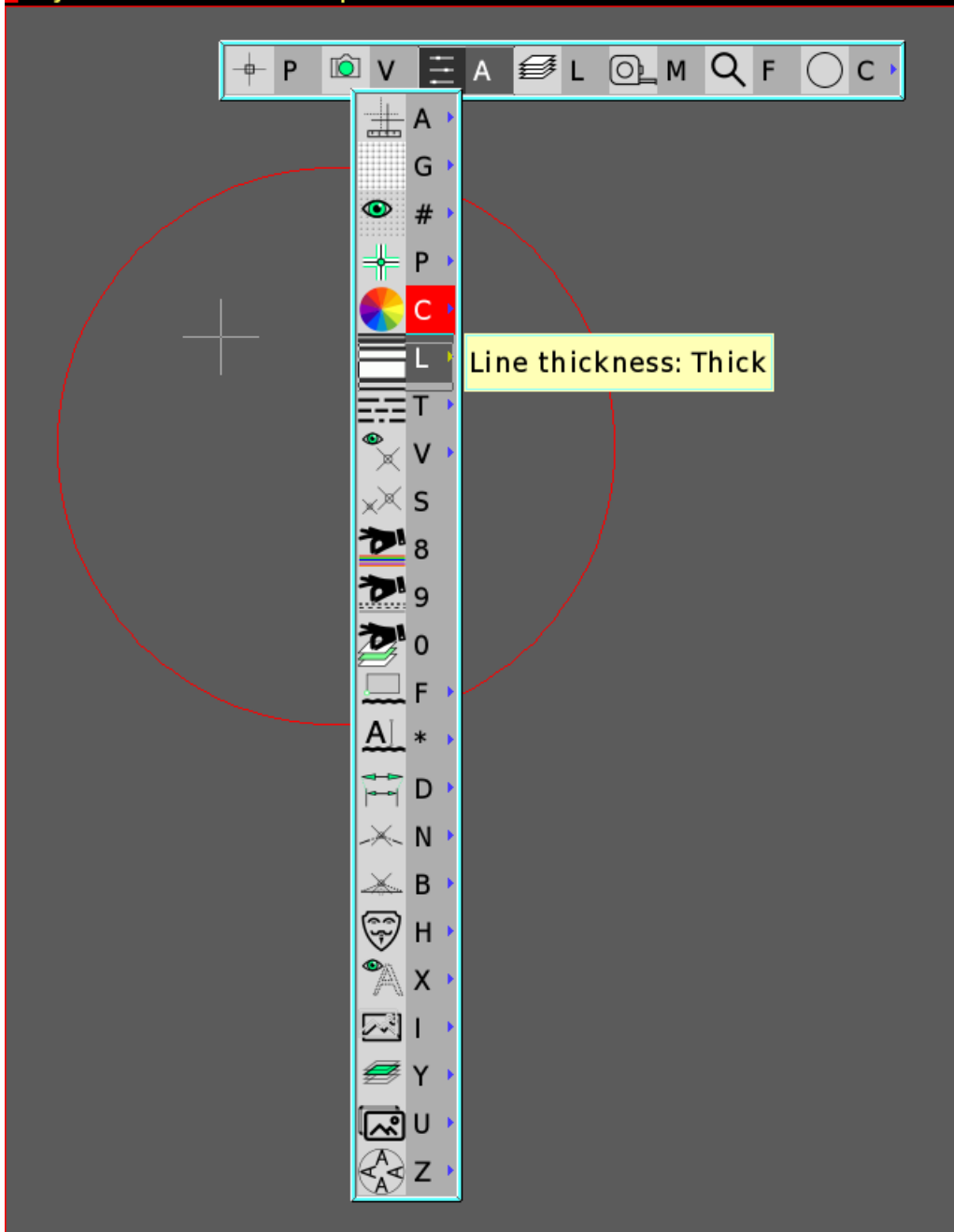
Graphic User Interface

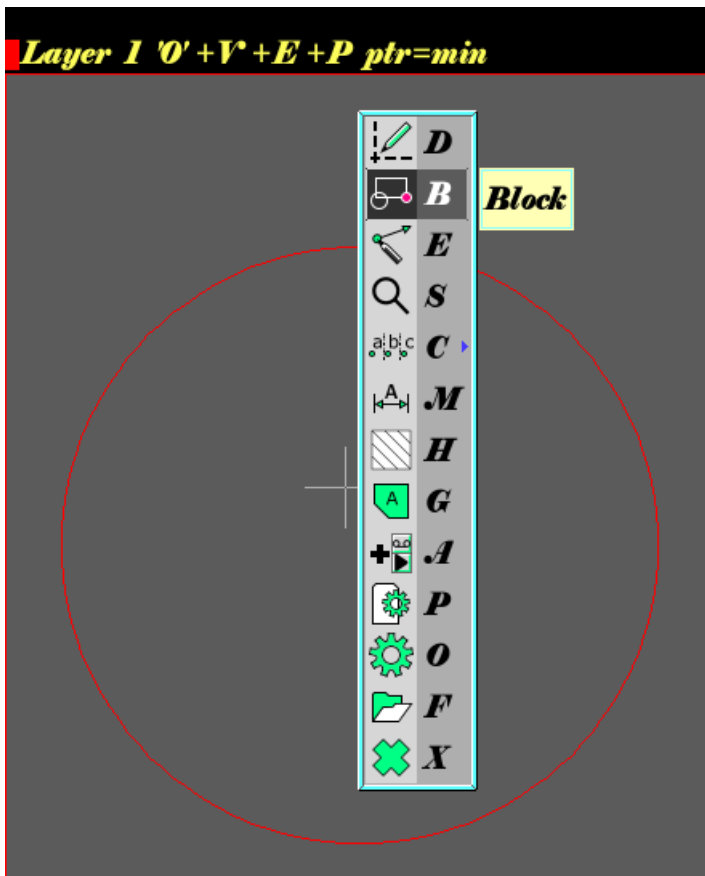
Drawing desktop TruType font, as well as fonts high and width factor can be (almost) freely selected in **Font** function in **Options** menu. Pointing to Font function and waiting for a tip 2 seconds lets you know what font is currently installed. The same information comes in tip window for height and with factor. Height can be selected between 12 pxl (insanely small) to 30 pxl (insanely big). Recommended height is between 17 and 24 pxl, so 21 is the best choice.



