

AlfaCAD ver. 2.4. A brief description of what's new (in reverse chronological order backward to v 2.0)

## 1. Plugins



AlfaCAD is equipped with an interface that allows you to add software plug-ins that expand the existing set of functions with new, unlimited in creativity additional functions. Plug-ins are divided into 4 groups and, depending on their character, they are located in the function group: Draw, Edit, Block or in the auxiliary menu, available from each function. For demonstration purposes, 4 plug-ins have been added in version 2.4: in the group: Draw - a function for drawing bent reinforcement bars (rebars) of reinforced concrete elements with their automatic dimensioning, in the group: Edit - a function for rounding the corners of polylines (including rectangles and regular polygons), and in the auxiliary menu group - a function of changing the angle of the local coordinate system and a simple function of redrawing the drawing.

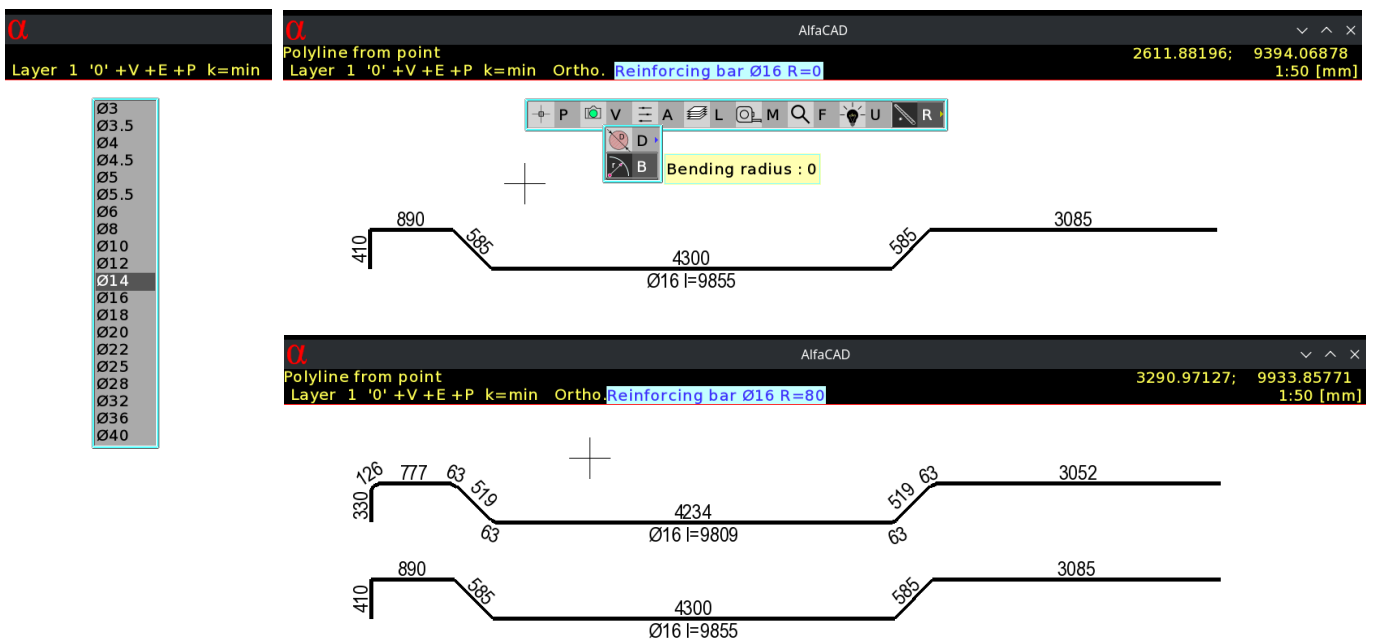
Like all AlfaCAD, software plug-ins are also designed in C/C++, with access to all drawing elements via an interface that provides a number of auxiliary functions, freeing the plug-in creator from the need to have detailed knowledge of the rules for managing the entire drawing data segment.

Plugins can initiate a parameter selection menu in text or icon form using icons attached by the user, place descriptions of commands and parameter editing fields, invoke any functions for pointing to objects or drawing them interactively and then perform any operations or additions on them, import blocks by initiating their interactive insertion etc. etc.

Knowledge of the form of drawing data, which can be obtained on the basis of the attached structure header file, allows the user - the plug-in author - for unlimited creativity, through interaction with existing and attached program functions.

The plug-in interface is constantly being developed and supplemented to provide the user with the best, simplest and most flexible programming tool possible. Demo plug-in source files, along with interface client source files and header files are provided with the entire AlfaCAD suite, for both Linux and Windows versions.

Example of “Rounded rebar” plugin: Bar diameter selection menu, dimensioned bar bypassing the bending radius and with a given radius



[sample plugin source code: cplug\\_alfa\\_rebar.c](#)

Plugin instructions, comments, and menus can be single or multilingual, as shown in the attached sample source code.

Light bulbs have captivated the minds of thinkers and writers, becoming powerful symbols that convey a range of ideas. Here are a few key symbolic meanings associated with light bulbs:

- Enlightenment: The invention of the light bulb by Thomas Edison symbolized the dawn of a new era, illuminating the world with knowledge and progress.
- Ideas and creativity: Light bulbs often represent the birth of a brilliant idea or a moment of creative inspiration.
- Curiosity and discovery: Like a light bulb's glow illuminating the darkness, it symbolizes the process of uncovering new knowledge and exploring uncharted territories.
- Innovation and progress: Light bulbs have come to represent technological advancements and human innovation, driving society forward.

This is a sufficient list to consider the light bulb as the most appropriate symbol for software plug-ins.

