isel®
From Components to Systems

isyCAM 2.5 (light)

Manual, Part 1
"Introduction"

isel Germany AG, D-36124 Eichenzell, Bürgermeister-Ebert-Str. 40 ☎ (06659)981-0 🖷 (06659)981-776

About this Manual:

Various symbols are to be found in this Manual to alert you to important information.

CAUTION: **NOTE:** **EXAMPLE:**

We do not assume any liability or warranty that the manual, the program package isyCAM 2.5 (light) or the supplementary tools provided are free of errors or suited for any special purpose. Any legal responsibility or liability for collateral damage shall be excluded.

All information provided in this manual is delivered without warranty. Subject to change without prior notice.

Manufacturer: ***isel*** Germany AG
Bürgermeister-Ebert-Strasse 40
D-36124 Eichenzell

Tel.: +49 (0) 6659 981-0
Fax: +49 (0) 6659 981-776
E-mail: automation@isel.com
http://www.isel.com

Version: 09/2004

Contents

GENERAL **5**

SCOPE OF SUPPLY AND SET-UP 5
HARDWARE AND SOFTWARE REQUIREMENTS 6
INTENDED PURPOSE OF THE PROGRAM PACKAGE ISYCAM 2.5 (LIGHT) 7

OPERATING ISYCAD/CAM 2.5 (LIGHT) **8**

OPERATING METHODS 8
OPERATING DEVICES 10
INPUTS 11
Numerical input / numerical display *11*
Blanks when entering commands *11*
Confusing similar characters *12*
Slash "/" instead of backslash "" *12*
Multi-key functions *12*
Editing command lines; command stack *12*
Key functions *12*
Acknowledge specification value "< ... >" with < Enter > *13*
KEYBOARD ASSIGNMENT WITH CUSTOMISED COMMANDS 13
PRINCIPLE OF POINT SELECTION - THE FIRST LINES 14
SETTING UP THE USER INTERFACE 19
The user interface *19*
The header *20*
The main view *20*
The command box *20*
TOOL MENUS 21
The individual tool menus *21*
ARRANGEMENT OF TOOL MENUS 24
Changing the size *24*
Placing tool menus *24*
SAVING THE CUSTOMISED USER INTERFACE 26
TIPS AND HINTS 26

VISUAL TOOLS **27**

ZOOMING 27
VIEWS 29
MAGNIFIER FUNCTION / DETAIL ZOOM 32
LIMITING THE VIEW TO SELECTED COLOURS (VARIABLE COLPLOT) 33
SPECIAL HIGHLIGHTING OF THE ACTIVE OBJECT 34
GRABBING OBJECTS INTELLIGENTLY - SUPPORT DURING AND AFTER POINT SELECTION 34
SNAPPING COORDINATES 35
Using the Continuous Grid Pattern *36*
Deleting a Coordinate Grid *41*
Using the Discontinuous Grid Pattern *41*
Saving and Loading of Grid Patterns *42*
Final Remarks Regarding the Coordinate Snapping *43*

FUNCTION KEYS AND MULTI-KEY FUNCTIONS **44**

PRELIMINARY REMARKS 44
GENERAL FUNCTION KEYS USED DURING SELECTION PROCESSES 44
SPECIAL KEY ASSIGNMENTS FOR INDIVIDUAL COMMANDS / FUNCTIONS 45
MULTI-KEY FUNCTIONS (SHORTKEYS / HOTKEYS) 45

DRAWING PARAMETERS **47**

PRELIMINARY REMARKS 47
SETTING, SAVING AND LOADING GENERAL PARAMETERS 47

SETTING, SAVING AND LOADING FILE-SPECIFIC PARAMETERS 49
ASSIGNING AND MANAGING GENERAL AND FILE-SPECIFIC PARAMETERS 50
EXAMPLE: SETTING, SAVING AND LOADING DEFINED LINE WIDTHS (LINE THICKNESSES) 53
AUTOMATED LOADING OF PREDEFINED SETTINGS 55
SUMMARY 55

NUMERICAL POINT SELECTION 56

PRELIMINARY REMARKS 56
FUNDAMENTALS - COORDINATE SYSTEMS 56
The orthogonal (Cartesian) coordinate system 56
The polar coordinate system 58
FUNDAMENTALS - ABSOLUTE AND RELATIVE COORDINATES 59
Absolute coordinates 59
Relative coordinates 60
THE DIALOG BOX "NUMERICAL COORDINATE SPECIFICATIONS" 60
NUMERICAL POINT SELECTION - A SIMPLE EXAMPLE 61
SUMMARY 65

SETTING THE LINE WIDTHS 66

LINE WIDTHS 66
Defining line widths immediately before printing 68
Defining line widths using the variable: COLOR ("Settings" dialog) 69

IMPORTANT VARIABLES 71

OVERVIEW OF THE MOST IMPORTANT SYSTEM VARIABLES 73

CONFIGURATION SETTINGS IN THE WINDOWS REGISTRY 77

OBJECT AND DATA STRUCTURE 79

WHAT ARE DRAWING OBJECTS? 79
WHAT ARE ELEMENTS? 80
WHAT ARE MACRO OBJECTS (OR OBJECT GROUPS)? 81
SUMMARY OF THE MOST IMPORTANT OBJECT STRUCTURE OPERATORS 83

CREATING CUSTOMISED TOOLBARS 87

WHAT ARE BUTTONS AND TOOLBARS ? 87

INDEX 90

General

Scope of Supply and Set-up

Scope of supply:

The scope of supply of the software product isyCAM 2.5 (light) includes the set-up CD with the isyCAM 2.5 software itself, as well as with "Remote" and the "RemoteTools". After starting set-up, all information required to install the software and a description how to proceed to obtain the license code are displayed.

Software components:

isyCAM 2.5	Windows-based CAM software for 2D and 2.5D engineering and manufacturing tasks
Remote	Operating and output software for executing NCP files
RemoteTools	Tool for coupling Remote to isyCAM 2.5, including extended help functions

The components "Remote" and "RemoteTools" can only be used if isyCAM 2.5 (light) is installed on your computer.

