

AVAX-CAD R3

https://gardos-software.com/

© 1993-2026 Gardos Software LTD

Table of Contents

Foreword 0

Part I AVAX-CAD R3 5

1 INTRODUCTION 5

2 Program installation 5

3 Program operations 8

File 8
Edit 11
View 18
Drawing 21
Select 26
Object snap 27
Modify 28
Tools 35
Add-Ons 45
Window 45
Help 45

4 COMMANDS 45

ALL 45
ANG 46
ARC 48
AREA 49
ARRAY 50
AUTOLIMITS 50
BEZIER 51
BLOCK 52
BMP_OUT 53
BORDER 54
BREAK 55
CALCEXTENDS 56
CANCEL 57
CBSPLINE 57
CENTROID 58
CHANGE_COLOR 59
CHANGE_FONT 60
CHANGE_LAYER 61
CHANGE_LTYPE 62
CHANGE_PTYPE 63
CHANGE_TEXT 64
CHANGE_WIDTH 65
CLOSE 66
CIRCLE 66
CIRCLE2P 67
CIRCLE3P 69
COPY 70
COPY_IN 71
COPY_OUT 72
CSPLINE1 73

© 1993-2026 Gardos Software LTD

CSPLINE2 74
DIM 75
DIVIDE 77
DOWN - UP - LEFT - RIGHT 77
DXF_IN 78
DXF_OUT 79
ELLIPSE 80
ENDPOINT 81
ERASE 82
EXIT 83
EXPLODE 84
EXPLODE_BLOCK 85
EXTEND 86
EXTENDS 87
FILLET 87
FREEZE 88
GRID 89
GRID_ON 90
GRID_OFF 90
HATCH 91
HELP 93
INFO 93
INSERT 94
INSERT_AVX 95
INTERSEC 95
INWINSEL 96
LASTSV 97
LINE 98
LIST 100
MAGNIFY 100
MAKESV 101
MEASURE 102
MIDPOINT 103
MIRROR 104
MOVE 105
NODE 106
NEAREST 107
OBSPLINE 108
OFFSET 109
OPEN 110
ORTHO - ORTHO_ON - ORTHO_OFF 111
PEDIT 112
PGDN - PGUP - PGLEFT - PGRIGHT 113
PIC_OUT 113
PLT_OUT 114
POINT 115
POLYGON 116
POLYLINE 117
PREVIOUS 118
PRINT 119
QUADRANT 121
RADIATE 122
RECTANGLE 123
REDO 124

© 1993-2026 Gardos Software LTD

REDRAW 125
ROTATE 125
SAVE 126
SAVE_AS 128
SAVE_OPEN 129
SCALE 130
SELECT 131
SELECTSUB 132
SELECTADD 133
SELECT_ALL 134
SELECTV 135
SET 136
SETGRID 138
SETSNAP 139
SETUP 140
SHELL 144
SNAP 144
STRETCH 145
TANGENT 146
TEXT 147
THAW 148
TRIM 149
UNDO 150
UNSELECT 151
USERLIMITS 153
VERTICAL 154
WALL 155
WINSEL 156
ZOOM 157
ZOOMUNDO_ON - ZOOMUNDO_OFF 158
CHANGE_3DELEVATION - CHANGE_3DHEIGHT 158

5 EXAMPLES 159

6 APPENDIX 169

Index 0

© 1993-2026 Gardos Software LTD

1 AVAX-CAD R3

1.1 INTRODUCTION

AVAX-CAD R3 is a stand alone 2-dimension ($2^{1/2}$D) drawing program, which addresses Architects, Civil Engineers, Electrician Engineers, Mechanical Engineers, Topographers, Designers, Students and generally everyone who is occupied with geometrical drawing. It constitutes an alternative solution to all these expensive and inconvenient market's CAD programs. It is a program simple to use and at the same time it has all drawing possibilities. Briefly, some of the capabilities AVAX-CAD R3 has, are the following:

- Drawing all the basic geometric objects (Line, Point, Polyline, Arc, Circle, Ellipse, Polygon, Bezier curve e.t.c.).
- Text with fonts of Windows.
- Filling of polygons with hatch lines, icons or other kinds of design patterns. Capability of definition layer's appearance of polygons that have fillings such as to regulate intercovering.
- Creation and insertion of complicated objects. (Blocks).
- Color pallets, Layers, drawing width, font lists, line and points styles.
- Zoom window, extends, pan, all, previous view, zoom in e.t.c.
- Object selection with frame, within frame, all object selection, selected object removal e.t.c.
- Grid, snap to grid.
- Copy, copy at a specified distance, array, move, scale, rotation, mirror and stretch objects.
- Object snap during drawing : to end, middle, intersection, perpendicular, tangent e.t.c.
- Trim, Fillet, Extend e.t.c.
- Dimensioning of straight line segments and angles.
- Imports image files (BMP, JPEG, ICO, WMF, PCX, SHP, RTF)
- Imports drawing files AutoCAD® DXF
- Exports drawings in several types of files DXF, SVG, SWF, PLT, PIC, BMP, WMF, PDF, XML, RTF e.t.c.
- Measurement of lengths and areas.
- Printing preview and printing under scale.
- Refutation / Reverse (Undo / Redo) every action.

The following pages provide a brief description of installation and operation of AVAX-CAD R3. Detailed description of the use of all drawing commands and examples, are available at program's "help", where there is the ability to print them.
You can find our new versions of program AVAX-CAD R3 at internet page: https://gardos-software.com/ with upgrade availability. Also, at the same page you can find a list with add-Ons (extra by-programs) which can enrich the drawing program with new operations.

1.2 Program installation

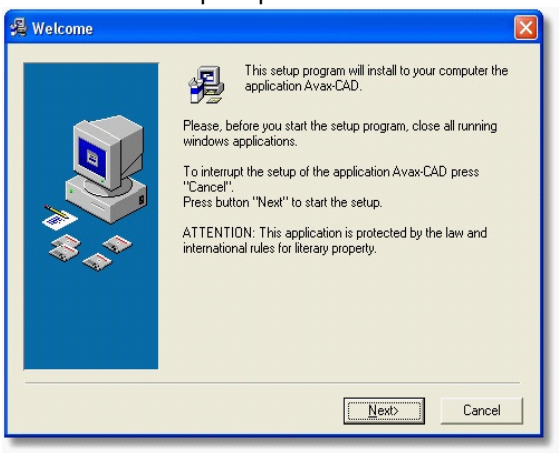
E.1. System requirements
For AVAX-CAD R3 proper operation it is required to install to your computer one of the following operation systems:
- Windows 98SE/Me
- Windows NT/2000/XP/Vista

E.2. Installation
Before you start the installation, close all running programs and note your serial number which is on your CD.
Put your installation CD into your computer's CD-ROM.
1. If installation doesn't start automatically, choose (start) ΰ (run) and type on your keyboard D: \Setup.exe where D the letter of your driver.
2. At the first prompt choose installation language (For Greek choose Greek).

© 1993-2026 Gardos Software LTD

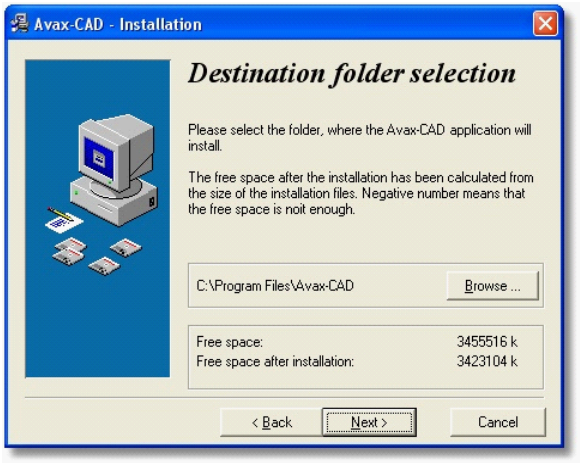


3. At the second prompt choose Next.

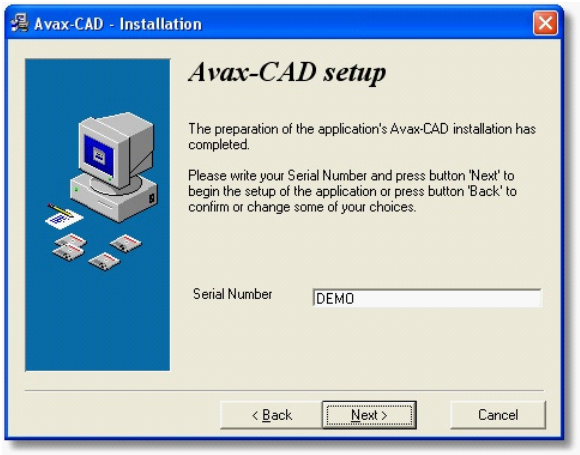


© 1993-2026 Gardos Software LTD

4. At the next prompt if you want choose Browse to change the folder where the program will be installed (the default position is C:\Program Files\AVAX-CAD R3). When you set the installation position you choose Next to continue the installation.

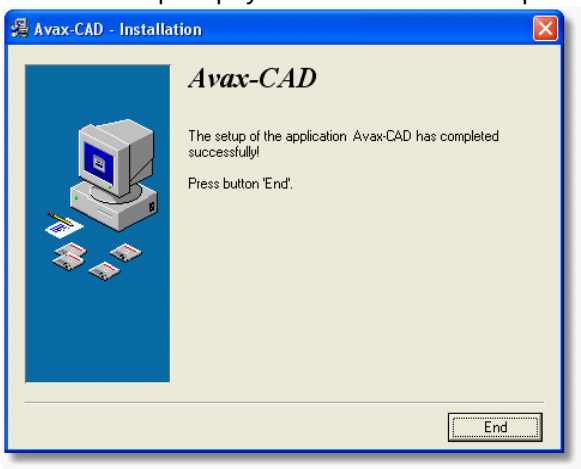


5. At the next prompt you enter your serial number which you were supplied with, when you bought the program, at the position Serial Number. After that you choose Next for installation to begin.



© 1993-2026 Gardos Software LTD

6. At the last prompt you choose End to complete installation.



E.3. Program starting

After the end of installation at (start) ů (all Programs) an AVAX-CAD R3 folder has been created with a shortcut by the same name. Also, a shortcut has been created at desktop.



1.3 Program operations

1.3.1 File

1.1. New drawing (NEW)

Command:> NEW
Shortcut: Ctrl+N



By this command we create a new drawing file.

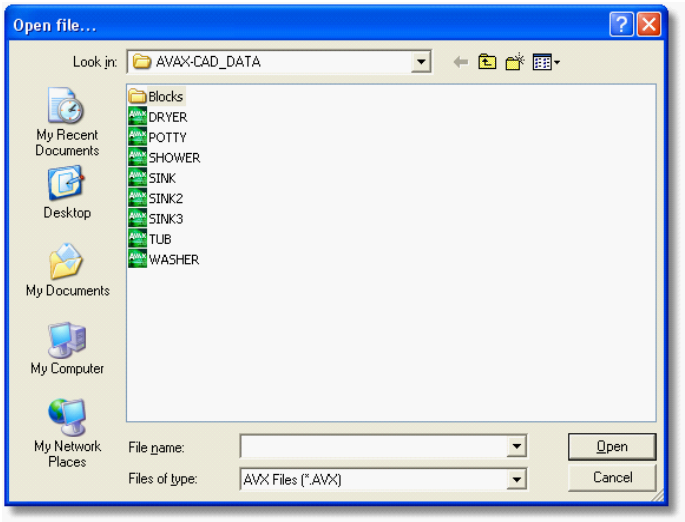
.2.Open (OPEN)

Command:> OPEN
Shortcut: Ctrl+O



By this command we open an existing file of AVX type.

© 1993-2026 Gardos Software LTD



1.3 Save (SAVE)

Command:> SAVE
Shortcut: Ctrl+S

By this command we save the current drawing.

1.4 Save as (SAVE_AS)

Command:> SAVE_AS

By this command we can save a drawing with another file name or in another folder.

1.5 Close (CLOSE_FILE)

Command:> CLOSE_FILE

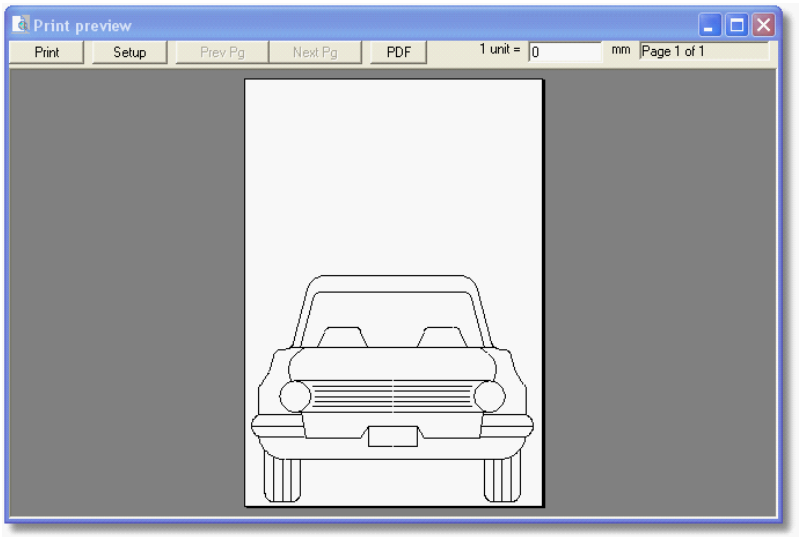
By this command we close the active file.

© 1993-2026 Gardos Software LTD

1.6 Print preview (PRINT PREVIEW)

Command:> PRINT_PREVIEW

By this command we can see a preview of the drawing to be printed.



Choosing the key *Print* we print the drawing to the current printer and choosing *Setup* we can change the settings of printing, and if we choose the key *PDF* we can export the drawing to a PDF file type.

1.7 Print (PRINT)

Command:> PRINT
Shortcut: Ctrl+P

By this command we print the current drawing.

1.8 Import from file (IMPORT)

Command:> IMPORT

By this command we can import to the program files of DXF, DWG, AVX, XML, SVG, SHP, PLT, WMF, RTF, BMP, JPG, GIF, ICO, PCX, PNG and TXT type.

1.9 Export to file (EXPORT)

Command:> EXPORT

By this command we export the active drawing to files of DXF, DWG, SVG, SWF, WMF, PDF, RTF, PLT, BMP, PIC, JPG, and XML type.

1.10 Sending e-mail (SEND_MAIL)

Command:> SEND_MAIL

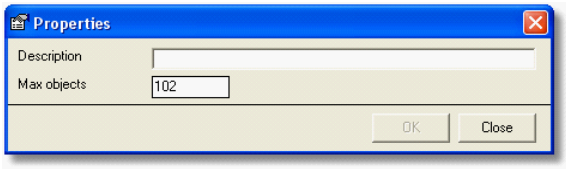
By this command we send the drawing by e-mail.

© 1993-2026 Gardos Software LTD

1.11 Properties (FILE_PROPERTIES)

Command:> FILE_PROPERTIES

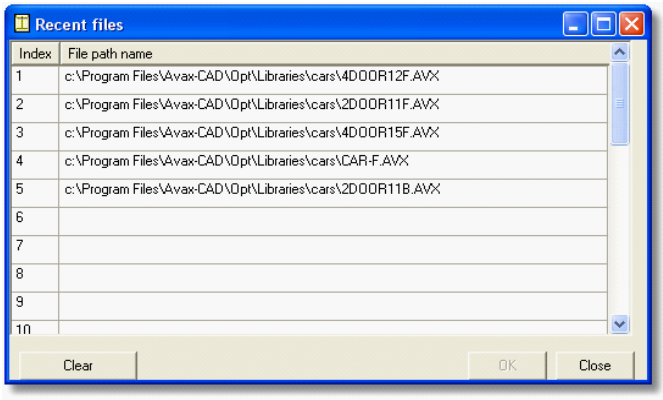
This command opens a window with the properties - description of the current drawing.



1.12 Recent files (RECENT_FILES)

Command:> RECENT_FILES

Choosing *Recent files* opens a window with all the recently used files from the program.



1.13 Exit (EXIT)

Command:> EXIT
Shortcut:> Alt+F4

By this command we terminate the program. Program asks us to save our drawing – if something has changed to it - and then returns to the operating system of our computer, closing all the used files. This is the ONLY PERMISSIBLE WAY OF CLOSING our work (and not e.g. switching off the computer , while we are in the program). By this command we release, all the memory (RAM) that the program was using, into the operating system

1.3.2 Edit

2.1 Undo (UNDO)

Command:> UNDO
Shortcut:> Ctrl+Z

By this command we can reverse the commands that we have already given (only those that can be reversed - UNDO), in the reverse turn that they have been given, taking constantly the drawing back to its previous condition.
Giving the UNDO command we recall the most recent command we've given, on the condition that this recalling is allowed. Giving again UNDO, we recall the exact previous command e.t.c.. After giving the UNDO command, drawing is coming back one step backward, while prompt is getting automatically refreshed, showing the new, every time, drawing.
Notes
(1) Program undoes 3000 commands. That means that it can recall by UNDO, every time, the 3000 last operations that can be recalled.
(2) When we close a drawing and then we open it again, UNDO is not valid for commands we had given before we closed it.
Example
We give command LINE from point (0,0) to point (1,1).
We give CANCEL (not another line).
We give UNDO. The line we've already given is being erased.

2.2 Redo (REDO)

© 1993-2026 Gardos Software LTD

Command:> REDO 
Shortcut: Ctrl+N

By this command we can reverse the effect of the last UNDO command, that was executed. If we use it once, we cannot use it a second time, unless we give a new Undo command.
Conclusive REDO command can recall only one UNDO command every time, in contrast with UNDO command that recalls 3000 commands. So we can say that every UNDO has only one REDO.

2.3 Cut (CUT)

Command:> CUT_AVAX 
Shortcut: Ctrl+X

By this command we cut selected objects from the current drawing. These objects have been saved in the computer's memory and there is the possibility to paste them later to the same or another drawing.

2.4 Copy (COPY)

Command:> COPY_AVAX 
Shortcut: Ctrl+C

This command creates a copy of selected objects which are saved in the computer's memory and there is the possibility to paste them later to the same or another drawing.

© 1993-2026 Gardos Software LTD

2.5 Paste (PASTE)

Command:> PASTE
Shortcut: Ctrl+V

By this command we place all cut (see command 2.3) or copied (see command 2.4) objects to the current drawing.

2.6 Erase (ERASE)

*[*Objects selection] We choose all objects we want to erase.
Command:> ERASE
Shortcut: Ctrl+Del

By this command we erase all selected objects, from the drawing.
First we choose all objects we want to erase from the drawing (see RELEVANT commands SELECT, WINSEL, INWINSEL which appear with capital letters). We remind that, if we have chosen more objects than we want, we can cancel some of them, using SELECT- and SELECT+ commands.
ERASE command, has the ability to be recalled by UNDO command.

Example



-We choose the circle (by SELECT)
-We give command ERASE

2.7 Orthogonal drawing (ORTHO)

Command:> ORTHO

By this command we activate (ORTHO ON) or inactivate (ORTHO OFF) the drawing of only horizontal and vertical lines.
(1) ORTHO command changes condition ORTHO_ON to ORTHO_OFF and vice versa.
(2) ORTHO_ON command. By command ORTHO_ON we return to orthogonal drawing condition, where the lines we draw are horizontal or vertical (parallel to X and Y axes).
(3) ORTHO_OFF command. By this command we cancel the orthogonal drawing, and we can draw oblique lines.
Note
When we are in condition ORTHO_ON and we give commands like MOVE, COPY e.t.c., the movement vector is parallel to X axis or Y axis.
ORTHO command, doesn't have the ability to be recalled by UNDO command.

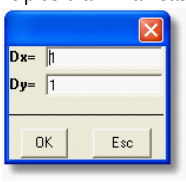
2.8 Grid Mode (GRID)

Command:> GRID
[--->Dx =
Dy =] At the dialog box we enter grid spacing (Where: Dx = horizontal spacing, Dy = vertical spacing).

By this command we create a grid of points on the prompt to DX, DY spaces between them, which we are going to use later, as a reference for objects drawing.

© 1993-2026 Gardos Software LTD

Giving GRID command, a dialog box appears on prompt where we have to give two variables, Dx = horizontal spacing between grid points, and Dy = vertical spacing between the same grid points. As the command is executed the grid appears (its points). Regardless of whether it pre-existed or not, the grid is redrawn according to the new Dx, Dy.
Then, SNAP command (see chapter 2.9) and proper Dx, Dy, restricts the movement of crosshair to intervals already defined, in order to help us draw in an easier way.



GRID command, doesn't have the ability to be recalled by UNDO command.

RELEVANT VARIABLES

Tools	General Options	Snap

→**'Grid' :** Specifies if the grid is active.
→**'Grid by lines':** Specifies if the grid appears by lines.
→**'Dx' :** Specifies the spacing between the points of the grid to axis X.
→**'Dy' :** Specifies the spacing between the points of the grid to axis Y.

→RELEVANT COMMANDS
< GRID_ON >, < GRID_OFF >, < SETGRID >, < SETSNAP >, < SNAP >

2.8.1 Grid mode activation (GRID_ON)

Command:> GRID_ON
[--->Dx =
Dy =] At the dialog box we enter grid spacing). Where: Dx = horizontal spacing, Dy = vertical spacing.

By this command the grid appears. If one or both of the values of DX, DY are equal to 0, then they are requested on the dialog box and the command operates exactly like GRID command.
Otherwise the grid appears and the points have DX, DY spacing between them.
GRID_ON command, doesn't have the ability to be recalled by UNDO command.

RELEVANT VARIABLES

Tools	General Options	Snap

→**'Grid' :** Specifies if the grid is active.
→**'Grid by lines' :** Specifies if the grid appears by lines.
→**'Dx' :** Specifies the spacing between the points of the grid to axis X.
→**'Dy' :** Specifies the spacing between the points of the grid to axis Y.

RELEVANT COMMANDS
< GRID >, < GRID_OFF >, < SETGRID >, < SETSNAP >, < SNAP >

© 1993-2026 Gardos Software LTD

2.8.2. TURNING OFF GRID MODE (GRID_OFF)

Command:> GRID_OFF

This command turns off the grid .It is exactly the opposite of command GRID_ON.
DX, DY (horizontal and vertical spacing between the points of the grid) remain as variables, so when we set the GRID MODE ON, the grid appears again in the previous form.
GRID_OFF command, doesn't have the ability to be recalled by UNDO command.

RELEVANT VARIABLES

Tools	General Options	Snap

→ **'Grid' :** Specifies if the grid is active.
→ **'Grid by lines' :** Specifies if the grid appears by lines.
→ **'Dx' :** Specifies spacing between the points of the grid to axis X.
→ **'Dy' :** Specifies spacing between the points of the grid to axis Y.

RELEVANT COMMANDS
< GRID >, < GRID_OFF >, < SETGRID >, < SETSNAP >, < SNAP >

2.9 Snap to Grid (SNAP)

Command:> SNAP
[--->Dx =
Dy =] At the dialog box we enter the horizontal and vertical step.
Where: Dx = horizontal step, Dy = vertical step.

By this command the cursor's movement is restricted to specified intervals (points of grid).
By SNAP command and DX, DY positive e.g. DX=1 DY=1, crosshair moves only on the points of the drawing that they have spacing between them DX and DY. If points of the grid are visible and DX, DY of the grid (GRID) equal to DX, DY of the SNAP, then crosshair's movement is restricted on the points of the grid, making drawing easier.
If we want to cancel the step movement of crosshair, we enter SNAP command again, with DX=0 DY=0 (or DX=DY=0).
SNAP command, doesn't have the ability to be recalled by UNDO command.

RELEVANT VARIABLES

Tools	General Options	Snap

→ **'Snap to grid' :** Specifies if the snap to grid mode is active.
→ **'Grid' :** Specifies if the grid is active.
→ **'Dx' :** Specifies spacing between the points of the grid to axis X.
→ **'Dy' :** Specifies spacing between the points of the grid to axis Y.

RELEVANT COMMANDS
< GRID >, < SETSNAP>

© 1993-2026 Gardos Software LTD

2.10 Text style

2.10.1 Bold
Command:> TEXT_BOLD
This command enables or disables the text Bold status.
2.10.2 Italic
Command:> TEXT_ITALIC
This command enables or disables the text Italic status.

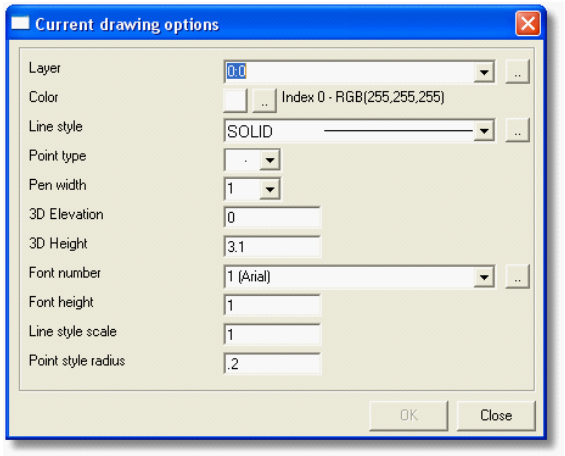
2.10.3 Underline
Command:> TEXT_UNDERLINE
This command enables or disables the text Underlined status.

2.10.4 Strike
Command:> TEXT_STRIKE
This command enables or disables the text Striked status.

2.11 Current options (SET)

```
Command:> SET
          [---> Layer =
                color     =
                  ...          =]  We specify the values of variables.
```

By this command we specify the values of variables on a dialog box. These variables are always valid and affect the drawing e.g. LAYER, COLOR, PEN WIDTH e.t.c.



→ **'Layer' :** We specify the drawing layer where drawing objects will belong.
→ **'Color' :** We specify layers color.
→ **'Line type' :** We specify line type (e.g. SOLID).
→ **'Point type' :** We specify point type (e.g. cross +).
→ **'Pen width' :** We specify pen width.
→ **'3D elevation' :** We specify objects elevation at axis Z.

© 1993-2026 Gardos Software LTD

→ **'3D height' :** We specify objects height at axis Z**.**
→ **'Font' :** We specify text font (e.g. Arial).
→ **'Text height' :** We specify text height.
→ **'Line scale by type' :** We specify the hidden line size, to lines by type.
→ **'Point radius by type' :** We specify the point size, to points by type.

During the time we are drawing, some variables specify the way we draw. One of these is e.g. the current drawing LAYER, where we enter everything we are drawing. What happens when we want to change Layer? Here comes SET command, to give us this capability. This command is simple. We enter SET and at the dialog box we change whatever we want.
SET command, doesn't have the ability to be recalled by UNDO command.

RELEVANT VARIABLES

Tools	General Options	Snap

→ **'Layer' :** We specify the drawing layer where drawing objects will belong.
→ **'Color' :** We specify layers color.
→ **'Line type' :** We specify line type (e.g. SOLID).
→ **'Point type' :** We specify point type (e.g. cross +).
→ **'Pen width' :** We specify pen width.
→ **'3D elevation' :** We specify objects elevation at axis Z.
→ **'3D height' :** We specify objects height at axis Z**.**
→ **'Font' :** We specify text font (e.g. Arial).
→ **'Text height' :** We specify text height.
→ **'Line scale by type' :** We specify the hidden line size, to lines by type.
→ **'Point radius by type' :** We specify the point size, to points by type.
→ **'Pixel width by mm':** We specify the pixel width by mm.
→ **'Width view by scale':** We specify if the pixel width will appear by scale.

RELEVANT COMMANDS
< SETUP >

2.12 Cancel (CANCEL)

Command:> CANCEL
Shortcut: Esc
MOUSE: Right key -> Canceling

By this command we cancel all actions and unselect all selected objects, if there are any.
The kind of command we are canceling is in proportion with the result E.g. when we are drawing by LINE command, command will stop execution and the line we were going to draw in the next step will not be drawn, but lines that we have drawn until now will remain the same. (Let's note that giving LINE command

© 1993-2026 Gardos Software LTD

only once, we can draw several lines until we stop). Generally, it cancels the next step we are going to do.
CANCEL command, doesn't have the ability to be recalled by UNDO command

1.3.3 View

3.1 Zoom by window (ZOOM)

Command:> ZOOM 
[First point] We enter the first point of the drawing area we want to zoom in.
[Second point] We give the second point of the area, on the diagonal.

By this command we "bring" visually closer an area of our drawing, so we can see more details, that is we 'enlarge' it to look exactly like it would be if we were looking at the drawing through a magnifying glass. The real size of the drawing remains the same. Then we can bring back the drawing to its previous condition ("keep back") by PREVIOUS command (see 3.2).
We give ZOOM command and then two points which define a window, into which there is the area we want to see better. This area is being enlarged, overtaking all the available prompts' drawing area. Then we can work there with emphasis on details and later on, either come back to the previous condition or give another one ZOOM.
ZOOM command, has the ability to be recalled by UNDO command, only if the variable at GENERAL OPTIONS "UNDO to ZOOM" is equal to 1.

RELEVANT COMMANDS
< PREVIOUS >, < ZOOMUNDO_ON >, < ZOOMUNDO_OFF >, < ZOOMWINDOW >

3.2 Previous view of the drawing (PREVIOUS)

Command:> PREVIOUS 

By this command the previous view of our drawing appears. Program has the ability to store in memory more than 50 views of the drawing.
Giving PREVIOUS command, if there was a ZOOM, this is recalled, and by redrawing, all objects have the size that they had before ZOOM. The real objects size remains the same (as well as ZOOM command). The only thing that changes is their relative distance from the user.
If we give PREVIOUS command again, the ZOOM before the last one is recalled and so on, until there is no other ZOOM active, and the result is nothing (we see relevant message).
PREVIOUS command has the ability to be recalled by UNDO command if the variable at GENERAL OPTIONS is equal to 1.

RELEVANT COMMANDS
< ZOOM >, < ZOOMUNDO_ON >, < ZOOMUNDO_OFF >, < ZOOMWINDOW >

3.3 View all drawing (ALL)

Command:> ALL 

By this command we have the view of all drawing objects within the limits of the prompt.

© 1993-2026 Gardos Software LTD

ALL command is useful to help us see the whole drawing by a single move, especially after a drawings part Zooming.
Note:
After that command, prompts limits are these that define from USERLIMITS command (see 3.11). If we haven't used the USERLIMITS command, Xmin, Xmax, Ymin, Ymax, Zmin, Zmax will be the default tablets limits at choice GENERAL PARAMETRES (see 8.7). For example Xmin=-5, Xmax=20, Xmin Ymin=-5,Ymax=10 and Zmin=Zmax=0.
ALL command doesn't have the ability to be recalled by UNDO command.

RELEVANT COMMANDS
< AUTOLIMITS >, < USERLIMITS >

3.4 View to the drawing extends (EXTENDS)

Command:> EXTENDS 

By EXTENDS command, drawing is being redrawn in such a way that it is displayed exactly at the prompts extends.
Note
There is a possibility, the drawing not to be fit at the prompts extends, even though we have given EXTENDS command. This could happen if in a moment we drew out of the limits, that had been given by the previous CALCEXTENDS command.
In this case we give the CALCEXTENDS command again, at the time when the real new drawing limits are calculated again, and then, EXTENDS command will refer to these new limits.
EXTENDS command doesn't have the ability to be recalled by UNDO command.