## 2. Ellipses, elliptical discs (filled ellipses) and elliptical arcs



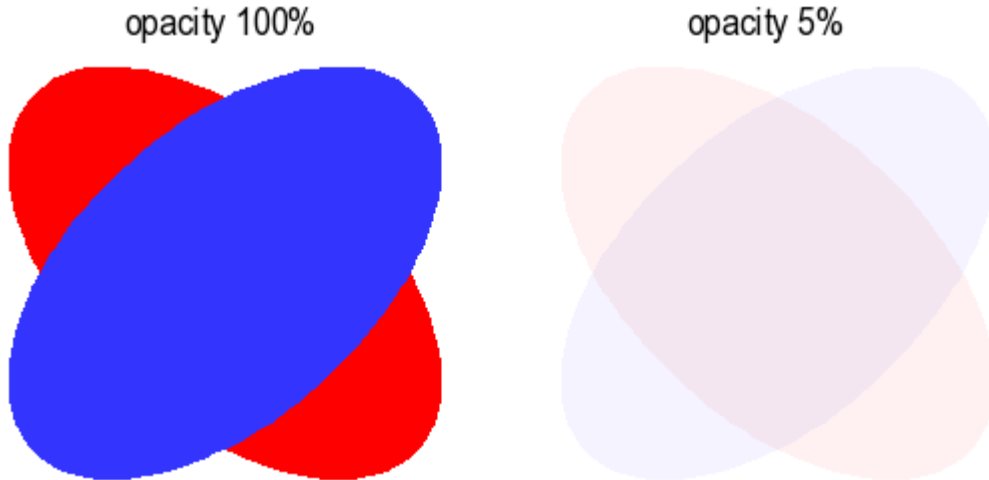
Finally, AlfaCAD has graphic primitives such as ellipses, drawn with a selected from 5 available line thicknesses, a selected from 32 available line type (pattern), and a selected from 255 available colors, as well as elliptical arcs with identical features. These primitives are objects drawn based on parameters such as the coordinates of the center of the ellipse or elliptical arc, the angle relative to the global coordinate system, and the length of the first semi-axis of the ellipse and the length of the second ellipse. Elliptical arcs additionally have set parameters of the initial angle determining the direction of the vector between the center of the ellipse and the starting point of the arc in the local coordinate system of the ellipse (where the X axis of the system coincides with the direction of the first semi-axis) and the final angle determining the direction of the vector between the center of the ellipse and the end point of the arc. When editing an elliptical arc, both the starting and ending angles can be declared as angles in the local coordinates of the ellipse, or in the global (or local) drawing coordinates. In each case, these angles are converted to angles in the local system of the ellipse, so any rotation of the ellipse in the Rotate function does not change these values.

The statement "finally" refers to the fact that previously AlfaCAD had ellipses as polylines, generated on the basis of identical parameters interactively entered by the user in the ellipse drawing function, and elliptical arcs could only be created as a result of cutting out a fragment of the ellipse, or were generated as open elliptical polylines when converting a file from the DXF format (now also DWG).

Ellipses and elliptical arcs are subject to all regimes available in the program for detecting intersections with other objects, including other ellipses and elliptical arcs, intersection points with lines perpendicular to the ellipse, points of line tangency, points closest to the indicated point, and finally, for elliptical arcs, detection of both ends of the arc, wherever a locating point (AKA "osnap point") is needed when drawing or editing.

Ellipses can be cut (resulting in the form of elliptical arcs), elliptical arcs can be cut (creating two elliptical arcs) or trimmed (where one of the arc angles changes, the initial or final one, and thus the position of the starting point, respectively initial or final).

Elliptical disks (or filled ellipses in other words) are ellipses that do not have an outline like ellipses, instead the interior of the ellipse is filled with one of the 255 available colors with a given opacity (understood as transparency inversion), selected from the 255 available ones, with this attribute to facilitate the selection. is determined on a scale from 100% to 5%, with an available increment of 5%. This allows for 20 different opacities for elliptical disks, similar to areas and color fills in the hatch function.



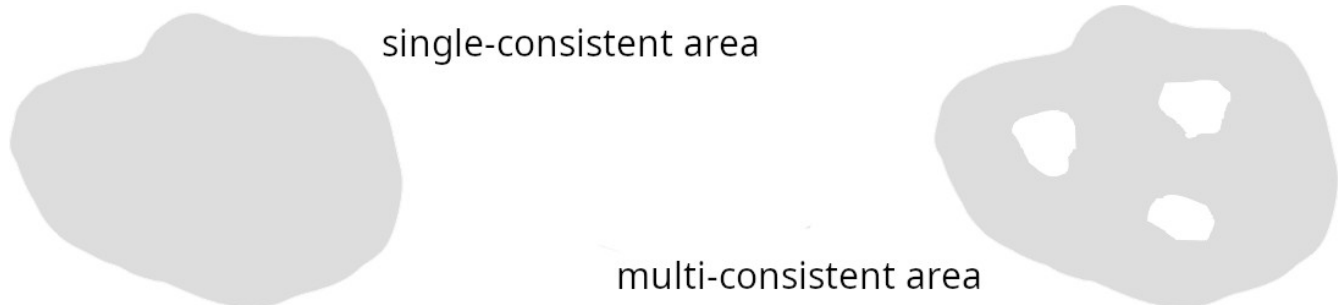
### 3. Hatching with linear thermal insulation pattern



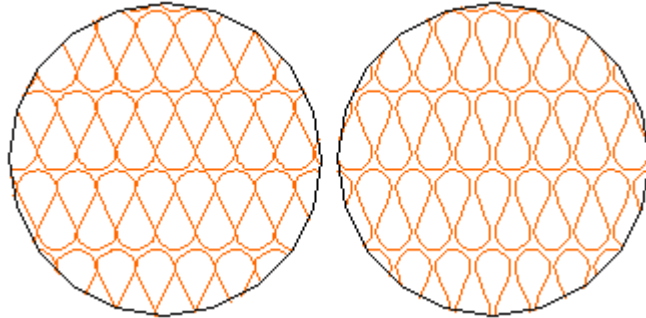
AlfaCAD has approximately 165 vector hatching patterns, which can be freely supplemented with additional patterns, as well as a catalog of over 700 bitmap patterns of photo-realistic quality, allowing for filling closed areas of any complexity, both single-consistent and multi-consistent.

From a mathematical point of view, the region  $O \subset C$  is called a single-consistent domain or region if each loop contained in it can be continuously deformed to a point, remaining in the region all the time (the loop in  $O$  is contractible to a point). The edge of such region then has one cohesion component. Let's call region an area for CAD purpose.