The isyCAM 2.5 software documentation consisting of the three manuals **Introduction, CAD** and **CAM** is included on the set-up CD in PDF format (3 documents.

This documentation including a comprehensive "Examples" chapter is also available as an online help to which access is granted to the user after installing the "RemoteTools".

Note:

The manuals can be ordered additionally as a package in paper form:

Art. no: 970Z13 HD001 isy CAM 2.5 (light) Manual in 3 parts, A4 b/w

Installing isyCAM 2.5:

We recommend you **to quit all applications currently running** before you start set-up.

Insert the set-up CD into the CD drive; the **SET-UP** program is started from the **Autostart** menu.

If the set-up does not start automatically, proceed as follows:
- Start the Windows Explorer.
- Select the drive with the set-up CD.
- Double-click on "set-up.exe".

Applying for the license code:

To be able to work with the program, you will need the license code. This code is coupled to the PC hardware. You are allowed to install the program on **only once on any computer**.
After you have started the set-up program from the "Autostart" menu by clicking on the "isyCAM 2.5" button, a hardware-dependent PC code is read out. Use this code to apply for your license code, either via

- fax: ++49 (0) 36964 / 84510,
- via e-mail: isy25register@isel.com or
- using the form on the website at www.isel.com.

For more detailed information with the appropriate links, click on the **Register** button in the "Autostart" menu.

After you have received the license code, you can install **isyCAM 2.5**.
Optionally, after installing the program, you can install the interpreter software **Remote** and - for the intermediate coupling to isyCAM 2.5 with the option of using additional online help features and examples - **RemoteTools**.
If you have installed the program successfully, you will find the icon of the program  on your desktop.

Tip: It is also possible to start the set-up from the Windows **Control Panel**:
To do so, observe the following sequence:

Start - Settings - Control Panel - **Software** - Install

Windows will browse for the set-up program automatically. After the set-up program has been found, click on the "Finish" button to start set-up.

Hardware and Software Requirements

System requirements: To be able to work with isyCAM 2.5, it is imperative to have:

- a **PC with Pentium II or higher**, at least 333 MHz

- min. 64 **MB free RAM**

- Windows 98, Windows NT, Windows 2000 or Windows XP

Calling the program: To start **isyCAM 2.5**, double-click on the **program icon**  .

Calling the program using the **Windows start menu**: Click on the **Start** button. Under **Programs** in the **Start** pop-up menu, you will find the program group "SchottSysteme".

Clicking with the mouse on **IsyCAM 2.5** loads the program, and
the **start screen** of isyCAM 2.5 will appear on your desktop.

see also:
Section "The user interface on page 19

Intended Purpose of the Program Package isyCAM 2.5 (light)

isyCAM 2.5 (light) is a Windows-based CAM software package tailored to control systems from "isel".

Starting out from the isyCAM 3.0 software, a "light" version is offered with isyCAM2.5, which provides, on the one hand, all functions required for the engineering and design in the field of 2D programming and, on the other hand, a powerful tool for creating machining data for all typical 2D and 2.5D manufacturing tasks for machines with max.
4 axes.

The "light" version is operated analogously to isyCAM 3.0, using tool boxes with hierarchically arranged buttons, input menus and dialog boxes.

The NCP files created in the isyCAM part directly from the engineering data can be output to the connected machine or control system using the integrated operating and output software "Remote".
The extended help functions, including comprehensive examples, which can be called by clicking on the "Help" button in the menu "General tools", offer an additional service for the isyCAM 2.5 application.

Note:

To activate the additional functions provided by isyCAM and described above, two new buttons have been implemented which can be chosen from "**All tools**".

starts the functions of the interpreter program "Remote"

starts the supplementary online help (BASICS, CAD, CAM, SAMPL)

Operating isyCAD/CAM 2.5 (light)

Operating Methods

Creating drawings using CAD systems:

Despite the rapid progress in hardware and software development in the last few years, for many designers, engineers and developers, the drawing-board (the drawing table) has remained the most important aid when creating complex drafts and drawings. The mode of working is practically exclusively determined by manual work, such as "the drawing of a line", "erasing" and so on.

These rigid, time-intensive and error-prone procedures have already been replaced for a long time by computer-aided designing and drawing, called **CAD (Computer Aided Design)**. As far as the method to create a drawing is concerned, there is principally nothing new. Merely the tools, such as pencil, dividers and paper, are replaced by cross-hair, screen and printer/plotter. The obvious advantages of a CAD system compared to conventional drawing techniques, are the significant gain in the speed, the extraordinary improvement in the precision, the automatic design and engineering aids, the straightforward possibilities of implementing changes and modifications, and the broad range of data exchange possibilities. Thus, the drawer can put more creativity into his own task, since tiresome routine work is substantially reduced.

The changeover to the new technique, however, is not always easy, as the activities which for many years have been done more or less automatically must be conveyed to the computer system observing a strict formalism. This could be best explained as if the drawer were to give an external person the order to perform his work, for example, in the following form:

- Take your pencil.
- Place the ruler vertically to a point.
- Set down the pencil.
- Draw a line with a certain length from the starting point, and so on.

Similarly, a specified logical sequence of activities must also be specified in the CAD program. It is quite clear that this is not done using normal "everyday" expressions, but by adhering to very formal and strict language rules. We call it command language.

The CAD drawer must naturally also be provided with appropriate possibilities of intervening manually beyond the command level. For example, he must set freehand points in the drawing area or identify drawing elements. In such a case, when talking about isyCAM 2.5 (light), we speak about a point selection.