Circle - CENTER - radius

Layer 1 '0' +V+E+P ptr=min

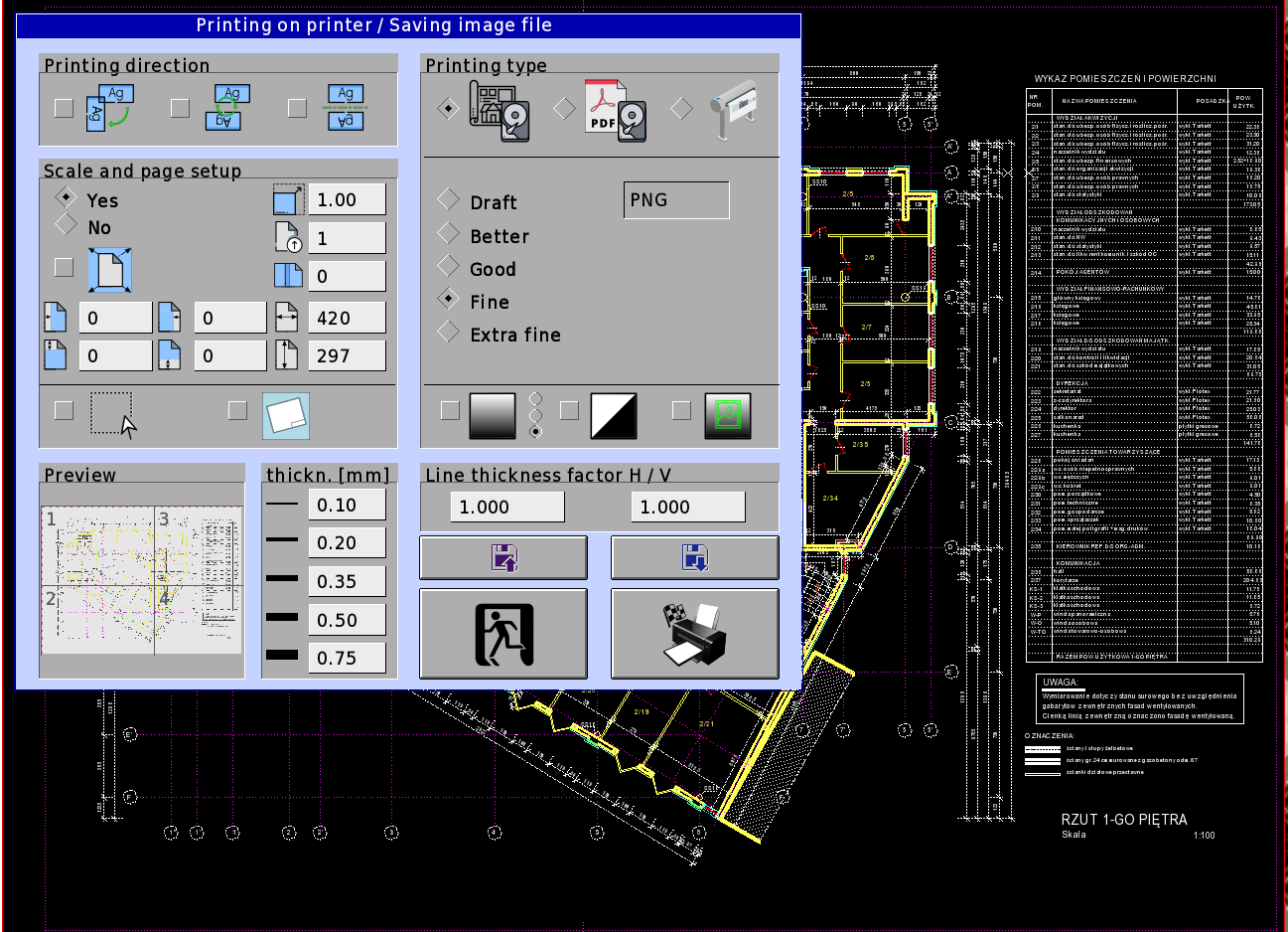




Font style can be freely selected among all Windows fonts, like on the example, where some fancy font style is used, if somebody likes it. By default DejaVuSans.ttf is used (free, public domain font style delivered with AlfaCAD). All menus, dialogue windows end edit fields are using active font, while dialogue windows are scaled to fit all texts depends of the weight and width of characters. On images below “reasonable” size of default font (21 pxl) then smallest ever available (12 pxl) and nearly biggest (28 out of maximum 30 pxl) filling up nearly whole notebook's screen to present printing dialogue window. Please use those setting reasonably.