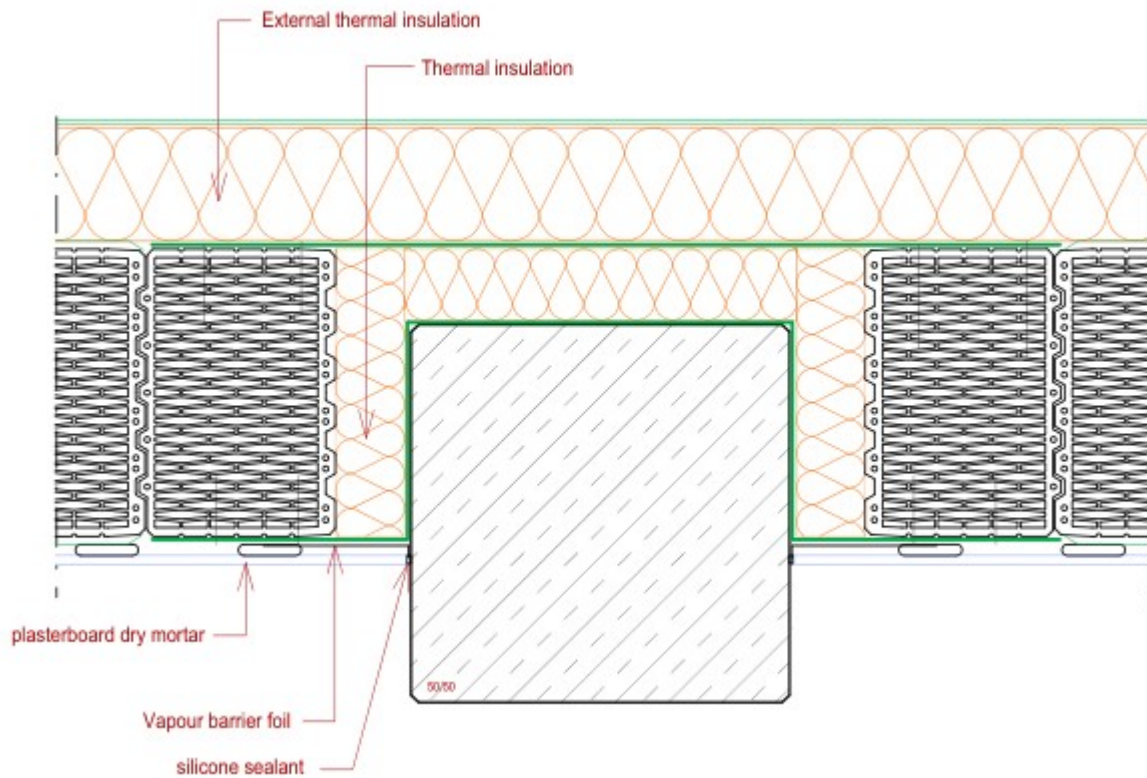
More generally, the edge of the area may have  $k$  components, where  $0 \leq k \leq \infty$ . If  $k > 1$ , then the area is called a multi-consistent area. The number  $k$  is called the order of consistency. If  $k = 2$ , the area is called a bi-consistent area, if  $k = 3$  - a three-consistent area, etc. If  $k < \infty$ , the area is called a finitely consistent area, and if  $k = \infty$  - an infinitely consistent area.



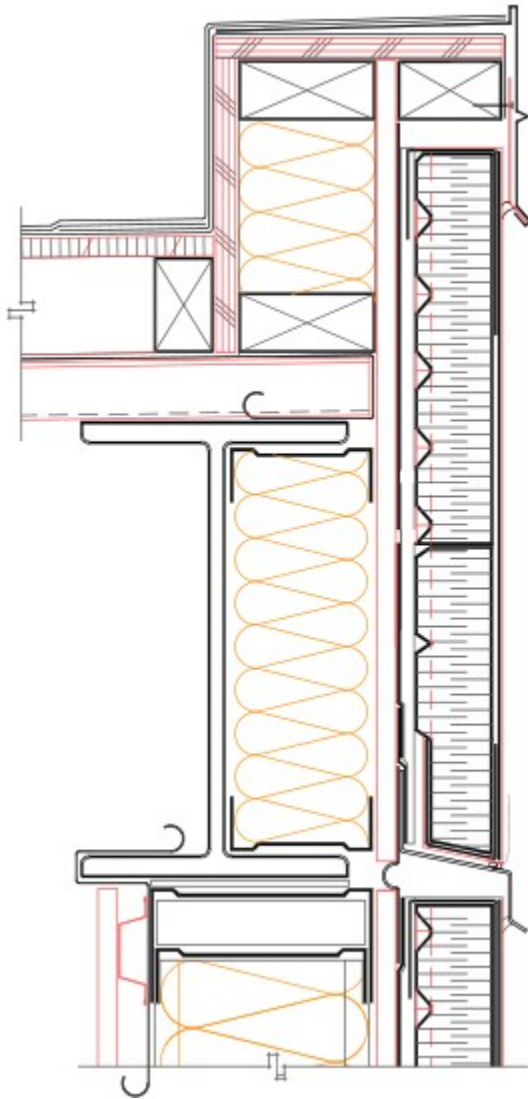
There are two hatch patterns in AlfaCAD called H1INSUN1 and BINSULATION. The names are not important, because AlfaCAD either offers patterns using their image, presented at a given scale and with a given rotation angle, selected line thickness and color, or, as in the case of the function of filling an area with a linear thermal/acoustic insulation pattern, the pattern is selected automatically. Both patterns are shown in the image below (in red).



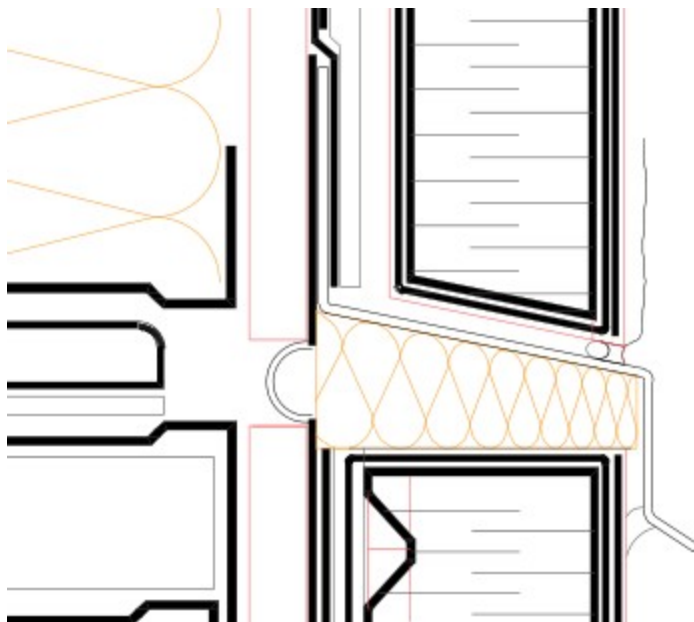
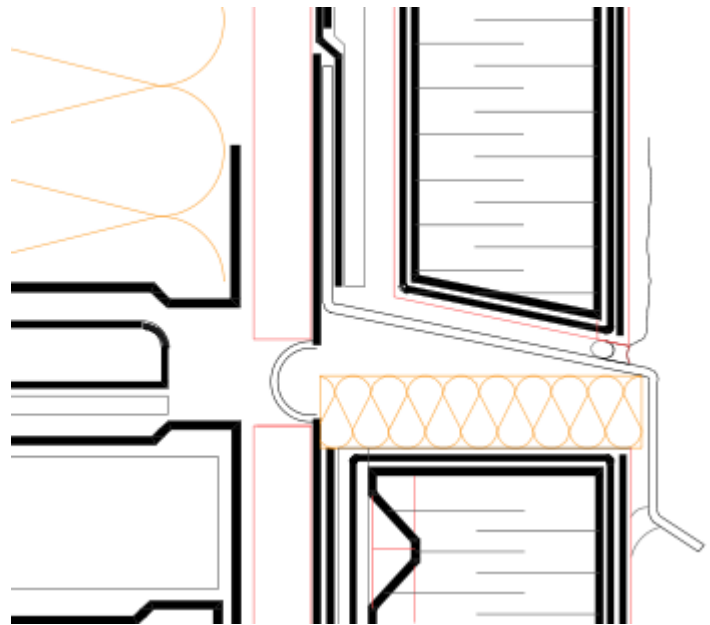
In the linear isolation pattern fill function, any specified rectangular area is filled in such a way that a single linear pattern segment fills the area along the long edge of the rectangle where a rotation angle of the pattern is equal to the angle of the edge vector, and a fill scale is corresponding to the ratio of the original height of the pattern to the length of the shorter edge. The result is fills as demonstrated in the example drawing below (red fill).