The handling of the CAD system includes the following modes of working:

- Command input (calling and execution of commands)
- Parameter input (input of numbers and text as necessary)
- Point selection (selection of geometric point positions)

Command input:

Generally, **commands** can be **input** using two methods:

- via the keyboard
- command selection using menus

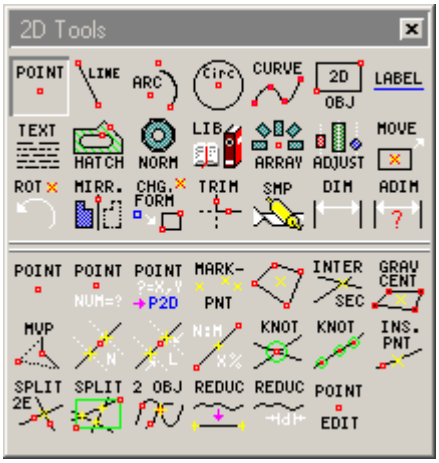
To input commands directly via the keyboard, it is imperative that the user possesses an exact knowledge of the formal CAD input language (command language) which - especially during the starting phase - causes difficulty. The menus, however, provide the user with a world of commands in a more simple, dialog-oriented form with graphics support. In practice, a mix of both methods of operation will usually be found. In particular, with growing knowledge on the part of the user, the direct command input will become more and more important because of its advantages as regards the speed.

Note:

To enter commands, the command box must be active, which can be recognised from the flashing cursor. The command box can be activated either by clicking on <Space> or by clicking with the mouse in the command box.

Tool menus have been implemented for the user to choose commands symbolised in the form of buttons on the monitor using a pointing device (such as a mouse). Due to the large variety of commands, however, not all functions can be displayed on the screen simultaneously. For this reason, some menus provide a tree-style structure. From higher-level menus, you can go down to more finely graduated functions by pulling up lower-level menus.

An example is the menu for the 2D drawing functions, which is shown below.



Clicked buttons are indicated clearly by way of a "pressed" or "raised" display. If you point with the pointing device on a button, a help text will appear after a few seconds.

Hierarchy levels:

Since each user may layout the menus and place them on the screen as he wants, the documentation only uses an appropriate short form to represent the individual hierarchy levels. This could look as follows:

Despite the confusing number of menus and buttons, the concept is based on a simple logic. We differentiate the following categories:

- **Global functions**, such as for zooming, deleting, selection, viewing, colour snapping, special points etc., are summarised in a separate tool menu.
- **Drawing functions** (for creating new objects) and
- **Change functions** (geometrical modifications to existing objects) are structured depending on the type of the object to be created/processed and included hierarchically in submenus.
- **Measuring functions** (for determining numbers and units from geometry objects) are also structured hierarchically and summarised with the
- **Functions** for conversion, geometry information and changing the structure, as well as with the
- **Drawing aids** etc. in the menu "General 2D Tools". Please note that we have assigned the dimensioning types, which could also be categorised as measuring processes, to the drawing functions.

The **parameter input** is performed either via keyboard in the command box or using either the pointing device or the keyboard in the dialog boxes.
Contrary to the command box which is available permanently, dialog boxes are opened only after triggering the appropriate function and disappear thereafter.

Note:

Since an exact separation between the individual tools is not possible in each individual case, some buttons are to be found in several menus.

Operating devices

Operation using the keyboard:

To be able to execute the actions mentioned in the previous section, the keyboard is an obvious handling device. The layout of the keyboards depends on the type of computer you are using and on the national language set.

With some keyboards, the control keys are labeled both in English and in German. For example, the key labeled with the German text <Strg> corresponds to the English <Ctrl>. All key designation equivalents are to be found in the table below:

Equivalent key designations			
English		**German**	
Esc	ESCAPE	Eing Lösch	EINGABE LÖSCHEN
Break	BREAK	Abbr	ABBRUCH
Home	HOME	Pos1	POSITION 1
End	END	End	ENDE
Ins	INSERT	Einfg	EINFÜGEN
Del	DELETE	Lösch	LÖSCHEN
PgUp	PAGE UP	Bild	BILD AUF
PgDn	PAGE DOWN	Bild	BILD AB
Ctrl	CONTROL	Strg	STEUERUNG
Print Screen	PRINT SCREEN	Druck	DRUCK

Enter	ENTER (CR-LF)	↵	EINGABE
Shift	SHIFT	Groß	GROSS
BkSp	BACKSPACE	←	RÜCKSCHRITT
Space	SPACE-BAR	Lange Taste	LEERSCHRITT
Tab	TABULATOR	↹	TABULATOR

Operation using the mouse:

As already described above, you can input commands and parameters using the keyboard. The graphical handling, such as the selection of geometrical point positions, is performed using the mouse. The mouse is an electronic input device which serves for the quick control of a pointer (e.g. a cross-hair) on the screen. It is therefore also called a **pointing device**.

In practice, the pointer is set on the desired position by moving the mouse, and the action is triggered by pressing a key. This process is commonly called "clicking". In this context, it is important to mention that the desired action is only triggered when the mouse button is released.

Inputs

Numerical input / numerical display

Number formats:

Please note that in **isyCAM 2.5 (light)** decimal numbers are input not with a comma (" , "), but with a decimal point (" . "). A comma is always used for separation between several numerical values, e.g. with a pair of coordinates x, y.

If a number starts with a leading zero, it can also be omitted in individual cases. In other words: Instead of "0.1234", you may also write ".1234".
For the representation of especially large or especially small numbers, the exponential notation is often also used. For example, the following numbers are equivalent:

```
1230000=1.23e+006
0.00001=1.00e-005
```

Please note also that negative numbers must always be included in round brackets for differentiating options in the case of input via the keyboard.

The notations for the most simple arithmetic operations are:

+	Addition
-	Subtraction
*	Multiplication
/	Division

Blanks when entering commands

Blanks:

Spaces when entering commands via the keyboard serve as a delimiter between the individual parameters. For example:

```
arc * -t -r50
```

If a parameter itself incorporates blanks, for example, in the case of text inputs consisting of several words, the appropriate term must be included in inverted commas.

label * "This is a text with blanks"

Confusing similar characters

Similar characters: Users with less experience often confuse the following characters:

```
0 (zero)            and    O (uppercase "o")
l (lowercase "l")   and    1 (one)
$ (dollar)          and    § (paragraph character)
```

In case of error, verify your command input once more with reference to the notation.