C:\Users\Marek\ALFACAD64\Drawings\Samples\Opole\BIURA_PZU\RZUTY\1PIETRO.OLF

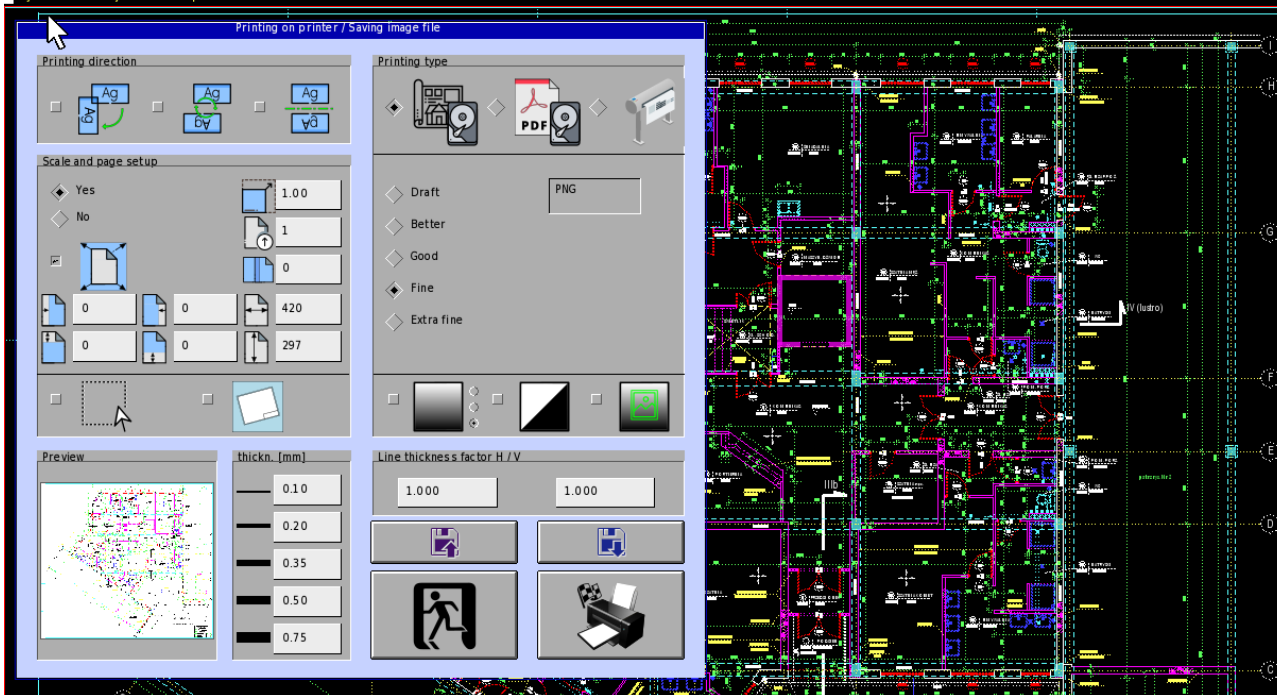
File C:\Users\Marek\ALFACAD64\Drawings\Samples\Opole\BIURA_PZU\RZUTY\1PIETRO.OLF
Layer 1 '1 PIĘTRO' +V+E+P Mode:Intersection ptr=min Ortho.