RELEVANT COMMANDS
<AUTOLIMITS>, < CALCEXTENDS >, <USERLIMITS>,

3.5 Zoom into the drawing (ZOOM_IN)

Command:> ZOOM_IN 
Shortcut: F2

By this command we can focus into the drawings area, we are interested in, as near as we want so we can see more details.

3.6 Zoom out of the drawing (ZOOM_OUT)

Command:> ZOOM_OUT 
Shortcut: F3

By this command we zoom out of the drawing enlarging the available drawing area.

3.7 Dynamic zoom (ZOOM_DYNAMIC)

Command:> ZOOM_DYNAMIC 

By this command the generated portion of the drawing is displayed, with a view box which you can shrink or enlarge.

3.8 Drawings removal (ZOOM PAN)

Command:> ZOOM_PAN

By this command the drawing is being moved in the current viewport.

© 1993-2026 Gardos Software LTD

3.9 Magnifying glass (MAGNIFY)

Command:> MAGNIFY

By this command a magnifying glass appears by which we can zoom to the objects we want.

3.10 Auto drawing limits (AUTOLIMITS)

Command:> AUTOLIMITS

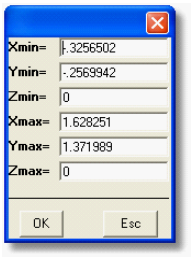
By this command we can increase the drawing limits and make space around the existing drawing.
AUTOLIMITS command has the intention to create the whole drawing vision on the prompt - regardless of how much stretched it is – in such a way, so that can fit at prompts limits and have margins around it.
Details:
This command searches to find an imaginary rectangle which encloses the drawing. Then put it (this and the enclosed drawing) to relative places with the down left and the upper right corner of the drawing area, so that they fit with the variables “down left factor” and “upper right factor”.
So the diagonal real co-ordinates which enclose the drawing are (0.0) and (20.15), and then the drawing will have width 20 and height 15. If the previous factors ("down left" and "upper right") are 0.5 then the distances will be DX=0.5*20=10 and DY=0.5*15=7.5.
So the rectangle that encloses the drawing will be distant DX=10 and DY=7.5 from the down left corner and DX>10 and DY=7.5 from the upper right corner. The whole drawing is placed into the imaginary rectangle which encloses the drawing and in this way we have a better vision. If we want of course another placing of the enclosed rectangle, we change the above factors from 0.5 to other values.
AUTOLIMITS command doesn't have the ability to be recalled by UNDO command.

RELEVANT COMMANDS
< CALCEXTENDS >, < EXTENDS >, < USERLIMITS >, < ZOOM >

3.11 User's drawing limits (USERLIMITS)

[Command] USERLIMITS
[---> Xmin
Ymin
Zmin
Xmax
Ymax
Zmax =] We give the down left (min) and the upper right corner (max).

By this command we can set the co-ordinates of the down left corner and the upper right corner of our drawing area, so as to have the vision of a specific drawing area.



© 1993-2026 Gardos Software LTD

We give the USERLIMITS commands. Then we give the dialog box the co-ordinates of the down left corner of drawing area (Xmin, Ymin, Zmin). Program has default values, which we either accept or change. Likewise, we change the co-ordinates of the over right corner (Xmax, Ymax, Zmax). The drawing is redrawn with the new co-ordinates, so as to be according to the new limits of the drawing area. (In fact, the drawing size doesn't change – the only thing that changes is the drawing part size that appears to prompt).
USERLIMITS command doesn't have the ability to be recalled by UNDO command
Note
At first there are the values of the variables Xmin, Xmax, Ymin, Ymax (approximately those of a signpost).

RELEVANT COMMANDS
< AUTOLIMITS >, < CALCEXTENDS >, < EXTENDS >, < ZOOM >

3.12 Redrawing (REDRAW)

Command: REDRAW 

By this command the active part of the drawing that appears to prompt is refreshed.
That command doesn't change anything. The only thing it does is to refresh the prompt by drawing again the visible part of the drawing. After this refreshing, all those little spots (points like "spots") that had been left on prompt, after corrections or other operations to the drawing, disappear.
REDRAW command doesn't have the ability to be recalled by UNDO command.
Notes
(1) This redrawing will take place automatically by the program, when this is necessary.
(2) In case of ZOOMWINDOW the REDRAW command cancels it.

1.3.4 Drawing

4.1 Line

By this command we create straight lines.
Command: LINE 

4.2 Circle arc

By this command we create arcs.
Command: ARC 

4.3 Circle

By this command we create a circle by center and radius.
Command: CIRCLE 

4.4 Circle from 2 points – endpoints of diameter

By this command we create a circle based on two endpoints of the diameter.
Command: CIRCLE2P 

4.Circle from 3 points – of perimeter

By this command we create a circle by three points given by the user.
Command: CIRCLE3P 

© 1993-2026 Gardos Software LTD

4.6 Rectangle

By this command we create a rectangle.
Command: RECTANGLE



4.7 Ellipse

By this command we create an ellipse.
Command: ELLIPSE



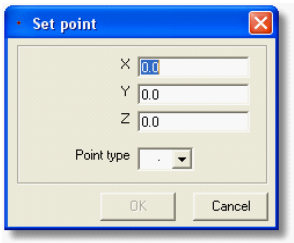
4.8 Point

By this command we create points.
Command: POINT

4.9 Point from co-ordinates

By this command we draw points based on its x, y, z co-ordinates. At the dialog box we have the ability to define the points type.
Command: POINTCOOR

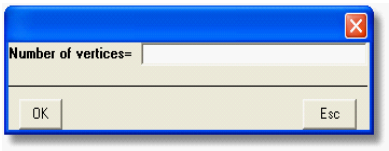




4.10 Polygon

By this command we create an equilateral closed polygon which has v vertexes. At the dialog box we give the number v of polygon vertexes.
Command: POLYGON





4.11 Polyline

By this command we create polylines which consist of adjoining parts.
Command: POLYGON

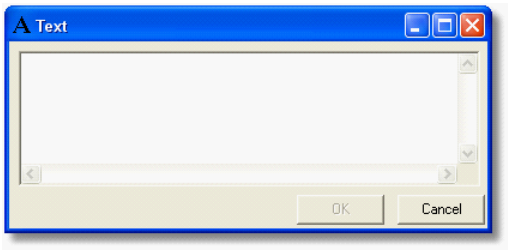


Creates polylines (adjoining straight parts of lines).
Command: POLYLINE

© 1993-2026 Gardos Software LTD

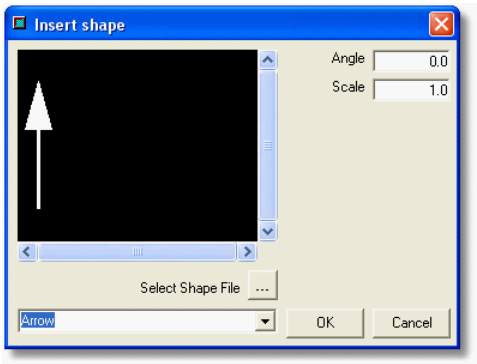
4.12 Text

Creates a single line of text.
Command: TEXT



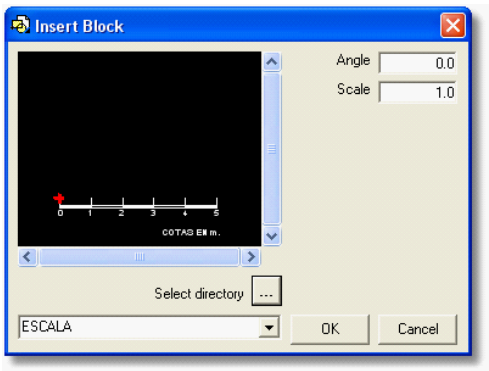
4.13 Shape

Inserts into the drawing, a shape by scale and angle.
Command: SHAPE



4.14 BLOCKS Insertion

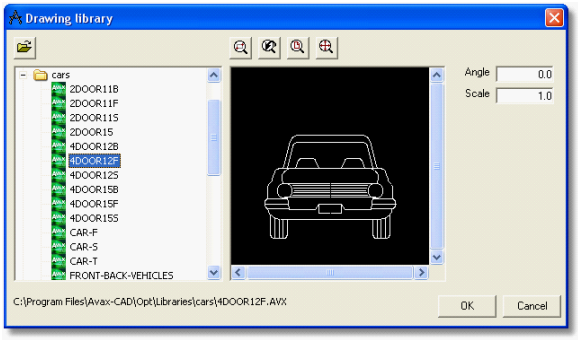
Inserts a block or a drawing in the current drawing using scale and angle.
Command: INSERT



© 1993-2026 Gardos Software LTD

4.15 Drawing library

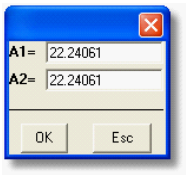
Opens a list of already made drawings to insert them in the drawing using scale and angle.
Command: SELECT_LIBRARY



4.16 Complex objects

4.16.1 Triangle

Creates a triangle choosing a straight line as the first side and giving the length of the other two sides.
Command: TRIANGLE

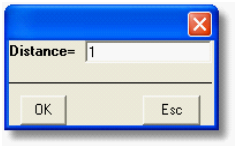


4.16.2 Angles dimensioning

Measures and draws the angle between two chosen lines.
Command: ANG

4.16.3 Line segment dimensioning

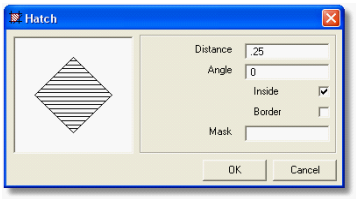
Measures the length of a line segment and the line will be drawn at a defined, by the user, distance.
Command: DIM



© 1993-2026 Gardos Software LTD

4.16.4 Hatch

Creates hatch lines from a selected polyline or a temporary polygon.



4.16.5 Polygon radii

Creates new lines from the center of a selected closed polyline or a temporary polygon to the vertices.
Command: RADIATE

4.16.6 BEZIER curve

Creates a Bezier curve from a selected polyline or a temporary polygon.
Command: BEZIER

4.16.7 Cubic SPLINE 1

Creates a Cubic SPLINE 1 curve from a selected polyline or a temporary polygon.
Command: CSPLINE1

4.16.8 Cubic SPLINE 2

Creates a Cubic SPLINE 2 curve from a selected polyline or a temporary polygon.
Command: CSPLINE2

4.16.9 Open B-SPLINE

Creates an Open B-SPLINE curve from a selected polyline or a temporary polygon.
Command: OBSPLINE

4.16.10 Closed B-SPLINE

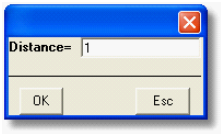
Creates a Closed B-SPLINE curve from a selected polyline or a temporary polygon.
Command: CBSPLINE

4.16.11 Vertices numbering

Numbers polygons vertices.
Command: VNCOUNT

4.16.12 Wall

Creates a wall.
Command: WALL



© 1993-2026 Gardos Software LTD

4.16.13 Wall from 3 points.
Creates a wall from three defined points.
Command: WALL3P

4.16.14 Border
Creates a new polyline to be the border of a selected polyline or a temporary polygon.
Command: BORDER

4.16.15 Centroid
Creates a new point in the center of a selected closed polyline or a temporary polygon.
Command: CENTROID

1.3.5 Select

5.1 Select object
Enables the select action, so user can add or remove objects (one object every time) from the selection set, using a small box which replaces the crosshair on the graphics cursor.
Command: SELECT

5.2 Select all objects
Selects all drawing objects.
Shortcut: Ctrl+A
Command: SELECT_ALL

5.3 Window select
Selects all drawing objects inside a rectangle defined by two points.
Command: WINSEL

5.4 In window select
Selects all drawing objects completely inside a rectangle defined by two points.
Command: INWINSEL

5.5 Sub selected
Switches to the “sub” selection method in which selected objects can be removed from the selection set.
Command: SELECTSUB

5.6 Add selected
Switches to the “add” selection method in which selected objects can be added to the selection set.
Command: SELECTADD

© 1993-2026 Gardos Software LTD

5.7 Select vertices – area creation
Select vertices to make a temporary polygon.
Command: SELECTV

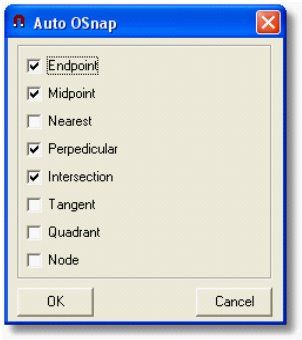
5.8 Last selected vertices
Selects the last temporary polygon.
Command: LASTSV

5.9 Make selected vertices
Creates a temporary polygon from a polyline object.
Command: MAKESV

5.10 Cancel objects selection
Unselects all selected objects if there are any.
Command: UNSELECT

1.3.6 Object snap

6.1 Auto OSnap
Enables or disables the auto-object snap operation and determines the kind of the auto-object snap points to be selected.
Command: AUTOOSNAP



6.2 End
Snaps to the closest endpoint of a line, polyline, arc e.t.c.
Command: ENDPOINT

6.3 Intersection
Snaps to an intersection of a line, polyline, arc e.t.c.
Command: INTERSEC

6.4 Middle or center of a circle
Snaps to the midpoint of a line, polyline, arc e.t.c. or to the center o a circle.
Command: MIDPOINT

© 1993-2026 Gardos Software LTD

6.5 Node
Snaps to a point.
Command: NODE

6.6 Nearest point
Snaps to the closest point of a line, polyline, arc e.t.c.
Command: NEAREST

6.7 Perpendicular
Snaps to a point perpendicular to a line, polyline, arc e.t.c.
Command: VERTICAL

6.8 Tangent
Snaps to the tangent of a circle and ellipse.
Command: TANGENT

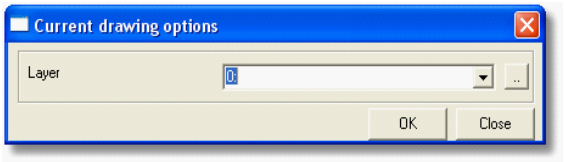
6.9 Quadrant
Snaps to a quadrant point of a circle and ellipse.
Command: QUADRANT

6.10 Close polyline
Draws a line segment from the current position to the starting point of the polyline, creating a closed polyline..
Command: CLOSE

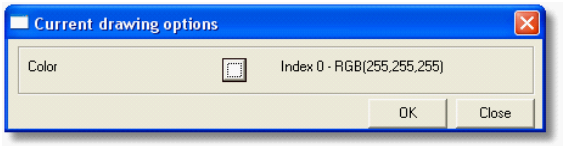
1.3.7 Modify

7.1 Change

7.1.1 Layer
Changes the selected object's layer.
Command: CHANGE_LAYER



7.1.2 Color
Changes the selected object's color.
Command: CHANGE_COLOR



© 1993-2026 Gardos Software LTD

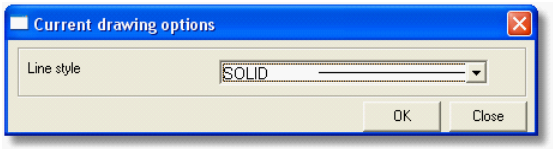
7.1.3 Width

Changes the selected object's Width.
Command: CHANGE_WIDTH



7.1.4 Line type

Changes the selected object's line type.
Command: CHANGE_LTYPE



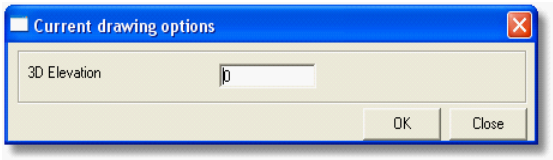
7.1.5 Point type

Changes the selected object's point type.
Command: CHANGE_PTYPE



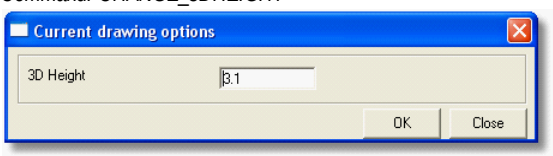
7.1.6 3D elevation

Changes the 3D elevation of the selected object.
Command: CHANGE_3DELEVATION



7.1.7 3D height

Changes the 3D height of the selected object.
Command: CHANGE_3DHEIGHT



© 1993-2026 Gardos Software LTD

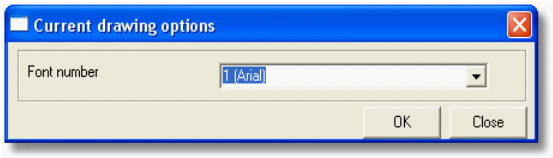
7.1.8 Text

Changes the selected text.
Command: CHANGE_TEXT



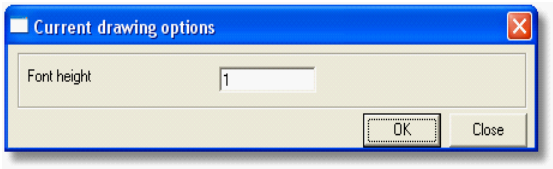
7.1.9 Font

Changes the selected objects' font.
Command: CHANGE_FONT



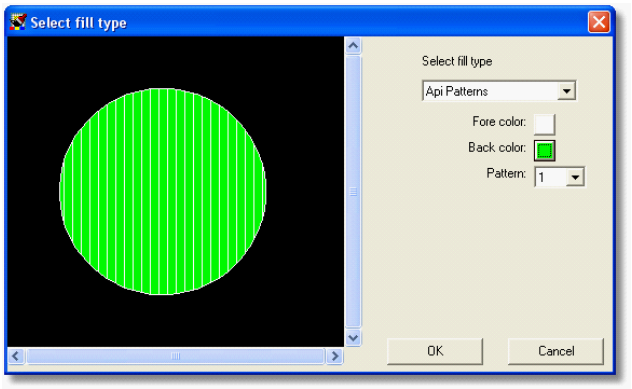
7.1.10 Text height

Changes the height of the selected text.
Command: CHANGE_TEXTHEIGHT



7.1.11 Fill

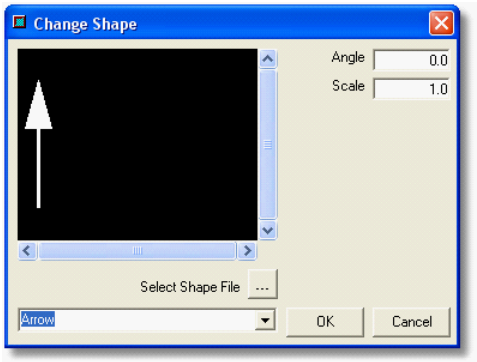
Changes the fill of a selected closed polyline.
Command: CHANGE_FILL



© 1993-2026 Gardos Software LTD

7.1.12 Shape

Replaces the shape with an other shape.
Command: CHANGE_SHAPE



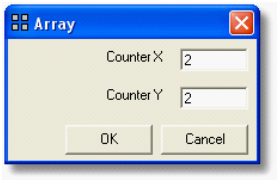
7.2 Copy

7.2.1 Copy

Duplicates selected drawing objects at a specified distance and direction defined by the user by a temporary line.
Command: COPY

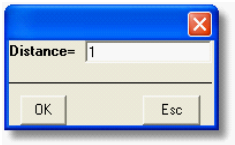
7.2.2 Array

Creates multiple copies of already selected objects.
Command: ARRAY



7.2.3 Copy offset

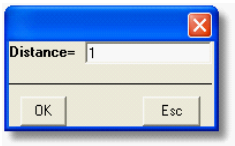
Creates parallel lines at a specified distance.
Command: OFFSET



© 1993-2026 Gardos Software LTD

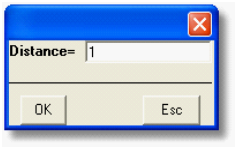
7.2.4 Copy offset in

Creates a new polyline inside a selected closed polyline or a temporary polygon at a specified distance.
Command: COPY_IN



7.2.5 Copy offset out

Creates a new polyline outside a selected closed polyline or a temporary polygon at a specified distance.
Command: COPY_OUT



7.3 Move

7.3.1 Move

Displaces selected drawing objects at a specified distance and direction defined by the user by a temporary line.
Command: MOVE

7.4 Transformation

7.4.1 Mirror

Creates a mirror copy of selected drawing objects about a temporary line given by the user.
Command: MIRROR

7.4.2 Rotate

Rotates the selected objects.
Command: ROTATE

7.4.3 Scale

Enlarges or reduces selected drawing objects by a temporary line given by the user.
Command: SCALE

7.5 Deformation

7.5.1 Stretch

Moves or stretches already selected lines or polylines.
Command: STRETCH

© 1993-2026 Gardos Software LTD

7.5.2 Trim

Trims lines at a cutting edge defined by other already selected lines.
Command: TRIM

7.5.3 Extend

Extends a line to meet another line.
Command: EXTEND

7.5.4 Fillet

Fillets the edges of selected lines.
Command: FILLET

7.6 Erase

7.6.1 Erase

Removes all selected objects from the drawing.
Command: ERASE

7.7 Measurement

7.7.1 Length

Estimates the length of a line or a polyline.
Command: MEASURE

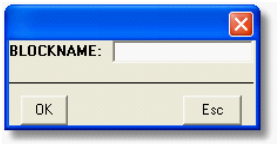
7.7.2 Area

Estimates the area of a selected closed polyline or a temporary polygon.
Command: AREA

7.8 Union-Split

7.8.1 Create Block

Creates a block definition from selected objects.
Command: BLOCK



7.8.2 Create polyline

Creates a polyline by the already selected lines.
Command: PEDIT

© 1993-2026 Gardos Software LTD

7.8.3 Divide

Places evenly spaced point objects along the length or perimeter of a selected object.
Command: DIVIDE



7.8.4 Explode

Breaks compound selected objects into their component objects.
Command: EXPLODE

7.8.5 Break line

Erases part of already selected line or splits this line in two, by a temporary line given by the user.
Command: BREAK

7.8.6 Explode Block

Breaks compound selected objects into their component objects.
Command: EXPLODE

7.8.7 Taxonomy

Shorts the selected polyline vertices when they are not continued.
Command: TAXIS

7.9 Alignment

7.9.1 Top - middle

Aligns the selected object with the top and the middle of the last selected object.
Command: Top_middle

7.9.2 Top - right

Aligns the selected object with the top and the right of the last selected object.
Command: Top_right

7.9.3 Right

Aligns the selected object with the right and the middle of the last selected object.
Command: Right_middle

7.9.4 Bottom - right

Aligns the selected object with the bottom and the right of the last selected object.
Command: Bottom_right

© 1993-2026 Gardos Software LTD

7.9.5 Bottom - middle

Aligns the selected object with the bottom and the middle of the last selected object.
Command: Bottom_middle

7.9.6 Bottom - left

Aligns the selected object with the bottom and the left of the last selected object.
Command: Bottom_left

7.9.7 Left - middle

Aligns the selected object with the left and the middle of the last selected object.
Command: Left_middle

7.9.8 Top - left

Aligns the selected object with the top and the left of the last selected object.
Command: Top_left

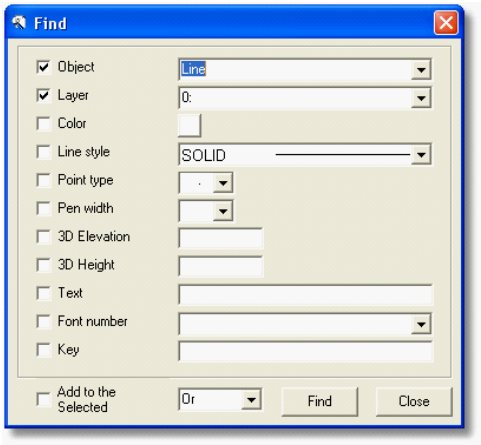
7.9.9 Center

Aligns the selected object with the center of the last selected object.
Command: Center

1.3.8 Tools

8.1 Find

Opens a window to find some objects by their properties.
Command: FIND



© 1993-2026 Gardos Software LTD

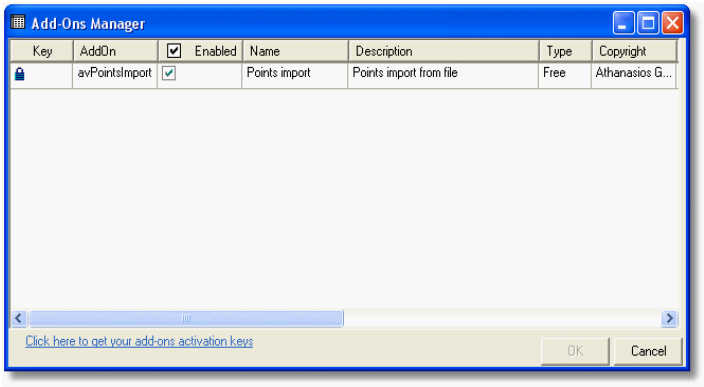
You can choose the property you want to be searched, using the selections at the left side of the prompt and then choose or enter the properties value at the right side..
Choosing **Add to selected** you can add the results of a searching to the results of the previous searching. The list at the bottom of the prompt defines the treaty of finding (**OR**: to be satisfied one of the chosen properties, **AND**: to be satisfied all the chosen properties, **NOT**: not to be satisfied any of the chosen properties).

8.2 Match properties

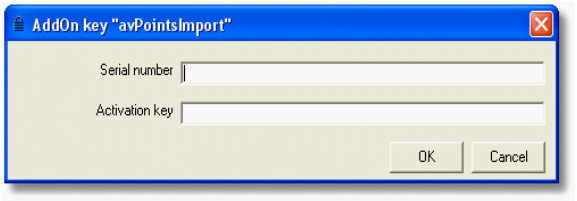
Matches the properties of the selected objects with the last selected object.
Command: MATCH_PROPERTIES

8.3 Add-Ons Manager

Opens a window to select the available add-ons to be included in the pull down menu.
Command: ADD-ONS_MANAGER



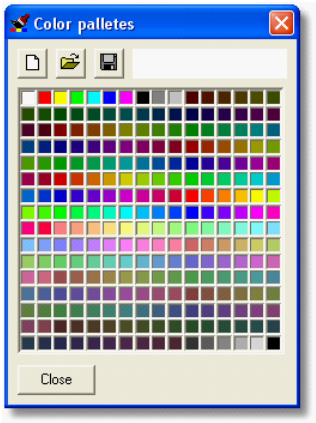
By double click on the cell **Key** you can enter the serial number and the activation code of every add-on.



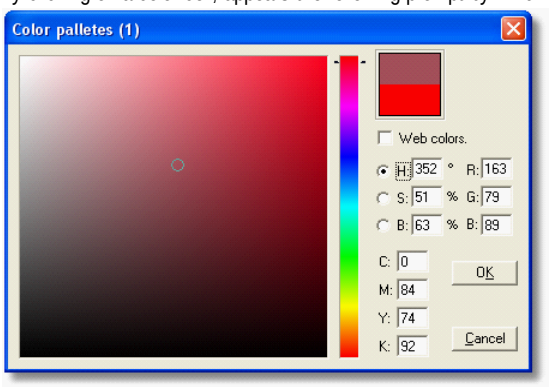
© 1993-2026 Gardos Software LTD

8.4 Color palettes

Opens a window to create and process the color palettes.
Command: COLOR_PALETTE



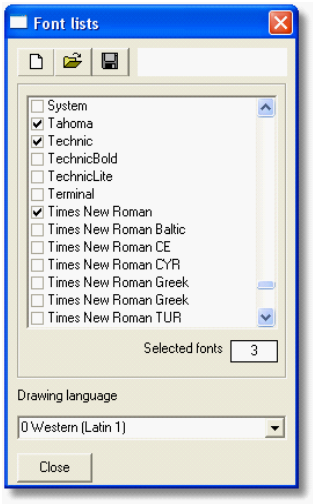
By clicking on a color cell, appears the following prompt by which you can choose and change a color.



© 1993-2026 Gardos Software LTD

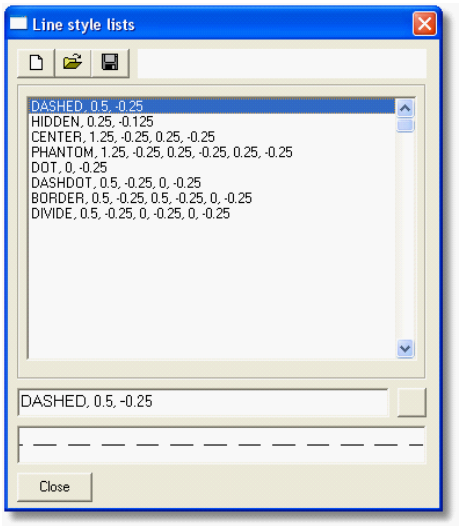
8.5 Font lists

Opens a window to create and process new line font lists.
Command: FONT_LIST



8.6 Line style lists

Opens a window to create and process new line style lists.
Command: LINETYPE_LIST

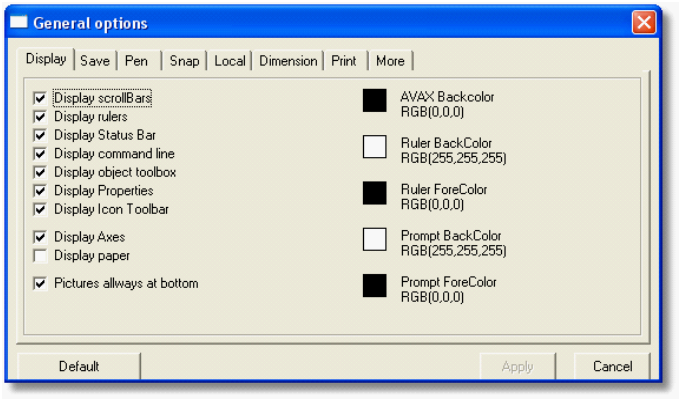


© 1993-2026 Gardos Software LTD

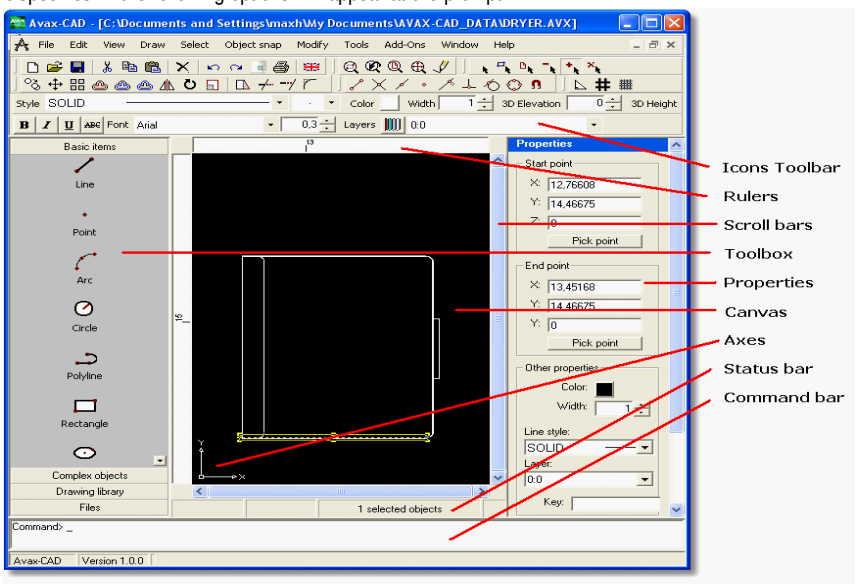
8.7 General options

Opens a window with many tabs, so as to enable the user to change the general options of the program. Command: PROGRAM_OPTIONS

1. It specifies the mode of programs and drawings area appearance.



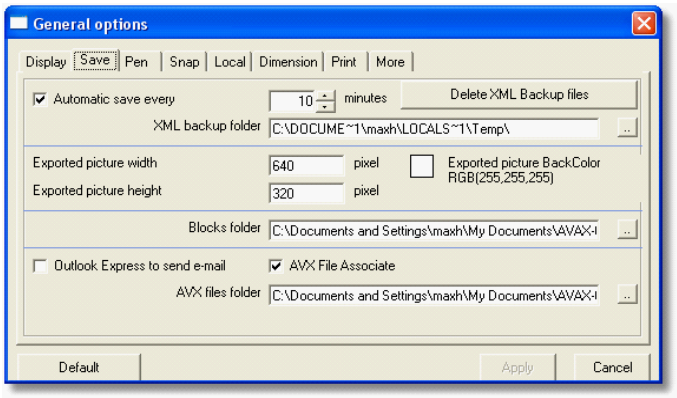
It specifies if the following options will appear at the prompt:



- → **'Pictures always at bottom' :** Specifies if pictures will appear always at the bottom (to the background).
- → **'AVAX BackColor' :** Specifies AVAX backcolor.
- → **'Ruler BackColor' :** Specifies rulers backcolor.
- → **'Ruler ForeColor' :** Specifies rulers pen color.
- → **'Prompt backcolor' :** Specifies command's line color.
- → **'Prompt ForeColor'' :** Specifies command line pen color.