#### 4. Quadrilateral transformation



If you need to fill not a rectangular but a trapezoidal area with an insulation pattern, the four-sided transformation function comes to the rescue. As in the example below, the space under the flashing should be insulated. In the first step, a rectangular area is temporarily created, filling it as shown in the figure below, and in the second step, a quadrilateral transformation function is performed on this rectangle.



There are many uses for this function, and it is also helpful in creating isometric projections.

## 5. Opening files in associated application based on file extensions and MIME types

In computing, a file association associates a file with an application capable of opening that file. More commonly, a file association associates a class of files (usually determined by their filename extension, such as .alf) with a corresponding application (such as AlfaCAD).

The Microsoft Windows series of operating systems support file extension-based associations.

Linux-based GUIs, such as KDE and GNOME, support MIME type-based associations.

When installing the AlfaCAD program, \*.alf files are associated with the AlfaCAD program, and files with this extension are assigned the "α" icon attached to the package. Additionally, files with the extension .alx and .axx are assigned icons "αb" and "αs", respectively.

Double-clicking on a file with the .alf extension in Windows file explorer or, for example, Dolphin, the KDE's file manager in Linux, will cause the AlfaCAD program, if it has not been started yet, to be started and the file will be loaded and displayed in the application window, ready for editing. If AlfaCAD was previously launched, it will stop editing the currently edited drawing file, leaving it in memory, and as a multi-document application - it will load the drawing indicated in the file manager, placing it in the program window as ready for editing.

If AlfaCAD is currently executing a longer instruction (e.g. printing), it will load the indicated drawing immediately after the previous instruction ends. If AlfaCAD was in the middle of some interaction, such as drawing objects, editing, etc., it will successively abandon all these operations until the starting position (i.e. the state in which the program is waiting for the selection of a function from the main menu) and will load the drawing.

In the current version, AlfaCAD allows you to load up to 16 drawings at the same time. This is not justified by any technical limitations, and this limit comes from the assumption that a reasonable number of drawings should reside in the program's memory, which, from a practical point of view, does not drastically exceed the number of drawings that are alternately edited and that are in some way related to each other associated, so as not to unnecessarily deplete system resources. It should be mentioned that many instances of the program can be running at any time, independent copies that manage their own resources, and each copy of the program can store up to 16 drawings in memory. If more than one instance of the program is running, they share in the process of opening the files passed in the file explorer command. The instance that is first ready to load a file does so if the number of open files in that instance does not exceed the limit. A new instance of the program is launched when it is launched without providing a parameter such as the name of the drawing file to be opened, or when it is the only instance.

It makes sense to open more than one instance of the program when you want to view a given drawing while editing another drawing. Block Copy and Block Paste operations can be shared between instances, so it is easy to transfer entire drawing structures between instances (drawings).

The principle of the AlfaCAD interface is to minimize the amount of information appearing in the program window that is not important from the point of view of editing the current drawing, and all functions are easily accessible from the menu, including the auxiliary menu (opened using the Spacebar, Ins key, or middle mouse button). Changing the current drawing when other drawings reside in memory is made in the File - Desktop function, where thumbnails of the drawings are shown on the program's desktop, with the option of selecting each of them.

The program checks whether the edited drawings have been saved to disk, preventing the user from leaving the program without updating the changes.

## 6. Drag and drop

Drag and drop – one of the facilities offered by graphical user interface window managers. It allows the user to perform specific operations on two different objects with a single movement of the computer mouse. Some applications offer an "alternate" method of performing an operation, allowing you to "grab" an item with the right mouse button. Then, after releasing the key, a menu unfolds that allows you to specify the operation that the user wants to perform by moving it, while in the case of grabbing with the left mouse button, the default action is performed. An example is Windows Explorer, where if you move a file with the left mouse button, the default action is performed, i.e. moving (if within the same device) or copying (if between devices), while in the case of moving the right mouse button, releasing the mouse button will display a menu with the choice of operation to complete. You can often also perform actions other than the default by holding down one of the modifier keys (Ctrl, Alt, or Shift).

As a feature, drag-and-drop support is not found in all software, though it is sometimes a fast and easy-to-learn technique. However, it is not always clear to users that an item can be dragged and dropped, or what is the command performed by the drag and drop, which can decrease usability.

Drag and drop functionality is not a feature offered by all window managers in graphical user interfaces, for example the X Window System (X11 or simply X) common in Unix-like operating systems such as Linux does not provide this facility as a feature of the system, although allowing the use of the "Xdnd" protocol for an identical purpose. X provides the basic structure of a GUI environment: drawing and moving windows on the display device and interacting with the mouse and keyboard. X does not impose a user interface - individual programs take care of that. As such, the visual style of X-based environments varies greatly; different programs can present radically different interfaces.

AlfaCAD also works in this environment, which, starting from version 2.4, has implemented the Xdnd protocol for Linux.

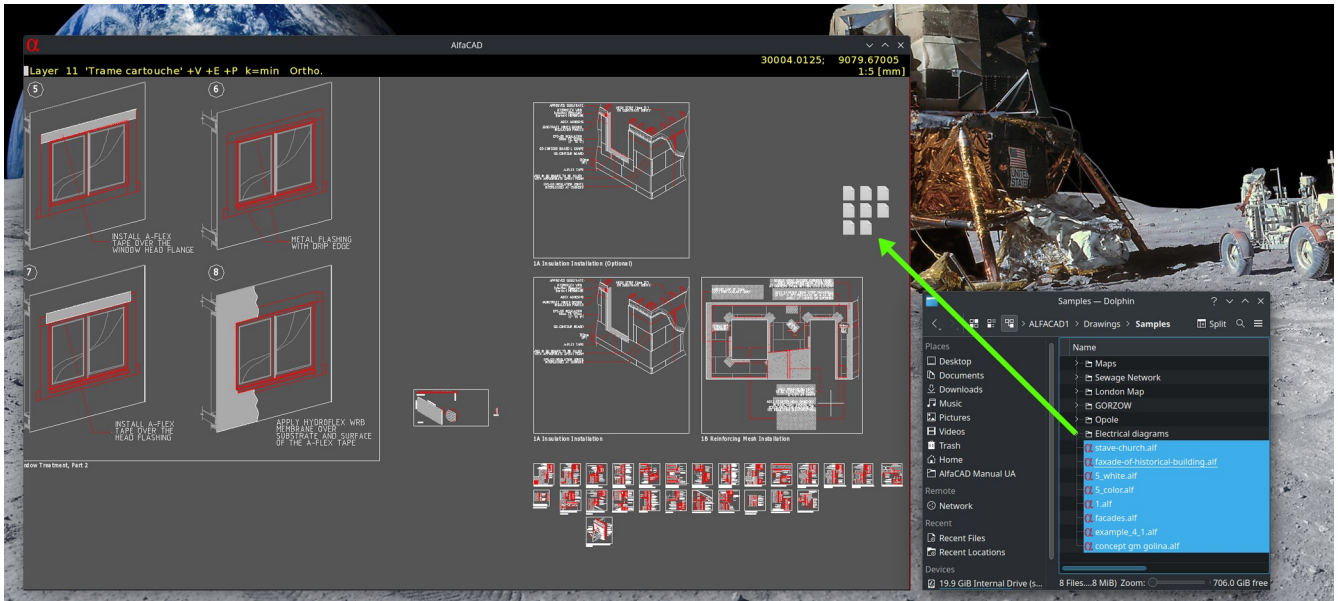
In Microsoft Windows, AlfaCAD uses drag-and-drop mechanisms built into the graphical user interface.