C:\Users\Marek\ALFACAD64\Drawings\Samples\example_4_1.alf

File C:\Users\Marek\ALFACAD64\Drawings\Samples\example_4_1.alf

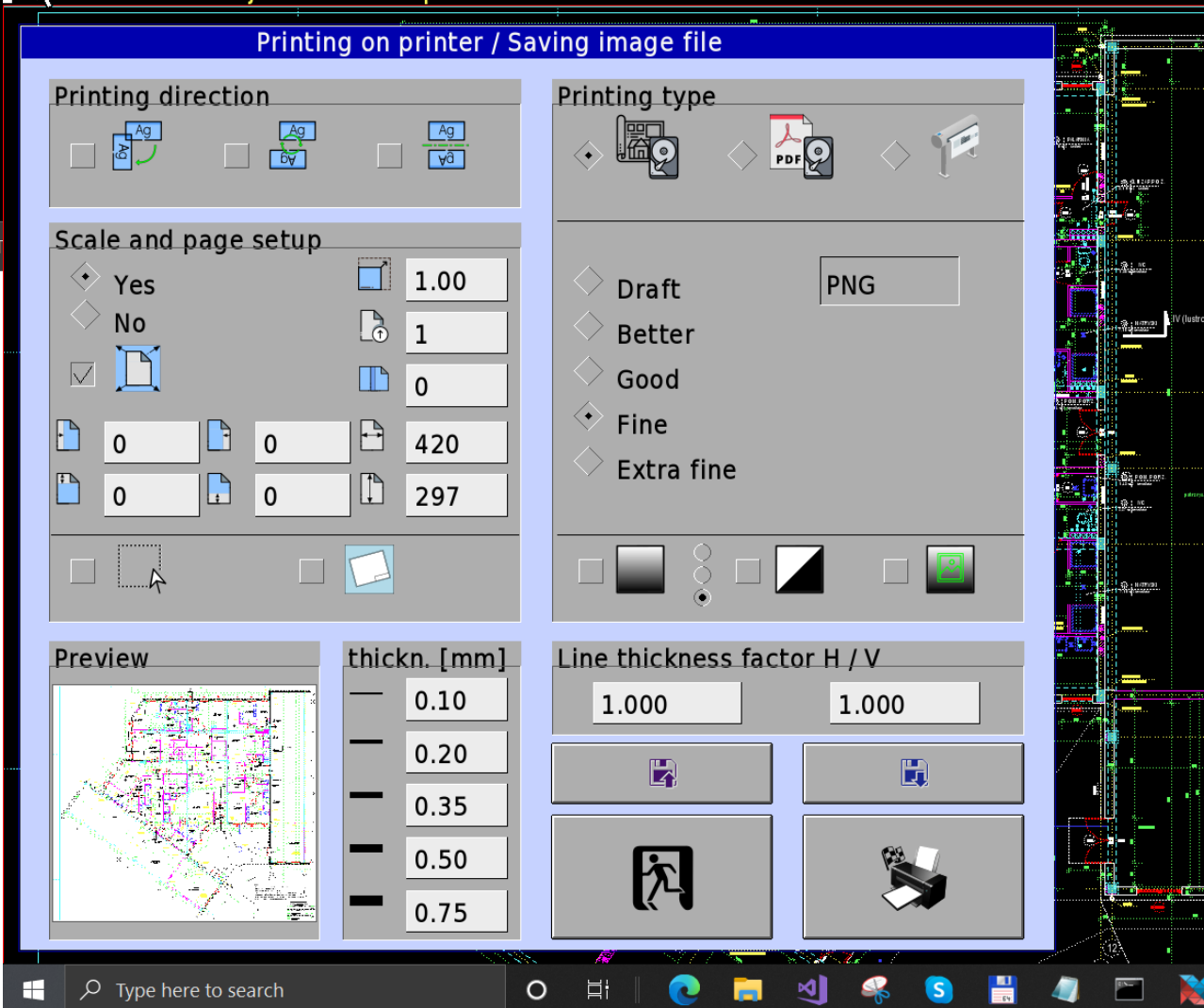
Layer 14 'Instalacje' +V+E+P ptr=min

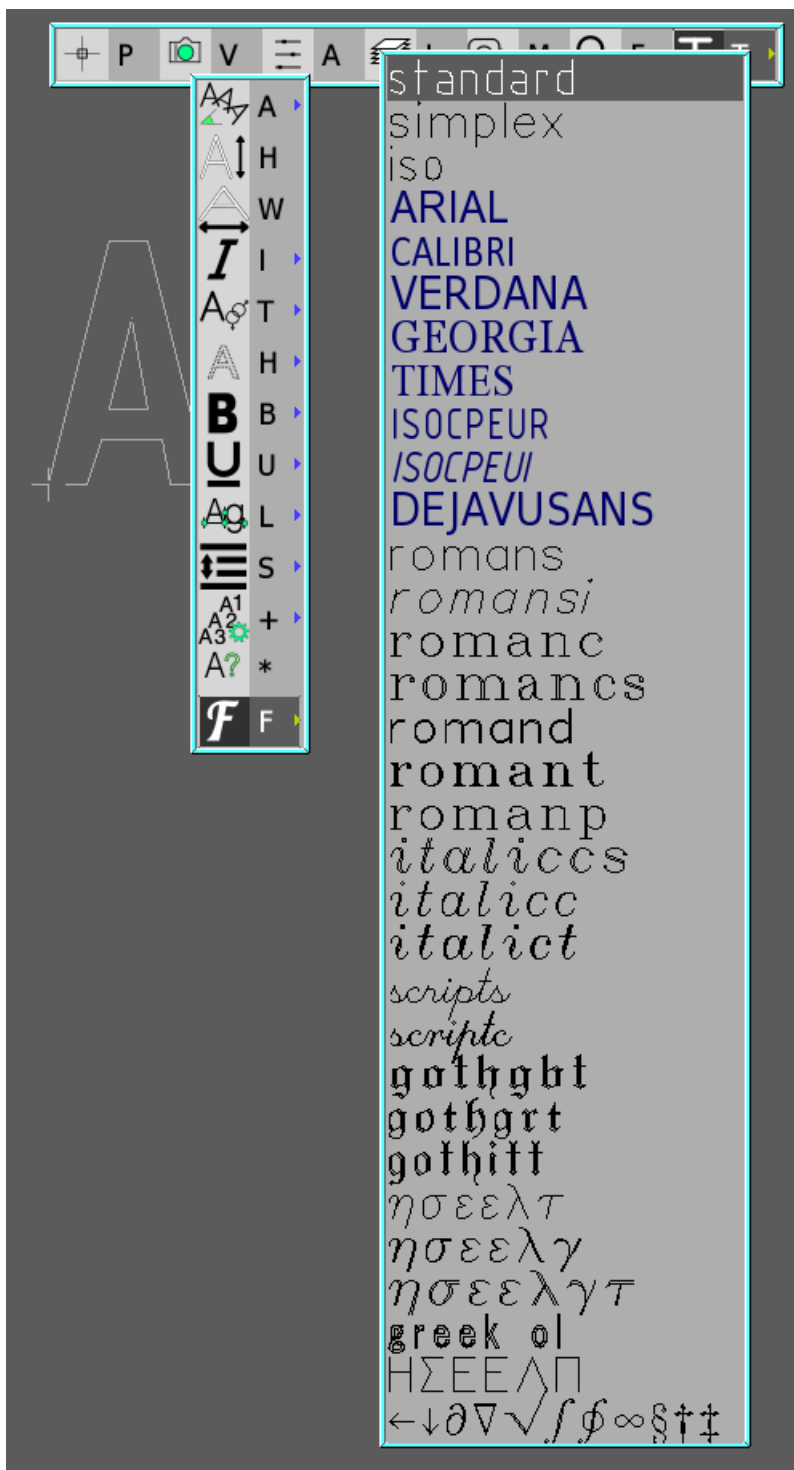


C:\Users\Marek\ALFACAD64\Drawings\Samples\example_4_1.alf

File C:\Users\Marek\ALFACAD64\Drawings\Samples\example_4_1.alf

Layer 14 'Instalacje' +V+E+P ptr=min



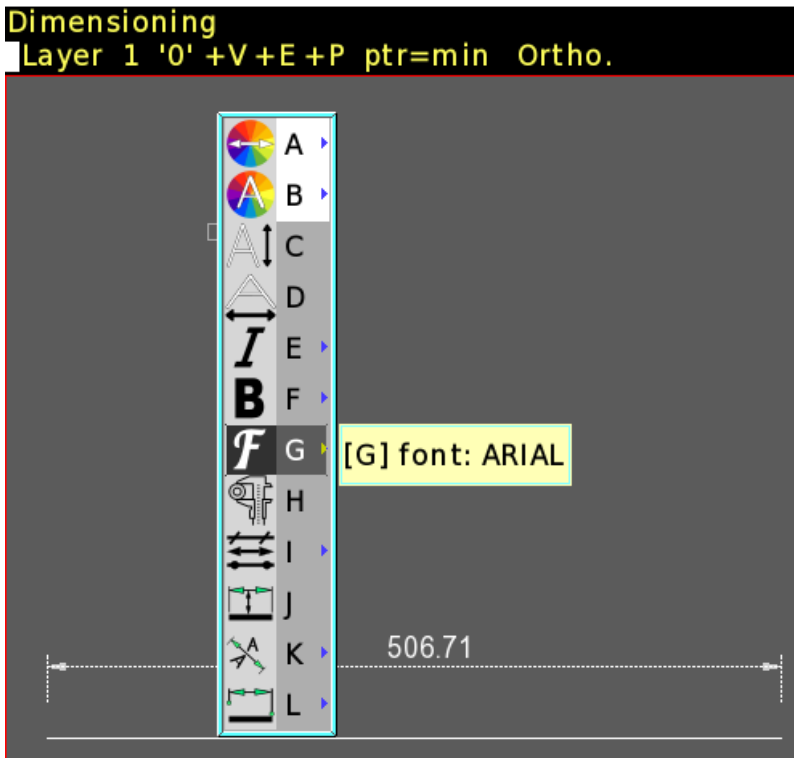


In function **Draw Text**, in auxiliary menu (just after text is typed) **Text** submenu appears where one of option is **Font**. In new version all (vector and TrueType) are shown as they look like, and as a