Slash "/" instead of backslash "\"

When using commands for the handling of files (dir, copy, mv, etc.), the backslash " \ " is often required. In **isyCAM 2.5 (light)** it is also allowed to use the normal slash " / " (located above the number "7") in the same places.

Multi-key functions

When working with multi-key functions, such as <Ctrl> <C>, you should always first press the control key, i.e. <Ctrl>, and then **additionally** the corresponding key, e.g. <C>. Subsequently in this documentation, the multi-key functions will always be marked by a plus sign between the key designations, e.g. <Ctrl>+<C>.

Editing command lines; command stack

The keys <ArrowLeft>, <ArrowRight>, <Del>, <Backspace>, <Home>, <End> serve for editing and cursor control within the command line. The keys <ArrowUp> and <ArrowDown> permit browsing in the command stack (command repetition memory) in the direction of commands entered previously or later.

Commands activated from the tool menus are also accepted into the command stack.

Key functions

Key	Action
<ArrowUp>	Fetch older command from stack
<ArrowDown>	Fetch newer command from stack
<ArrowLeft>	Moves the cursor to the next position on the left
<ArrowRight>	Moves the cursor to the next position on the right
<Home>	Cursor to the beginning of the line
<End>	Cursor to the end of the line

<Backspace>	Delete character left of the cursor
<Del>	Delete the character on which the cursor is positioned
<Shift>+<ArrowLeft>	The character left of the cursor is highlighted, and the cursor jumps to the left by one character (continuously when holding down the arrow key)
<Shift>+<ArrowRight>	The character on which the cursor is positioned is highlighted, and the cursor jumps to the right by one position (continuously when holding down the arrow key)
<Ctrl>+<C>	copies the highlighted characters to the clipboard
<Ctrl>+<M>	red mark for highlighting the cursor position regarding previous system messages and stack commands for accepting into the clipboard
<Ctrl>+<V>	pastes the contents of the clipboard at the current cursor position
<Ctrl>+<X>	moves the highlighted characters to the clipboard (cutting)

Note:

It is naturally also possible to position the cursor using the pointing device. If you move the mouse over the command line with the left mouse button pressed, for example, characters over which the mouse button passes are highlighted. If the mouse pointer is located in the command box, after pressing the right mouse button, a menu appears for copying, cutting and pasting.

Acknowledge specification value "< ... >" with < Enter >

With operation via the keyboard, some of the procedures of **isyCAM 2.5 (light)** offer default values enclosed in acute brackets "<...>". These values can be accepted by pressing <Enter> without further input.

```
Enter angle of rotation <90>:
```

Keyboard assignment with customised commands

Customised keyboard assignment:

User-specific key combinations can be stored in the file *pictures.skd*. Subsequently, **isyCAM 2.5 (light)** must be restarted.

Finally, the most important keys/actions for program operation are listed below:

a) in the command input mode (if keyboard input is possible in the command line):

<Enter>	to complete commands entered manually via the keyboard
<F1>	Help function for information regarding the operation
<Ctrl+Break>	interrupts a drawing process or an action (not always supported)

b) in the point selection mode (if a graphical cursor appears on the screen)

Left mouse button	for clicking on (selecting) points/objects (when working with a digitiser, press the pen onto the pad).
Right mouse button	or <F6> to quit the selection mode - POLYEND**.** When working with the mouse, the cross-hair must be located within the screen window.

Principle of Point Selection - The First Lines

Point selection using the mouse:

After introducing the operating devices, let's now study the basic principle of drawing to create simple polygons (poly-lines). In this phase, we will use the command input to learn also the relevant syntax. We will start our activities from the command line and begin by creating a polygon with the command

```
poly *
```

In this command, " `poly` " constitutes the command keyword, and " `*` " a parameter (here: *object*). A blank (space) is used as the delimiter between the two words, as with any parameter separation under **isyCAM 2.5 (light)**. Converted to everyday language, the command reads approximately as follows:

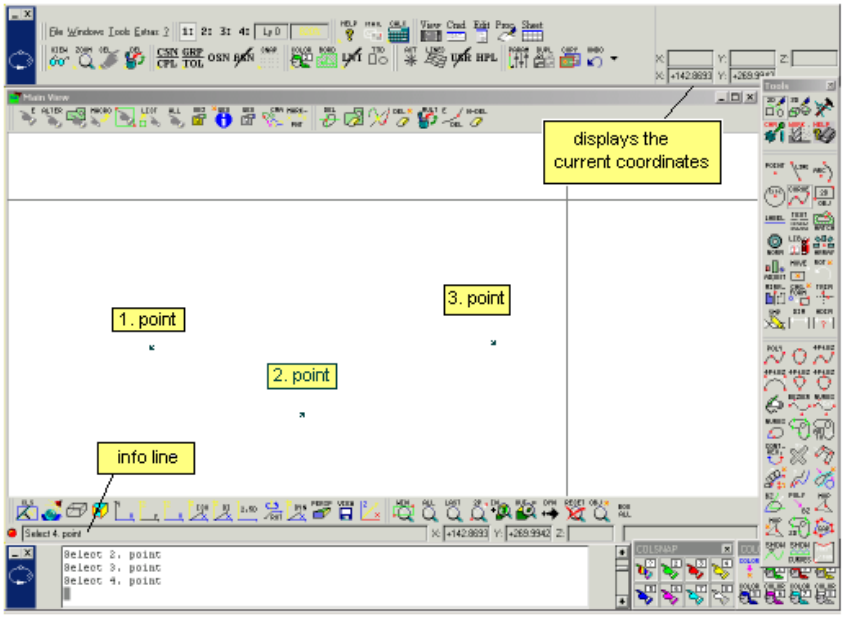
```
Create a polygon and assign an object name
automatically for identification
```

By specifying "`*`", the computer will use the prefix "`obj`" (for "object") and append there a number, incrementing any further objects automatically. If we complete the polygon creation in our example successfully, this first object is assigned the name "`obj1`" in the internal database, the next "`obj2`", etc.