In the current version of AlfaCAD, the drag and drop function applies to drawing files. The internal protocol for handling the function is as follows:

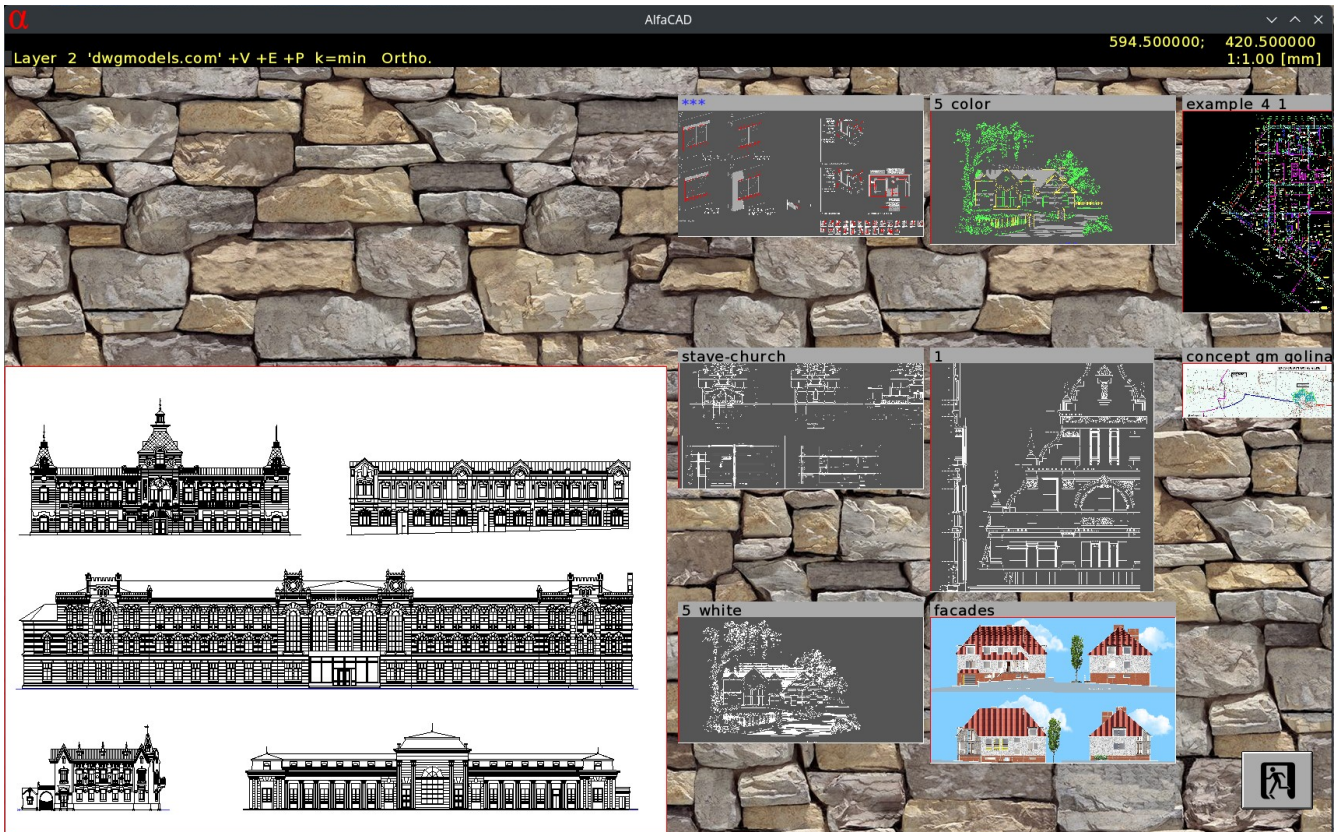
- AlfaCAD recognizes that its window (drawing desktop) can be a drop target, exchanging necessary information with the source window when the drag and drop process is initiated. The drop target can be any AlfaCAD instance if more than one is running.
- When the cursor is in the target window and the dragged object is dropped, the target AlfaCAD window accepts the object by assessing whether it is a list of file paths (in the case of selecting multiple names simultaneously by the source window) or a single path
- AlfaCAD tries to open a new drawing each time by loading the file based on the path taken from the list, leaving previously loaded drawings in memory.
- In each case, whether a single file or a list of files, this process is carried out until the limit of the maximum number of drawings that can be stored in the memory of each instance of the program is used. In version 2.4 it is the number 16.
- After loading a drawing, AlfaCAD displays it in its desktop window and then repeats the process for the next file on the list until it is completed or ignored when the limit on the number of drawings is reached.

In subsequent versions of the program, the "drag and drop" function will be extended to include functions for inserting (importing) blocks, graphic images (photos and bitmaps) and text blocks.





(in the File - Desktop function)