sample the name of the font is drawn. TTF fonts are shown in dark blue colour to distinguish them from others. Scrolling to the end of menu, “add new font” appears.



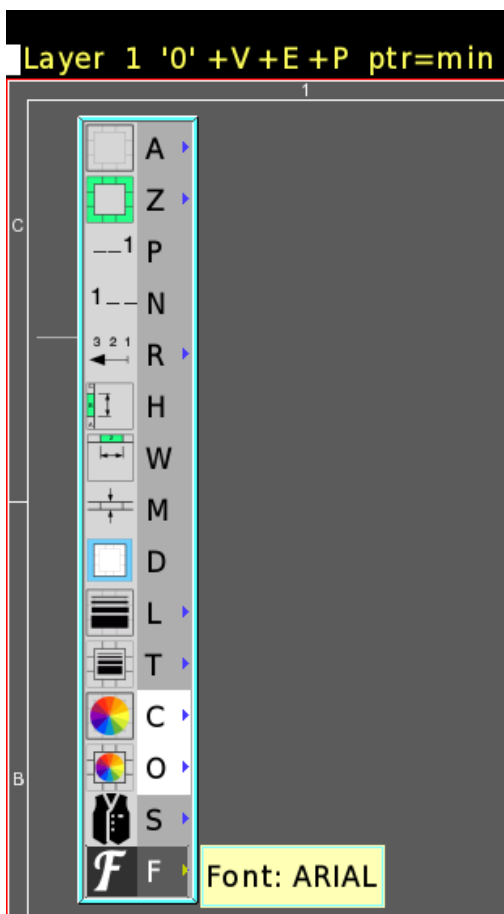
Font menu is also shown in two another menus, **Dimensioning - Parameters** (after initializing dimensioning), and in **Parameters - Zones** (mostly used on electrical diagrams).

All three font styles (for text drawing, for dimensioning and for zones numbering) are saved in each drawing individually, and restored after starting re-edition of drawing.



(Dimensioning - Parameters)

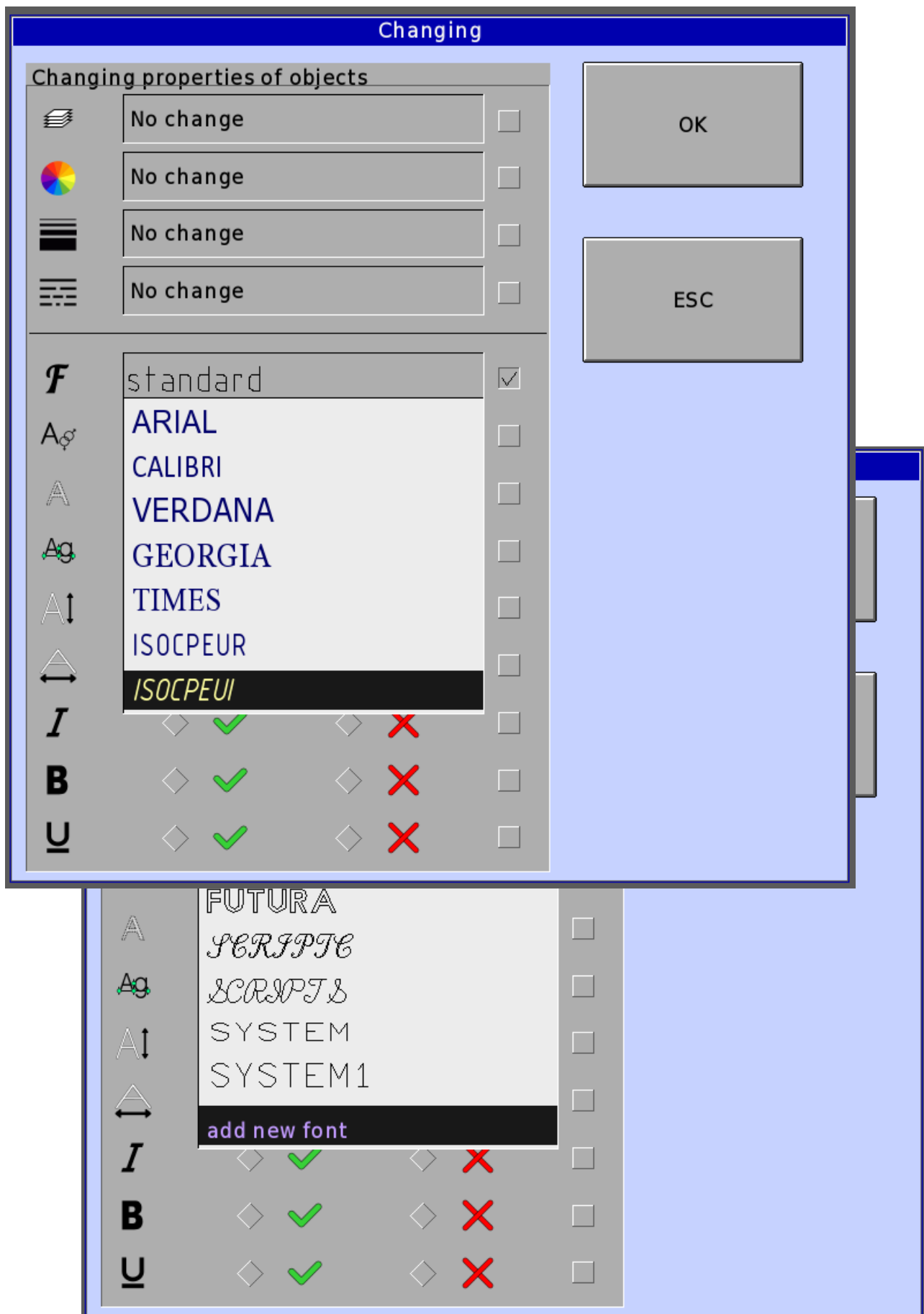
(Parameters - Zones)