© 1993-2026 Gardos Software LTD

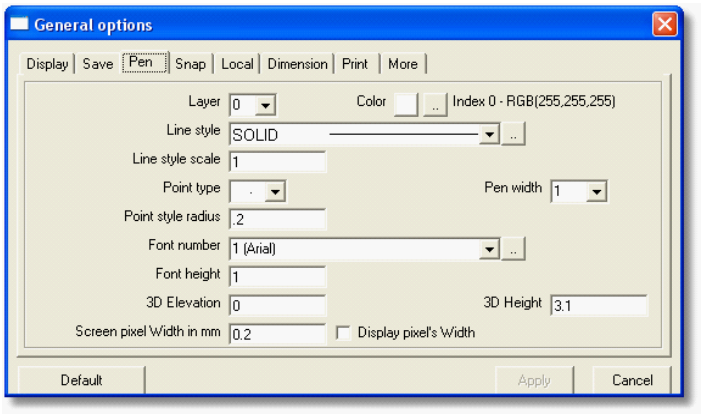
2. Saving and exporting options are specified.



- → **'Automatic save every' :** Specifies if the program will save the drawing automatically during the design processing.
- → **'minutes' :** Specifies the space of time in which the automatic drawing save will take place (e.g. every 10 minutes).
- → **'Delete XML backup files' :** Specifies the deleting of XML backup files, which are saved by the program.
- → **'XML backup folder' :** Specifies the computers disk place where the drawing XML files will be saved.
- → **'Exported picture width' :** Specifies the width of the export picture to pixels.
- → **'Exported picture height' :** Specifies the height of the export picture to pixels.
- → **'Exported picture back color' :** Specifies the background color of the export picture to pixels.
- → **'Block's folder' :** Specifies the computers disk place where will be the list of blocks.
- → **'Outlook Express to send e-mail' :** Specifies if the send e-mail command of Outlook Express will be active.
- → **'AVX file Associate' :** Specifies if programs connection command with files AVX type, will be active.
- → **'AVX files folder' :** Specifies the computers disk place where the list of AVX files will be.

© 1993-2026 Gardos Software LTD

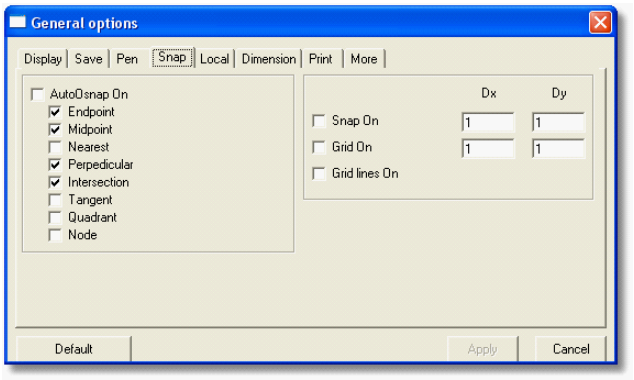
3. The new objects' pen properties are specified.



- → **'Layer' :** Specifies the drawing layer where the drawing objects will belong.
- → **'Color'** Specifies the layer's color.
- → **'Line style' :** Specifies the line style (e.g. SOLID).
- → **'Lines style scale ' :** Specifies hidden lines size, to lines by type.
- → **'Point type' :** Specifies the style of points' appearance (e.g. cross +).
- → **'Point style radius' :** Specifies points' size by type.
- → **'Pen width' :** Specifies pen's width.
- → **'Font number' :** Specifies text font appearance (e.g. Arial).
- → **'Text height' :** Specifies text height.
- → **'3D elevation' :** Specifies object's drawing elevation at axis Z.
- → **'3D height' :** Specifies objects' height at axis Z.
- → **'Screen pixel width in mm'** Specifies the drawing pixel width in mm.
- → **'Display pixel's width :** Specifies if pixel width will appear by scale.

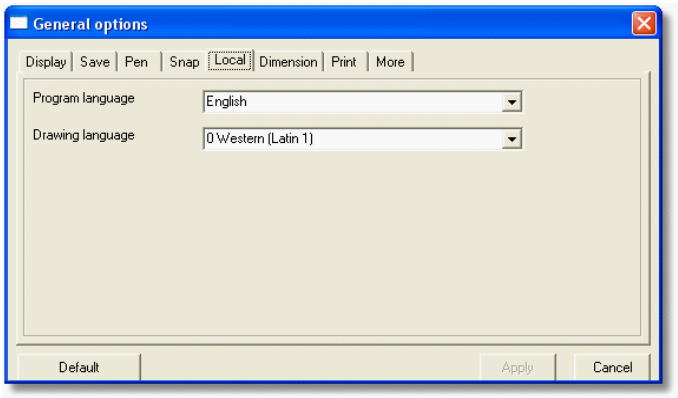
© 1993-2026 Gardos Software LTD

4. The points' snap properties are specified.



- → **'AutoOsnap on' :** Specifies crosshairs snap points, with choices : end, middle, near, perpendicular, intersection, tangent, quadrant, node.
- → **'Snap on' :** Specifies if ruler's snap to grid will be active.
- → **'Grid on' :** Specifies if grid will be active.
- → **'Grid lines on' :** Specifies if grid will appear by lines..
- → **'Dx' :** Specifies the spacing between the points of the grid to axis X.
- → **'Dy':** Specifies the spacing between the points of the grid to axis Y.

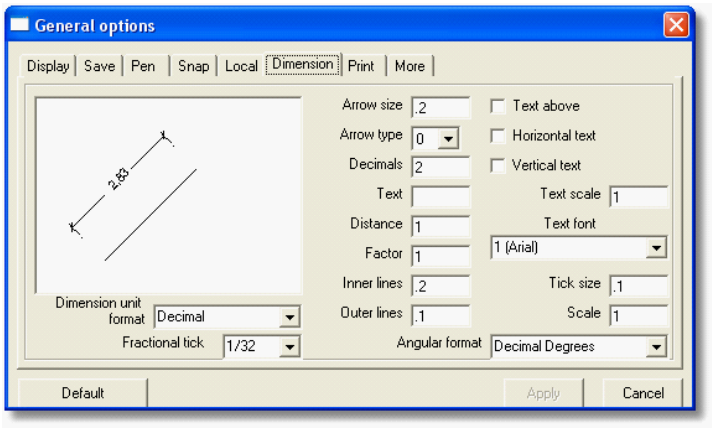
5. The local language properties of programs and drawings are specified.



- → **'Program language' :** Specifies the program language (Greek or English).
- → **'Drawing language' :** Specifies the drawing text language.

© 1993-2026 Gardos Software LTD

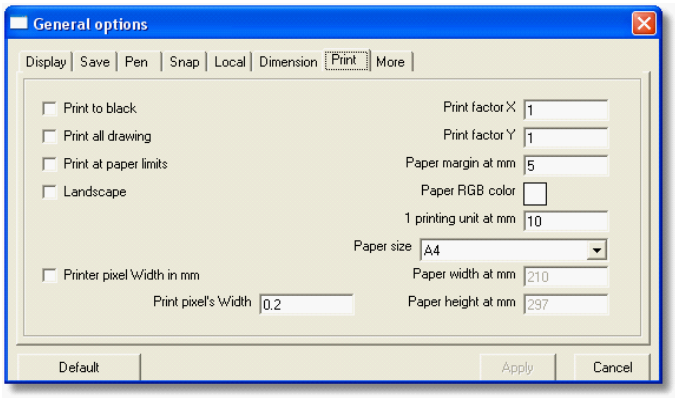
6. New drawing dimension properties are specified..



- → **'Arrow size' :** Specifies dimension arrow size (to length).
- → **'Arrow type' :** Specifies the arrow type (0 – line, 1 – arrow, 2 – circle).
- → **'Decimals' :** Specifies the number of decimals of the dimension.
- → **'Text' :** Specifies the dimension text.
- → **'Distance' :** Specifies the dimension of distance from the straight line segment that is being dimensioned.
- → **'Factor' :** Specifies the dimension factor increase.
- → **'Inner lines' :** Specifies the protrusion of lines that define the dimension, to the straight line segment side.
- → **'Outer lines' :** Specifies the protrusion of lines that define the dimension, to the straight line segment outside.
- → **'Text above' :** Specifies if the text will appear above the dimension line.
- → **'Horizontal text' :** Specifies if the text will appear horizontal.
- → **'Vertical text' :** Specifies if the text will appear vertical.
- → **'Text scale' :** Specifies the size of dimension text.
- → **'Text font' :** Specifies the font type of the dimension text (e.g. Arial).
- → **'Tick size' :** Specifies the tick size (line, arrow, circle) which limits the dimension.
- → **'Scale' :** Specifies dimension size (text and arrow).
- → **'Angular format' :** Specifies the format of angles (e.g. degrees).
- → **'Dimension unit format' :** Specifies the format of dimensions (e.g. decimal).
- → **'Fractional tick' :** Specifies the fractional tick when dimensions appear in fractional type.

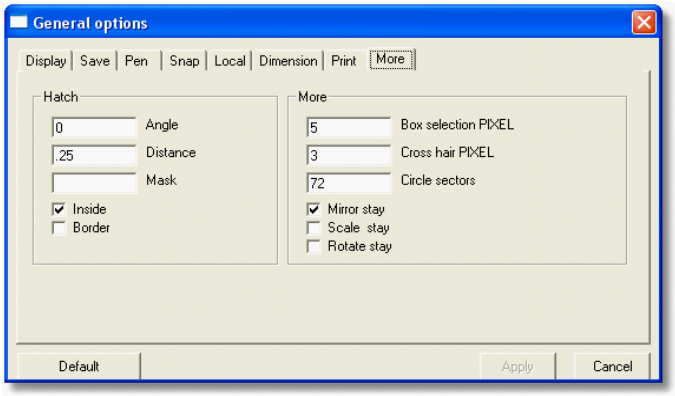
© 1993-2026 Gardos Software LTD

7. New drawing printing properties are specified.



- → **'Print to black' :** Specifies if the drawing's printing color will be black.
- → **'Print all drawing' :** Specifies if the whole drawing is to be printed.
- → **'Print at paper limits' :** Specifies if drawing printing will be within the chosen paper limits..
- → **'Landscape' :** Specifies if the position of the paper during the printing will be horizontal.
- → **'Printer pixel width in mm' :** Specifies if objects' width printing by scale will be active.
- → **'Print pixel's width in mm' :** Specifies the pixel printing width of the drawing in mm, if the choice "Printer pixel width in mm' is selected.
- → **'Print factor X' :** Specifies the scale factor at X.
- → **'Print factor Y' :** Specifies the scale factor at Y.
- → **'Paper margin at mm' :** Specifies the margin perimetrically of the drawing in mm, so as not to be printed at the edge.
- → **'Paper RGB color' :** Specifies the color of the paper to be printed.
- → **'1 printing unit at mm' :** Specifies the real printing dimension - correspondence to the drawing unit (e.g. if we choose that 1 drawing unit = 10 mm, the drawing will be printed by scale 1:100).
- → **'Paper size' :** Specifies the size of the printing paper in mm.
- → **'Paper width at mm' :** Specifies the width of the printing paper in mm.
- → **'Paper height at mm' :** Specifies the height of the printing paper in mm.

8. Several other properties are specified.



- → **'Hatch'**
  - → **'Angle' :** Specifies the hatch angle (e.g. 45o).
  - → **'Distance' :** Specifies the distance between the hatch lines.
  - → **'Mask' :** Specifies the hatch shape.
  - → **'Inside' :** Specifies if the hatch will appear inside the chosen area.
  - → **'Border' :** Specifies if the border will appear in case of hatching outside the object.
- → **'More'**
  - → **'Box selection PIXEL' :** Specifies the number of pixel of the rectangular box selection.
  - → **'Crosshair PIXEL' :** Specifies the number of pixel of the crosshair.

© 1993-2026 Gardos Software LTD

- → **'Circle sections' :** Specifies the number of circle segments.
- → **'Mirror stay':** Specifies if the mirror command will be active.
- → **'Scale stay' :** Specifies if the scale command will be active.
- → **'Rotate stay' :** Specifies if the rotate command will be active.

1.3.9 Add-Ons

From this menu you can use all the installed add-ons.

1.3.10 Window

From this menu you can manage the drawings you are processing..

1.3.11 Help

11.1 Contents
Opens a window with the program manual.
Shortcut: F1
Command: HELP

11.2 Avax-CAD Web site
Opens the AVAX-CAD R3 Web site.
Command: WEB_SITE

11.3 Check for updates
Checks the web for updates.
Command: CHECK_FOR_UPDATES

11.4 About
Opens a window with details relevant to the program.
Command: ABOUT

1.4 COMMANDS

1.4.1 ALL

QUICK REFERENCE

By **ALL** command the whole drawing appears within the prompt limits.

SYNTAX

[Command] **ALL**

PROCEDURE

ALL command is useful to see the whole drawing by making only one movement, especially after a **ZOOM**.
Note
After using this command, the prompt limits are those defined by **USERLIMITS** or **AUTOLIMITS** commands.

ICON

© 1993-2026 Gardos Software LTD

MENU

View	Zoom all

RELEVANT COMMANDS

<AUTOLIMITS>, <USERLIMITS>

UNDO

NO

1.4.2 ANG

QUICK REFERENCE

By **ANG** command we can dimension an angle which is formed by two non-parallel straight segments (it is not necessary to be intersected).

SYNTAX

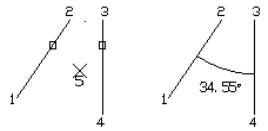
[Selection of the a' straight segment] only by **SELECT.**
[Selection of the b' straight segment] holding **Shift** and **SELECT.**

© 1993-2026 Gardos Software LTD

[Command] **ANG**
[Point of placing angle dimension]

PROCEDURE

Select only by **SELECT** (not **WINSEL** or **INWINSEL**) a straight segment and then another one, non-parallel, the angle of which we want..
Then, we give **ANG** command and the cursor of crosshair becomes a small circle (waiting for a point). We give a point where the circles arc will pass through and give us the angle dimension. If we give a point between the two lines, we will take the inside angle, otherwise we will take the outside angle.
Figure



We select segment 1-2 by **SELECT.**
We select segment 3-4 holding **Shift** .
and **SELEC.T**
We give **ANG.**
We give point 5.
The dimension arc of the inside angle and its value (e.g. 34.55 degrees) appears.

Note
The size and the text style, are depending of the active font type.

ICON



MENU

Draw	**Complex objects**	**Angle Dimensioning**

RELEVANT VARIABLES

Tools	**General Options**	**Dimension**

- → **'Decimals' :** Specifies the number of decimals of the dimension
- → **'Text scale' :** Specifies the size of dimension text.
- → **'Text font' :** Specifies the font type of the dimension text (e.g. Arial).
- → **'Tick size' :** Specifies the tick size (line, arrow, circle) which limits the dimension.
- → **'Scale' :** Specifies dimension size (text and arrow).
- → **'Angular format' :** Specifies the format of angles (e.g. degrees).

RELEVANT COMMANDS

<DIM>

© 1993-2026 Gardos Software LTD

UNDO

YES

1.4.3 ARC

QUICK REFERENCE

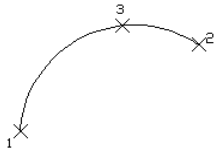
By **ARC** command we can draw a circle arc.

SYNTAX

[Command] **ARC**
[Point-1] The start-point of the chord of a circle.
[Point-2] The end-point of the chord of a circle.
[Point-3] The point on the arc.

PROCEDURE

3 points are being asked: The start and the end points of the chord of a circle and 1 point on the arc. The drawing direction is defined by the point on the arc.
Figure



Note
We can give the 3 points using one of the well-known methods (mouse, keyboard: absolute – relative co-ordinates).
We can, also call the **ARC** command, while we've already called and we are executing **POLYLINE** command, in order to draw an arc as segment of the polyline and then to continue drawing lines.

ICON



MENU

Draw	**Circle Arc**

RELEVANT VARIABLES

Tools	**General Options**	**More**

→ **'Arc segments' :** Specifies the number of arc segments.

© 1993-2026 Gardos Software LTD

RELEVANT COMMANDS

<POLYLINE>

UNDO

YES

1.4.4 AREA

QUICK REFERENCE

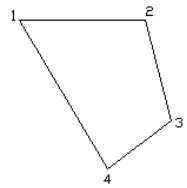
By this command is being calculated an objects area.

SYNTAX

[Creation of a sum of vertices or selection of a polygonal line]
[Command] **AREA**

PROCEDURE

First we choose the object we want to find its area. Then, we give **AREA** command and the calculation is being executed and appears the result.
Notice
The only shapes that give us its area, after we choose them are polyline , circle and ellipse. When we want to find other shapes area we have to select one by one all the vertices using **SELECTV** command (from menu: **Select -> SelectV-region creation**), instead o giving **SELECT** command, and then we give the **AREA** command. Vertices selection have to be with one direction (left or right).
Example
Figure



We select from menu **Select -> SelectV-region creation.**
Then we select the vertices with the direction as they are noted (1, 2, 3, 4).
Then we select from menu **Modify -> Measurement -> Area**
And appears a box with the calculated area.
If we want to erase the shape that has temporary appeared we give **REDRAW** command.

MENU

Modify	**Measurement**	**Area**

RELEVANT COMMANDS

<SELECTV>

© 1993-2026 Gardos Software LTD

UNDO

NO

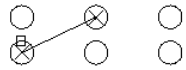
1.4.5 ARRAY

QUICK REFERENCE

By this command we draw a shape in one array (m x n) = (m) lines by (n) columns. That is to say the same shape m x n times in predefined distances at x and y.

SYNTAX

[Objects selection]
[Command] **ARRAY**
[--->Counter X=] We give the number of the shapes to be drawn in axis x.
[--->Counter Y=] We give the number of the shapes to be drawn in axis y.
[Reference point] We give the start point of a vector.
[End point] We give the end point of the vector.
The vector we give defines the dx and dy distances between the shapes.
Figure



Notes
(1) The chosen objects are not repeated in the position they have already been.
(2) Objects' drawing direction is the same as the vector's drawing.
(3) Vector's drawing is temporary. After the command's end, there is no vector anymore.

ICON



MENU

Modify	**Copy**	**Array**

RELEVANT COMMANDS

<COPY>

UNDO

YES

1.4.6 AUTOLIMITS

QUICK REFERENCE

AUTOLIMITS command aims to create the whole drawing vision on the prompt - regardless of how much stretched it is – in such a way, so that it can fit within the prompt's limits and have margins around it.

SYNTAX

[Command] **AUTOLIMITS**

PROCEDURE

© 1993-2026 Gardos Software LTD

We simply give **AUTOLIMITS** command and the drawing is being automatically redrawn and appears on prompt as centered as possible.
Details:
This command searches to find an imaginary rectangle which encloses the drawing. Then it places it (this and the enclosed drawing) in places relative to the down left and the upper right corner of the drawing area, such that they fit with the variables **"down left factor"** and **"upper right factor".**
So, if the real diagonal co-ordinates which enclose the drawing are (0.0) and (20.15), then the drawing will have width 20 and height 15. If the above factors (**"down left"** and **"upper right"**) are 0.5 then the distances will be DX=0.5*20=10 and DY=0.5*15=7.5.
So the rectangle that encloses the drawing will be distant DX=10 and DY=7.5 from the down left corner and DX>10 and DY=7.5 from the upper right corner. The whole drawing is placed into the imaginary rectangle which encloses the drawing and in this way we have a better vision. If we want of course another placing of the enclosed rectangle, we change the above factors from 0.5 to other values.

MENU

View	**Autolimits**

RELEVANT VARIABLES

"down left factor", "upper right factor" with default value 0.25

RELEVANT COMMANDS

<CALCEXTENDS>, <EXTENDS>, <USERLIMITS>, <ZOOM>

UNDO

NO

1.4.7 BEZIER

QUICK REFERENCE

A **BEZIER** curve is drawn based on a polygonal leader. The leader is defined with the selection of the vertices (by **SELECTV**) or with the selection of a polygonal line.

SYNTAX

[Creation of a sum of vertices / Selection of the polygonal line]
[Command] **BEZIER**
[CANCEL] (Cancelling in case of vertices' selection).

PROCEDURE

By **SELECTV** command we select the vertices of the polygonal leader at one direction (clockwise movement or counter clockwise) or select by **SELECT** command a **POLYLINE** that creates a polygon. Then we give **BEZIER** command, and a **BEZIER** curve is formed. After that, if it is necessary, we give **CANCEL** command to stop the process of selecting vertices.
Note
The polygonal leader doesn't have the ability to have more than twenty (20) vertices.

ICON



MENU

© 1993-2026 Gardos Software LTD

Draw	Complex objects	BEZIER curve

RELEVANT COMMANDS

<CBSPLINE>, <CSPLINE1>, <CSPLINE2>, <OBSPLINE>

UNDO

YES

1.4.8 BLOCK

QUICK REFERENCE

By **BLOCK** command we can group some objects into a new , united and complicated object, which – after we give it a name – we can use in our drawing or in other drawings, as an object of our library. An object like this could be a door, a window or anything else that would be useful.

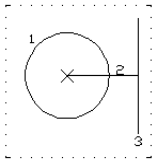
© 1993-2026 Gardos Software LTD

SYNTAX

[Objects selection]
[Command] **BLOCK**
[---> BLOCK name:] We give BLOCK'S name, up to 8 letters.
[Insertion-reference point:] We give the reference point of the BLOCK, useful for its insertion in drawings.

PROCEDURE

We select the objects we are interested in and we give **BLOCK** command. Then we give the name, by which we will call the **BLOCK** in the future. At the end, we define a point, usually on one of the objects of the **BLOCK** (not necessarily), based on which, all **BLOCK** objects will be drawn to its every insertion (by **INSERT** command).
Figure



We select the 3 objects, we give **BLOCK** command, by name e.g.. CIRC1L2 and insertion point the center of the circle (by **MIDPOINT** command on the circle).

Notes
(1) **BLOCK** acts as an object. This means that if we try to select one of the objects that it consists of, the whole **BLOCK** (**SELECT** command, **WINSEL** e.t.c.) will be selected. If something like this is undesired, then we can break it to its pieces by **EXPLODE** or **EXPLODE_BLOCK** command.
(2) **BLOCKS** are common files with the ending BLK and they are saved in the directory AVXLIB (the same for all drawings) e.g. if a drawing contains a BLOCK "B1" then a B1.BLK file is created which is saved in the directory AVXLIB (library)

MENU

Draw	**Insert Block**

RELEVANT COMMANDS

<BLOCK_LABEL>, <EXPLODE>, <EXPLODE_BLOCK>, <INSERT>

UNDO

YES

1.4.9 BMP_OUT

QUICK REFERENCE

By this command we can create a BMP file type which can contain our drawing and thus enable it to be read by other programs e.g. PAINT, CORELDRAW®, MS-WORD® e.t.c. in WINDOWS®. The BMP file is an image file.

SYNTAX

[Command] **BMP_OUT**

PROCEDURE

We give **BMP_OUT** command and on the dialog box that appears we write a name. The program creates, based on picture saving parameters (picture width and height), a picture file BMP type with the above name, that contains the current drawing.

© 1993-2026 Gardos Software LTD

RELEVANT VARIABLES

Tools	General Options	Save

- → **'Exported picture width' :** Defines the width of the exported picture in pixel.
- → **'Exported picture height' :** Defines the height of the exported picture in pixel.
- → **'Exported picture color" :** Defines the background color of the exported picture.

RELEVANT COMMANDS

<DXF_OUT>, <PIC_OUT>, <PLT_OUT>

UNDO

NO

1.4.10 BORDER

QUICK REFERENCE

By this command we can automatically draw a parallelogram rectangle around objects, so that it includes them precisely.

SYNTAX

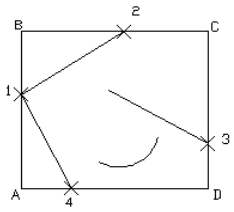
[Creation of a sum of vertices / Selection of the polygonal line]

© 1993-2026 Gardos Software LTD

[Command] **BORDER**
[CANCEL] (to stop vertices creation).

PROCEDURE

We select (in one direction) the most distant object vertices, around on which we want to draw the border, and then we give **BORDER** command (from the keyboard or from menu with selections **SelectV–Region creation**).
We cancel the command action by **CANCEL** command. Optionally we give **REDRAW** command.
Figure



We select from menu **SelectV–Region creation** and we select 1, 2, 3, 4 vertices.
Then we give **BORDER** command and at the end **CANCEL** command.
The parallelogram rectangle ABCD is automatically drawn.

ICON



MENU

Draw	**Complex objects**	**Border**

RELEVANT COMMANDS

<SELECTV>

UNDO

YES

1.4.11 BREAK

QUICK REFERENCE

It erases part of already selected line or splits this line in two by a temporary line given by the user.

SYNTAX

[Object selection]
[Command] **BREAK**
[Point-1] We give the first point of the temporary line.

© 1993-2026 Gardos Software LTD

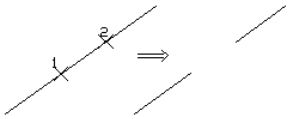
[Point-2] We give the second point of the temporary line or we cancel by CANCEL command , so only the first point takes part in the procedure. In case we give second breaking point different from the first one, the line segment that is defined by these two points is erased.

PROCEDURE

We select the line and give the command, then we give the one or the two points. With reference to the points selection, the program has the default way of selection **"NEAREST".**

Example

Figure



We select the line segment, we give **BREAK.**
We select points 1 and 2.
The line segment between points 1 and 2 is being erased.

MENU

Modify	Union - Split	Break line

UNDO

YES

1.4.12 CALCEXTENDS

QUICK REFERENCE

This command gives us a new calculation of drawing limits and the whole drawing is being drawn in such a way as to fit within prompts limits.

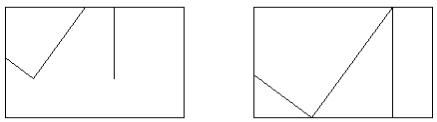
SYNTAX

[Command] **CALCEXTENDS**

PROCEDURE

After we have finished a part of our drawing, we can see its whole view within prompts limits.

Figure



© 1993-2026 Gardos Software LTD

Note
Simple command, that calculates the limits , which **EXTENDS** command will call back later (for more speed).

RELEVANT COMMANDS

<EXTENDS>

UNDO

NO

1.4.13 CANCEL

QUICK REFERENCE

By this command we cancel another command, which is active at the moment we give **CANCEL** command.

SYNTAX

[Command] **CANCEL**

PROCEDURE

By this command we cancel all actions and unselect all selected objects, if there are any.
The kind of command we are canceling is in proportion with the result E.g. when we are drawing by LINE command, command will stop execution and the line we were going to draw in the next step will not be drawn, but lines that we have drawn until now will remain the same. (Let's note that giving LINE command
only once, we can draw several lines until we stop). Generally, the next step we are going to take is cancelled. .
Cancel command can be activated from keyboard by Esc key.

MOUSE

Right key -> Cancel

UNDO

NO

1.4.14 CBSPLINE

QUICK REFERENCE

In order to draw **CBSPLINE** curve we use a polygonal leader as a basis. The leader is set by the vertices selection (by **SELECTV**) or by selection of a polygonal line.

SYNTAX

[Creation of a sum of vertices / Selection of the polygonal line]
[Command] **CBSPLINE**
[CANCEL] (Cancelling in case of vertices selection).

PROCEDURE

By **SELECTV** command we select the vertices of the polygonal leader at one direction (clockwise movement or counter clockwise) or

© 1993-2026 Gardos Software LTD

select by **SELECT** command a **POLYLINE** that creates a polygon. Then we give **CBSPLINE** command, and a **CBSPLINE** curve is created. After that, if it is necessary, we give **CANCEL** command to cancel the action of selecting vertices.
Note
The polygonal leader cannot have more than twenty (20) vertices.

ICON



MENU

Draw	**Complex objects**	**Closed B-SPLINE**

RELEVANT COMMANDS

<BEZIER>, <CSPLINE1>, <CSPLINE2>, <OBSPLINE>

UNDO

YES

1.4.15 CENTROID

QUICK REFERENCE

It creates a new point in the center of a selected closed polyline or a temporary polygon. The definition of this region takes place by selecting vertices or a circle or ellipse **POLYLINE.**

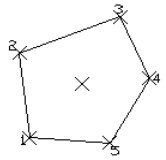
© 1993-2026 Gardos Software LTD

SYNTAX

[Creation of a sum of vertices / Selection of the polygonal line]
[Command] **CENTROID**
[CANCEL] (Cancelling in case of vertices selection).

PROCEDURE

By **SELECTV** command we select the vertices of the region the center of which we want to find. Then we give **CENTROID** command and a point (whose appearance depends on **"point type"**) is automatically placed in the center of the region. After that we give **CANCEL** command to cancel the action of selecting vertices.
Figure



We give **SELECTV**
We select 1, 2, 3, 4, 5 vertices
We give **CENTROID**
We give **CANCEL**

Note
The same thing will happen if we select a **POLYLINE** with no intersecting segments, instead of vertices selection.