Let's begin: upon completion of the command, the cross-hair appears, together with the prompt "Choose the 1st option!". The user is prompted to define the first point on the drawing area by manually positioning the cross-hair and triggering interactively manually. To fix the point, press the first button when working with the mouse, or when working with the digitiser either the pen or key 1 on the cross-hair magnifier.

Set a total of four points at any screen positions in the manner described above. Please also observe the display of the running coordinates in the header also while selecting the points.

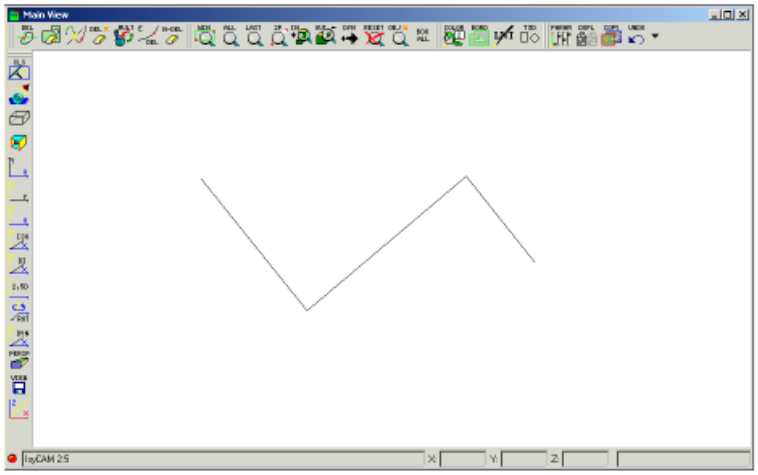
Screen while selecting the points:



We will complete the poly-line after the fourth point. To do so, we will have to cancel the process of point selection subsequently with POLYEND. Although already described above, the two different methods will be explained here once more in brief:

- using the keyboard: by pressing the function key <F6>
- using the pointing device: Position the cross-hair on the drawing area and click on the **second mouse button.**

After completion with POLYEND, the polygon is drawn on the screen. This means that the "poly" command was executed successfully.



Scaling the graphics:

By the way: POLYEND can also be used to cancel an inadvertently initiated graphical action prior to its execution.

To get to know also the other two **types of the graphical cursor** used when selecting points, let's do another exercise.

Please enter the command

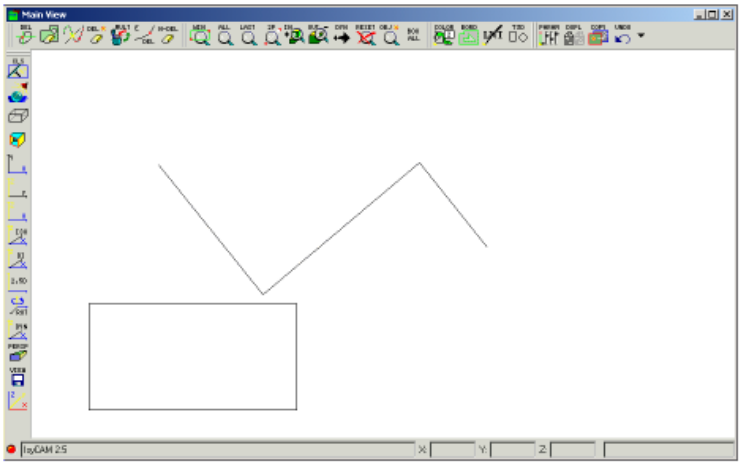
```
scale * -r
```

The cross-hair will again appear on the screen, but now with the message

```
Select the first corner point of the new object
box.
```

Let's simply position a point into the bottom left corner of your drawing area to test the effect. After activation, first the cross-hair disappears, and then the prompt "Select the second corner point of the new object box" appears in the info line.

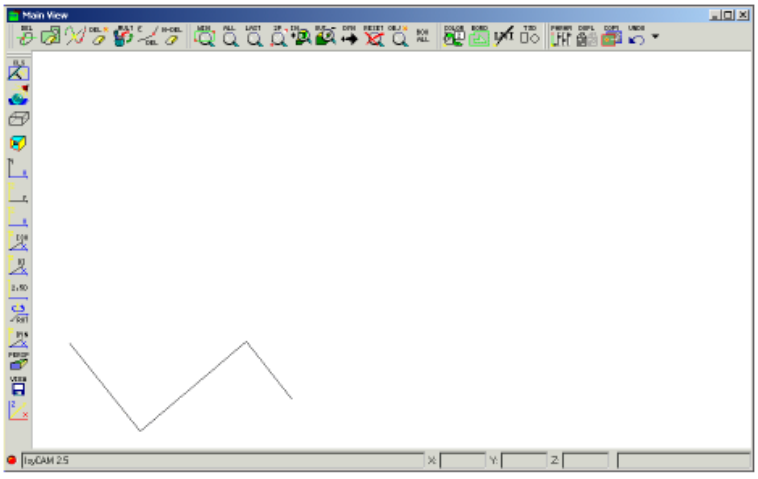
Variable window cursor:



If you now move the pointing device to the right and upwards, for example, a rectangle results which is variable in its size and whose bottom left corner has been defined by the point selected first. In this case, we speak of a **variable window cursor**, as the size of the resulting rectangle is determined by the selection of the second point. Set the second point as usual.

After releasing the second point, the polygon created above is fitted into the rectangular window such that only its size, but not its height / side ratio is changed. In other words: We have reduced (or enlarged) our object.

Scaled polygon:



In such a case, we speak of "scaling", therefore also the command keyword **scale**. The most important thing in our action, however, was not the scaling, but the handling of the variable window cursor. Therefore, repeat the execution of the last command, but now with enlargement of the object.

Rigid window cursor:

Now let's show you the third type, the **rigid window cursor**. It is activated on the screen by entering the command

```
scale * -m
```

The graphical cursor displayed now has the form of a rectangle (BOX), which is as large as the maximum dimensions of our polygon. This frame can be moved around the screen as usual using the pointing device.

After clicking on it, the polygon is shifted as expected to the newly selected window position.