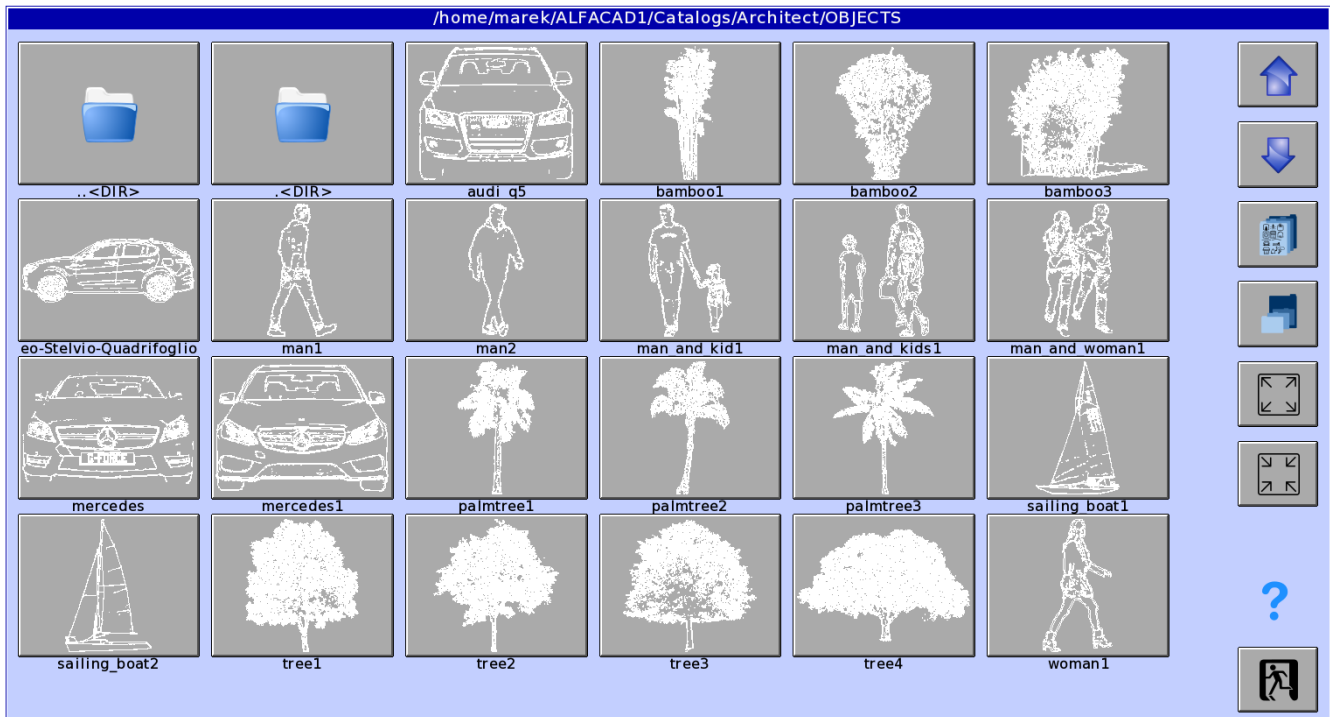
## 7. Vectorization of raster images and photos



In the vectorization function, any graphic file (jpg, png, bmp, pcx, etc.) can be converted into a monochromatic vector form and as a regular block inserted into a drawing at any scale (also in tainted scale where the scale factor in the X axis is different from the factor in Y axis) and at any given angle. Such a block can be subjected to any graphic operations, such as clipping, cutting, etc.).



Created blocks can be added to the catalog folder and available from the Block - Catalog function.



## 8. OpenType fonts



In addition to TTF (True Type Font) fonts, OpenFont fonts are also available. The ALFACADxx.INI configuration file contains paths to font folders, and for Linux the default paths are:

**TTF Fonts=Fonts/TTF ;TTF fonts folder, Windows fonts**

**OTF Fonts=/usr/share/fonts ;OTF fonts folder**

## 9. Custom hatch patterns in DXF files

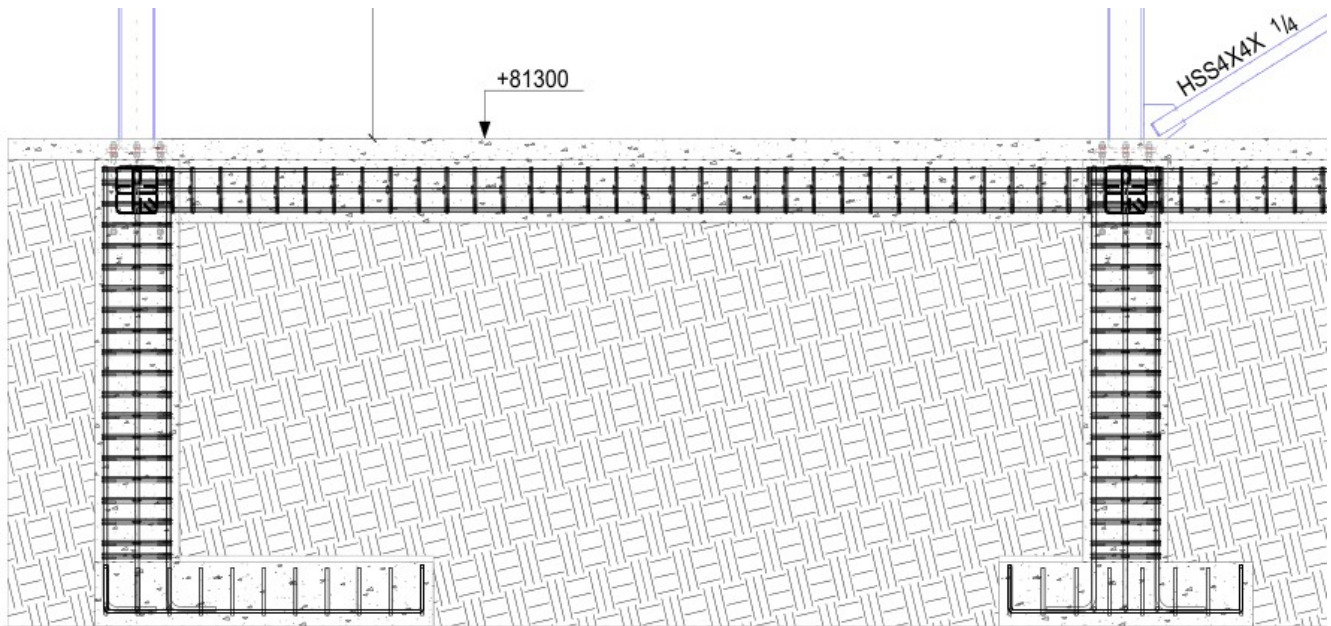
AlfaCAD in the current version reads HATCH objects when converting drawing files from DWG and DXF formats, inserting them into the drawing in a way analogous to native hatch objects, both with vector patterns and SOLID fill, with defined color and opacity. Due to the specificity of this type of objects, AlfaCAD saves fills in the DXF format as vector blocks, recognized natively (during reverse conversion) as HATCH objects, thus allowing for successive editing (e.g. changing the hatch pattern).

Below is an example of converting a fragment of a drawing containing areas filled with patterns defined in a DWG/DXF file.

The file was read using the File - Import - DWG function



(the LibreDWG module is used internally under the GNU General Public License published by the Free Software Foundation, Inc.).



A feature of DWG/DXF files is that the definitions of hatch patterns used in the drawing are embedded in the source format (DWG) file and the drawing exchange format (DXF) file, which allows AlfaCAD to accurately reproduce the drawing despite the differences in the nature of these objects (AlfaCAD generates hatch image as a block of lines, while defining hatch boundaries, used during re-editing, while other programs can display the fill block dynamically, generating the fill image "on the fly"). Work on improving the procedure of filling an area with a pattern in AlfaCAD is still ongoing, with the consistent goal of relieving the drawing regeneration/redrawing function from the need to generate an image of the filled area each time.