ICON



MENU

Draw	**Complex objects**	**Centroid**

RELEVANT VARIABLES

Tools	**General Options**	**Pen**

→ **'Point type" :** Defines the point type appearance (e.g. cross +)

RELEVANT COMMANDS

<RADIATE>

UNDO

YES

1.4.16 CHANGE_COLOR

QUICK REFERENCE

By this command we have the ability to change the color of certain objects.

SYNTAX

[Object(s) selection]
[Command] **CHANGE_COLOR**
[---> color number=] We select the color we want from the colors table.

PROCEDURE

© 1993-2026 Gardos Software LTD

We select the objects the color of which we want to change. Then we give **CHANGE_COLOR** command (using menu or mouse). A dialog box appears where we can see the active color of the objects we have chosen. Selecting the arrow at the right side a color table appears. We select the color we want, and then **[OK]** to complete the command.

MENU

Modify	**Change**	**Color**

RELEVANT VARIABLES

Tools	**General Options**	**Pen**

→ **'Color' :** Defines the layer's color.

Tools	**General Options**	**Display**

→ **'AVAX desktop color' :** Defines desktop's color.
→ **'Ruler color' :** Defines ruler's color.
→ **'Rulers pen color' :** Defines ruler's pen color.
→ **'Command line color' :** Defines command line color.
→ **'Command lines pen color' :** Defines command line pen color.

RELEVANT COMMANDS

<CHANGE_LAYER>, <CHANGE_LTYPE>, <CHANGE_PTYPE>, <CHANGE_TEXT>, <CHANGE_FONT>, <CHANGE_WIDTH>, <CHANGE_3DELEVATION>, <CHANGE_3DHEIGHT>

UNDO

YES

1.4.17 CHANGE_FONT

QUICK REFERENCE

By **CHANGE_FONT** command, we can change a drawing's text font, after selecting it
SYNTAX

[Select text] We select the text whose font we want to change.
[Command] **CHANGE_FONT**
[---> a/a font=] Using consecutive number, we select the font we want, from a list of available fonts.

PROCEDURE

First we select the text by command **SELECT**. Then we give **CHANGE_FONT** command. A dialog box appears where we can see the active font. Selecting arrow at the right side a list with the available font and their consecutive number appears. We select the consecutive number we want and then **[OK]** to complete the command. (All the available programs fonts are at **<APPENDIX C>**).

MENU

Modify	**Change**	**Font**

RELEVANT VARIABLES

Tools	**General Options**	**Pen**

→ **'Font' :** Defines the text font (e.g. Arial).

© 1993-2026 Gardos Software LTD

→ **'Font height' :** Defines the font height.

Tools	**General Options**	**Dimension**

→ **'Font' :** Defines the dimension text font (e.g.. Arial).
→ **'Text height' :** Defines the size of the dimension text.

RELEVANT COMMANDS

<CHANGE_COLOR>, <CHANGE_LTYPE>, <CHANGE_PTYPE>, <CHANGE_LAYER>
<CHANGE_TEXT>, <CHANGE_WIDTH>, <CHANGE_3DELEVATION>, <CHANGE_3DHEIGHT>

UNDO

YES

1.4.18 CHANGE_LAYER

QUICK REFERENCE

This command gives us the ability to move an object from one drawing layer (LAYER) to another one, so that we can organize our work better.

SYNTAX

[Object(s) selection]
[Command] **CHANGE_LAYER**
[--->layer=] We give the a/a of the layer (LAYER).

PROCEDURE

First we select all objects whose layer we want to change. We give **CHANGE_LAYER** command (using menu or mouse). A dialog box appears where we can select the a/a of the layer (0 to 255) where our objects will be moved. After commands end our objects have been moved to the new layer.

Note
The need of moving objects to LAYERS, depends on how efficiently we want to organize our drawing. So we can imagine every LAYER, as one drawing layer where we draw some objects with a (probably) logical relation between them, e.g. to LAYER 0 walls, to LAYER 1 doors-windows, to LAYER 2 hydraulic installation, to LAYER 3 electric installation e.t.c.
Then by **FREEZE** command we can hide ("freeze") a LAYER, so that we can see every time, only the things we are interested in. Later by **THAW** command we can bring forward all LAYERS we have hidden.

MENU

Modify	**Change**	**Layer**

RELEVANT COMMANDS

<CHANGE_COLOR>, <CHANGE_LTYPE>, <CHANGE_PTYPE>, <CHANGE_TEXT>,
<CHANGE_FONT>, <CHANGE_WIDTH>, <CHANGE_3DELEVATION>, <CHANGE_3DHEIGHT>, <FREEZE>, <THAW>

UNDO

YES

© 1993-2026 Gardos Software LTD

1.4.19 CHANGE_LTYPE

QUICK REFERENCE

By **CHANGE_LTYPE** command, we can change line type.

SYNTAX

[Objects selection] We select the lines whose type we want to change.
[Command] **CHANGE_LTYPE**
[--->line type=] We select the line type we want, from a list of line types.

PROCEDURE

First we select the lines whose type we want to change, (relevant commands : **SELECT**, **WINSEL**, **INWINSEL**). We remind that if we have selected more objects than we want, we can unselect some of them, using **SELECTSUB** and **SELECTADD** commands. Then we give **CHANGE_LTYPE** command and a dialog box appears where we can see the line type that is active. By selecting the arrow at the right side a list with all the available line types appears. We select the one we want and then **[OK]** to complete the command.

MENU

Modify	**Change**	**Line Type**

RELEVANT VARIABLES

Tools	**General Options**	**Pen**

→ **'Line type' :** Defines the type in which the line will appear. (e.g.. SOLID).

RELEVANT COMMANDS

<CHANGE_COLOR>, <CHANGE_PTYPE>, <CHANGE_LAYER>, <CHANGE_TEXT>, <CHANGE_FONT>, <CHANGE_WIDTH>, <CHANGE_3DELEVATION>, <CHANGE_3DHEIGHT>

UNDO

YES

© 1993-2026 Gardos Software LTD

1.4.20 CHANGE_PTYPE

QUICK REFERENCE

By **CHANGE_PTYPE** command, we can change the point type.

© 1993-2026 Gardos Software LTD

SYNTAX

[Points selection] We select the points whose type we want to change.
[Command] **CHANGE_PTYPE**
[--->point type=] We select the point type we want, from a list of point types

PROCEDURE

First we select the points whose type we want to change, (relevant commands : **SELECT**, **WINSEL**, **INWINSEL**). We remind that if we have selected more objects than we want, we can unselect some of them, using **SELECTSUB** and **SELECTADD** commands. Then we give **CHANGE_PTYPE** command and a dialog box appears where we can see the point type that is active. By selecting the arrow at the right side a list with all the available point types appears. We select the one we want and then **[OK]** to complete the command.

MENU

Modify	**Change**	**Point type**

RELEVANT VARIABLES

Tools	**General Options**	**Pen**

→ **'Point type' :** Defines the type of appearance of the points (e.g. cross +).

RELEVANT COMMANDS

<CHANGE_COLOR>, <CHANGE_LTYPE>, <CHANGE_LAYER>, <CHANGE_TEXT>, <CHANGE_FONT>, <CHANGE_WIDTH>, <CHANGE_3DELEVATION>, <CHANGE_3DHEIGHT>

UNDO

YES

1.4.21 CHANGE_TEXT

QUICK REFERENCE

By **CHANGE_TEXT** command, we can change a drawing text (which we had written e.g. by **TEXT** command).

SYNTAX

[Text selection] We select the text we want to change.
[Command] **CHANGE_TEXT**
[Action] A window with the text appears. We change the text.

© 1993-2026 Gardos Software LTD

[Escape] We exit by using [ESC].

PROCEDURE

First we select the text we want to change (e.g. by **SELECT** command). Then we give **CHANGE_TEXT**, command and a window with the text appears. It is a small text processor, where except for correcting the existing text lines, we can insert new lines or erase old text lines. So we have an important ability to edit the drawing texts. We exit using [ESC]

MENU

Modify	**Change**	**Text**

RELEVANT COMMANDS

<CHANGE_COLOR>, <CHANGE_LTYPE>, <CHANGE_PTYPE>, <CHANGE_LAYER>
<CHANGE_FONT>, <CHANGE_WIDTH>, <CHANGE_3DELEVATION>, <CHANGE_3DHEIGHT>, <TEXT>

UNDO

YES

1.4.22 CHANGE_WIDTH

QUICK REFERENCE

By this command we can change the objects' line width.

SYNTAX

[Object(s) selection]
[Command] **CHANGE_WIDTH**
[--->width=] We give the line's width in prompts pixels
(typical width=1).

PROCEDURE

We select the objects whose line width we want to change and we give **CHANGE_WIDTH** command (also we can give it by mouse from menu). At the dialog box that appears, we give the line width as a number of pixels. Typically the width is what has been saved in the variable ***"Pen Width"*** and it is one pixel (=1). Usually 2 or 3 pixels are enough.

© 1993-2026 Gardos Software LTD

MENU

Modify	Change	Width

RELEVANT VARIABLES

Tools	General Options	Pen

→ **'Pen width' :** Defines pen width.

RELEVANT COMMANDS

<CHANGE_COLOR>, <CHANGE_LTYPE>, <CHANGE_PTYPE>, <CHANGE_TEXT>, <CHANGE_FONT>, <CHANGE_LAYER>, <CHANGE_3DELEVATION>, <CHANGE_3DHEIGHT>

UNDO

YES

1.4.23 CLOSE

QUICK REFERENCE

By this command we can close a polygonal line in progress, that is draw a straight line from the last to the first point – which we had given at the start of our drawing by **LINE** or **POLYLINE** command.

SYNTAX

[Command] **CLOSE**

PROCEDURE

We give **CLOSE** command as we draw lines and the program connects the last point with the first point with a straight line. Then that command terminates the process of **DRAW** command (as if we have given **CANCEL** command).

ICON



MENU

Object snap	Close Polyline

UNDO

YES

1.4.24 CIRCLE

QUICK REFERENCE

By this command we draw a circle, giving its center and radius.

SYNTAX

[Command] **CIRCLE**

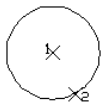
© 1993-2026 Gardos Software LTD

[Point-1 = circles center] We give the first point as center.
[Point-2 = point of perimeter] We give the second point, and if this belongs to the perimeter of the circle, it also defines its radius.

PROCEDURE

A very simple command. We give **CIRCLE** command, then a point as circle center, using the mouse or keyboard, and then the second point which belongs to the perimeter. Here we can use the mouse to select a point that is in relation with other objects using **"ENDPOINT"**, **"NEAREST"**, **"VERTICAL"** e.tc. (relevant commands **ENDPOINT**, **NEAREST, VERTICAL** e.t.c.). Also we can give directly the radius using the keyboard. For example as follows # 0,1 (e.g. angle = 0, radius = 1).
Drawing the circle using the mouse, as we move it we can see it constantly changing size, until we give the second point.

Figure



ICON



MENU

Draw	**Circle**

RELEVANT COMMANDS

<CIRCLE2P>, <CIRCLE3P>, <ELLIPSE>

UNDO

YES

1.4.25 CIRCLE2P

QUICK REFERENCE

By this command we draw a circle, using the ends of its diameter.

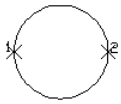
SYNTAX

[Command] **CIRCLE2P**
[Point-1] We give the first diameter point.
[Point-2] We give the second diameter point.

PROCEDURE

Simple command. We give **CIRCLE2P** command, then the first end-point of its diameter, either using mouse or keyboard, and then we repeat the same with diameter second end-point. Program draws the circle.

Figure



We give CIRCLE2P command.
We give the point 1 and then the point 2.

© 1993-2026 Gardos Software LTD

ICON



MENU

Draw	Circle from 2 points

RELEVANT COMMANDS

<CIRCLE>, <CIRCLE3P>, <ELLIPSE>

UNDO

YES

© 1993-2026 Gardos Software LTD

1.4.26 CIRCLE3P

QUICK REFERENCE

By this command we draw a circle, giving three points of its perimeter.

© 1993-2026 Gardos Software LTD

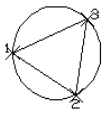
SYNTAX

[Command] **CIRCLE3P**
[Point-1] first point of circle's perimeter.
[Point-2] second point of circle's perimeter.
[Point-3] third point of circle's perimeter.

PROCEDURE

Simple command. We give the command and then the three points from which we want the circle to pass. We give the points in the same way, either using keyboard or mouse, and they can be in absolute or relative co-ordinates.

Figure



We give CIRCLE3P command
We give points 1, 2 and 3
The circle Is being created, to the imaginary triangle with sides 1-2, 2-3, 3-1.

ICON



MENU

Draw	**Circle from 3 points**

RELEVANT COMMANDS

<CIRCLE>, <CIRCLE2P>, <ELLIPSE>

UNDO

YES

1.4.27 COPY

QUICK REFERENCE

By this command we can copy objects that have been drawn, from one part of the drawing to another.

SYNTAX

[Object(s) selection]
[Command] **COPY**
[Point-1] We give the a' point of the moving vector.
[Point-2] We give the b' point of the moving vector.

© 1993-2026 Gardos Software LTD

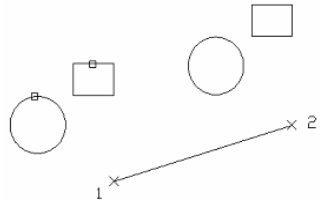
[Point-3] We give a new point. The previous one is the start and the current is the end of the moving vector.

[Point-N] ...
[CANCEL] Process canceling.

PROCEDURE

First we select the objects we want to copy. We give **COPY** command. The copy will take place in a specific distance and by a relative angle from the selected objects.
These two components (distance and angle)emerge from the direction and the length of one imaginary vector, which we set as follows: After giving the command we also give the start and the end point of this vector (moving vector) and the objects are being drawn in the new position. We give, if we want, a new point and the objects are being copied there also. We continue giving more points – or we stop by giving **CANCEL**.

Figure



We select the circle and the rectangle. We give **COPY** and then points 1 and 2. Objects are being copied in the new position. We give **[ESC]** or **CANCEL** to stop copying.
Note
This command is similar to **MOVE** command. With **COPY** command, however, along with the copy the original object is maintained as well and we can also have more than one copies of the object.

ICON

MENU

Modify	**Copy**	**Copy**

RELEVANT COMMANDS

<ARRAY>, <MOVE>

UNDO

YES

1.4.28 COPY_IN

QUICK REFERENCE

By this command we can copy an area's outline in a specific distance from it to the interior of this outline.

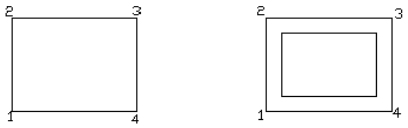
SYNTAX

[Vertices creation / Polygonal line selection]
[Command] **COPY_IN**
[--->Distance=] We give the distance inside the area.
[CANCEL] for canceling vertices selection.

© 1993-2026 Gardos Software LTD

PROCEDURE

For example we have a rectangle as in the following figure:
Figure



We give **SELECTV** command and we select (using e.g. **ENDPOINT**) 1, 2, 3, and 4 points. Then we give **COPY_IN** command. A dialog box appears where the program asks for a distance (from the shape's sides) to be copied inside. Let us give 0.2. The rectangle is being copied as you can see in the figure.

Note
The figure that is being drawn inside, is now a **POLYLINE** (ibeing now one object). We can break it to its pieces using **EXPLODE** command.

ICON



MENU

Modify	**Copy**	**Copy offset in**

RELEVANT COMMANDS

<COPY>, <COPY_OUT>, <LASTSV>, <MAKESV>, <SELECTV>

UNDO

YES

1.4.29 COPY_OUT

QUICK REFERENCE

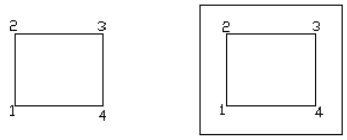
By this command we can copy an area's outline in a specific distance from it to the exterior of this outline.

SYNTAX

[Vertices creation / Polygonal line selection]
[Command] **COPY_OUT**
[--->Distance=] We give the distance outside the area.
[CANCEL] for canceling vertices selection.

PROCEDURE

For example we have a rectangle as in the following figure:
Figure



© 1993-2026 Gardos Software LTD

We give **SELECTV** command and we select (using e.g. **ENDPOINT**) 1, 2, 3, and 4 points. Then we give **COPY_OUT** command. A dialog box appears where the program asks for a distance (from the shape's sides) to be copied outside. Let us give 0.2. The rectangle is being copied as you can see in the figure.

Note
The figure that is being drawn outside, is now a **POLYLINE** (being now one object). We can break it to its pieces using **EXPLODE** command.

ICON



MENU

Modify	**Copy**	**Copy offset out**

RELEVANT COMMANDS

<COPY>, <COPY_OUT>, <LASTSV>, <MAKESV>, <SELECTV>

UNDO

YES

1.4.30 CSPLINE1

QUICK REFERENCE

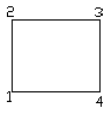
By this command we can copy an area's outline in a specific distance from it to the exterior of this outline.

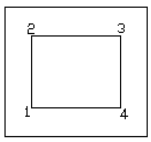
SYNTAX

[Vertices creation / Polygonal line selection]
[Command] **COPY_OUT**
[--->Distance=] We give the distance outside the area.
[CANCEL] for canceling vertices selection.

PROCEDURE

For example we have a rectangle as in the following figure:
Figure





We give **SELECTV** command and we select (using e.g. **ENDPOINT**) 1, 2, 3, and 4 points. Then we give **COPY_OUT** command. A dialog box appears where the program asks for a distance (from the shape's sides) to be copied outside. Let us give 0.2. The rectangle is being copied as you can see in the figure.

Note
The figure that is being drawn outside, is now a **POLYLINE** (being now one object). We can break it to its pieces using **EXPLODE** command.

ICON

© 1993-2026 Gardos Software LTD



MENU

Modify	Copy	Copy offset out

RELEVANT COMMANDS

<COPY>, <COPY_OUT>, <LASTSV>, <MAKESV>, <SELECTV>

UNDO

YES

1.4.31 CSPLINE2

QUICK REFERENCE

By this command we can draw a cubic Spline curve with free ends condition, using a polygonal leader (adjoining vertices). The leader's setting takes place with the vertices' selection (by **SELECTV**) or with the selection of a polygonal line.

© 1993-2026 Gardos Software LTD

SYNTAX

[Vertices selection / Polygonal line selection]
[Command] **CSPLINE2**
[CANCEL] (Cancelling in case of vertices selection).

PROCEDURE

By **SELECTV** command we select the vertices of the polygonal leader at one direction (clockwise movement or counter clockwise) or select by **SELECT** command a **POLYLINE** that creates a polygon. Then we give **CSPLINE2** command, and a **CSPLINE2** curve with free ends condition is created. After that, if it is necessary, we give **CANCEL** command to cancel the action of selecting vertices.
Note
The polygonal leader cannot have more than twenty (20) vertices.

ICON


MENU

Draw	**Complex objects**	**Cubic SPLINE2**

RELEVANT COMMANDS

<BEZIER>, <CBSPLINE>, <CSPLINE1>, <OBSPLINE>

UNDO

YES

1.4.32 DIM

QUICK REFERENCE

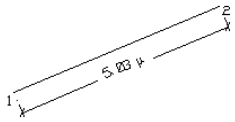
By **DIM** command, we can dimension a straight line segment.

SYNTAX

[Straight line segment selection]
[Command] **DIM**
[--->Distance=] We give the distance from the line segment, where the dimension will appear.
[Point – sign of the direction] We give a point towards the side we want the dimension to be drawn.
[CANCEL] Canceling.
or
[Vertices creation / Polygonal line selection]

© 1993-2026 Gardos Software LTD

[Command] **DIM**
[--->Distance=] We give the distance. Dimensions of polygonal sides will automatically appear outside the polygon.
[CANCEL] (Cancelling in case of vertices' selection).
Example
Figure



We want segments 1.2 dimension, towards the right side and at distance 0.5 units.
-We select segment 1-2
-We give **DIM**
-At the dialog box where the distance is being asked we give 0.5
-We give a point towards the right side of segment 1-2
-We cancel other objects' selection by **CANCEL** command.

ICON

MENU

Draw	**Complex objects**	**Dimension**

RELEVANT VARIABLES

Tools	**General Options**	**Dimension**

- → **'Arrow size' :** Specifies the dimension arrow size (to length).
- → **'Arrow type' :** Specifies the arrow type (0 – line, 1 – arrow, 2 – circle).
- → **'Decimals' :** Specifies the number of decimal of the dimension.
- → **'Text' :** Specifies the dimension text.
- → **'Distance' :** Specifies the dimension of distance from the straight line segment that is being dimensioned.
- → **'Factor' :** Specifies the dimension factor increase.
- → **'Inner lines' :** Specifies the protrusion of lines that define the dimension, towards the straight line segment side.
- → **'Outer lines' :** Specifies the protrusion of lines that define the dimension, towards the straight line segment outside.
- → **'Text above' :** Specifies if the text will appear above the dimension line.
- → **'Horizontal text' :** Specifies if the text will appear horizontal.
- → **'Vertical text' :** Specifies if the text will appear vertical.
- → **'Text scale' :** Specifies the size of dimension text.
- → **'Text font' :** Specifies the font type of the dimension text (e.g. Arial).
- → **'Tick size' :** Specifies the tick size (line, arrow, circle) which limits the dimension.
- → **'Scale' :** Specifies dimension size (text and arrow).
- → **'Angular format' :** Specifies the format of angles (e.g. degrees).
- → **'Dimension unit format' :** Specifies the format of dimensions (e.g. decimal).
- → **'Fractional tick' :** Specifies the fractional tick when dimensions appear in fractional form.

© 1993-2026 Gardos Software LTD

RELEVANT COMMANDS

<ANG>

UNDO

YES

1.4.33 DIVIDE

QUICK REFERENCE

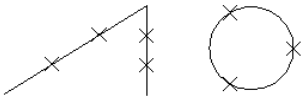
By **DIVIDE** command we can divide (imaginarily) an object (e.g. line or circle) and points (**NODES**) are placed on the equally – distanced subdivisions. Then we can use these points as reference points for further drawing.

SYNTAX

[Object selection]
[Command] **DIVIDE**
[--->Divider=] We give the number of subdivisions.

PROCEDURE

We select objects and give **DIVIDE** command. A dialog box appears, where we give the number of subdivisions N. N-1 points appear on every straight line segment, while N points appear on every circle or ellipse, (closed Shape).
Figure



We select 3 objects (2 line segments and 1 circle).
We give **DIVIDE** command and number of subdivisions 3. 2 points on every line segment and 3 points on the circle appear.

MENU

Modify	**Union-Split**	**Divide**

RELEVANT COMMANDS

< >

UNDO

YES

1.4.34 DOWN - UP - LEFT - RIGHT

QUICK REFERENCE

By these commands we can move the drawing in a short distance down, up, left or right.

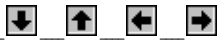
SYNTAX

[Command] **DOWN** or **UP** or **LEFT** or **RIGHT**

PROCEDURE

© 1993-2026 Gardos Software LTD

We give these commands using the mouse mainly and pressing the arrows on the scroll bars and the drawing is being moved in a short distance, (usually as long as a text character), down or up or left or right.
We can also give them using the keyboard with the movement arrows.

KEYS 

Arrows on the scroll bars

RELEVANT COMMANDS

<PGDN>, <PGUP>, <PGLEFT>, <PGRIGHT>

UNDO

NO

1.4.35 DXF_IN

QUICK REFERENCE

By this command we can import to our drawing another drawing which is in a DXF type file. These drawing files are created by other drawing programs, such as AUTOCAD of AUTODESK, the company that has established DXF type files. Today it is the most widespread way of communication of drawing programs.

SYNTAX

[Command] **DXF_IN**
[--->DXF file:] We select the file's name.

© 1993-2026 Gardos Software LTD

PROCEDURE

We give **DXF_IN** command and a window appears where we can select one of the available DXF files.

MENU

File	**Import**

RELEVANT COMMANDS

<DXF_OUT>, <EXPLODE>, <EXPLODE_BLOCK>

UNDO

YES

1.4.36 DXF_OUT

QUICK REFERENCE

By this command we can create a DXF file which will contain our drawing and will make it readable by other programs e.g. AUTOCAD® e. t.c. The DXF file is a drawing file.
In combination with **DXF_IN** command we can achieve a two - way communication with any drawing program (even with picture or text processing programs- usually within a WINDOWS® environment).

SYNTAX

[Command] **DXF_OUT**

PROCEDURE

We give **DXF_OUT** command and on the window that appears we write the file name we want. Program creates a DXF type drawing file with the above name, based on the file saving options. This file includes the current drawing.

MENU

File	**Export**

RELEVANT COMMANDS

<BMP_OUT>, <DXF_I>, <PIC_OUT>, <PLT_OUT>

© 1993-2026 Gardos Software LTD

UNDO

NO

1.4.37 ELLIPSE

QUICK REFERENCE

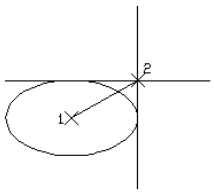
By this command we can draw an ellipse using its center and its two half-axis.

SYNTAX

[Command] **ELLIPSE**
[Point-1] Ellipse center.
[Point-2] Point over ellipse perimeter by which its x, y half-axis are indirectly defined.

PROCEDURE

We give **ELLIPSE** command and a point – its center. Then, while we move the crosshair, we notice the ellipse being continuously drawn and being tangent to the crosshairs axis. Distances between the first point and the crosshairs two axis are the half-axis of the ellipse. At the end we give the second point and the ellipse is being drawn. Points can be given – as we already know – using either the mouse or the keyboard and coordinates can be either absolute or relative.
Example
Figure



We give **ELLIPSE** command.
We give Point 1 and then Point 2.

Notes
(1) If we give equal half-axis then a circle is drawn.
E.g. **ELLIPSE**
[Point-1] 10,5
[Point-2] 1,1 (distance x = 1 – relative to 10,5 / distance y = 1 – relative to 10,5)
(2) If we are in condition **ORTHO_ON** then a circle is drawn again.

ICON



© 1993-2026 Gardos Software LTD

MENU

Draw	**Ellipse**

RELEVANT COMMANDS

<CIRCLE>, <CIRCLE2P>, <CIRCLE3P>

UNDO

YES

1.4.38 ENDPOINT

QUICK REFERENCE

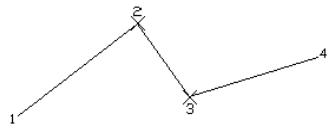
We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a line's segment). The purpose is the selection of the endpoint of a line segment – useful to the active command.

SYNTAX

[Command] e.g. **LINE**
[Command] **ENDPOINT**
[Point on the line segment] We give a point near a line segment's endpoint and on that line.
[Continue with the other command] e.g. **LINE**

Example

Figure



We want to connect vertices 2 and 3 with a line.
We give **LINE**, then **ENDPOINT**, select vertex 2, we give **ENDPOINT**, select vertex 3.
Press **CANCEL**.

ICON



MENU

Object Snap	**End**

RELEVANT COMMANDS

<CLOSE>, <INTERSEC>, <MIDPOINT>, <NEAREST>, <NODE>, <QUADRANT>, <TANGENT>, <VERTICAL>

© 1993-2026 Gardos Software LTD

UNDO

YES (Undoes the step of the active command).

1.4.39 ERASE

QUICK REFERENCE

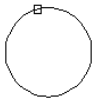
By **ERASE** command, we can erase objects.

SYNTAX

[Object(s) selection] We select the objects we want to erase.
[Command] **ERASE**

PROCEDURE

First we select all objects we want to erase from the drawing (relevant commands **SELECT**, **WINSEL**, **INWINSEL**). We remind that, if we have selected more or fewer objects than we want to erase, we can remove or add some of them, using **SELECTSUB** and **SELECTADD** commands, respectively.
Example
Figure



-We select the circle (using **SELECT** command)
-We give **ERASE** command

ICON



MENU

Edit	**Delete**	**Erase**

or

Modify	**Erase**

UNDO

YES

© 1993-2026 Gardos Software LTD

1.4.40 EXIT

QUICK REFERENCE

By this command the program's operation is terminated.

© 1993-2026 Gardos Software LTD

SYNTAX

[Command] **EXIT**

PROCEDURE

When we want to stop our work, we give **EXIT** command. Program prompts us to save our drawing – if something has changed to it. This is the **ONLY PERMISSIBLE WAY OF TERMINATING** our work (and not e.g. turning off our P/C, while our program is active).

MENU

File	**Exit**

KEYBOARD

Alt + F4

RELEVANT COMMANDS

<SAVE>, <SHELL>

UNDO

NO

1.4.41 EXPLODE

QUICK REFERENCE

By this command we can break a **POLYLINE** into its component parts, or break **BLOCKS**, into its component objects which can be lines or other simple objects or **POLYLINES** or even **BLOCKS** (case of complicated **BLOCKS**).
This command as regards **POLYLINES**, is the opposite of **PEDIT** command while as regards **BLOCKS it** is the opposite of **BLOCK** command.

SYNTAX

[POLYLINE or BLOCK selection] e.g using **SELECT** command
[Command] **EXPLODE**

© 1993-2026 Gardos Software LTD

PROCEDURE

We select a **POLYLINE** (e.g. a wall - see **WALL** command – or a rectangular parallelogram – see **RECTANGLE** command e.t.c.) or a **BLOCK**. We give **EXPLODE** command. The object remains the same on prompt, but if we now try to select it by **SELECT**, only one of its components that consist it will be selected, components that are now separated objects. On the contrary, if we want to join them to a **POLYLINE**, we select them and give **PEDIT** command. Now if, selecting one of these component objects, we notice that more objects are selected, then all those together consist another **POLYLINE** or **BLOCK** (inside the initial **BLOCK** or **POLYLINE**) and if we want to break it we continue using **EXPLODE** command as many times as it's necessary.

MENU

Modify	**Union-split**	**Explode**

or

Modify	**Union-split**	**Explode Block**

RELEVANT COMMANDS

<BLOCK>, <BREAK>, <DIVIDE>, <DXF_IN>, <EXPLODE_BLOCK>, <INSERT>, <POLYLINE>

UNDO

YES

1.4.42 EXPLODE_BLOCK

QUICK REFERENCE

This command is the opposite of **BLOCK** command. By **EXPLODE_BLOCK** command we can break a group of objects (**BLOCK**), to the component objects it consisted of (**BLOCK** command). Using this command, we cannot break a **POLYLINE** to its component objects.

SYNTAX

[Selection BLOCK]
[Command] **EXPLODE_BLOCK**

PROCEDURE

We select a **BLOCK** (e.g. using **SELECT**) and then we give **EXPLODE_BLOCK** command. Objects are remaining the same on prompt, but from now on every object is autonomous, that is if we select (e.g. by **SELECT**) one of them, the rest are not selected.
Notes
(1) If **BLOCK** consists of other **BLOCKS** (also simple objects) then after **EXPLODE_BLOCK**, these **BLOCKS** aren't broken. We have to repeat **EXPLODE_BLOCK** command to

© 1993-2026 Gardos Software LTD

Break them to their component objects. In that case (well known as complicated **BLOCKS**), using **EXPLODE_BLOCK** command we move one step every time, for breaking complicated to simple **BLOCKS**.