Object and element:

The last little exercise will clarify the difference between **object** and **element**.

The object created in our example (polygon) consists of three line segments (three straight lines between two points each). We say: "The object obj1 consists of three elements". In **isyCAM 2.5 (light)**, we call the smallest geometrical element, i.e. the smallest geometry unit, an element.

The differentiation between object and element is of great importance, in particular, with deletion and trimming processes.

Here is an example: We will delete a line segment of the polygon, i.e. an element, using the command

```
trimx -d
```

After input, the cross-hair appears. Direct it, for example, to the centre of the line element you want to delete, and click on the activation button. The selected partial polygon is deleted and disappears from the screen.

If you want delete the object, however, i.e. the entire polygon, the required command is:

```
del *
```

Please note that the last command can be undone in case of error by entering

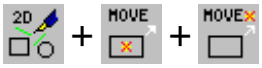
```
undo
```

Now let's mention the appropriate steps of the mouse operation for drawing, scaling and deleting:

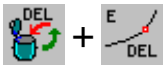
Drawing the polygon 

(Please observe that a function with "rubber band" is called here, i.e. the resulting lines are displayed already when drawing)

Proportional change in the size using the window 

Moving the object using the window 

Delete element: 

alternatively 

Delete the active object 

UNDO (the last action is undone 

Setting Up the User Interface

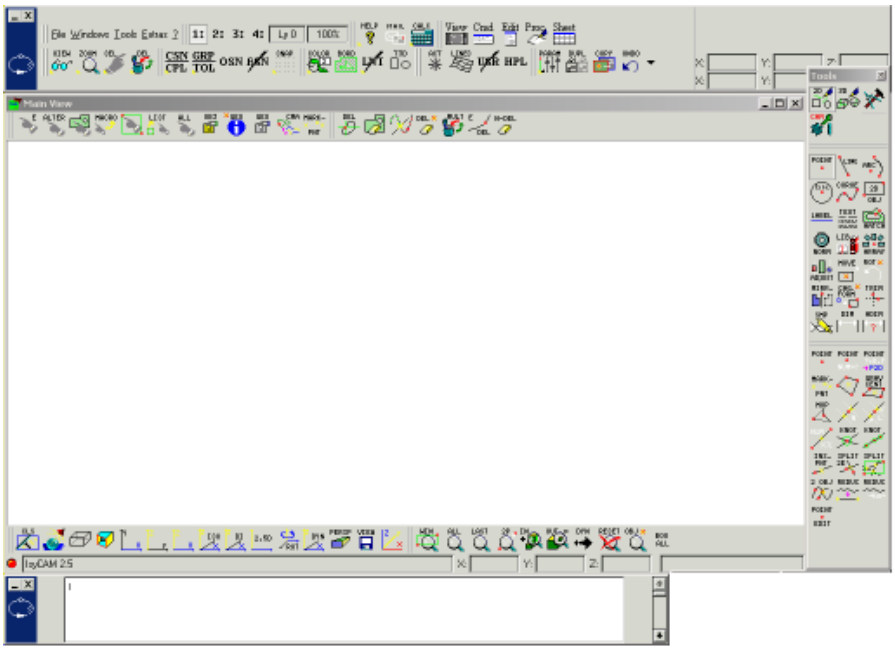
The user interface

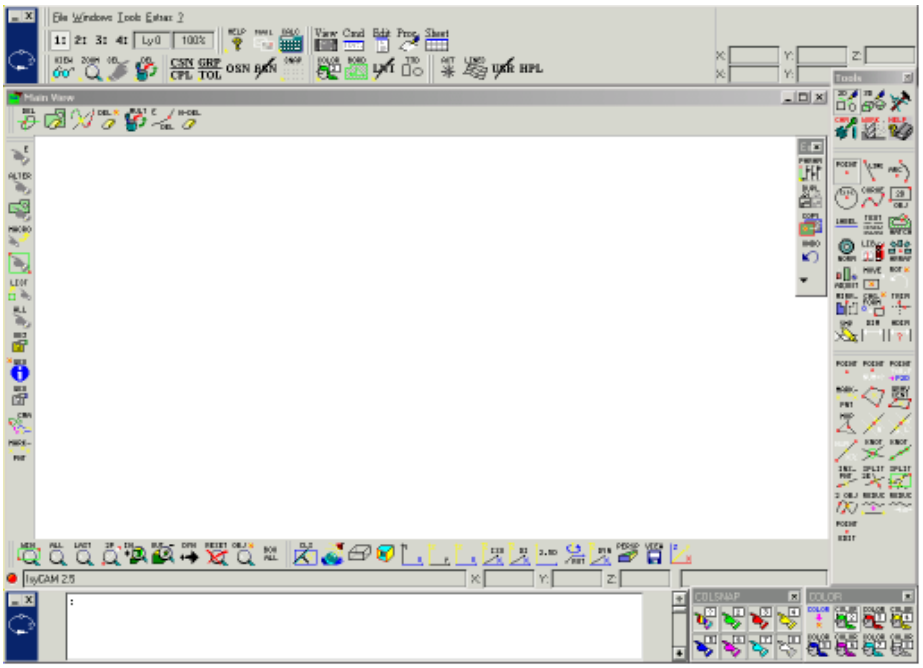
First start:

When the program is started for the first time, a standard user interface appears (depending on the screen resolution).

With each subsequent start, the user interface appears in the state in which **isyCAM 2.5 (light)** was quit. Two examples are shown below.

Each user may generate and use his own, customised user interface for effective working.