## 10. Conversion of DWG and DXF format files

In the current version of AlfaCAD, ellipses, filled ellipses and elliptical arcs are converted to a new form, as primitives, unlike the previous version in which these objects took the form of polylines. This reduces the volume of the drawing without compromising the regeneration/redraw speed. Below is an example of the facades of a classical building, with subsequent close-ups showing the complexity of the drawing, which contains over 360,000 blocks, composed mainly of Bézier curves, circular and elliptical arcs, and a countless number of lines and polylines. The fact that AlfaCAD treats each block individually results in an increase in the volume of the drawing, because each block is cloned with the attributes of the insertion point, scale and rotation angle, and as a clone it can be individually subjected to any editing operations, which in the case of this particular drawing does not have much justification. AlfaCAD, however, treats each block as a group of primitives, although it distinguishes an anonymous group of primitives from a block that has its own name and as such is available from the list of blocks in the Block - List function.

Block cloning, however, does not affect speed or memory consumption, so it will not exhaust system resources. Here is a summary of memory usage before and after loading this drawing (Linux Arch, AlfaCAD and another program), reported by the "top" command, a Linux tool that provides a dynamic view of system performance in real time, where:

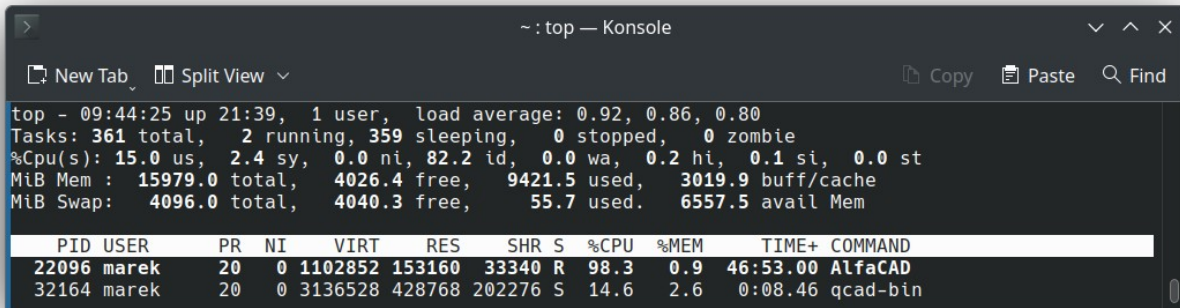


VIRT: total virtual memory used by the task

RES: how much physical RAM the process is using, measured in kilobytes

SHR: represents the Shared Memory size (kilobytes) used by a task

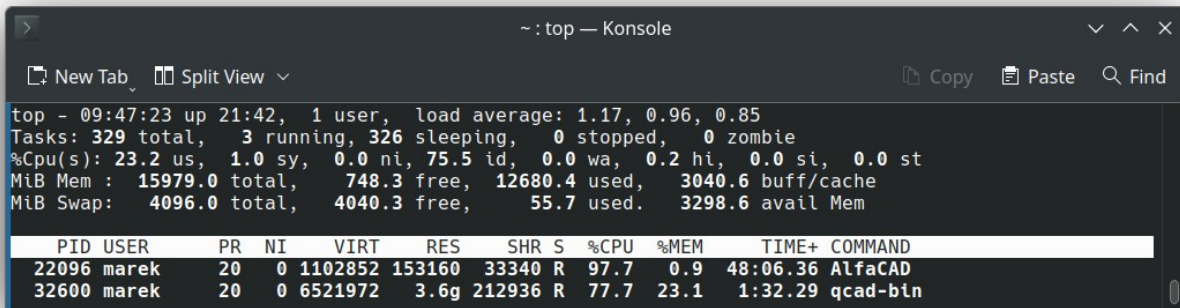
Before AlfaCAD loads the drawing:



```
top - 09:44:25 up 21:39, 1 user, load average: 0.92, 0.86, 0.80
Tasks: 361 total, 2 running, 359 sleeping, 0 stopped, 0 zombie
%Cpu(s): 15.0 us, 2.4 sy, 0.0 ni, 82.2 id, 0.0 wa, 0.2 hi, 0.1 si, 0.0 st
MiB Mem : 15979.0 total, 4026.4 free, 9421.5 used, 3019.9 buff/cache
MiB Swap: 4096.0 total, 4040.3 free, 55.7 used. 6557.5 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 22096 marek    20   0 1102852 153160 33340 R  98.3   0.9   46:53.00 AlfaCAD
 32164 marek    20   0 3136528 428768 202276 S  14.6   2.6    0:08.46 qcad-bin
```

and after loading:



```
top - 09:47:23 up 21:42, 1 user, load average: 1.17, 0.96, 0.85
Tasks: 329 total, 3 running, 326 sleeping, 0 stopped, 0 zombie
%Cpu(s): 23.2 us, 1.0 sy, 0.0 ni, 75.5 id, 0.0 wa, 0.2 hi, 0.0 si, 0.0 st
MiB Mem : 15979.0 total, 748.3 free, 12680.4 used, 3040.6 buff/cache
MiB Swap: 4096.0 total, 4040.3 free, 55.7 used. 3298.6 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 22096 marek    20   0 1102852 153160 33340 R  97.7   0.9   48:06.36 AlfaCAD
 32600 marek    20   0 6521972  3.6g 212936 R  77.7  23.1    1:32.29 qcad-bin
```

AlfaCAD will not allocate any additional memory, although the drawing size is approximately 70 MB, the entire drawing fits into the originally allocated 640 MB buffer (which can be dynamically configured by the user, and which is also used for another operations, like hatching). The image of the drawing in memory has the same form as the image in the disk file. To load a drawing, another program (as an example of a similar program) allocated several GB of RAM, and loading a DWG file took many times longer than loading a native .ALF file through AlfaCAD.

As a matter of facts is, however, that AlfaCAD does not use a virtual screen in the case of drawings containing only vector primitives, which is only used in the presence of bitmap objects. This explains why the appetite for memory resources is so small, although the result is the need to redraw the drawing with each zoom or panning operation. The speed of these operations compensates for this planned "imperfection".



AlfaCAD



Layer 2 'dwgmodels.com' +V +E +P k=min Ortho.