(2) **EXPLODE_BLOCK** command differs from **EXPLODE** command, to the point that the first one breaks, selectively, only **BLOCKS** and not **POLYLINES**.

RELEVANT COMMANDS

<BLOCK>, <BREAK>, <DIVIDE>, <DXF_IN>, <EXPLODE>, <INSERT>, <POLYLINE>

UNDO

NO

1.4.43 EXTEND

QUICK REFERENCE

BY **EXTEND** command we can extend line segments until they meet another line segment.

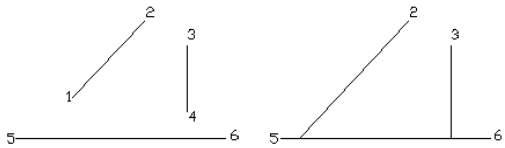
SYNTAX

[Objects selection] We select the line segments we want to extend.
[Command] **EXTEND**
[Objects-boundaries] We select the line segments UP TO WHICH, the line segments we have selected in the first step will be extended.
[Objects-boundaries selection / CANCEL] We continue (optionally) to select other line segments as in the previous step, or we stop further selection by [**CANCEL**].

PROCEDURE

First we select the line segments we want to extend. Then we give **EXTEND** command and we select the line segment up to which we want to extend the initially selected, line segments. At the end we stop selecting by [**CANCEL**].

Example
Figure



-We select segments 1-2 and 3-4
-We give **EXTEND** command
-We select segment 5-6
Segments 1-2 and 3-4 are being extended up to 5-6 line segment

© 1993-2026 Gardos Software LTD

ICON


MENU

Modify	**Deformation**	**Extend**

UNDO

YES

1.4.44 EXTENDS

QUICK REFERENCE

EXTENDS command has the intention to create the whole drawing vision on the prompt in such a way, so that it can fit within the prompt's limits precisely.

SYNTAX

[Command] **EXTENDS**

Note
It is possible that the whole drawing cannot fit within the prompt's limits after giving **EXTENDS** command. This may happen if at a certain point, we drew outside the limits we had set while using a previous **CALCEXTENDS** command.
In this case we repeat **CALCEXTENDS** command, and the new real limits of the drawing are recalculated. Then using **EXTENDS** command, the drawing will fit within the prompt's limits precisely.

RELEVANT COMMANDS

<AUTOLIMITS>, <CALCEXTENDS>, <USERLIMITS>,

ICON


MENU

View	**Extends**

UNDO

NO

1.4.45 FILLET

QUICK REFERENCE

By this command we can fillet two non-parallel line segments on their intersection point.
Segments are extended (in order to reach their intersection point) or cut in such a way that, after the command's execution only TWO line segments begin from their intersection point.

SYNTAX

[Object selection] We select (by SELECT command only) the a' line segment.

© 1993-2026 Gardos Software LTD

[Object selection] We select (by SELECT command only) the b' line segment.
[Command] **FILLET**
[CANCEL] Stop further selection of objects.
Example

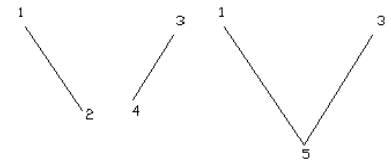
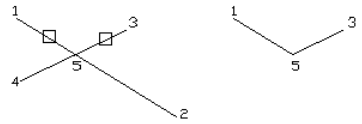


Figure A *Figure B*

We want to connect segments 1-2 and 3-4.
We select using **SELECT** command segment 1-2.
We select using **SELECT** command segment 3-4.
We give **FILLET** command.
We stop selection using **CANCEL**.
The result is on *Figure B*, with the two segments connected to the point 5.

Note
This command can also operate like **TRIM** command if the two segments are already intersected. In this case what matters is the selection point of each line segment. The side that is defined by this and intersection point, will remain after **FILLET**, and the other side will be cut.
Example



(Before FILLET) (After FILLET)

We want to **SELECT** points A and B.
After **FILLET**, segments1-5 and 3-5 are left.

ICON



MENU

Modify	**Deformation**	**Fillet**

RELEVANT COMMANDS

<TRIM>

UNDO

YES

1.4.46 FREEZE

QUICK REFERENCE

This command "**freezes**" a **LAYER** (drawing transparent), in such a way that all objects which belong to it become invisible. The opposite of this command is **THAW** command ("**thaw**" **LAYER**).
SYNTAX

[Command] **FREEZE** (freeze the current **LAYER**)

PROCEDURE

This command is very simple. We give **FREEZE** and it immediately freezes the current **LAYER** and all objects that belong to it become invisible. Later we can bring back the **LAYER** using **THAW** command.
Note
In order to freeze another **LAYER** (not the current one) we change the current **LAYER** to the **LAYER** we want to freeze (and make this current), then we give **FREEZE** command and we change again the current **LAYER** giving the initial. The change of the current **LAYER** takes place using **SET** command.

© 1993-2026 Gardos Software LTD

Example
The current **LAYER** is "0" and we want to freeze "1" layer.
We give **SET** command. We change **LAYER** 0 to 1, on the relevant dialog box.
We give **FREEZE** command. All objects of **LAYER-1** disappear from the prompt. (They aren't erased from the drawing, though).
We bring back the initial **LAYER**, giving **SET** command and the number "0" to the variable "**LAYER**", on the relevant dialog box

ICON



RELEVANT VARIABLES

Tools	General Options	Pen

- → **'Layer' :** We specify the drawing layer where drawing objects will belong
- → **'Color' :** We specify layer's color.

RELEVANT COMMANDS

<SET>, <THAW>

UNDO

YES

1.4.47 GRID

QUICK REFERENCE

By this command we create a grid of points on the prompt in DX, DY spaces between them, which we are going to use later, as a reference for objects drawing.

[Command] GRID
[--->Dx =
Dy =] At the dialog box we enter grid spacing (Where: Dx = horizontal spacing, Dy = vertical spacing).

PROCEDURE

Giving GRID command, a dialog box appears on prompt where we have to give two variables, Dx = horizontal spacing between grid points, and Dy = vertical spacing between the same grid points. As the command is executed the grid appears (its points). Regardless of whether it pre-existed or not, the grid is redrawn according to the new Dx, Dy.
Then, SNAP command and proper Dx, Dy, restrict the movement of crosshair to intervals already defined, in order to help us draw in an easier way.

ICON



RELEVANT VARIABLES

Tools	General Options	Snap

- → **'Grid' :** Specifies if the grid is active.
- → **'Grid by lines':** Specifies if the grid appears by lines.
- → **'Dx' :** Specifies the spacing between the points of the grid to axis X.
- → **'Dy' :** Specifies the spacing between the points of the grid to axis Y.

RELEVANT VARIABLES

© 1993-2026 Gardos Software LTD

<GRID_ON>, <GRID_OFF>, <SETGRID>, <SETSNAP>, <SNAP>

UNDO

NO

1.4.48 GRID_ON

QUICK REFERENCE

By this command the grid appears. If one or both of the values of DX, DY are equal to 0, then they are requested on the dialog box and the command operates exactly like GRID command. The opposite of this command is **GRID_OFF** command.

SYNTAX

[Command] **GRID_ON** Grid activation
If DX=0 or DY=0:
[--->DX =
DY =] At the dialog box we enter grid spacing). Where: DX = horizontal spacing, DY = vertical spacing.

PROCEDURE

By this command the grid appears. If one or both of the values of DX, DY are equal to 0, then they are requested on the dialog box and the command operates exactly like GRID command.
Otherwise the grid appears and the points have DX, DY spacing between them.

RELEVANT VARIABLES

Tools	General Options	Snap

→**'Grid' :** Specifies if the grid is active.
→**'Grid by lines':** Specifies if the grid appears by lines.
→**'Dx' :** Specifies the spacing between the points of the grid to axis X.
→**'Dy' :** Specifies the spacing between the points to the grid to axis Y.

RELEVANT VARIABLES

<GRID>, <GRID_OFF>, <SETGRID>, <SETSNAP>, <SNAP>

UNDO

NO

1.4.49 GRID_OFF

QUICK REFERENCE

This command turns off the grid .It is exactly the opposite of command GRID_ON.
DX, DY (horizontal and vertical spacing between the points of the grid) remain as variables, so that when we set the GRID MODE ON, the grid appears again in the previous form.

SYNTAX

[Command] **GRID_OFF** Grid turning off

© 1993-2026 Gardos Software LTD

PROCEDURE

We give **GRID_OFF** command and the grid disappears from prompt. We can reactivate the grid using **GRID_ON** command.

RELEVANT VARIABLES

Tools	General Options	Snap

→**'Grid' :** Specifies if the grid is active.
→**'Grid by lines':** Specifies if the grid appears by lines.
→**'Dx' :** Specifies the spacing between the points of the grid to axis X.
→**'Dy' :** Specifies the spacing between the points to the grid to axis Y.

RELEVANT VARIABLES

<GRID>, <GRID_ON>, <SETGRID>, <SETSNAP>, <SNAP>

UNDO

NO

1.4.50 HATCH

QUICK REFERENCE

By this command we can fill with a hatch pattern a surface that has been created by a **POLYLINE** (closed or open) or a region that has been created by vertices (using **SELECTV** – vertices selection).

SYNTAX

[surface selection]
[Command] **HATCH**
[--->Distance= Distance between hatching lines.
Angle= Angle of hatching lines (in degrees).
Inside= Internal or external hatching lines.
Border= Hatching border.

© 1993-2026 Gardos Software LTD

Mask: It is a sequence of numbers 1 and 0 in which the program creates or not hatching lines (e.*g* number "101" means that the program will not create the middle line every three lines).

PROCEDURE

We first select the surface we want to fill with hatching lines. This can be a **POLYLINE** or a region, which we create using **SELECTV** command (we select all vertices that define this region). Then we give **HATCH** command and at the dialog box that appears, we give all necessary data (or we accept the default). These data are the distance between the hatching lines, the angle of hatching lines (all of them parallel) and X axis, the internal or external hatching lines, (external for the hatching region and internal for the rectangle parallelogram which encloses the region), our purpose to create or not rectangle border which is going to enclose hatching lines and the mask, if we want to leave some of the hatching lines out or not.

Note

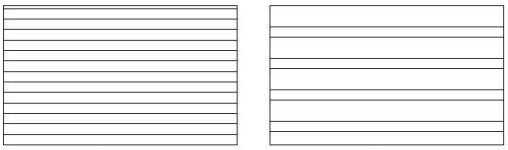
If we are asked for a border then a border appears which is identical to the border given by **RECTANGLE** command (rectangle, with parallel sides to X and Y axis). This is not, however, a **POLYLINE** object but 4 separated lines (**LINE** objects).

Example



Figure A *Figure B* *Figure C*

We have the **POLYLINE** of Shape A.
We select (e.g. by **SELECT**)
We give **HATCH**.
We give Distance = 0.25
" Angle = 0
" Inside = 1
" Border = EMPTY – We don't check the little box
And we have Shape B.
We give Mask = EMPTY (that means to draw all lines – Drawing 1) or e.g. 011 (that means not to draw 1 line and then to draw 2 lines – Drawing 2)



Drawing 1 *Drawing 2*

If we've given Border = YES – We check the little box and we will have Shape C.

ICON



© 1993-2026 Gardos Software LTD

MENU

Draw	Complex objects	Hatch

RELEVANT VARIABLES

Tools	General Options	More

→ 'Hatch'
- → **'Angle' :** Specifies the hatch lines angle (e.g. 45o).
- → **'Distance :** specifies the distance between hatch lines.
- → **'Mask' :** Specifies the style of hatch lines (see Drawing 1 & 2).
- → **'Inside' :** Specifies if the hatch lines appear inside the selected area.
- → **'Border' :** Specifies if the border will appear, in case of hatch lines outside the object. (see Figure C)

UNDO

YES

1.4.51 HELP

QUICK REFERENCE

This command provides us with explanatory help for every program's command, on prompt.

SYNTAX

[Command] **HELP**

PROCEDURE

The simplest command. We give **HELP** and on the window that opens information about the program appears. The first page that appears is that of contents where we can select the chapter we are interested in.

MENU

Help	Contents

KEYBOARD

F1

1.4.52 INFO

QUICK REFERENCE

By this command we can have information relevant to an object we select.

SYNTAX

[Object selection] We select the object we want information about.
[Command] **INFO**

PROCEDURE

© 1993-2026 Gardos Software LTD

We select (e.g. **SELECT**) the object we want information about and we give **INFO** command. Then the program gives us some information about this object such as the object's kind, the LAYER that belongs to, its color, its width and its type.

UNDO

NO

1.4.53 INSERT

QUICK REFERENCE

This command allows the insertion of drawing segments that have been saved using **BLOCK** command, or whole drawing files (that have been first transformed into a **BLOCK**).

SYNTAX

[Command] **INSERT**
[---> BLOCKS name:] We give BLOCK'S name.
[BLOCKS insertion point] We give BLOCK'S insertion point.

PROCEDURE

We give **INSERT** command. Then using mouse or keyboard we define a point into our drawing where we want the Block to be inserted. This point is being automatically correspondent to BLOCK'S insertion point (which is information of the Block itself and the point to which are referring all its objects – see **BLOCK** command).
Note
With **INSERT** command we can insert drawings and details to our drawing.
By this way we have the ability to have libraries of symbols as BLOCKS and to use them often assuring more productivity. These symbols can be anything such as furniture, apparatuses, plants e.t.c. Organization and name codification of these symbols is required of course so that their names can be easily recognized.

© 1993-2026 Gardos Software LTD

RELEVANT COMMANDS

<BLOCK>, <DXF_IN>, <EXPLODE>, <EXPLODE_BLOCK>, <INSERT_AVX>

UNDO

YES

1.4.54 INSERT_AVX

QUICK REFERENCE

By this command we can insert a whole drawing (not a BLOCK) to another drawing.

SYNTAX

[Command] **INSERT_AVX**
[--->BLOCKS name :] We give the drawing file name (without AVX extension)
[Insertion Point] We give a point to the current drawing, where the other drawing will be inserted.

PROCEDURE

We give **INSERT_AVX** command. To the dialog box that appears, we give the drawing file name, which we want to insert. We don't use the extension AVX, we give only the name e.g. NEW.
Then we define the insertion point, giving a point into the drawing. The other drawing is inserted there as a BLOCK.
If we want, we break it later, using **EXPLODE_BLOCK** command.
Note
The inserted file now exists as a BLOCK (e.g. NEW.BLK) in the library (AVXLIB directory) and in [file name] AVF directory.

RELEVANT COMMANDS

<BLOCK>, <DXF_IN>, <EXPLODE>, <EXPLODE_BLOCK>, <INSERT>

UNDO

YES

1.4.55 INTERSEC

QUICK REFERENCE

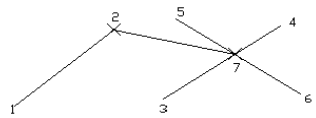
We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a line's segment). The purpose is the selection of other objects' intersection point.

© 1993-2026 Gardos Software LTD

The Intersection point is indicated by the user, giving one point NEAR it. The point thus selected is used for finishing the current command.

SYNTAX

[Any Command e.g. ***LINE****]*
[Command] **INTERSEC**
[Reference point near the object] We give a point OVER the intersection point (e.g. line segments).
[Continuing the current command e.g. ***LINE****.*
[Finishing of command e.g. ***LINE*** *by* ***CANCEL****]*
Example
Figure



We want to draw a line segment from vertex 2 to point 7.
We give **LINE**
We give **ENDPOINT**
We give a point NEAR the vertex 2 over the line segment 1-2
We give **INTERSEC**
We give a point NEAR point 7 and over line segment 3-4 or 5-6
We cancel draw command by **CANCEL**.

ICON



MENU

Object Snap	**Intersection**

RELEVANT COMMANDS

<ENDPOINT>, <MIDPOINT>, <NEAREST>, <NODE>, <QUADRANT>, <TANGENT>, <VERTICAL>

UNDO

YES

1.4.56 INWINSEL

QUICK REFERENCE

By this command we can select all objects, which are whole inside an objects' selection window. Then we can apply any action to the selected objects (e.g. moving to another place).

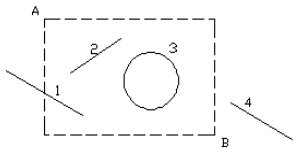
© 1993-2026 Gardos Software LTD

SYNTAX

[Command] **INWINSEL**
[First Point of window selection] We give a point in the drawing.
[Second Point of window selection] We give a second point which, together with the first point define the diagonal of the window selection. All objects have been selected.
[... we continue with another handle – command of the selected objects].

PROCEDURE

We give **INWINSEL** command. By this command, we will define a rectangle border (window) objects' selection. So, we give two points on the drawing, defining a diagonal of the objects' selection window therefore and the window itself (its sides are parallel to X and Y axis). Now all objects that are WHOLE inside the window, are being selected (and change their Color). We can continue selecting other objects (giving again a selecting objects' command such as **WINSEL** or I**NWINSEL** or **SELECT**), or handle in some way the selected objects (e.g. copying them somewhere else).
Note
If some of the objects that are inside the window selection, have one or more segments outside of it, then we don't select them with **INWINSEL** command, but using **WINSEL** or **SELECT** (one by one).
Example



We have the objects of this Figure
We give **INWINSEL**
We give Point A
We give Point B
We select objects 2 and 3.
Object 1 is not selected, because it has a segment outside the window selection.

ICON



MENU

Select	**In Window Select**

RELEVANT COMMANDS

<SELECT>, <SELECTADD>, <SELECTSUB>, <UNSELECT>, <WINSEL>,

UNDO

YES

1.4.57 LASTSV

QUICK REFERENCE

By this command we recall the area that we have created before in our drawing, using **SELECTV** command and continue selecting more vertices, or we handle immediately the region.

SYNTAX

[Command] **LASTSV**
[Handle - Command of the region and select more vertices]
PROCEDURE

© 1993-2026 Gardos Software LTD

We give **LASTSV** command. If we have already created a region, before with **LASTSV** command, then this region (or the last one, if there were more than one) appears. Then we can select more new vertices and at the end handle the region properly, using some relevant command (see relevant commands of **SELECTV**).

MENU

Select	**Last Selected Vertices**

RELEVANT COMMANDS

<MAKESV>, <SELECTV>, <UNSELECT>

UNDO

NO (CANCELING USING **CANCEL**)

1.4.58 LINE

QUICK REFERENCE

It is one of the most basic commands, which allows us to draw line segments.

SYNTAX

[Command] **LINE**
[Start Point] We give the first Point of the line.
[Endpoint] We give the second Point of the line – the first line is formed.
[Endpoint] If we want, we continue drawing lines. The endpoint of the previous line automatically becomes the start point of the next one, and giving another point we have a second line. We can continue in this way for as long as we want.

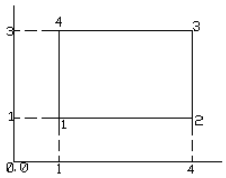
© 1993-2026 Gardos Software LTD

[Endpoint] ...
[Cancel] By **CANCEL** the Draw command is stopped.

PROCEDURE

We give **LINE** command and then we give points that are being joined one by one, by the turn we are giving them, creating lines. Every new point creates one line with the previous point. At the end we stop with **CANCEL** (**CTRL+C** or clicking the right button of the mouse).
We can give the points using all the well known methods, that is with mouse or directly to the drawing, or with relevant to other points commands (e.g. **ENDPOINT**, **MIDPOINT**) or with keyboard. In the last case we give absolute coordinates e.g. 2,3 (X=2, Y=3) or the angle and the distance from the last point e.g. #45,4 (angle 45 , distance 4 units), or at last, movement DX, DY always from the last point e.g. @4,8 (DX=4, DY=8).
Let's note that we can give also a third number (Z coordinate) e.g. 1,1,4.

Example 1
We draw a simple line from (0,0) to (5,4).
We give **LINE** command
We give a' Point (keyboard) = 0,0
We give b' Point (keyboard) = 5,4
We stop with **CANCEL**
Example 2
We draw a parallelogram rectangle with length of horizontal side= 3, vertical=2 and position of the down left corner 1,1.



We give **LINE**
We give the first Point = 1,1
We give the second Point, right = @3,0
We give the third Point, over = @0,2
We give the fourth Point, left = @-3,0
We give the fifth Point, down = @0,-2
We stop with **CANCEL**.
(DX=3, DY=0
DX=0, DY=2
DX=-3, DY=0
DX=0, DY=-2)

ICON



MENU

Draw	**Line**

RELEVANT VARIABLES

Tools	**General Options**	**Pen**

- → **'Layer' :** We specify the drawing layer where drawing objects will belong.
- → **'Color' :** We specify layer's color.
- → **'Line scale with type' :** We specify line scale with type (e.g. SOLID).
- → **''Pen width' :** We specify pen width**.**

RELEVANT COMMANDS

<POLYLINE>, <POINT>

UNDO

YES

© 1993-2026 Gardos Software LTD

1.4.59 LIST

QUICK REFERENCE

It is a command, which gives us a general description of all the existing commands of the program.

SYNTAX

[Command] **LIST**

PROCEDURE

Giving list command, a window with all the existing commands appears on prompt, as well as a general description of each command. By double clicking on any command, the selected command is activated.

UNDO

NO

1.4.60 MAGNIFY

QUICK REFERENCE

By this command, we have the ability to see, in a moving window in detail a part of our drawing, and the rest of it remains the same.

SYNTAX

[Command] **MAGNIFY**

© 1993-2026 Gardos Software LTD

PROCEDURE

Sometimes it is necessary for us to work in detail on a part of our drawing and at the same time to have the best possible view of the whole drawing. That happens only if we have, at the same time, two different views of our drawing , one in condition **ZOOM** and the other normal. That can be possible using **MAGNIFY** command, which is similar to a magnifying glass and has the following procedure: we give **MAGNIFY** command and we notice that a detail window appears on which we can see a part of our drawing magnified while the rest of it, remains the same. We can move this window freely over the entire surface of the drawing focusing on a different part of it, each time.
By double clicking on this window we can have a better zoom of the drawing and so on.
The detail window disappears with **CANCEL** or **REDRAW**.

RELEVANT VARIABLES

<ZOOM>

UNDO

NO

1.4.61 MAKESV

QUICK REFERENCE

This command creates a group of selected vertices identical to those given by **POLYLINE** command (or circle or ellipse).

SYNTAX

[Object selection] We select (e.g. by **SELECT**) a **POLYLINE**
[Command] **MAKESV** (Creation of selected vertices):
[Handle selected vertices – selection of new vertices] We handle properly the region of the selected vertices (e.g. **AREA**), or we also select new vertices.

PROCEDURE

We first select the object - **POLYLINE**. Then we give **MAKESV** command and a group of selected vertices is automatically created – identical to the ones given by **POLYLINE** command and we are in a selecting vertices condition (such as **SELECTV**). From now on we continue in the same way using **SELECTV** command.
Notes

(1) Selected objects can be, except for **POLYLINE**, rectangle (RECTANGLE), circle or ellipse, because they can be drawn as a **POLYLINE** (the latter consist of as many line segments as the variable ***"circle sectors"*** gives at the selection **GENERAL OPTIONS** (e.g. 72).

© 1993-2026 Gardos Software LTD

(2) **MAKESV** command influences the **LASTSV** command, changing the last selected group of vertices, to the current.

MENU

Select	**Make selected vertices**

RELEVANT VARIABLES

Tools	**General Options**	**More**

→ **'Circle sectors' :** We specify the number of circle segments.

RELEVANT COMMANDS

<LASTSV>, <SELECTV>

UNDO

NO (CANCELING USING **CANCEL**)

1.4.62 MEASURE

QUICK REFERENCE

By this command, we have the ability to measure the sides' length of one or more objects together.

SYNTAX

[Object selection] We select the objects the length of which we want to know.
[Command] **MEASURE**

PROCEDURE

We select as many objects as we want. Then we give **MEASURE** command and the program answers giving us a number. This number is the sum of the sides' length of all selected objects. If we have selected circles or circle arcs, by length we mean the length of the evolution of its perimeter (that means the perimeter of the circle and proportionately if it's circle arc).
Note
It is obvious that this command operates additively for all selected objects.

MENU

Modify	**Measurement**	**Length**

© 1993-2026 Gardos Software LTD

RELEVANT COMMANDS

<ANG>, <DIM>

UNDO

NO

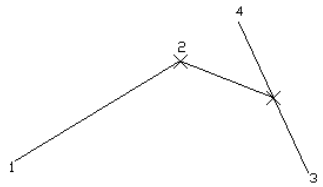
1.4.63 MIDPOINT

QUICK REFERENCE

We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a lines segment). The purpose is the selection of the midpoint of an object. We indicate the midpoint, giving a point NEAR the object and as soon as this is found (the midpoint) it is used for the completion of the active command.

SYNTAX

[Command] e.g. **LINE**
[Command] **MIDPOINT**
[Reference point near the object] We give a point on the object (e.g. to line segment).
[Continue with the other command] e.g. **LINE**
Example
Figure



We want to draw a line segment from vertex 2 to the **MIDPOINT** of line segment 3-4.
We give **LINE**
We give **ENDPOINT**
We give a point NEAR to vertex 2
We give **MIDPOINT**
We give a point OVER the line segment 3-4
We cancel the Draw command by **CANCEL**.

Note
If we give **MIDPOINT** command to the perimeter of a circle we will have its **CENTER**.

ICON



MENU

Object Snap	**Middle or circles center**

© 1993-2026 Gardos Software LTD

RELEVANT COMMANDS

<ENDPOINT>, <INTERSEC>, <NEAREST>, <NODE>, <QUADRANT>, <TANGENT>, <VERTICAL>

UNDO

YES

1.4.64 MIRROR

QUICK REFERENCE

By this command we can copy some model objects, symmetrical to an imaginary axis. These models can remain or be erased, after the command.

SYNTAX

[Objects selection] We select the objects (we want to copy using mirror)
[Command] **MIRROR**
[First Point of symmetry - copy axis]
[Second Point of symmetry - copy axis]

PROCEDURE

We select the objects we want e.g. using **WINSEL** command. We give **MIRROR** command. Then we have to define a symmetry axis, to which we will have the copy. For this reason we give two points, (1) and (2), and the Figure is being copied.
Example
We have the Figure A. We want its symmetrical Figure to axis (1)-(2).

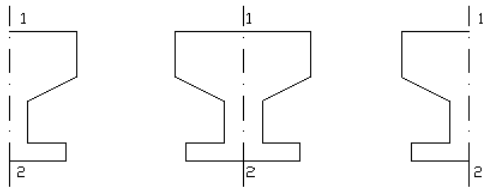


FIGURE A FIGURE B FIGURE C

We select the object of Figure A (e.g. with **WINSEL**).
We give **MIRROR** command and points (1) and (2). We have the result at Figure B.
Notes
(1) If after the copying, the model object will remain or not, it depends on the variable ***"MIRROR stay"*** of **SETUP**. If it is "1" it will remain, if it is "0" it will be erased.
So on the above example the variable ***"MIRROR stay"*** = "1", otherwise the result will be that shown in the Figure C.
(2) If we want horizontal or vertical symmetry axis, we activate the feature **ORTHO**, using **ORTHO_ON** command and then we give the second point for symmetry axis.

© 1993-2026 Gardos Software LTD

ICON



MENU

Modify	Transformation	Mirror

RELEVANT VARIABLES

Tools	General Options	More

→ **'Mirror stay' :** We specify if the command of mirror objects will be active.

RELEVANT COMMANDS

<COPY>, <MOVE>, <ROTATE>, <SCALE>

UNDO

YES

1.4.65 MOVE

QUICK REFERENCE

By this command we can move the objects we have selected, from one position to another.

SYNTAX

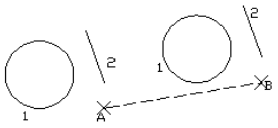
[Objects selection] Using e.g. **WINSEL** command, we select objects that will be moved.
[Command] **MOVE**
[First Point of moving vector]
[Second Point of moving vector]

PROCEDURE

We select the objects we want to move, using any objects' selection command (e.g.. **SELECT**, **WINSEL** e.t.c.). then we give **MOVE** command. After that, we have to give the direction and the moving distance. For this reason, we give two points which define a vector. The moving angle is the vector's angle to axis X and the moving distance, the vector's length.
Objects are being moved to the new position, (vector is imaginary and automatically disappears). For the specification of vector's points, we can use commands type **ENDPOINT**, **MIDPOINT** e.t.c., or mouse or keyboard, giving absolute or relative moving coordinates.

Example
We have the Figure A

© 1993-2026 Gardos Software LTD



We select the line segment (e.g. with **SELECT**)
We give **MOVE** command
We give points (A) and (B) and we have Figure B.

FIGURE A FIGURE B

ICON

MENU

Modify	**Move**	**Move**

RELEVANT COMMANDS

<COPY>, <MIRROR>, <ROTATE>, <SCALE>

UNDO

YES

1.4.66 NODE

QUICK REFERENCE

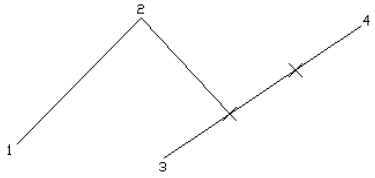
We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a lines segment). The purpose is the selection of a POINT - NODE that has been drawn using **POINT**, **DIVIDE** or **MEASURE** commands. The Point NODE, is being defined by the user, giving a Point NEAR it, and as soon as the program selects it, we use it to end the active command.

SYNTAX

*[E.g. **LINE** command]*
[Command] **NODE**
[Reference point near the object] We give a Point OVER (or NEAR) the NODE
*[Continuing the active command e.g. **LINE**]*
*[End of command e.g. **LINE** with **CANCEL**]*

© 1993-2026 Gardos Software LTD

Example



We want to draw a line segment from vertex 2 to the one third from the left, of line segment 3-4.
First we divide segment 3-4 to three equal segments:
-We select by **SELECT** (or using the relevant key and the mouse), segment 3-4,
-We give **DIVIDE** and then number 3.
After that we draw the line segment form vertex 2 to the first node from the left, which has been created with **DIVIDE** command.
-We give **LINE**,
-We give **ENDPOINT**,
-We give a Point NEAR vertex 2,
-We give **NODE**,
-We give a Point NEAR the first left node of segment 3-4 in such a way that this node will be included in the selection box of the cursor,
-We cancel the action of Draw command by **CANCEL**.

ICON

MENU

Object Snap	**Node**

RELEVANT COMMANDS

<ENDPOINT>, <INTERSEC>, <MIDPOINT>, <NEAREST>, <POINT>, <QUADRANT>, <TANGENT>, <VERTICAL>

UNDO

YES

1.4.67 NEAREST

QUICK REFERENCE

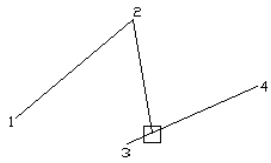
We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a lines segment). The purpose is the selection of a Point that belongs to another object, and it is as near a point we show as possible. We use this point to end the active command.