All fonts initially loaded to AlfaCAD are declared in group [Fonts] in ALFACAD.INI configuration file. Here is a part of it:

```
[Fonts]
AFF=FONTS\AFF\STANDARD.AFF ,0.1 ,txt 1.0 ;
AFF=FONTS\AFF\SIMPLEX.AFF ,0.1 ,txt 1.0 ;
AFF=FONTS\AFF\ISO.AFF ,0.1 ,txt 1.0 ;
TTF=Arial.ttf , 0.075, arial.ttf 1.0 ;
TTF=Calibri.ttf, 0.075, Calibri.ttf 1.0 ;
TTF=Verdana.ttf, 0.075, Verdana.ttf 1.0 ;
TTF=Georgia.ttf, 0.075, Georgia.ttf 1.0 ;
TTF=Times.ttf, 0.075, Times.ttf 1.0 ;
TTF=isocpeur.ttf, 0.075, isocpeur.ttf 1.0 ;
TTF=isocpeui.ttf, 0.075, isocpeui.ttf 1.0 ;
TTF=DejaVuSans.ttf, 0.075, DejaVuSans.ttf 1.0 ;
AFF=FONTS\AFF\ROMANS.AFF ,0.1 ,txt 1.0 ;
AFF=FONTS\AFF\ROMANSI.AFF ,0.05 ,txt 1.0 ;
AFF=FONTS\AFF\ROMANC.AFF ,0.05 ,txt 1.0 ;
AFF=FONTS\AFF\ROMANCS.AFF ,0.05 ,txt 1.0 ;
AFF=FONTS\AFF\ROMAND.AFF ,0.05 ,txt 1.0 ;
AFF=FONTS\AFF\ROMANT.AFF ,0.05 ,txt 1.0 ;
AFF=FONTS\AFF\ROMANP.AFF ,0.05 ,txt 1.0 ;
etc. ....
```

When necessary, new TTF font can be added to the session, or by selecting “add new font” in **Draw Text** function, as shown above, or in **Edit – Change properties**, as shown below.



In both cases Fonts folder is opening (in default Windows folder) offering any font to choose. Once is chosen, can be used until the end of AlfaCAD session. Initial configuration is not changed in ALFACAD.INI configuration file however. To load new font always at start, its name should be inserted into configuration file.

Please notice that every font loaded to AlfaCAD needs memory resources, there is always limit how many fonts can be loaded. To exploit acceleration of texts procedures, fonts have to be loaded to the memory once, then used and they are never dismissed, until the end of program session.

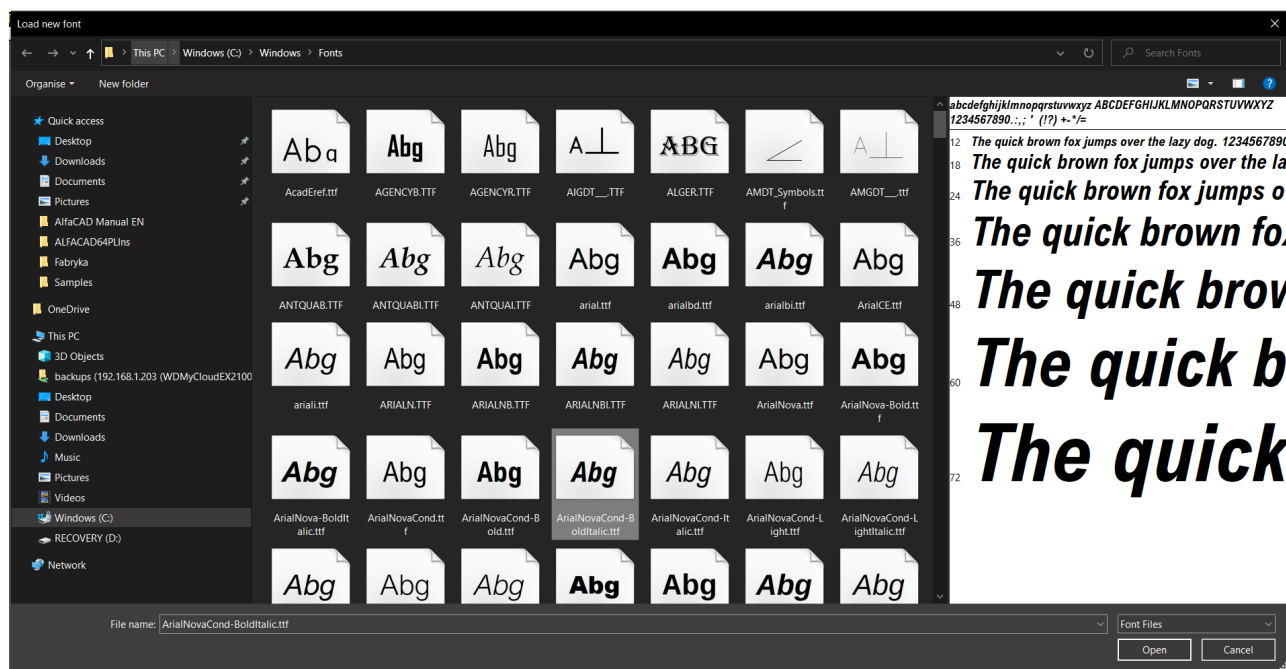
Even if the particular font style is not loaded at start, but it was used in the drawing currently loaded to AlfaCAD, this font will be loaded too, so drawing will never look different then it looked when was created. The only exception is when the font is no longer available for any reason. Using common Windows TrueType fonts doesn't bring such risk.