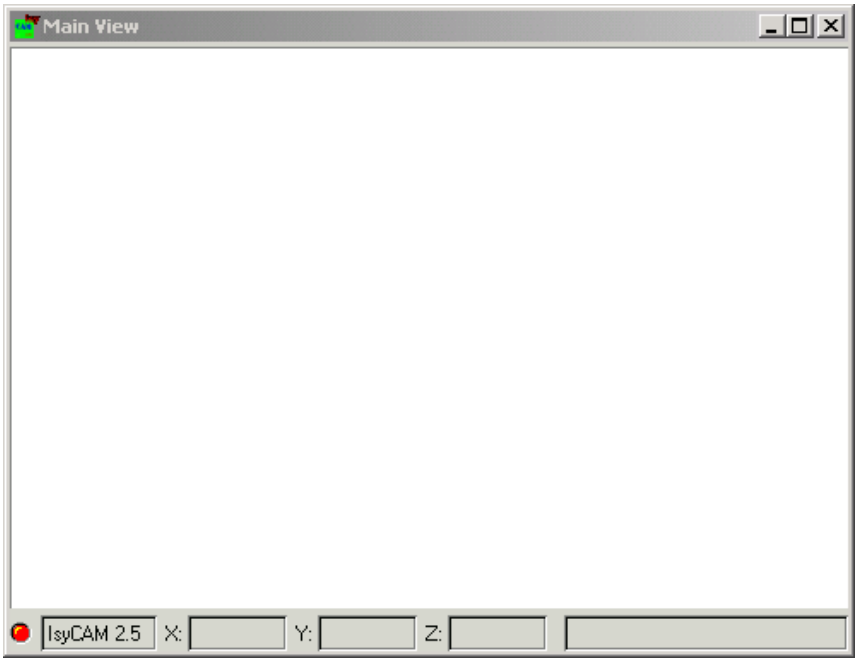
The following sections describe the basic components of the user interface and how to handle them. The description of the mouse buttons (left/right) always refer to a mouse configured for right-handed persons.

The header

The header incorporates the action buttons for minimising or quitting the program in the top left corner. The running coordinates of the current drawing plane are displayed with selection processes in the top right corner (constant coordinate = axis-parallel drawing plane). Either the 2D coordinates of the current drawing plane (with the 2.5D grid pattern activated -> T3D) or the running 2D absolute coordinates according to the screen level are displayed in the top right corner. For estimating distances, the 2D coordinates can be reset by pressing <F3>. The standard menu and various tool menus can be placed in the header.

The main view

Window of the main view:



Views are surfaces for visualising and manipulating geometry objects. If further views are opened next to the main view, the view currently active is highlighted by a red point in the bottom left corner. Standard and tool menus can be docked to particular views.

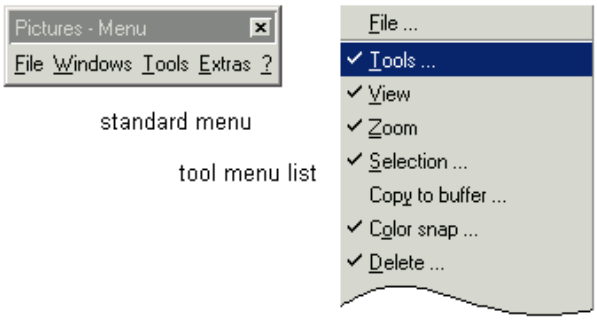
The command box

The command box incorporates the buttons for minimising (the box) and quitting the program in the left top corner. It serves for communication with the PC (operator and error messages by the program, inputs via the keyboard by the user). The activation by the user is performed via mouse click or by pressing the spacebar.
When clicking on a button, the appropriate command and input prompts (if any) are

displayed in a box. Pressing the "ArrowUp" key recalls the commands executed last and displays them in the last line. Copying / pasting of texts from / to the clipboard is possible.

Tool menus

To operate the program, various tool menus sorted by topics are available. They can be arranged freely on the screen. They are activated by clicking on the appropriate entry in the menu selection <Tools> from the standard text menu.



The individual tool menus

File ...

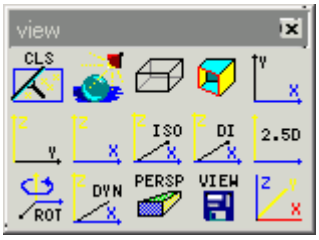
File ...
Functions for handling files (New / Open / Save) and for plotting

Tools ...

Clicking on a button opens one of the following four menus (View / Zoom / Selection / Delete)

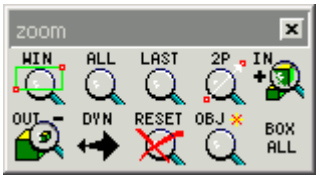
View ...

Various 2D/3D views, Shading, Hidden Edges, Rotation, Perspective...



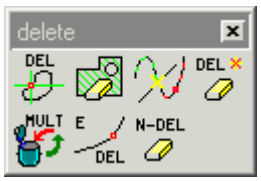
Zoom ...

Zoom window, dynamic, PAN, closer, further, last ...

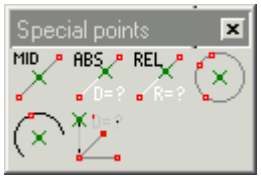


Selection ... Element selection via cross-hair, special selections (hatching, macro, name ...)

Delete ... Deletion of elements and objects, multiple deletion, ...

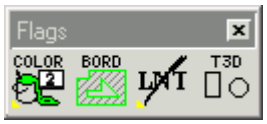


Special points ... Special point selection with 3D operations; is only activated with execution of 3D operations



Parameters ... Colour display / snapping, object snapping and grab tolerance, snap to matrix

Flags ... Selection and display of the current drawing colour, margin / hatching, line types, 2.5D drawing plane

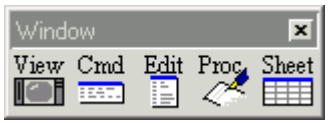


More ... Display of the active object, of grid lines, of user coordinates, of hatching

Helps ... Online help, calculator, support mail

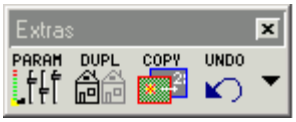
Window ...

Open new window / command box, switching to editor and sheet



Tools ...

Parameter settings (font, hatching, dimensioning, colours, line types...), duplication, copying to alternative buffer, UNDO function

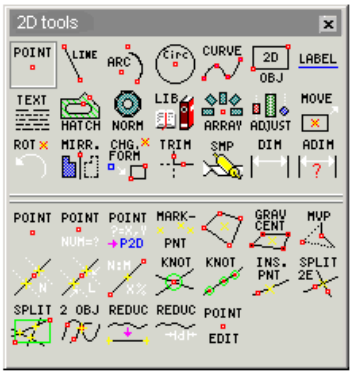


Status display ...