SYNTAX

*[Any command e.g. **LINE**]*
[Command] **NEAREST**

© 1993-2026 Gardos Software LTD

[Reference point near the object] We give a point NEAR the object (e.g. line segment)
[Continuing of command that is active e.g. ***LINE****]*
[End of command e.g. ***LINE*** *by* ***CANCEL****]*
Example
Figure



We want to draw a line segment from vertex 2 to a point over line segment 3-4.
We give **LINE**,
We give **ENDPOINT**,
We give a Point NEAR vertex 2 in order to begin our line from vertex 2,
We give **NEAREST**,
We show a point on line segment 3-4 and the program will find the nearest point to it, which will be precisely over the line segment 3-4. We stop draw command using **CANCEL**.

ICON



MENU

Object Snap	**Nearest**

RELEVANT COMMANDS

<ENDPOINT>, <INTERSEC>, <MIDPOINT>, <NODE>, <QUADRANT>, <TANGENT>, <VERTICAL>

UNDO

YES

1.4.68 OBSPLINE

By this command we can draw an open B-Spline, using a polygonal leader (adjoining vertices). Leader's setting takes place with the vertices' selection (by **SELECTV**) or with the selection of a polygonal line.

SYNTAX

[Vertices selection / Polygonal line selection]
[Command] **OBSPLINE**
[CANCEL] (Cancelling in case of vertices' selection).

PROCEDURE

By **SELECTV** command we select the vertices of the polygonal leader at one direction (clockwise movement or counter clockwise) or select by **SELECT** command a **POLYLINE** that creates a polygon. Then we give **OBSPLINE** command, and is being created an **OBSPLINE** curve with open node vector. After that, if it is necessary, we give **CANCEL** command to cancel the action of selecting vertices.

© 1993-2026 Gardos Software LTD

Note
The polygonal leader cannot have more than twenty (20) vertices.

ICON



MENU

Draw	**Complex objects**	**OBSPLINE**

RELEVANT COMMANDS

<BEZIER>, <CBSPLINE>, <CSPLINE1>, <CSPLINE2>

UNDO

YES

1.4.69 OFFSET

QUICK REFERENCE

By this command we can copy some line segments in a specific distance from the model objects.

SYNTAX

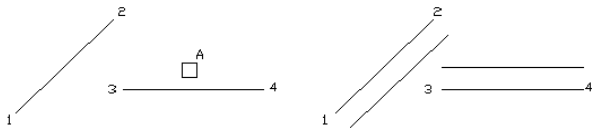
[Objects selection] We select line segments, which are going to be copied.
[Command] **OFFSET**
[---> Distance=] We give the distance with a number.
[Reference point] We give a point to show the direction where the copying will be.
[Cancel] We stop **SELECT** by **CANCEL**.

PROCEDURE

We select drawing segments, which are going to be copied. Then we give **OFFSET** and to the dialog box that appears, we give the copying distance from the model objects.
We notice that the crosshairs cursor is being changed into point selection cursor (little circle) and we give a point to show the direction where the copying will be. At the end, we stop objects' selection using **CANCEL**.

© 1993-2026 Gardos Software LTD

Example
We want to copy 1-2 and 3-4 line segments in a 0.5 distance from them.



We select 1-2 and 3-4
We give **OFFSET** command
We give value 0.5 units
We give the A to show the direction
We stop using **CANCEL**

ICON



MENU

Modify	**Copy**	**Copy offset**

RELEVANT COMMANDS

<COPY>, <COPY_IN>, <COPY_OUT>

UNDO

YES

1.4.70 OPEN

QUICK REFERENCE

By this command we open an existing drawing or create a new file. If we are already working on a drawing, to which we have made some changes, the program will ask us to save it, before it opens the new file.

SYNTAX

[Command] **OPEN**
[Selection of drawing file] From the list of drawing files, we select one of the existing or give a new file.

© 1993-2026 Gardos Software LTD

PROCEDURE

We give **OPEN** command. The list of our drawing files appears. We select one of them. By selecting a file, the drawing we were already working on is saved (after we answer positively to the relevant program questions). The new drawing is loaded to the current window, and the old drawing is unloaded from it.

ICON



MENU

File	**Open**

RELEVANT COMMANDS

<SAVE>, <SAVE_OPEN>,

UNDO

NO

1.4.71 ORTHO - ORTHO_ON - ORTHO_OFF

QUICK REFERENCE

Using these commands we activate or inactivate the drawing of only horizontal and vertical lines.

SYNTAX

[Command] **ORTHO** or **ORTHO_ON** or **ORTHO_OFF**

PROCEDURE

(1) **ORTHO** command. This command changes the condition ORTHO_ON to ORTHO_OFF and vice versa.
(2)**ORTHO_ON**. ORTHO_ON command turns to orthogonal drawing condition, where we draw lines horizontal or vertical (parallel to X and Y axis).
(3) **ORTHO_OFF** command. This command cancels the orthogonal drawing condition and we can draw now oblique lines too.
Note
When we are in condition ORTHO_ON and give **MOVE**, **COPY** e.t.c. commands, the moving vector is parallel to axis X or axis Y.

© 1993-2026 Gardos Software LTD

ICON



UNDO

NO

1.4.72 PEDIT

QUICK REFERENCE

By this command we can draw straight a POLYLINE, using some already selected lines, or some selected vertices, while **SELECTV** command (vertices selection) is active.

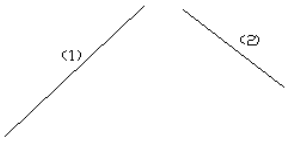
SYNTAX

[Lines selection] We select all lines we want, to create a POLYLINE
[Command] **PEDIT**

PROCEDURE

We select all lines we want, to create the POLYLINE (by **SELECT**, **WINSEL** e.t.c.). These lines don't have to be adjoining, not even create one united line, not even more a closed surface. Then we give **PEDIT** command and all separate lines, belong now to a POLYLINE. Drawing doesn't change that time – but next time we will select one of these lines, all lines will be selected together as one.
The opposite process (POLYLINE breaking) works out with **EXPLODE** command. Also, instead of selecting lines we can create POLYLINE (using **PEDIT**), giving first **SELECTV** command and selecting some vertices (real or not) and at the end giving **PEDIT** command.

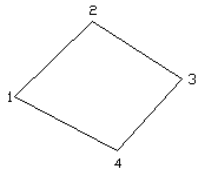
Notes
(1) After **PEDIT**, all lines are automatically moved to the current LAYER.
(2) If the POLYLINE is ad-joining, we can create a group of selected vertices with **MAKESV** command. (This process is exactly the opposite of the series of commands **SELECTV** + **PEDIT** = POLYLINE creation from vertices, that is we have here **SELECT** (POLYLINE) + **MAKESV** = vertices creation from POLYLINE).
Example 1



We select lines 1 and 2.
We give **PEDIT**
The two lines now are one POLYLINE, and from now on they are selected as one line, and in **MOVE** commands they both move together.

© 1993-2026 Gardos Software LTD

Example 2



We will create POLYLINE from vertices.
We give SELECTV command and then vertices 1, 2, 3 and 4
We give PEDIT command.
The POLYLINE has been created.

MENU

Modify	**Union-Split**	**Create a polyline**

RELEVANT COMMANDS

<EXPLODE>, <EXPLODE_BLOCK>, <MAKESV>, <POLYLINE>, <SELECTV>

UNDO

YES

1.4.73 PGDN - PGUP - PGLEFT - PGRIGHT

QUICK REFERENCE

These commands move our drawing by one “page” (one prompt) down, up, left or right.

SYNTAX

[Command] **PGDN** or **PGUP** or **PGLEFT** or **PGRIGHT**

PROCEDURE

These commands are given using the mouse mainly, pressing on the scroll bars right vertical, down horizontal). After keys pressing, our drawing is moved by one prompt, according to the command (down, up, e.t.c.).

KEYS

Right scroll bar for up-down and horizontal scroll bar for right-left.

RELEVANT COMMANDS

<DOWN>, <UP>, <LEFT>, <RIGHT>

UNDO

NO

1.4.74 PIC_OUT

QUICK REFERENCE

By this command we can create a file PIC type, which will include our drawing, in that way that can be possible to be read from other programs e.g. CORELDRAW® e.t.c. PIC file is an icon file.

SYNTAX

© 1993-2026 Gardos Software LTD

[Command] **PIC_OUT**

PROCEDURE

We give **PIC_OUT** command and on the window that appears, we write the file name we want. The program creates, based on the saving icon options (icons width and height) a PIC icon file with the above name, which includes the current drawing.

MENU

File	**Export**

RELEVANT COMMANDS

<BMP_OUT>, <DXF_OUT>, <PLT_OUT>

UNDO

NO

1.4.75 PLT_OUT

QUICK REFERENCE

By this command we can create a file PLT type, which will include our drawing, in that way that can be possible to be read from other programs e.g. CORELDRAW® e.t.c. PLT file is a drawing file.

SYNTAX

[Command] **PLT_OUT**

© 1993-2026 Gardos Software LTD

PROCEDURE

We give **PLT_OUT** command and on the window that appears, we write the file name we want. The program creates, based on the saving drawing options, a PLT drawing file with the above name, which includes the current drawing.

MENU

File	**Export**

RELEVANT COMMANDS

<BMP_OUT>, <DXF_OUT>, <PIC_OUT>

UNDO

NO

1.4.76 POINT

QUICK REFERENCE

By this command, we create a point, which is the smallest drawing object. This can be used later as a reference point for other objects' drawing.

SYNTAX

[Command] **POINT**
[A point to the drawing] We give a point to the drawing, where the object "Point" will be placed.

PROCEDURE

Simple command. We give **POINT** command and then a point in the drawing. The object "Point" with the point type which has been defined in the selection **GENERAL OPTIONS** is placed there.
(Point type= 0 : simple point, = 1 : x, = 2 : +)
Note
The most obvious way of using a point is as a permanent reference of other objects' drawing. It can be selected during draw command using **NODE** command.

ICON

© 1993-2026 Gardos Software LTD

MENU

Draw	Point

RELEVANT VARIABLES

Tools	General Options	Pen

- → **'Point type' :** Specifies the style of point appearance (e.g. cross+).
- → **'Point style radius' :** Specifies points' size by type.

RELEVANT COMMANDS

<LINE>, <NODE>

UNDO

YES

1.4.77 POLYGON

QUICK REFERENCE

By this command we can draw a polygon, giving the number of its sides and the radius of the outlined circle.

SYNTAX

[Command] **POLYGON**
[---> number of vertices=] We give the number of vertices (sides).
[Point-center of the outlined circle] We give a point – the center of polygon.
[Point-on the perimeter of the outlined circle] We give another point the distance between which and the previous,is as long as the radius of the outlined circle.

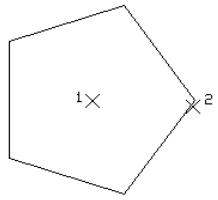
PROCEDURE

We give **POLYGON** command. A dialog box appears, where the program asks us for the number of polygons vertices (e.g. 5 for pentagon). Then we give two points – polygon's center and its radius – and the polygon is being drawn.

Notes
(1) Polygon is a a POLYLINE (e,g, we can use later the **MAKESV** command or the **EXPLODE** command).
(2) We can define a polygon which will have about 60 vertices.

© 1993-2026 Gardos Software LTD

Example
Pentagon's drawing



We give POLYGON command
Number of vertices = 5
We give points 1 and 2
Is being created a pentagon.

ICON



MENU

Draw	**Polygon**

RELEVANT COMMANDS

<RECTANGLE>, <MAKESV>, <POLYLINE>, <WALL>

UNDO

YES

1.4.78 POLYLINE

QUICK REFERENCE

This command is being used for the creation of one united line which consists of contiguous line segments or arcs. The final line acts as a united object (it is selected, moved, copied e.t.c. as one).

SYNTAX

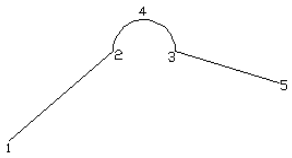
[Command] **POLYLINE**
[First Point ARC]
[Second Point ARC]
[Third Point ARC]
[Last Point ARC]
[CANCEL CLOSE] End or close POLYLINE.

© 1993-2026 Gardos Software LTD

PROCEDURE

We give **POLYLINE** command. Then we can give points, drawing contiguous line segments (as **LINE** command), or we can even draw arcs (using arc key). At the end we stop with **CANCEL**.

Example
We want to draw a POLYLINE like this in the Figure:



We give **POLYLINE**
We give points (1) and (2)
We press the drawing key arc (to the mouse)
We give point (3) and then (4)
We give point (5) (the drawing line condition has automatically returned)
We give **CANCEL** and the drawing stops

Notes
(1) During the drawing we can give the points, as at the LINE command, that means using mouse or keyboard, in absolute or relative co-ordinates, or in positions relevant to other objects (see **ENDPOINT**, **INTERSEC**, **MIDPOINT** e.t.c.).
(2) POLYLINE is like a BLOCK.
(3) Using a POLYLINE we can create a group of selected vertices with **MAKESV** command (corresponding with **SELECTV**).
(4) We can find the area of the enclosed surface, of one closed (or not) POLYLINE, with non intersected segments or arcs, with **AREA** command.
(5) On a POLYLINE (with contiguous segments) we can create hatch lines with **HATCH** command.

ICON



MENU

Draw	**Polyline**

RELEVANT COMMANDS

<ARC>, <CLOSE>, <EXPLODE>, <LINE>, <PEDIT>

UNDO

YES

1.4.79 PREVIOUS

QUICK REFERENCE

This command undoes the previous **ZOOM**. A second **PREVIOUS** undoes a second **ZOOM** (and not the first **PREVIOUS**).

SYNTAX

[Command] **PREVIOUS**

PROCEDURE

We give **PREVIOUS** command. If at some point we gave ZOOM command, that is refuted and also the drawing is redrawn, and the objects have the size that they had before ZOOM. The real objects' size remains the same (such as in ZOOM command). The only thing that changes is the relative distance from the user.
If we give **PREVIOUS** again, the ZOOM before the last is refuted and so on, until there is no ZOOM active.

© 1993-2026 Gardos Software LTD

ICON

RELEVANT COMMANDS

<ZOOM>, <ZOOMUNDO_ON>, <ZOOMUNDO_OFF>, <MAGNIFY>

UNDO

YES, if ZOOMUNDO_ON command has preceded.

1.4.80 PRINT

QUICK REFERENCE

By this command we can print our drawing (the whole or a part of it) to a printer, (DOT MATRIX LINE PRINTER, LASER, PLOTTER) or we can save it as an image file, processed by image processor programs (to DOS or to WINDOWS®, e.g. BMP) or finally as a PLOTTER file ready for plotting.

SYNTAX

[Command] **PRINT**

© 1993-2026 Gardos Software LTD

PROCEDURE

This command is very simple. The only thing we need is a ready printer (turned on and supplied with paper) and give **PRINT** command. The abilities of **PRINT** command are many, and they are defined by the values of some variables in the selection **GENERAL OPTIONS** , which are being described below.

Tools	General Options	Print

- → **'Print to black' :** Specifies if the drawing's printing color will be black.
- → **'Print all drawing' :** Specifies all drawing printing.
- → **'Print at paper limits' :** Specifies if drawing printing will be within the chosen paper limits..
- → **'Landscape' :** Specifies if the position of the paper during the printing will be horizontal.
- → **'Printer pixel width in mm' :** Specifies if objects' printing width by scale will be active.
- → **'Print pixel's width in mm' :** Specifies the pixel printing width of the drawing in mm, if it selected the choice "Printer pixel width in mm'
- → **'Print factor X' :** Specifies the scale factor at X.
- → **'Print factor Y' :** Specifies the scale factor at Y.
- → **'Paper margin at mm' :** Specifies the margin perimetrical of the drawing in mm, so as not to be printed at the edge.
- → **'Paper RGB color' :** Specifies the color of the paper to be printed.
- → **'1 printing unit at mm' :** Specifies the real printing dimension - correspondence to the drawing unit (e.g. if we choose that 1 drawing unit = 10 mm, the drawing will be printed by scale 1:100).
- → **'Paper size' :** Specifies the size of the printing paper in mm.
- → **'Paper width at mm' :** Specifies the width of the printing paper in mm.
- → **'Paper height at mm' :** Specifies the height of the printing paper in mm.

Notes

(1) We recommend you save your work, before you print using **<SAVE>** or **<SAVE_AS>** commands.

(2) ***"Pens width"*** is taken into account only when we have printing to a LINE PRINTER or LASER and also the creation of image files PCX or BMP. To the PLOTTER pens width replaces the correspondent pen that is related with ***"Color".***

(3) We can stop printing or files creation using the key [ESC].

ICON



MENU

File	Print

© 1993-2026 Gardos Software LTD

KEYBOARD

Ctrl + P

RELEVANT VARIABLES

All the above variables (**GENERAL OPTIONS**)

RELEVANT COMMANDS

<BMP_OUT>, <PLT_OUT>, <PIC_OUT>

UNDO

NO

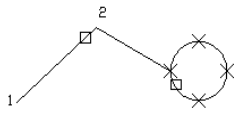
1.4.81 QUADRANT

YES **QUICK REFERENCE**

We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a line's segment). The purpose is the selection of a POINT that is on one of the quadrants of the circle (angles 0,90,180,270). We indicate the circle, giving a point ON it, and as soon as the nearest quadrant is selected, by the program, we use it to end the active command.

SYNTAX

[Command e.g. ***LINE****]*
[Command] **QUADRANT**
[Point on the circle] We give a point on the circle
[We continue with the other command e.g. ***LINE****]*
Example
Figure



We want to draw a line segment, from point 2 to the circle, so that it passes from the left quadrant.

-We give **LINE**,
-We give **ENDPOINT**,
-We give a Point NEAR vertex 2,
-We give **QUADRANT**,
-We give a Point ON circles perimeter, NEAR the left quadrant,
-We stop Draw command with **CANCEL**.

© 1993-2026 Gardos Software LTD

ICON



MENU

Object Snap	Quadrant

RELEVANT COMMANDS

<ENDPOINT>, <INTERSEC>, <MIDPOINT>, <NEAREST>, <NODE>, <TANGENT>, <VERTICAL>

UNDO

YES

1.4.82 RADIATE

QUICK REFERENCE

By this command, we can find the center of a polygonal region and draw radii from its center to its vertices. (Center isn't drawn as a point). It is a general command, and is also used to non closed – non regular polygons (without intersected segments). It is used to POLYLINES or to regions that have been created from selected vertices (e.g. **SELECTV** command).

SYNTAX

[POLYLINE Selection] / Creation of selected vertices]
[Command] **RADIATE**
[CANCEL] For canceling of vertices selection.

PROCEDURE

RADIATE command is used to a region that is created from selected vertices (or to POLYLINE) when there aren't intersected segments defining the region (or the POLYLINE, which moreover isn't allowed to include any arc). We create the region selecting vertices (with **SELECTV** or with **LASTSV** commands) or we select the POLYLINE (e.g. a polygon using **POLYGON** command or a RECTANGLE) and we give **RADIATE** command. The program finds automatically the center of the polygonal region (exactly like **CENTROID** command, with the difference that now the center isn't drawn as a point) and all radii are drawn to its vertices from there.
These radii are a POLYLINE (when you select one – all of them are selected). In case the command isn't possible to apply, we hear from the program a characteristic sound and we see an error message.

© 1993-2026 Gardos Software LTD

Example

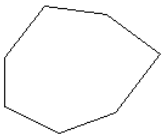


Figure A

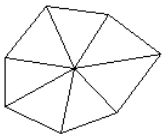


Figure B

We select the polygon (it has to be POLYLINE – and if it's not we change it into a Polyline using **PEDIT** command).
We give **RADIATE** command and we have the Figure B.

ICON



MENU

Draw	**Complex objects**	**Polygon radii**

RELEVANT COMMANDS

<CENTROID>, <POLYLINE>, <SELECTV>

UNDO

YES

1.4.83 RECTANGLE

QUICK REFERENCE

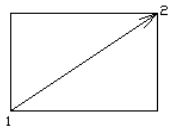
By this command we draw a rectangle.

SYNTAX

[Command] **RECTANGLE**
[Point-1] We give the first Point of diagonal
[Point-2] We give the second Point of diagonal

PROCEDURE

We give **RECTANGLE** command and then the two points of rectangle's diagonal, which we want to draw. Points are given in the same way, with mouse or keyboard, using absolute or relative co-ordinates.
Figure



© 1993-2026 Gardos Software LTD

Note
It is a very easy and useful command, and because of this there is a relevant key. With **SELECT** command the whole is selected, as a **POLYLINE**.

ICON



MENU

Draw	**Rectangle**

RELEVANT COMMANDS

<POLYLINE>, <WALL>

UNDO

YES

1.4.84 REDO

QUICK REFERENCE

REDO command brings back only the last command that has been recalled using **UNDO**, that is, it is the opposite of UNDO command.

SYNTAX

[Command] **REDO**

PROCEDURE

It is a very simple command. We can use it only if there were **UNDO**. If we use it once we cannot use it again, except if we give another **UNDO**.
Conclusively, **REDO** command can bring back only one command at a time, while **UNDO** brings back up to 2000 commands. To each **UNDO** there is the corresponding **REDO**.

ICON

MENU

Edit	**Redo**

© 1993-2026 Gardos Software LTD

RELEVANT COMMANDS

<UNDO>

UNDO

YES

1.4.85 REDRAW

QUICK REFERENCE

This command doesn't change anything to the drawing. The only thing it does, is to renew the prompt by redrawing the visible drawing segment. After this renewal, all marks (points like spots) that were on prompt have disappeared, after corrections or other operations to the drawing.

SYNTAX

[Command] **REDRAW**

PROCEDURE

Simple command. We give **REDRAW** and prompt is being redrawn, without spots, giving us a completely "clean" drawing.
Notes
(1) This redrawing can be done automatically from the program, when this is necessary.
(2) In case of **MAGNIFY, REDRAW** command cancels it.

ICON


MENU

View	Redraw

UNDO

NO

1.4.86 ROTATE

QUICK REFERENCE

By this command we rotate one or more selected objects around a specified point, by an angle.

SYNTAX

[Objects selection] We select the objects we want to rotate (e.g. **WINSEL**)
[Command] **ROTATE**
[Rotation Point] We specify a Point in the drawing, around which the rotation will be.
[Second Point] The second Point we give, together with the first one define a vector which, by its turn, defines, to X axis, an angle. This is the object's rotation angle.

PROCEDURE

We select the objects (e.g. with **SELECT** or **WINSEL** e.t.c.). Then we specify a point in the drawing, the

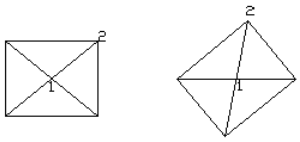
© 1993-2026 Gardos Software LTD

rotation point (or symmetry center). After that, giving another Point we define a vector, whose angle (to X axis) is being set from the program as the rotation angle (reference angle) and it rotates the objects. Objects are being erased from their initial position.

Example

Figure

Figure A Figure B



We select the rectangle (not the diagonals) at Figure A.
We give **ROTATE**.
We select the intersection of diagonal, Point (1) as the rotation Point.
We select Point (2) (the upper endpoint of the right diagonal) defining as the rotation vector the (1)-(2) and rotation angle the angle with X axis. The result is Figure B.

ICON



MENU

Modify	**Transformation**	**Rotate**

RELEVANT COMMANDS

<COPY>, <MIRROR>, <MOVE>, <SCALE>

UNDO

YES

1.4.87 SAVE

QUICK REFERENCE

Saving / updating of our drawing in a file, on disk (or diskette).

SYNTAX

[Command] **SAVE**

PROCEDURE

Simple command. We give **SAVE** command and our drawing is being saved in the current file (that is being shown on the upper left corner of the prompt).

Note

It is better to save the drawing we are working on regularly, using **SAVE** command, avoiding losing our work, in case of a drop in voltage or generally another unforeseen event that would stop our PC or our program operation. Also we have to save our drawing, before we exit, by either opening another program with **OPEN** command, or exiting from the program with **EXIT** command.

ICON



MENU

File	**Save**

© 1993-2026 Gardos Software LTD

KEYBOARD

Ctrl + S

RELEVANT COMMANDS

<OPEN>, <SAVE_OPEN>, <SAVE_AS>

UNDO

NO

© 1993-2026 Gardos Software LTD

1.4.88 SAVE_AS

QUICK REFERENCE

We save our drawing in a file, on disk with another file name.

© 1993-2026 Gardos Software LTD

SYNTAX

[Command] **SAVE_AS**
[--->file name:] We give the file name, under which we want to save the current drawing.

PROCEDURE

We give **SAVE_AS** command and at the dialog box the file name under which we want to save our drawing. If a similar file already exists, the program will give us a warning message to protect us from wrong overwriting it.
Note
After **SAVE_AS** command, we now work on the new file and not on the previous one.

MENU

File	**Save as..**

RELEVANT COMMANDS

<OPEN>, <SAVE>, <SAVE_OPEN>

UNDO

NO

1.4.89 SAVE_OPEN

QUICK REFERENCE

This command is a combination of **SAVE** and **OPEN** commands, without changing the drawing. The purpose is to save the drawing and at the same time to reduce as much as possible the space it takes up in the computer's ram (especially in case of a big drawing).

SYNTAX

[Command] **SAVE_OPEN**

PROCEDURE

SAVE_OPEN command is a combination of **SAVE** and **OPEN** commands. First it saves the drawing on disk. Then it opens – the same drawing – file, by-passing the file list which was going to appear in a **SAVE** and **OPEN** commands consequence.
Practically, the drawing is being saved using the least possible space in computers ram, and this also happens because all actions' history up to this moment is being erased, ,
So, we cannot use **UNDO** for actions that took place before **SAVE_OPEN**. Also, all the useless BLOCKS of the drawing are erased (in the relevant AVF catalog – directory).

© 1993-2026 Gardos Software LTD

Note
It is a very useful command for big drawings. We remind that after **SAVE_OPEN** command**, UNDO** is not valid for commands that had been given before this. Of course **UNDO** is valid for commands which are going to be given from now on.

RELEVANT COMMANDS

<EXIT>, <OPEN>, <SAVE>

UNDO

NO

1.4.90 SCALE

QUICK REFERENCE

By this command we can change the size of one or more objects (at the same time).

SYNTAX

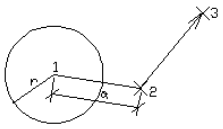
[Objects selection] We select the objects whose size we want to change.
[Command] **SCALE**
[Point-1] We give the α' Point of a size vector
[Point-2] We give the b' Point of a size vector
Vector's length is the scale of enlargement / shortening of the selected objects

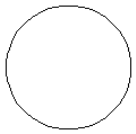
PROCEDURE

We select the objects. Then we give **SCALE** command and a size vector (we first give a point and then another one and we define the vector). If vector's length is less than 1, objects are being diminished, if it is bigger than 1, objects are being enlarged and if it is equal to 1, they retain their first size.
The a' point of the size vector is the reference point, to which objects' size will change and the movement that will result if scale is different of 1.

© 1993-2026 Gardos Software LTD

Example
We want to enlarge Figure's circle.





We select the circle.
We give **SCALE** command.
We give the a' point of size vector, to distance (a).
We give the b' point of vector. If vector's length is 1.7 units then the circle is being enlarged (1.7>1) and its radius now is r'= 1.7*r.
The new position of the circle is at the direction 1.2 and distance 1.7*a.

Note
If we want the object to retain its position in relation to a point of it, we select this point as a' point of the vector.

ICON



MENU

Modify	**Transformation**	**Scale**

RELEVANT COMMANDS

<COPY>, <MIRROR>, <MOVE>, <ROTATE>

UNDO

YES

1.4.91 SELECT

QUICK REFERENCE

By this command we can select every time, an object – or a complicated object (BLOCK, POLYLINE).

SYNTAX

[Command] **SELECT**
[Selection Point] We give a point on an object, to select it.

PROCEDURE

We give **SELECT** command. The crosshair turns into a little moving rectangle. We are moving wherever we want on the drawing, using the mouse. We are moving on an object and we press the left key of the

mouse. The object changes color to show that it has been selected. We can continue, if we want, to select more objects, by simply holding **SHIFT** or **CTRL** key pressed. It is also obvious, that we are in a selection condition, because the crosshair has turned into a window objects' selection cursor. When we select all objects we want, we can either move them somewhere else or copy them, erase them, make them BLOCK e.t.c. If we regret and want to stop, we give **CANCEL**.
Note
If during the time we are selecting objects, we change our mind and we want to remove some of those we've already selected, we press **SELECTSUB** and the last selected object is being removed (if we want to continue removing objects we hold **SHIFT** or **CTRL** key

© 1993-2026 Gardos Software LTD

pressed). Then we press **SELECTADD** and we are now in a selection condition again.

ICON

MENU

Select	**Object selection**

RELEVANT VARIABLES

Tools	**General Options**	**More**

→ **'Box selection PIXEL' :** Specifies the number of pixel of the objects' selection rectangle box.

RELEVANT COMMANDS

<INWINSEL>, <SELECTADD>, <SELECTSUB>, <UNSELECT>, <WINSEL>

UNDO

YES

1.4.92 SELECTSUB

QUICK REFERENCE

This command removes the selection of some objects that we have already selected.

SYNTAX

[Objects' selection command] We give an objects' selection command (**SELECT**, **WINSEL** e.t.c.)
[Objects' selection] We select some objects. Let's suppose that we have selected more objects than we wanted.
[Command] **SELECTSUB**
[Objects' removing] We continue selecting one by one, all the objects we want to remove.

© 1993-2026 Gardos Software LTD

[... We continue with the selected objects, giving another handling command e.g. ***MODE****, or we return to selection condition using* ***SELECTADD****]*

PROCEDURE

SELECTSUB command is given while we are in an objects' selection condition, and never on its own.
That is, we have to give first an objects' selection command e.g. **SELECT**, then select some objects and after that, if we want to remove some of them, we can give **SELECTSUB** command. Then we notice that the key [-] is being pressed and remains pressed while **SELECTSUB** is active. We select all objects we want to remove (in the same way we use **SELECT**). These objects are not selected anymore.
At the end we can either give **SELECTADD** command, so that we return to selection condition, or handle the already selected objects using another command (**COPY**, **MOVE** e.t.c.).

ICON



MENU

Select	**Sub selected**

RELEVANT COMMANDS

<INWINSEL>, <SELECTADD>, <SELECT>, <SELECT_ALL>, <UNSELECT>, <WINSEL>

UNDO

NO

1.4.93 SELECTADD

QUICK REFERENCE

This command is a **SELECT** type command, that is, objects' selection. It is being given only after **SELECTSUB** command (removing selected objects).