To make sure that the drawing can be printed using any printing service, in case of saving drawing in PDF document, all TrueType fonts are embedded (saved in the document file). All texts with vector fonts are sets of lines only, so no needs to embed anything.

By default Fonts folder (usually C:\Windows\Fonts) has attribute "System", for that reason folder is invisible in Windows Explorer, as well as would be invisible in AlfaCAD. For that reason, during installation or upgrading, AlfaCAD installation program is trying to change attributes removing "System" attribute, however leaving "Read only" attribute intact, or rather setting up this attribute instead. If after installation AlfaCAD wouldn't show that folder in "add new font" function, run command window (CMD) as administrator, and use command:

attrib -r -s c:\windows\fonts

(please note the location of Fonts folder can be different).



UTF8 encoding

In new version AlfaCAD introduced UTF-8 encoding of all texts and parameters.

When older drawings are loaded, they are automatically converted from propriety encoding system to UTF-8. Implemented UTF-8 is however limited to 2 bytes among multi-bytes characters, what covers all European languages alphabets, including Greek alphabet, and also Arabic alphabet (there are all together 1920 characters). Full implementation of multi-bytes UTF-8 (up to 4 bytes, [UTF-8 - Wikipedia](#)), covering all Unicode characters ([Supported Scripts \(unicode.org\)](#)) so the majority of scripts (in practice – languages), will be done soon, in next iteration of AlfaCAD version.