7934427.96; 317896.133  
1:50 [mm]

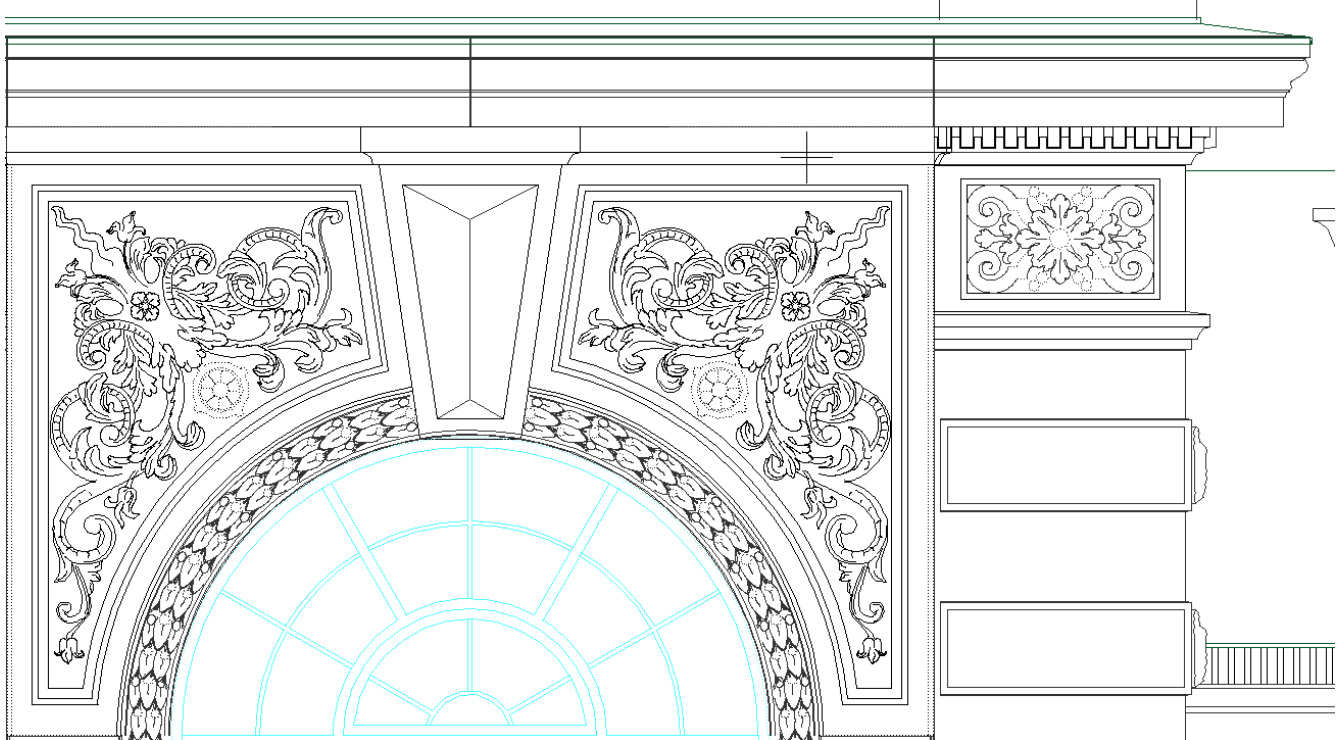


AlfaCAD

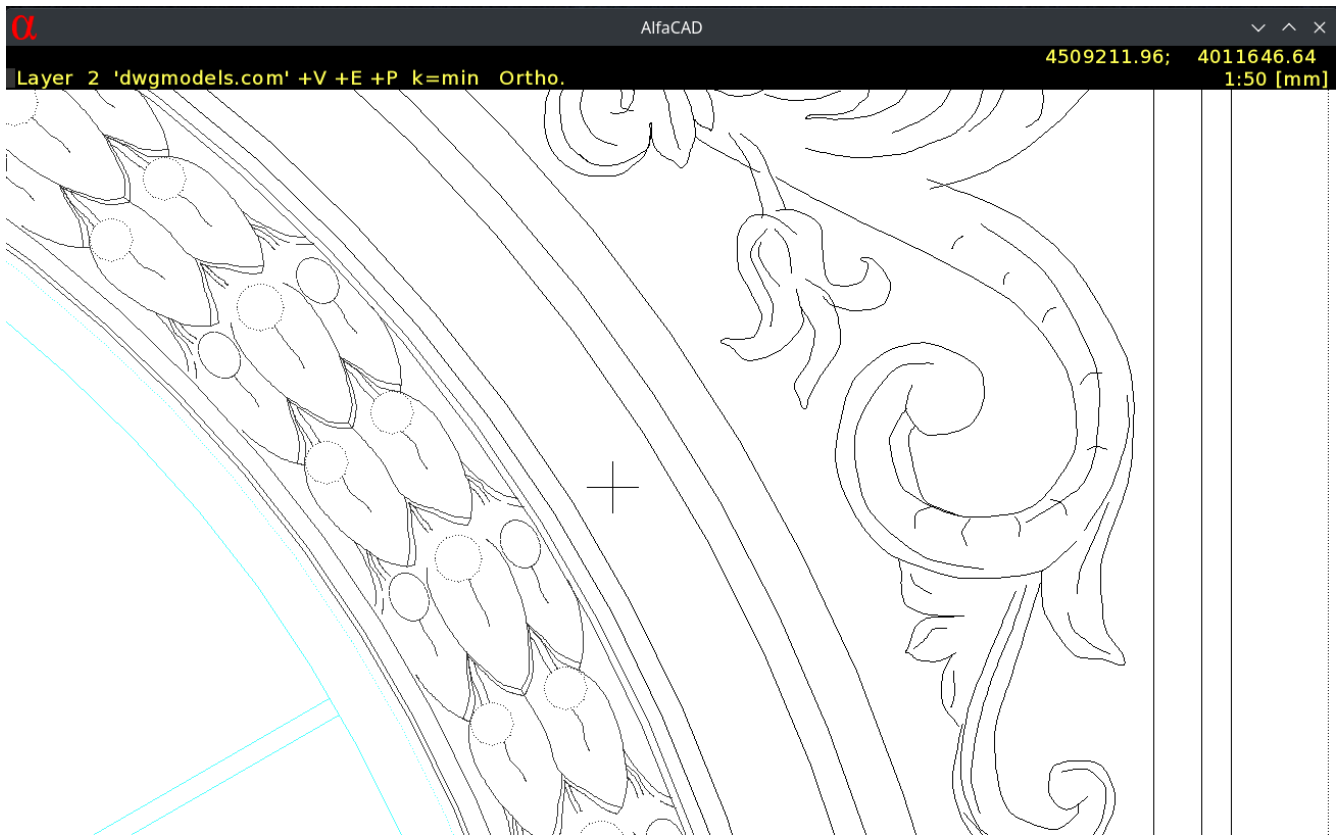


Layer 2 'dwgmodels.com' +V +E +P k=min Ortho.

4516052.83; 4106732.31  
1:50 [mm]







**Some archive files:**

[AlfaCAD v. 2.0](#)

[AlfaCAD leaders](#)

[AlfaCAD v. 1.9](#)

[AlfaCAD v. 1.8](#)

[AlfaCAD v. 1.6](#)

[AlfaCAD v. 1.4 Ergonomics](#)

[AlfaCAD v. 1.1](#)

[AlfaCAD TrueType](#)

[AlfaCAD 05/02/2021](#)

[AlfaCAD Introduction](#)

Enjoy AlfaCAD  
author