SYNTAX
[Object's selection command] We give an objects' selection command (**SELECT**, **WINSEL** e.t.c.)
[Objects' selection] We select some objects.
[Removing selected objects' command] **SELECTSUB**
[Objects' removing] We continue selecting one by one, all the objects we want to remove.
[Command] **SELECTADD**
[... We continue selecting objects, one each time]

© 1993-2026 Gardos Software LTD

PROCEDURE

SELECTADD command is being given, only if we are already in an objects' selection (or removing selected objects) condition. That is, we have to give first an objects' selection command e.g. **SELECT**, then select some objects and after that, if we want to remove some of them, we can give **SELECTSUB** command. After that we use **SELECTADD** command (we notice that the key [-] is being pressed). As we give **SELECTADD** we can continue selecting one by one the objects we want (the key [-] is not pressed anymore).

ICON



MENU

Select	**Add selected**

RELEVANT COMMANDS

<INWINSEL>, <SELECTSUB>, <SELECT>, <SELECT_ALL>, <UNSELECT>, <WINSEL>

UNDO

NO

1.4.94 SELECT_ALL

QUICK REFERENCE

By this command, we have the ability to select all objects of the drawing, at the same time (except those which are at frozen drawing layers).

SYNTAX

[Command] **SELECT_ALL**

PROCEDURE

We give **SELECT_ALL** command and all objects of the drawing are automatically selected. Then we can continue handling the selected objects using any command **BLOCK**, **ERASE**, **MOVE** e.t.c..
Notes
(1 The objects which are at "frozen" LAYERS (using **FREEZE** command) are not selected .
(2) **SELECT_ALL** command is useful because with **SELECT**, **WINSEL**, **INWINSEL** commands we can totally select up to 2000 objects.

© 1993-2026 Gardos Software LTD

ICON



MENU

Select	**Select all objects**

RELEVANT COMMANDS

<INWINSEL>, <SELECT>, <SELECTADD>, <SELECTSUB>, <SELECT_ALL>, <UNSELECT>, <WINSEL>

UNDO

YES (operates as **CANCEL**)

1.4.95 SELECTV

QUICK REFERENCE

This command **(SELECT VERTICES)** gives us the ability of regions creation, aiming at their hatching or even POLYLINE creation.

SYNTAX

[Command] **SELECTV**
[Point in the drawing] We give a point either on an object or not.
[Point in the drawing] The same as before.
[Another Command] e.g. **HATCH** or **PEDIT**

PROCEDURE

We give **SELECTV** command (vertices selection). Then, as many times as we want, we can select points on the drawing.
These points constitute always, a border of an imaginary closed surface, which we are going to name “region”. Using key [Backspace] we can reduce the selected vertices by one.
This border, exists only during the time we create the region. However, at the end and as we have selected all vertices we want, we can make it permanent, by giving **PEDIT** (POLYLINE creation). Then the border is a usual POLYLINE. We can also make it permanent, if we give **HATCH** and BORDER type = 2 or 3. On the contrary, if we give **CANCEL** command, while we select vertices, this disappears and the selecting vertices process, is being canceled.

© 1993-2026 Gardos Software LTD

Example 1

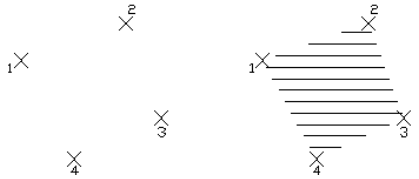


Figure A Figure B

We give **SELECTV** command
We select points 1, 2, 3, 4 by this row. The region has been created. Let's suppose, we want to hatch it.
We give **HATCH** command , distance = 0.25, angle = 0, inside = 1, border = 0. The result is shown in Figure B.

Example 2

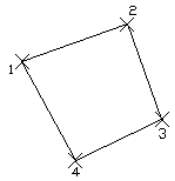


Figure C

If we give **PEDIT** instead of **HATCH** to the previous Example 1, the result is shown in Figure C.
The four-sided that has been created is a closed POLYLINE.

Notes

(1) There are many handling commands of the selected vertices (using **SELECTV**). These are **HATCH** and **PEDIT** we saw before, but also **AREA**, **COPY_IN**, **COPY_OUT**, **BORDER**, **CENTROID**, **RADIATE**.
(2) We use **LASTSV** (Last Selected Vertices) for the direct calling up of the last region we've created / handled. Reversely, we create selected vertices from a POLYLINE with **MAKESV** command.

MENU

Select	**Select vertices**

RELEVANT COMMANDS

<AREA>, <BORDER>, <CENTROID>, <COPY_IN>, <COPY_OUT>, <HATCH>, <LASTSV>,<MAKESV>, <RADIATE>

UNDO

NO (Stop with **CANCEL**=[ESC])

1.4.96 SET

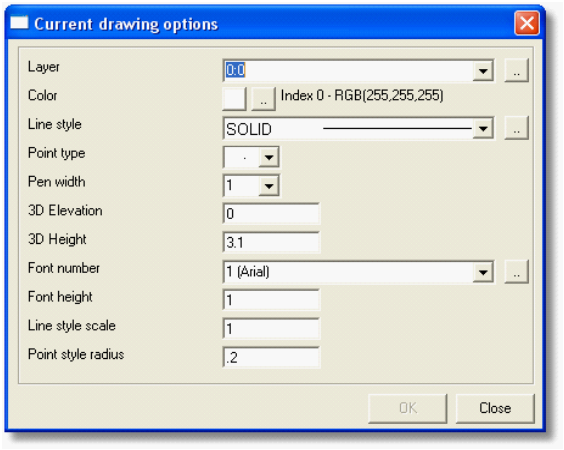
QUICK REFERENCE

By this command we specify the values of some variables, through a dialog box. These variables are always active and influence drawing e.g LAYER, COLOR, WIDTH, PEN WIDTH e.t.c.

SYNTAX

[Command] **SET**

© 1993-2026 Gardos Software LTD



[---> Layer=
Color =
... =] We specify the variables values.

PROCEDURE

During the time we are drawing, some parameters specify the way we are drawing. One of these is e.g. the current drawing LAYER, to which everything we are drawing is entered. What happens,though, if we want to change LAYER? Here comes **SET** command, to give us this ability. This command is simple. We give **SET** and to the dialog box we change everything we want.

RELEVANT VARIABLES

Tools	General Options	Pen

- → **'Layer' :** Specifies the drawing layer where the drawing objects will belong.
- → **'Color'** Specifies the layer's color.
- → **'Line style' :** Specifies the line style (e.g. SOLID).
- → **'Lines style scale ' :** Specifies hidden lines size, to lines by type.
- → **'Point type' :** Specifies the style of points' appearance (e.g. cross +).
- → **'Point style radius' :** Specifies points' size by type.
- → **'Pen width' :** Specifies pen's width.
- → **'Font number' :** Specifies text font appearance (e.g. Arial).
- → **'Text height' :** Specifies text height.
- → **'3D elevation' :** Specifies object's drawing elevation at axis Z.
- → **'3D height' :** Specifies objects' height at axis Z.
- → **'Screen pixel width in mm'** Specifies the drawing pixel width in mm.
- → **'Display pixel's width :** Specifies if pixel width will appear by scale.

RELEVANT COMMANDS

<SETUP>

UNDO

NO

© 1993-2026 Gardos Software LTD

1.4.97 SETGRID

QUICK REFERENCE

By this command we can specify the spaces DX and DY of the grids points and we activate or inactivate the grid on prompt.

SYNTAX

[Command] **SETGRID**
If there is not grid:
[---> DX =
DY =] We give spaces DX and DY of the grids points
If, on the contrary there is a grid, it disappears.

PROCEDURE

We give **SETGRID** command. If that moment the grid is active, giving the above command the grid disappears. That is, this command acts like **GRID_OFF**. If there is no grid, then by giving **SETGRID** a dialog box appears, where the values DX and DY (spaces between grids points) are asked.
By giving these two values, the grid appears. That is, this command acts like **GRID**.

© 1993-2026 Gardos Software LTD

RELEVANT VARIABLES

Tools	General Options	Snap

→ **'Grid' :** Specifies if the grid is active.
→ **'Grid by lines' :** Specifies if the grid appears by lines.
→ **'Dx' :** Specifies the spacing between the points of the grid to axis X.
→ **'Dy' :** Specifies the spacing between the points of the grid to axis Y.

RELEVANT COMMANDS

<GRID>, <GRID_ON>, <GRID_OFF>, <SNAP>

UNDO

NO

1.4.98 SETSNAP

QUICK REFERENCE

This command is similar to **SNAP** command, however, it operates reversely. That is, it cancels SNAP, if this is active.

SYNTAX

[Command] **SETSNAP**
If there is SNAP, then it is canceled
If there is no SNAP, then it operates as **SNAP**:
[---> DX =
DY =] We give spaces DX and DY, on which crosshair will be moving.

PROCEDURE

SETSNAP command is similar to **SNAP**. If SNAP (crosshair's step movement) is not active, then **SETSNAP** command acts like **SNAP**.
On the contrary, if SNAP is active, then the command inactivates the crosshair. That is, it is moving on prompt freely.
Spaces between DX and DY maintain their values, and they are going to appear on a dialog box, at the next **SNAP** or **SETSNAP** command.

RELEVANT VARIABLES

Tools	General Options	Snap

→ **Grid' :** Specifies if the grid is active.
→ **'Dx' :** Specifies the spacing between the points of the grid to axis X.
→ **'Dy' :** Specifies the spacing between the points of the grid to axis Y.

© 1993-2026 Gardos Software LTD

RELEVANT COMMANDS

<GRID>, <SNAP>

UNDO

NO

1.4.99 SETUP

QUICK REFERENCE

By this command, according to **SET** command, we set the values of more variables, which were necessary to be as parameters, so that the user will be able to intervene on them. These parameters affect the way of certain commands' operation and define some of the first typical values for all drawings.

SYNTAX

[Command] **SETUP**
[--->.
. =] We give values for variables. We select prompts variables using mouse. Pressing key [Apply] we accept all changes, regardless of what prompt they are on, and we return to the drawing. On the contrary using [Cancel] all previous variable values remain the same.

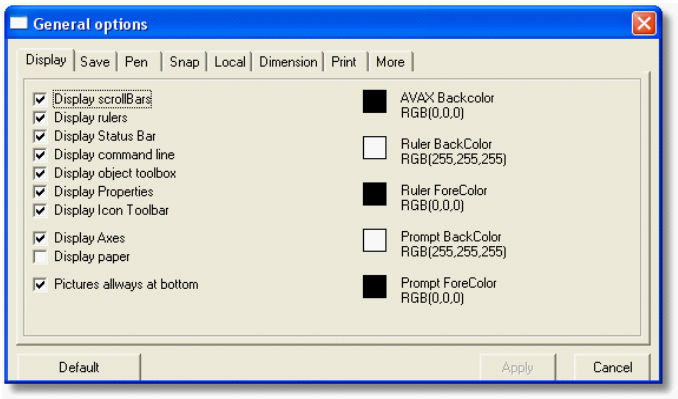
PROCEDURE

When we enter the program and open any drawing or we run **SETUP**, all the typical values of variables that are valid are defined. If we change variables values (on the dialog box that appears to some commands), while we are working on a drawing, then the new values are valid, e.g. **HATCH** command, variable ***"HATCH distance"***. When the program starts, this variable has its default value, from the corresponding variable in SETUP. If we then give **HATCH** command and change the value of variable ***"HATCH distance"***, the new value remains the same until we change it again by **HATCH** or **SETUP.** In the latter case, all the corresponding of ***"HATCH distance"*** variables, which had been changed during commands' running, are updated.

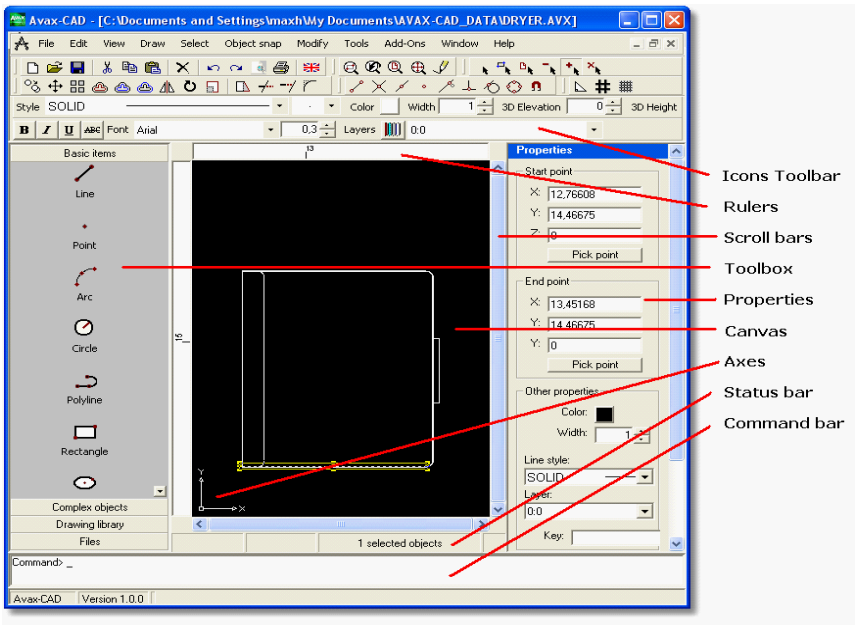
© 1993-2026 Gardos Software LTD

Tools	General Options	Display

1. It specifies the mode of programs and drawing area appearance.



It specifies if the following options will appear on the prompt:



- **'Pictures always at bottom' :** Specifies if pictures will appear always at the bottom (to the background).
- **'AVAX BackColor' :** Specifies AVAX backcolor.
- **'Ruler BackColor' :** Specifies ruler's backcolor.
- **'Ruler ForeColor' :** Specifies ruler's pen color.
- **'Prompt backcolor' :** Specifies command's line color.
- **'Prompt ForeColor'' :** Specifies command line pen color.

© 1993-2026 Gardos Software LTD

Tools	General Options	Save

- **'Automatic save every' :** Specifies if the program will save the drawing automatically during the design processing.
- **'minutes' :** Specifies the space of time in which the automatic drawing save will take place (e.g. every 10 minutes).
- **'Delete XML backup files' :** Specifies the deleting of XML backup files, which are saved by the program.
- **'XML backup folder' :** Specifies the computer's disk place where the drawing XML files will be saved.
- **'Exported picture width' :** Specifies the width of the export picture in pixels.
- **'Exported picture height' :** Specifies the height of the export picture in pixels.
- **'Exported picture back color' :** Specifies the background color of the export picture in pixels.
- **'Block's folder' :** Specifies the computer's disk place where the list of blocks will be.
- **'Outlook Express to send e-mail' :** Specifies if the send e-mail command of Outlook Express will be active.
- **'AVX file Associate' :** Specifies if program's connection command with files AVX type, will be active.
- **'AVX files folder' :** Specifies the computer's disk place where the list of AVX files will be.

Tools	General Options	Pen

- **'Layer' :** Specifies the drawing layer where the drawing objects will belong.
- **'Color'** Specifies the layer's color.
- **'Line style' :** Specifies the line style (e.g. SOLID).
- **'Lines style scale ' :** Specifies hidden lines size, to lines by type.
- **'Point type' :** Specifies the style of points' appearance (e.g. cross +).
- **'Point style radius' :** Specifies points' size by type.
- **'Pen width' :** Specifies pen's width.
- **'Font number' :** Specifies text font appearance (e.g. Arial).
- **'Text height' :** Specifies text height.
- **'3D elevation' :** Specifies object's drawing elevation at axis Z.
- **'3D height' :** Specifies objects' height at axis Z.
- **'Screen pixel width in mm'** Specifies the drawing pixel width in mm.
- **'Display pixel's width :** Specifies if pixel width will appear by scale.

Tools	General Options	Snap

- **AutoOsnap on' :** Specifies crosshair's snap points, with options : end, middle, near, perpendicular, intersection, tangent, quadrant, node.
- **'Snap on' :** Specifies if ruler's snap to grid will be active.
- **'Grid on' :** Specifies if grid will be active.
- **'Grid lines on' :** Specifies if grid will appear in lines.
- **'Dx' :** Specifies the spacing between the points of the grid to axis X.
- **'Dy':** Specifies the spacing between the points of the grid to axis Y.

Tools	General Options	Local

- **'Program language' :** Specifies the program language (Greek or English).
- **'Drawing language' :** Specifies the drawing text language.

Tools	General Options	Dimensions

- **Arrow size' :** Specifies dimension arrow size (to length).
- **'Arrow type' :** Specifies the arrow type (0 – line, 1 – arrow, 2 – circle).
- **'Decimals' :** Specifies the number of decimals of the dimension.
- **'Text' :** Specifies the dimension text.
- **'Distance' :** Specifies the dimension of distance from the straight line segment that is being dimensioned.
- **'Factor' :** Specifies the dimension increase factor.
- **'Inner lines' :** Specifies the protrusion of lines that define the dimension, to the straight line segment side.
- **'Outer lines' :** Specifies the protrusion of lines that define the dimension, to the straight line segment outside.

© 1993-2026 Gardos Software LTD

- **'Text above' :** Specifies if the text will appear above the dimension line.
- **'Horizontal text' :** Specifies if the text will appear horizontal.
- **'Vertical text' :** Specifies if the text will appear vertical.
- **'Text scale' :** Specifies the size of dimension text.
- **'Text font' :** Specifies the font type of the dimension text (e.g. Arial).
- **'Tick size' :** Specifies the tick size (line, arrow, circle) which limits the dimension.
- **'Scale' :** Specifies dimension size (text and arrow).
- **'Angular format' :** Specifies the format of angles (e.g. degrees).
- **'Dimension unit format' :** Specifies the format of dimensions (e.g. decimal).
- **'Fractional tick' :** Specifies the fractional tick when dimensions appear in fractional type.

Tools	General Options	Print

- **'Print to black' :** Specifies if the drawing's printing color will be black.
- **'Print all drawing' :** Specifies if the whole drawing is to be printed.
- **'Print at paper limits' :** Specifies if drawing printing will be within the chosen paper limits..
- **'Landscape' :** Specifies if the position of the paper during the printing will be horizontal.
- **'Printer pixel width in mm' :** Specifies if objects' width printing by scale will be active.
- **'Print pixel's width in mm' :** Specifies the pixel printing width of the drawing in mm, if the choice "Printer pixel width in mm' is selected.
- **'Print factor X' :** Specifies the scale factor at X.
- **'Print factor Y' :** Specifies the scale factor at Y.
- **'Paper margin at mm' :** Specifies perimetrically the margin of the drawing in mm, so as not to be printed at the edge.
- **'Paper RGB color' :** Specifies the color of the paper to be printed.
- **'1 printing unit at mm' :** Specifies the real printing dimension - correspondence to the drawing unit (e.g. if we choose that 1 drawing unit = 10 mm, the drawing will be printed by scale 1:100).
- **'Paper size' :** Specifies the size of the printing paper in mm.
- **'Paper width at mm' :** Specifies the width of the printing paper in mm.
- **'Paper height at mm' :** Specifies the height of the printing paper in mm.

'Tools	General Options	More

- **'Hatch'**
  - **'Angle' :** Specifies the hatch angle (e.g. 45o).
  - **'Distance' :** Specifies the distance between the hatch lines.
  - **'Mask' :** Specifies the hatch shape.
  - **'Inside' :** Specifies if the hatch will appear inside the chosen area.
  - **'Border' :** Specifies if the border will appear in case of hatching outside the object.
- **'More'**
  - **'Box selection PIXEL' :** Specifies the number of pixel of the rectangular box selection.
  - **'Crosshair PIXEL' :** Specifies the number of pixel of the crosshair.
  - **'Circle sections' :** Specifies the number of circle segments.
  - **'Mirror stay':** Specifies if the mirror command will be active.
  - **'Scale stay' :** Specifies if the scale command will be active.
  - **'Rotate stay' :** Specifies if the rotate command will be active.

RELEVANT VARIABLES

All the above variables (**GENERAL OPTIONS**)

RELEVANT COMMANDS

<SET>

© 1993-2026 Gardos Software LTD

1.4.100 SHELL

QUICK REFERENCE

By this command we can have a temporary exit to the operation system (DOS). The program remains open and we can return to it using DOS command **EXIT**.

SYNTAX

[Command] **SHELL**

PROCEDURE

Sometimes, while we are working on the program, we want to do some work outside of it without exiting. This operation situation is being succeeded using **SHELL** command (operation system shell). Then the "classic" prompt of DOS appears, where we can normally execute all DOS commands e.g. **DIR**, **COPY**, **DEL** e.t.c., or moreover load another program (which we recommend to be avoided). When we finish, we return to program using **EXIT** command.

Notes

(1) After **SHELL** command, program is being loaded in computers ram, even if it isn't obvious. That means, we have less available space for the execution of other programs. So, if we try to load another program, there is a possibility to be impossible, and DOS informs us with a relevant error message.

(2) Because of all the above, it isn't permitted after **SHELL** to give **AVAX** command, but we can return to it giving **EXIT** command.

(3) Before we execute **SHELL** command, it will be proper to save our drawing using **SAVE**, because, exiting temporarily to DOS , our P/C may "get stuck"(from a handling error) and we may lose the work we had done.

RELEVANT COMMANDS

<EXIT>

UNDO

NO

1.4.101 SNAP

QUICK REFERENCE

This command defines the horizontal and vertical step movement of crosshair and in combination with **GRID** command, can give enough help to drawing.

SYNTAX

[Command] **SNAP**
[---> DX =
DY =] We give the step DX and DY to drawing units.

PROCEDURE

By **SNAP** command and DX, DY positive e.g. DX=1 DY=1, crosshair moves on drawing points the distances between which are DX and DY. If we also have the grid visible with DX, DY GRIDS equal to DX, DY of SNAP, then the crosshair moves only on the points of the grid, thus facilitating drawing.
When we want to terminate the movement step of crosshair, we give again **SNAP** command, with DX=0 DY=0 (or DX=DY=0).

RELEVANT VARIABLES

© 1993-2026 Gardos Software LTD

Tools	General Options	Snap

- **'Snap on' :** Specifies if ruler's snap to grid will be active.
- **'Grid on' :** Specifies if grid will be active.
- **'Dx' :** Specifies the spacing between the points of the grid to axis X.
- **'Dy':** Specifies the spacing between the points of the grid to axis Y.

RELEVANT COMMANDS

<GRID>, <SETSNAP>

UNDO

NO

1.4.102 STRETCH

QUICK REFERENCE

By this command we can have a result similar to "stretching" of an object. Mainly we deform the selected object, moving some of its vertices, while others remain stable. After giving the command, all objects connections remain stable.

SYNTAX

[Object selection] with **WINSEL**
[Command] **STRETCH**
[First Point] Start of moving vector.
[Second Point] End of moving vector

© 1993-2026 Gardos Software LTD

PROCEDURE

This command is useful to change some objects' relative positions, retaining, however, their local relations (connections, shape). Its use with **WINSEL** command (objects' selection with window selection as many as are inside and those which intersect the window selection) is successful. In this case the selected objects which have a part of them outside the window, will keep the vertices that are outside the window selection stable, while they'll move the internal vertices to vector's direction and distance. The latter is valid for those objects that are whole inside the window selection, as well. For command's application it is required to select objects and define movement vector.
Example
We have Figure A and we want to change it as it is in Figure B:

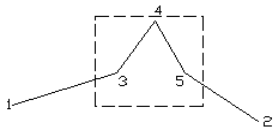


Figure A

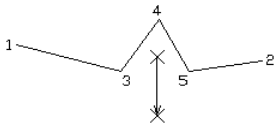


Figure B

We select the object using **WINSEL** command, as it is shown in Figure A.
We give a reference point and a second point down and right and we have the result, as it is shown in Figure B. Both objects retain their vertices 1 and 2 stable, while vertices 3, 4 and 5 have been moved. In spite of these, segment (3)-(4)-(5) is the same in both cases, but in another position.

ICON



MENU

Modify	**Deformation**	**Stretch**

UNDO

YES

UNDO

YES

1.4.103 TANGENT

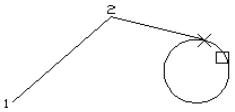
QUICK REFERENCE

We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a line's segment). The purpose is the selection of a POINT which is on a circle's periphery and, along with another given point, it defines a line segment tangent to the circle. The circle is being indicated by giving a point on it.
and as the tangent point is selected from the program, we use it to end the active command.

SYNTAX

[Command e.g. LINE)
[Command] **TANGENT**
[Point on a circle] We give a Point ON circle's periphery towards the side we want the tangent line to be.
[We continue with the other command e.g. LINE]
Example
Figure

© 1993-2026 Gardos Software LTD



We want to have a tangent line segment from point 2, to the circle.
-We give **LINE**,
-We give **ENDPOINT**,
-We give a point NEAR vertice 2,
-We give **TANGENT**
-We give a point ON circle's periphery,
-We cancel Draw command with **CANCEL**.

ICON



MENU

Object Snap	**Tangent**

RELEVANT COMMANDS

<ENDPOINT>, <INTERSEC>, <MIDPOINT>, <NEAREST>, <NODE>, <QUADRANT>, <VERTICAL>

UNDO

YES

1.4.104 TEXT

QUICK REFERENCE

By this command we can merge texts in our drawings. It is a very flexible command, as it allows us to write as many text lines as we want, directly on our drawing.

© 1993-2026 Gardos Software LTD

SYNTAX

[Command] **TEXT**
[---> text:] We write our text (if we want to change line, we press **ENTER**).
[Reference point] We give a Point and our text is placed there, with all current options (scale, text font, color e.t.c.).

PROCEDURE

We give **TEXT** command and a dialog box appears, where we can write a text, as big as we want (if we want to change line, we press **ENTER**). When we finish the text, we press **OK.** At the end, we give a point in the drawing, where the text is inserted.
Note
If text is big, during insertion we can stop its appearance on prompt, pressing **ESC**. Actually, the text has been inserted, but its whole appearance was simply cancelled. Giving the next **REDRAW**, the whole will appear.

ICON


MENU

Draw	**Text**

RELEVANT VARIABLES

Edit	**General Options**

→ **'Font' :** We specify text font (e.g. Arial).
→ **'Text height' :** We specify text height.

RELEVANT COMMANDS

<CHANGE_FONT>, <CHANGE_TEXT>

UNDO

YES

1.4.105 THAW

QUICK REFERENCE

This command activates a "frozen" LAYER (by **FREEZE** command). After **THAW** command, all objects of this LAYER, are becoming visible and processing again. It is the opposite of **FREEZE** command.

© 1993-2026 Gardos Software LTD

SYNTAX

[Command] **THAW** ("thaw" the current **LAYER**)

PROCEDURE

This command is very simple. It operates exactly like **FREEZE**, with the opposite result. We give **THAW** and appear all objects of the current **LAYER** immediately**.**

Note

In order to "thaw" another **LAYER** (not the current one) we change the current **LAYER** to the **LAYER** we want to thaw (and make this current), then we give **THAW** command and we change again the current **LAYER** giving the initial. The change of the current **LAYER** takes place using **SET** command.

Example

The current **LAYER** is "0" and we want to activate "1" layer.
We give **SET** command. We change **LAYER** 0 to 1, on the relevant dialog box.
We give **THAW** command. All objects of **LAYER-1** appear again.
We bring back the initial **LAYER**, giving **SET** command.

ICON



RELEVANT VARIABLES

Edit	**General Options**

→ **'Layer' :** Specifies the drawing layer where the drawing objects will belong.

RELEVANT COMMANDS

<FREEZE>, <SET>

UNDO

YES

1.4.106 TRIM

QUICK REFERENCE

By this command, we can cut off parts of line segments, using other line segments, as leaders – boundaries.

SYNTAX

[Object selection] We select a boundary line segment (**SELECT**)
[Object selection] We select a boundary line segment (**SELECT**)

© 1993-2026 Gardos Software LTD

[Command] **TRIM**
[Object selection] We select the line segment we want to cut off (**SELECT**)
[Object selection] We select the line segment we want to cut off (**SELECT**)
[Cancel] We stop with **CANCEL**

PROCEDURE

First we select (with **SELECT**) one by one all line segments we want to be cutting off boundaries. We give **TRIM** command and then (with **SELECT** again one by one) we give the line segments we want to cut off. At the end we stop using **CANCEL**.
Example

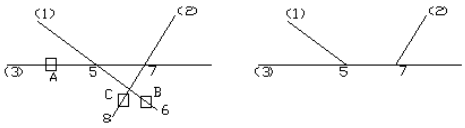


Figure A Figure B

We want to cut off segments 5-6 and 7-8 of line segments (1) and (2) of Figure A. We use segment (3) as cutting boundary
We select the cutting boundary with **SELECT** to the point (A).
We give **TRIM** command.
We select the first segment (1) which is going to be cut off with **SELECT**, to point (B).
We select the second segment (2) which is going to be cut off, with **SELECT**, to point (C).
We stop with **CANCEL**.
The result is shown in Figure B.
Notes
(1) We can use **TRIM** command with POLYLINES as boundaries.
(2) Cutting off is being applied only in line segments.

ICON



MENU

Modify	**Deformation**	**Trim**

RELEVANT COMMANDS

<FILLET>

UNDO

YES

1.4.107 UNDO

QUICK REFERENCE

By this command we can reverse the commands that we have already given (only those that can be reversed - UNDO), in the reverse turn that they have been given, taking constantly the drawing back to its previous condition.

SYNTAX

[Command] **UNDO**

PROCEDURE

© 1993-2026 Gardos Software LTD

Giving the UNDO command we recall the most recent command we've given, on the condition that this recalling is allowed. Giving again UNDO, we recall the exact previous command e.t.c. After giving the UNDO command, drawing is coming back one step backward, while prompt is getting automatically renewed, showing the new, each time, drawing.
Example
We give command LINE from point (0,0) to point (1,1).
We give CANCEL (not another line).
We give UNDO. The line we've already given is being erased.
Notes
(1) Program undoes 2000 commands. That means that it can recall by UNDO, each time, the 2000 last operations that can be recalled.
(2) When we close a drawing and then we open it again, UNDO is not valid for commands we had given before we closed it.
(3) We can regulate in **SETUP**, if **ZOOM** command will have or not the ability of UNDO.

ICON



MENU

Edit	**Undo**

KEYBOARD

Ctrl + Z

RELEVANT COMMANDS

<REDO>

UNDO

NO

1.4.108 UNSELECT

QUICK REFERENCE

UNSELECT command cancels the selection of all selected objects and stops the selection process.

SYNTAX

[Command] **UNSELECT**

PROCEDURE

This command is being given only when we are in an objects' selection condition. For example, if we had given **SELECT** command, and we had selected some objects, when we give **UNSELECT** command, all selected objects are being unselected and the process is being canceled.
Notes
(1) The same thing happens if we give **CANCEL**.
(2) With **UNSELECT** the ability to recall the last selected vertices (**LASTSV** command) is lost.

ICON



MENU

© 1993-2026 Gardos Software LTD

Select	Cancel Selection

RELEVANT COMMANDS

<INWINSEL>, <SELECT>, <SELECTSUB>, <SELECTADD>, <SELECT_ALL>, <WINSEL>

UNDO

NO

© 1993-2026 Gardos Software LTD

1.4.109 USERLIMITS

QUICK REFERENCE

By this command we can place the down left and up right corner co-ordinates of the drawing area, in order to have a concrete view of it.