Changing

Changing properties of objects

No change

No change

No change

No change

F

ARIAL

A

No change

A

A

No change

A

No change

A

0.85

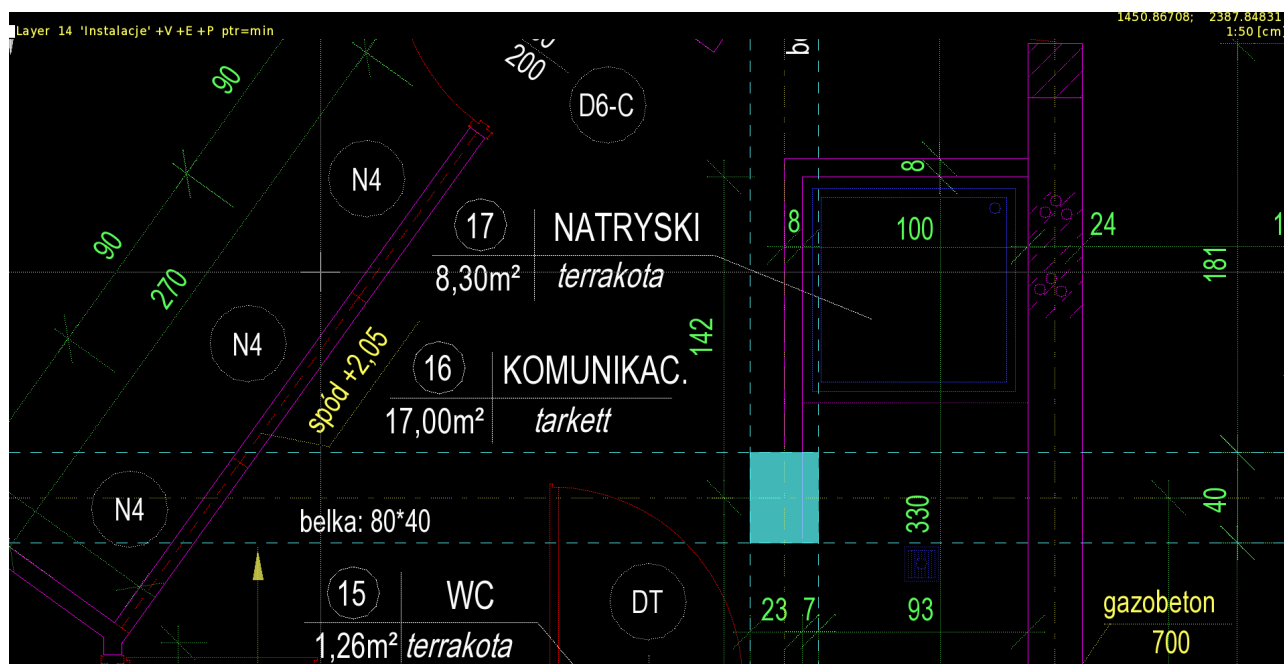
I

B

U

OK

ESC



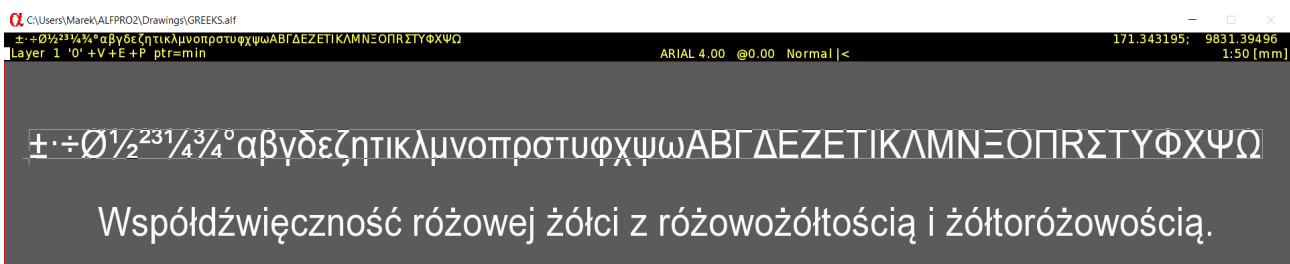
Keyboard layouts

In new version AlfaCAD relies on the keyboard setup inherited from Windows interface. If you use multiple setup (e.g. US, UK, Spanish, Polish or whichever else layout) select it during typing texts or multiline texts by pressing Windows Key – Space. To type diacritics, most of keyboard layouts use AltGr key combined with Latin letter, e.g. AltGr-a makes Polish diacritic “ą”. In some keyboard there are different method, in Spanish layout to get letter with accent first acute accent key ' (single appostrophe) has to be pressed, than letter I.i. E.,e,O,o, e.g. **Í í é ó**, but not simultaneously. Making technical drawings in all engineering specialties very often requires the use of special characters such as \pm ° ² ³ $\frac{1}{2}$ $\frac{1}{4}$ $\frac{3}{4}$ Ø as well as **Greek** letters. In AlfaCAD, independently of currently used keyboard layout, those extra letters and signs are easy to type using **not** AltGr rather **left Alt** key combined with number or Latin letter, and so **Alt-a** makes Greek **α**, **Alt-b** makes **β** etc. To avoid interfering with system hotkeys, is not even necessary to keep pressing Alt when Latin letter is pressed. It reminds Spanish keyboard for accents. Just press **Alt** (then you can release it), then press letter **a**, you will get Greek **α**. Extra **Shift** key changes letter to its capital, however Greek capital letters with identical look to Latin letters are omitted.

Special charcters:

Alt-1 $\frac{1}{2}$, Alt-2 ², Alt-3 ³, Alt-4 $\frac{1}{4}$, Alt-5 $\frac{3}{4}$, Alt-8 °, Alt-0 Ø, Alt-+ ±

Below the example of special characters, Greek letters, and example of fancy sentence using substantial number of diacritics.



Limitations

Fonts installed on your system are licensed by their manufacturers for specific uses. Font license might allow you to use the font to display on your monitor and print on your desktop printer, however, the license might restrict you from embedding the font when you save your work, for example PDF document. Restricted fonts cannot be loaded to AlfaCAD (examples: technic_.ttf, techb_.ttf, techl_.ttf, sanss_.ttf, sanssb_.ttf, sansso_.ttf, 0.1, sanssbo_.ttf). There are just a few of them in the system. Restricted fonts can be unlocked (please „google” for the solution), but this is not job to be done by AlfaCAD, such ublocking can potentially breach copyrights.

Rendering TrueType font glyphs on a screen or printer requires fast processing and relatively large resources. In some situations, it requires a trade-off between speed and usability.

All TrueType fonts have the same parameters as vector fonts: character height, width factor, italics, bold, text angle in relation to the coordinate axis, text adjustment, underlining, hiding, and finally the spacing between lines in multiline texts. There is one limitation when saving an output to a PDF document. The PDF document does not allow the use of bold text. This attribute will be disregarded, however, this attribute will be replaced with a proportional increase in the height and width of the font, which results in a similar effect to bold text.

There is another limitation. In AlfaCAD, texts with TrueType fonts are plotted on legacy vector plotters as letter outlines, omitting glyph fillings.

Code Pages

Unicode (also UTF-8 as a way of storing data) is an effort to include all characters from all currently and historically used human languages into single character enumeration (effectively one large single code page), removing the need to distinguish between different code pages when handling digitally stored text. Unicode tries to retain backwards compatibility with many legacy code pages, copying some code pages 1:1 in the design process. An explicit design goal of Unicode was to allow round-trip conversion between all common legacy code pages, although this goal has not always been achieved. Some vendors, especially Microsoft, have anachronistically assigned code page numbers to Unicode encodings. This convention allows code page numbers to be used as metadata to identify the correct decoding. It has some important consequences, that's why also AlfaCAD has to know what code page is used in the system.

In ALFACAD.INI configuration file there are two parameters in [Directory] section:

Code Page=1250 ; 1250=Latin2, 1252=Western, 28592=ISO-8859-2

DXF Encoding=1250 ; 1250=Latin2, 1252=Western, 28592=ISO-8859-2, 852=DOS Latin-2

Set it properly to get appropriate encoding and decoding of file names, texts imported from another programs (e.g. to multiline text editing box) as well as texts within imported and exported drawings from and to DXF format files.

Other changes and enhancements

Other big change, not related to TrueType fonts and UTF-8 encoding, is extension of **Block – Open catalog** function to DXF blocks (before only native AlfaCAD blocks were displayed).

For that purpose adopted some small, but still usefull catalog of DXF blocks found in LibreCAD (all resources of LibreCAD are declared as public domain, so also DXF catalog can be freely used for any commercial or non-commercial purpose).

Final thought

The author used Kirill Kryukov's Glyph Keeper package to render the TrueType fonts ([Glyph Keeper - A text rendering library \(kirill-kryukov.com\)](http://kirill-kryukov.com)). Kirill's work, a very intelligent and extremally useful implementation of the FreeType programming library, not only allowed the author to undertake the task of implementing TrueType fonts as part of the graphical interface, but above all as graphic objects of the developed drawings for which the program was created, with a whole range of parameters and requirements accompanying such a comprehensive implementation, including printing on printers and saving in graphic files of various formats, in very high resolutions.

Thank you Kirill for your contribution to AlfaCAD development on the way you didn't even know, and however creation of that nice piece of code got stuck some time ago, there is AlfaCAD's author who hopes this work will be continued. There are still a lot to do, for the benefits of many. Good code is like a gold. Will never get rust.

Enjoy AlfaCAD

author