© 1993-2026 Gardos Software LTD

SYNTAX

[Command] **USERLIMITS**
[---> Xmin
Ymin
Zmin
Xmax
Ymax
Zmax =] We give the down left corner (min) and the up right corner (max).

PROCEDURE

We give **USERLIMITS** command and to the dialog box that appears, we give the co-ordinates of the down left corner of the drawing area (Xmin, Ymin, Zmin). Program has already some default values, which we either accept or change. In the same way we change the co-ordinates of the up corner (Xmax, Ymax, Zmax). Drawing is being redrawn with the new co-ordinates, so as to be accordant to the new drawing limits. (Actually, the drawing size doesn't change – the only thing that changes is the size of drawing's part which appears on prompt).

MENU

View	**User limits**

RELEVANT COMMANDS

<AUTOLIMITS>, <CALCEXTENDS>, <EXTENDS>, <ZOOM>

UNDO

NO

1.4.110 VERTICAL

QUICK REFERENCE

We use this command in combination with another command that is already active (e.g. **LINE** – drawing of a line's segment). The purpose is the selection of an object's point, vertical to another point. The object is being shown by the user, giving a point NEAR it. The point that is selected by this way VERTICALLY and ON the object, is useful to end the active command.

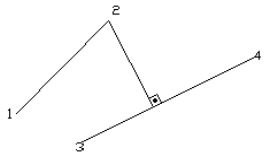
SYNTAX

[Command] e.g. **LINE**
[Command] **VERTICAL**
[Reference point near the object] We give a point on the object (e.g. to a straight line segment)

[Continue with the other command] e.g. **LINE**

Example
Figure

© 1993-2026 Gardos Software LTD



We want to have a vertical line to line segment 3-4, from vertex 2.
We gvie **LINE**,
We give **ENDPOINT**,
We give a point NEAR vertex 2,
We give **VERTICAL**,
We give a point NEAR segment 3-4
We stop with **CANCEL**.

ICON



MENU

Osnap	**Vertical**

RELEVANT COMMANDS

<ENDPOINT>, <INTERSEC>, <MIDPOINT>, <NEAREST>, <NODE>, <QUADRANT>, <TANGENT>

UNDO

YES

1.4.111 WALL

QUICK REFERENCE

By this command we draw a wall which has a defined width.

SYNTAX

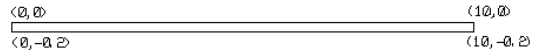
[Command] **WALL**
[---> Distance =] We give the drawing width of the wall (in drawing units).
[Start point] We give the start point of a straight line segment.
[End point] We give the end point of a straight line segment.
[Direction point] We give a point that shows the side (direction) where the other side of the wall will be drawn, at the defined distance (the first line of the wall, has been defined by the two first points).

PROCEDURE

We give **WALL** command and the distance between internal and external side of the wall. Then we give the start and the end of the one side, giving start point and end point.

© 1993-2026 Gardos Software LTD

What is left is to give a third point, where the other side (parallel to the first one)will pass from.
Note
Wall is a closed POLYLINE.
Example
We give **WALL** command
We give distance 0.2
We give the first point (0,0), the second point (10,0)
We give the third point (10,-1) and we have the Figure below:



ICON



MENU

Draw	**Complex objects**	**Wall**

RELEVANT COMMANDS

<RECTANGLE>, <COPY_IN>, <COPY_OUT>, <POLYGON>, <POLYLINE>

UNDO

YES

1.4.112 WINSEL

QUICK REFERENCE

By this command we can select all objects, at least a part of which is inside an objects' selection window. Then we can process the selected objects anyway we want (e.g. moving to another place).

© 1993-2026 Gardos Software LTD

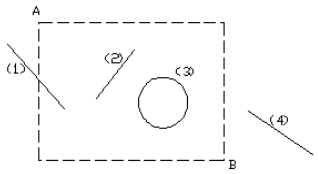
SYNTAX

[Command] **WINSEL**
[First Point of window selection] We give a point on the drawing which is the first point of the window selection.
[Second Point of window selection] We give a second point which, together with the first point defines the diagonal of the window selection. Objects that have at least a part in that window, have been selected.
[... we continue with another handle – command of the selected objects].

PROCEDURE

It is a command similar to **INWINSEL** command. Their basic difference is that it selects more objects. That is because it selects not only these objects which are WHOLE inside the window selection (that INWINSEL also selects) but those which are being intersected from the window selection, as well. So the process is as follows:
We give **WINSEL** command
We give the first point of the objects' selection window
We give the second point of the objects' selection window
(We define by this way a diagonal of the objects' selection window with sides parallel to X and Y axis.
Now all objects that are wholly or partly inside the window, are being selected.
We can continue selecting other objects (giving again a selecting objects' command such as **WINSEL** or **INWINSEL** or **SELECT**), or handle in some way the selected objects (e.g. **MOVE**, **COPY**, **ERASE** e.t.c.). if we want to stop, we give **CANCEL**.

Example (correspondent to **INWINSEL**)



We have the objects of the Figure
W give WINSEL
We give the a' Point A of the window
We give the β' Point B of the window
Objects (1), (2) and (3) are selected.

ICON



MENU

Select	**Window select**

RELEVANT COMMANDS

<INWINSEL>, <SELECT>, <SELECTADD>, <SELECTSUB>, <UNSELECT>

UNDO

YES

1.4.113 ZOOM

QUICK REFERENCE

By this command we "bring" visually closer an area of our drawing, so we can see more details, that is we 'enlarge' it to look exactly like it would be if we were looking at the drawing through a magnifying glass. The real size of the drawing remains the same. Then we can bring back the drawing to its previous condition ("keep back") by PREVIOUS command.

SYNTAX

© 1993-2026 Gardos Software LTD

[Command] **ZOOM**
[First Point] We give the first Point of the drawing area we want to magnify.
[Second Point] We give the second Point of the area, on the diagonal.

PROCEDURE

We give ZOOM command and then two points which define a window, into which there is the area we want to see better. This area is being enlarged, overtaking all the available prompts' drawing area. Then we can work there with emphasis on details and later on, we can either come back to the previous condition (with **PREVIOUS**) or give another **ZOOM**.

ICON

MENU

View	**Zoom**

RELEVANT COMMANDS

<PREVIOUS>, <ZOOMUNDO_ON>, <ZOOMUNDO_OFF>, <MAGNIFY>

UNDO

YES, if **ZOOMUNDO_ON** command has preceded.

1.4.114 ZOOMUNDO_ON - ZOOMUNDO_OFF

QUICK REFERENCE

When we recall giving **UNDO** command, **ZOOM** command is being recalled based on the value of variable ***"UNDO and ZOOM"*** (0=NO UNDO, 1=YES UNDO).
The same happens with **ZOOMUNDO_ON** command, which gives us the ability to recall **ZOOM**. We succeed the opposite, using **ZOOMUNDO_OFF** command which recalls the ability of **ZOOM UNDO.**

SYNTAX

[Command] **ZOOMUNDO_ON** or **ZOOMUNDO_OFF**

RELEVANT COMMANDS

<UNDO>, <ZOOM>
UNDO

NO

1.4.115 CHANGE_3DELEVATION - CHANGE_3DHEIGHT

QUICK REFERENCE

By **CHANGE_3DELEVATION**, **CHANGE_3DHEIGHT** commands, we can change an object's elevation start and height.

SYNTAX

[Objects' selection] We select the objects whose elevation or height we want to change.
[Command] **CHANGE_3DELEVATION** (or) **CHANGE_3DHEIGHT**
[Dialog box] We give the object's elevation or height respectively.

© 1993-2026 Gardos Software LTD

PROCEDURE

By **CHANGE_3DELEVATION** command, object takes the elevation which is defined to the dialog box, by the user. With **CHANGE_3DHEIGHT** command, the object takes the height which is defined to the dialog box, by the user. For example, if we change a circle's height we will have a cylinder, a point's height we will have a line parallel to Z axis and a line's height we will have a parallelogram. First we select all objects whose elevation or height we want to change. Then we give **CHANGE_3DELEVATION** command or **CHANGE_3DHEIGHT** and we give to the dialog box, where it is asked, the elevation or the height respectively.

MENU

Modify	**Change**	**3D ELEVATION**

RELEVANT COMMANDS

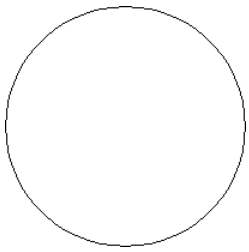
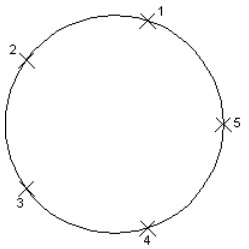
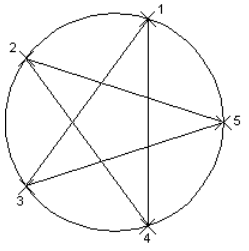
<CHANGE_COLOR>, <CHANGE_LTYPE>, <CHANGE_PTYPE>,<CHANGE_TEXT>,<CHANGE_FONT>, <CHANGE_LAYER>, <CHANGE_WIDTH>, <SET>, <SETUP>

UNDO

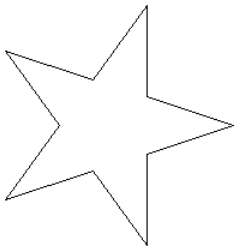
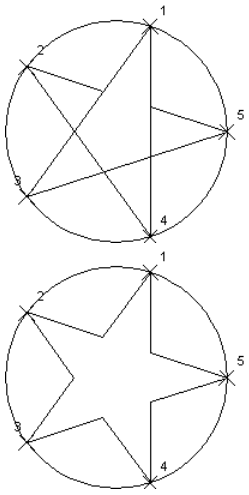
YES

1.5 EXAMPLES

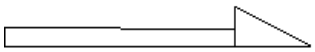
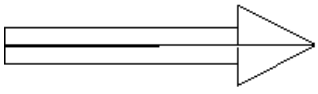
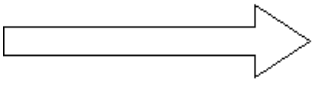
Stars construction

1. We create a circle Giving to the command line, circle command or selecting from the basic menu <Draw>-Circle and giving its center and radius.	
1. Divide the circle to 5 equal arcs. We select the circle, and from the basic menu we select <Modify>-<Union-Split>-Divide. Then we give to the dialog box Divide number=5, and five new points are created, which divide circle to 5 equal arcs.	
1. We create the helping lines of the star with line command. We select «Object snap - node» or from menu «Auto Snap» the selection «Point» and we create lines between vertices 1-3, 3-5,5-2 2-4.and 4-1.	

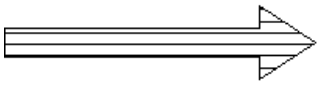
© 1993-2026 Gardos Software LTD

1. A' way We select from «Auto Snap» menu the selection «Point» and «intersection» and we draw a polyline (polyline command, or <Draw>-polyline) which is the outline of the star, when we connect all vertices and intersection points of the helping lines. Then we select the helping points, lines and the circle and we erase them using ctrl+delete command, and thus the star remains.	
1. B' way We select two by two the helping lines e.g. we select 1-3 and 1-4. We go to the basic menu <Modify><Deformation> and we select «Trim». We select the segment of helping line 2-5 which is between lines 1-3 and 1-4 and the segment is cut out. Then we press esc so we stop the cutting out process. We do the same for all the other helping lines. At the end we erase the circle and the helping points, and the star is ready.	

Arrow creation

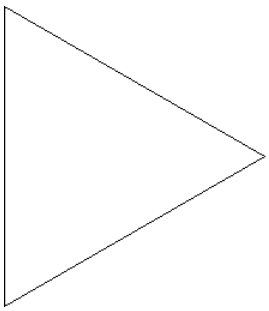
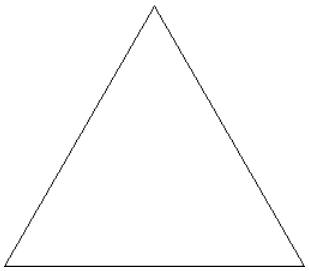
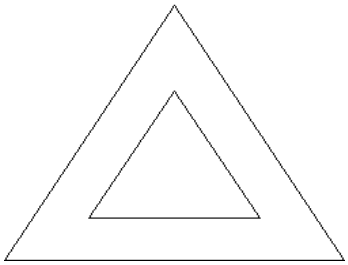
1. We create a line With line command (on command line) or selecting from the basic menu <Draw>- Line and giving the start and the end point of it.	
1. We create two lines vertical to the first line. On menu "Auto Osnap" we check the selection "nearest" and "endpoint". Using Line command we draw a line vertical to the down part of the arrow and a smaller line to the end of the big line.	
1. We create a vertical line from the end of the small vertical line to the big vertical line and from the end of the big vertical line to the end of the big line.	
1. We select with a window all the above lines. We select from the basic <Modify>-<Transformation>-Mirror and we give as "mirror" axis the big line (giving two points on it).	
1. We make the border of the arrow creating a polyline and we erase all helping lines. What is left is its border.	

© 1993-2026 Gardos Software LTD

1. Hatching arrows area. We select the arrow polyline. From the basic menu we select <Draw>-<Complex objects>-<hatch> and the internal area of the arrow is hatched.	

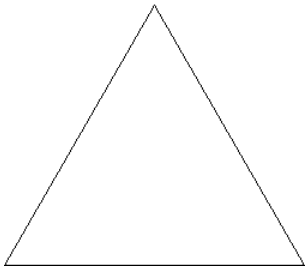
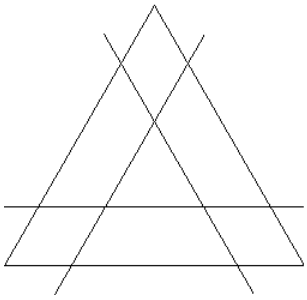
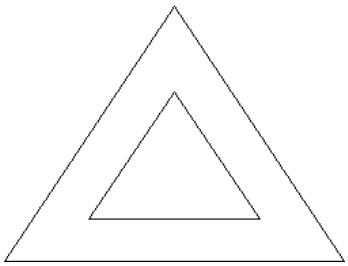
Triangle creation

a' way

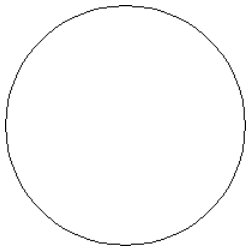
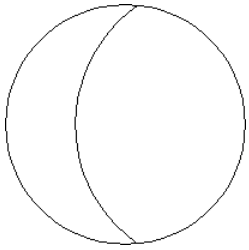
1. We create a triangle with Command polygon (to the command line) or selecting <Draw>-Polygon from basic menu. We give to the dialog box the sides number n=3 and to the command line the side's length of the equilateral triangle e.g. 5. An equilateral triangle with side length 5m is created.	
1. We rotate the triangle to 90°. We select the triangle, and to the command line we give rotate command or we select <Modify>-<Transformation>-Rotate, from the basic menu. We give the rotation center and rotation angle 90°.	
1. We select the triangle and we also select <Modify>-<Copy>-Copy offset in, from the basic menu. To the dialog box we give the inside copying distance e.g. 1.00 and the triangle is ready.	

© 1993-2026 Gardos Software LTD

β' way

1. We create a triangle We create a line, using Command line, and we select it. From <Complex objects> we select Triangle. We give the direction where the equilateral triangle will be created, which will have side's length equal to line's length.	
1. We select the triangle and we select <Modify>-<Copy>-Copy offset. We give the distance to the dialog box e.g. 1.00 and the direction where we want to have the offset lines.	
1. We select two by two the triangle sides and we also select <Modify>-<Deformation>-Fillet, from the basic menu. The triangle is ready.	

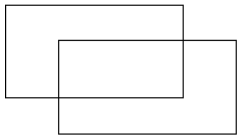
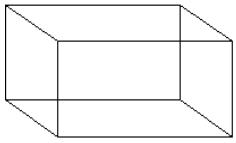
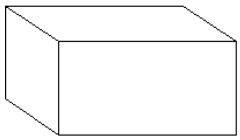
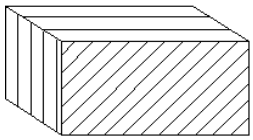
Half-moon creation

1. We create a polygon, using polygon command, with a big number of sides e.g. 60.	
1. We create an arc using arc Command giving the start Point, arc's center and the end point. The start and the end point are the polygon's vertices.	

© 1993-2026 Gardos Software LTD

1. We select the polygon which is polyline and we also select <Modify>-<Union – Split>-Explode and polyline is separated to 60 line segments. Then we select and erase, with ctrl+delete, line segments which are left over. The half-moon is ready.	

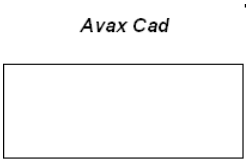
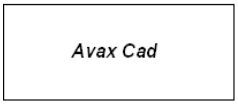
Rectangular solid creation

1. We create a rectangular solid Using rectangle command (to the command line) or selecting <Draw>-rectangle, from basic menu.	
1. We create a copy of the above rectangle. We select the rectangle, and we also select <Modify>-<Copy>-Copy from the basic menu and we give the start and the end point of the copy.	
1. We join the vertices of the two rectangles creating lines using line command.	
1. We create a polyline for every visible side of the rectangular solid. Then we erase all helping lines with ctrl+delete command.	
1. Rectangular solid sides hatching. We select successively the sides. We also select <Draw>-<Complex objects>- Hatch from the basic menu and the sides of rectangular solid are hatched. To the dialog box, we select 45° angle for the front side, 90° angle for the left side and 0° angle for the top side.	

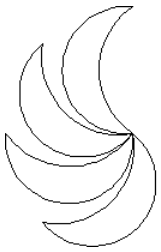
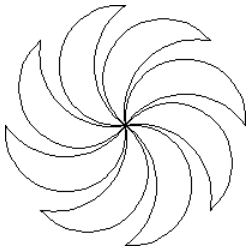
Text in the center of a rectangle

1. We select <Edit> «Grid» from the basic menu and we give dx and dy distances e.g. dx=2 and dy=2, for the grid which will be created on prompt.	
1. We select <Edit> «Snap to grid», from the basic menu. Now the cursor will be pulled by the grid's vertices.	
1. With rectangle selection and with the help of the grid points we create a rectangle parallelogram 10.00X4.00.	

© 1993-2026 Gardos Software LTD

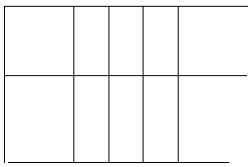
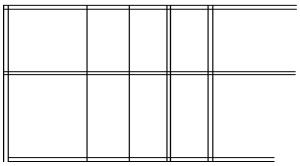
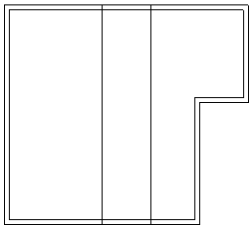
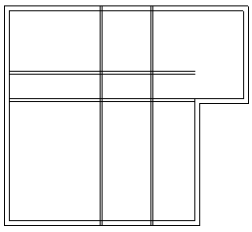
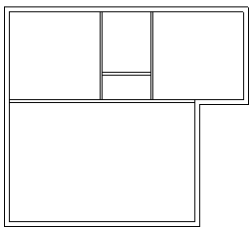
1. At a casual point of the rectangle we write a text. We select <Draw>-«Text» from the basic menu. A form opens where we write the text e.g. Avax –Cad, we press ok and then we give the text's insertion point.	
1. We first select the text. Then we can give text height, font type or we change the text. Also we can underline it, make it bold, or italics or striked out. All the above operations are in the Properties menu on the right side of the prompt.	
1. We select the text we want to align, and keeping the Shift key pressed, we select the parallelogram. Then we select <Modify>-<Alignment> - Center and the text is placed in the center of the rectangle.	

Wind mill creation

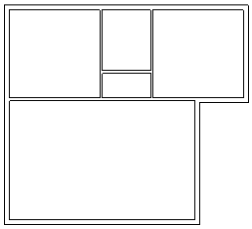
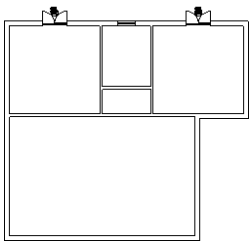
1. We use the half-moon, which we have already created, in a previous example.	
1. First we give <Tools>-<General Options>-<More> and we check the selection «Mirror Stay». We select the half-moon and we give"rotate" to the command line or we select <Modify>-<Transformation>-Rotate, from the basic menu. We give as the rotation center the down edge of the half-moon, which will be the center of the mill and to command line rotation angle #45,1. As a result, the original half-moon and its rotation will remain in the drawing.	
1. We select the two half-moons and we give Rotate command, to the command line or we select <Modify>-<Transformation>-Rotate, from the basic menu. We give the rotation center and to command line, rotation angle #90,1.	
1. We select the four half-moons and we give Rotate, to the command line or select <Modify>-<Transformatio>-Rotate, from the basic menu. We give the rotation center and to command line, rotation angle #180,1. The wind mill is ready.	

A ground plan creation

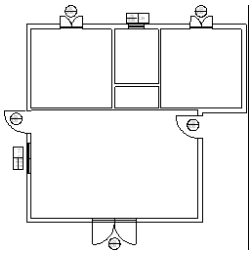
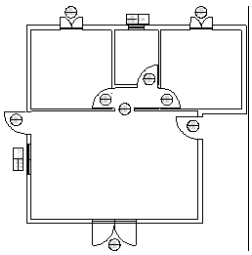
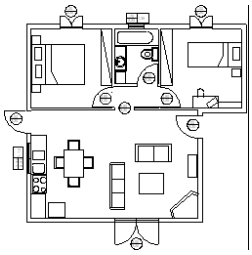
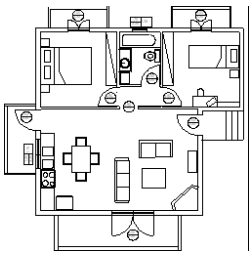
© 1993-2026 Gardos Software LTD

1. We create the basic grid of the ground plan. We draw a line with Line command. We select <Modify>-<Copy> and «copy offset» from the basic menu. We copy the line to distance 4 and 9 m. Then we create vertical line to the start points of the lines. We select the vertical line and with "copy offset" we copy it to distance 4, 6, 8, and 10m.	
1. We select all lines which consist the exterior sides of our plan, and we copy them to distance 0.20m. with direction inside it, so we create the exterior walls. We select the above lines and from <Modify> <Copy> we select «copy offset». To the dialog box we give dx=0.20.	
1. We fillet the exterior sides of our plan. We select two by two the exterior sides of our plan and we give <Modify>-<Deformation>-«Fillet» and the two sides are cut off to their intersection point.	
1. We create the interior walls. With Line command we create lines and with Offset command we copy them to the direction we want (copy offset) to distance 0.1m. (which will be the width of the interior walls).	
1. We cut off the segments of the interior walls, which are useless. We select the cutting edge, and using Trim command or selecting <Modify>-<Deformation>-Trim, we select the segments which we want to be cut off.	

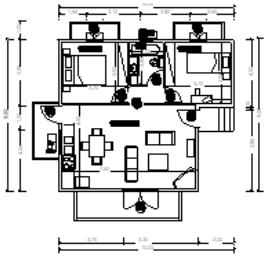
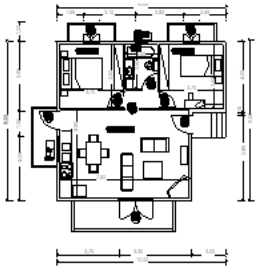
© 1993-2026 Gardos Software LTD

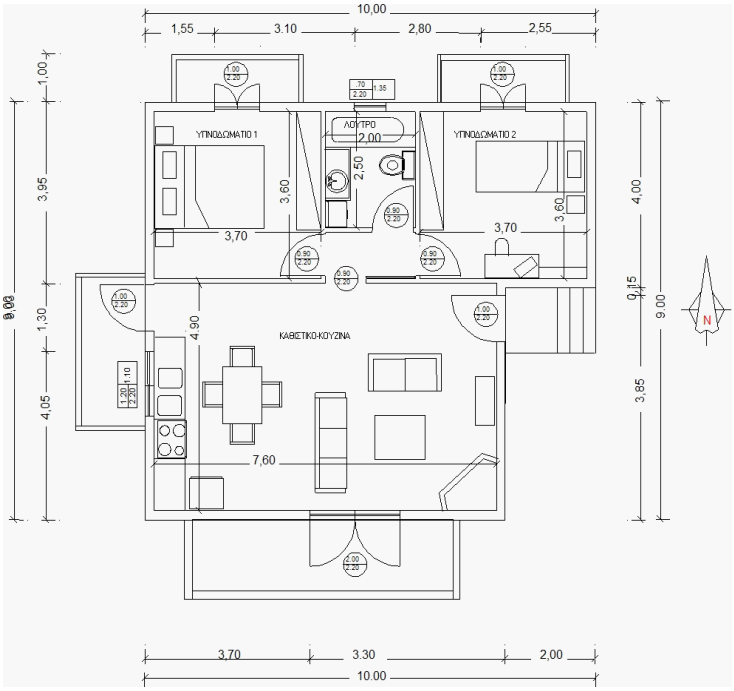
1. To the intersection points of two vertical walls, we select the two segments (keeping the Shift key pressed) and with Trim command we cut off the party segment of the other wall.	
1. Exterior openings' insertion. We create an exterior opening, in the up left room (bedroom) , with dimensions 1.00X2.20. We create a vertical line segment on the wall where the midpoint of the opening will be placed e.g. to the midpoint of the rooms wall (interior side). With «copy offset» we copy the above line segment to distance e.g. 0.50 in order to create an opening with total length 1.00m. Using Line command we create a line segment parallel to the wall between the three above segments, which will be the interior side of the window sash. We copy the above line segment to distance e.g. 3cm as the width of the window sash is. Using as center the left endpoint of the window sash and radius R=0.50 we create an arc (quadrant) towards the center of the opening. We select the above arc, with Extend command (<Modify>-<Deformation>-<Extend>) we extend the left endpoint of the opening until the end of the arc. So we have created the first opening. Then we select the above half opening and <Modify>-<Transformation >-«Mirror» and giving as first copy point the midpoint of the opening and second point, a casual vertical one, we create the other half opening. Then using circle command we create a small circle near the midpoint of the opening, a line segment on the midpoint of the opening and with Text command we write the opening's dimensions (up the width and down the height). So we have created the opening and its label.	
1. We select the above opening. We give <Modify>-<Union-Split> "Create block". We give block's name e. g. porta and its insertion point e.g. the midpoint of the opening. Then, as the opening of the other bedroom is the same as the above, we give <Draw>- "Insert Block" and we insert the block "porta" to the midpoint of the second rooms wall. At the end we select the door, which we have inserted, and with Trim command we cut off the wall between the ends of the opening.	

© 1993-2026 Gardos Software LTD

1. Using line, arc, offset, trim e.t.c. commands, as above, we create all the exterior openings and its labels.	
1. Similarly, we create the interior openings of our ground plan.	
1. We place furniture in the rooms. We either select prepared drawings giving <Draw>-Drawing library, or create them by ourselves. E.g. in the left bedroom using rectangle command we create a cupboard, a double bed and two bed side tables. We can make all these furniture blocks, so we will have them already prepared to use in other drawings. In the other bedroom, in the same way, we create a single bed, a desk, an office chair and a bookcase. We copy from the first bedroom the cupboard and the bed side table using Copy command. We insert from the "drawing library" all the bath furniture for the bathroom. We select the kind of the «bath-furniture», the rotation angle in relation with the preview and an insertion point. This point can be a prompt's casual point. The object which is inserted is already selected and we can move it to another place, using Move command. For the kitchen we select from the "drawing library", electric cooker and refrigerator and we draw a kitchen-sink, which we can save as block, so that we can use it in other drawings. We also select furniture for the living room from the "Drawing library".	
1. We create balconies and entrance stair, using Line and Rectangle commands.	
1. We create polygons inside the balconies using Rectangle command and we hatch them using Hatch command.	

© 1993-2026 Gardos Software LTD

1. We successively select, all the exterior and interior line segments of the plan, and we give them dimensions using Dim command or <Draw>-<Complex objects>-«Dimension».	
1. Using dtext command or <Draw>-«Text» we write rooms' names e.g. bedroom, bathroom, e.t.c..	
1. We select <Draw>-<Drawing library> from the basic menu and we insert to our plan, the symbol of North, from drawing library.	
1. If we want to have the area of our plan, we create a polyline on the outline, then we select it and give <Modify>-<Measurement>-«Area», to the basic menu and we take the information that this area is 79.95m2.	



© 1993-2026 Gardos Software LTD

1.6 APPENDIX

1. KEYBOARD'S SHORTCUTS

KEY	COMMAND	PROCEDURE
Ctrl+N	**-**	Opens a new drawing
Ctrl+O	**OPEN**	Opens a drawing using file
Ctrl+S	**SAVE**	Saves the current drawing to a file
Ctrl+P	**PRINT**	Prints the current drawing
Alt+F4	**EXIT**	Program exit
Ctrl+Z	**UNDO**	Undoes the previous command
Ctrl+X	**CUT_AVAX**	Cuts off selected objects
Ctrl+C	**COPY_AVAX**	Copies selected objects
Ctrl+V	**PASTE_AVAX**	Pastes selected objects
Ctrl+Delete	**ERASE**	Erases selected objects
Esc	**CANCEL**	Action's cancel
F2	**ZOOM_IN**	Drawing's magnifying
F3	**ZOOM_OUT**	Drawing's diminution
Ctrl+A	**SELECT_ALL**	All objects' selection
F1	**HELP**	Program help

2. CO-ORDINATES SETTING FROM COMMAND LINE

For some commands the determination of a point is necessary e.g. POINT, LINE, COPY, MOVE e.t.c. This point can be set from command line, too. Co-ordinates' direct setting can be done using one of the following ways:

- Using absolute co-ordinates
  Point's co-ordinates (x,y,z) are given directly to the command line.
  Syntax : Command > x,y[,z]
  e.g. Command > 0,0 : Point to position (0,0,0)
  Command > 0,3,1 : Point to position (0,3,1)
- Polar co-ordinates
  The angle and the absolute distance are given, from the previous point (or from point (0,0,0) when there is no previous point).
  Syntax : Command > #a,d
  Where, a: the counter-clockwise angle to degrees and d: the absolute distance from the previous point,
  e.g. Command > #45,5 :The point distant 5 drawing units from the previous point and the angle between the points is 45o.
- Relative co-ordinates
  The point's relative distances from the previous point (or from point (0,0,0) are given, when there is no previous point)
  Syntax : Command > @dx[,dy][,dz]
  Where, dx : the relative distance from the previous point, to axis x
  dy : the relative distance from the previous point, to axis y
  dz : the relative distance from the previous point, to axis z
  e.g. Command > @1,5,0
  If the previous point has co-ordinates (5,3,0) then giving the above command, the new point will be in the position (6,8,0)

Note: [z], [dy], [dz] are given optionally.

© 1993-2026 Gardos Software LTD