Display and switching of the current buffer and of layers; display of the zooming factor

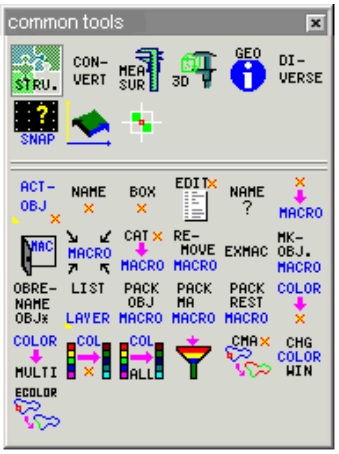
2D tools ...

Dialog box for 2D drawing and manipulation tools



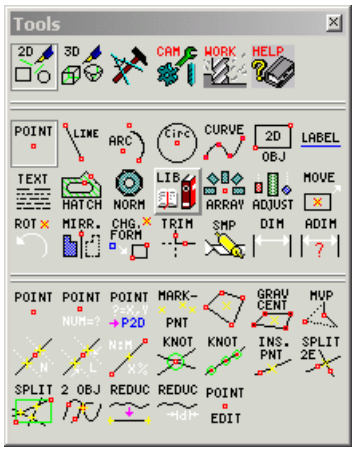
Common tools

Special functions for structuring drawings, converting, measuring, geometry information, plane setting...



All tools ...

Space-saving summary of 2D, 3D and common tools, switching in the top line

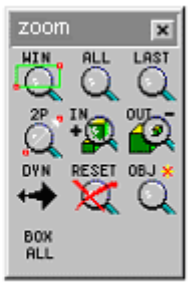


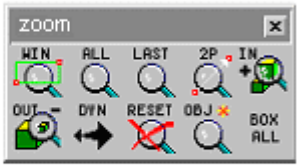
Arrangement of Tool Menus

Changing the size

Tool menus:

Tool menus can be changed in their size (number of columns/lines). To do so, position the mouse pointer on the margin or on one of the four corners until an arrow with two tips appears. With the left mouse button held down, you can change the size of the menu.



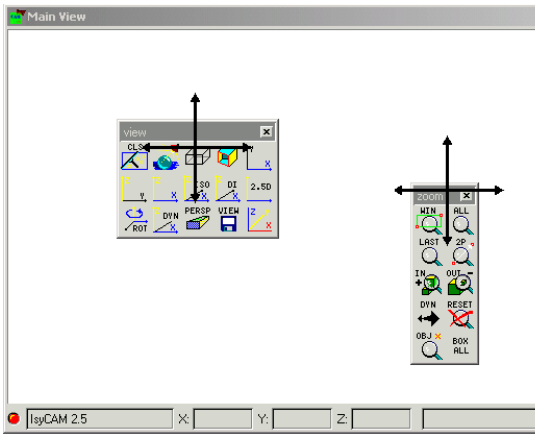


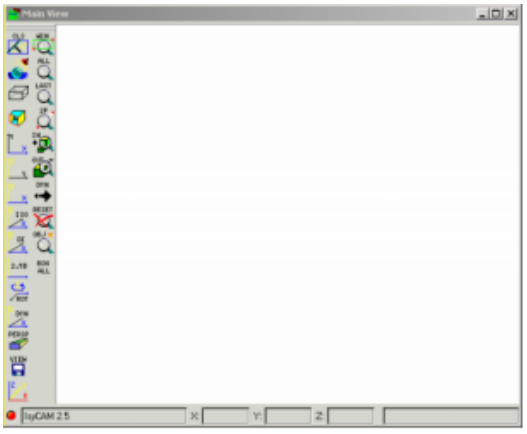
The sizes of header, view windows and command input can be changed similarly. If tool menus are placed in the view windows, they will determine their minimum size.

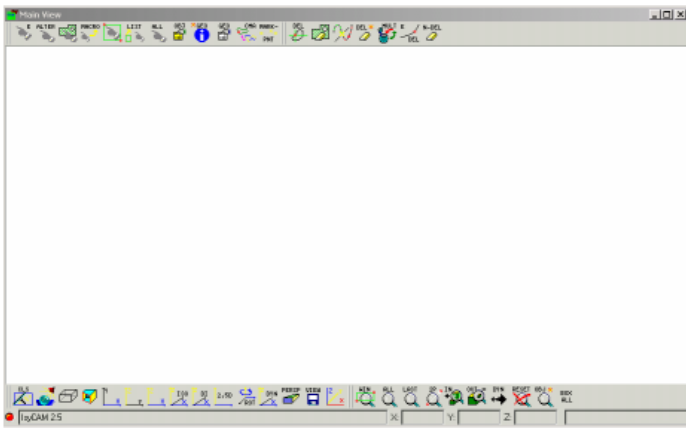
Placing tool menus

Arrangement of tool menus:

In addition to arranging tool menus on the desktop, it is also possible to dock tool menus to the header or to a view window. Please note that the menus docked to a view window, such as ZOOM, are limited in their function exclusively to this window. If the menu is not docked to a specific window, the function refers to the window currently active (marked by the red point).
To dock a menu, place the mouse pointer into the header of the appropriate menu. Thereafter, move the menu, for example, to the desired place at the margin of the view window with the mouse button held down until it is docked automatically. The docking to the header is performed analogously.

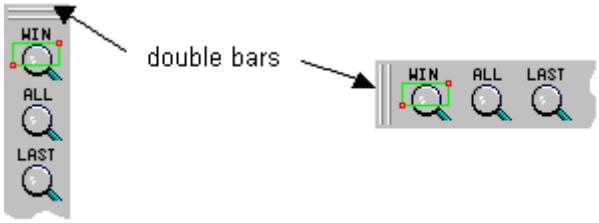






Undocking tool menus:

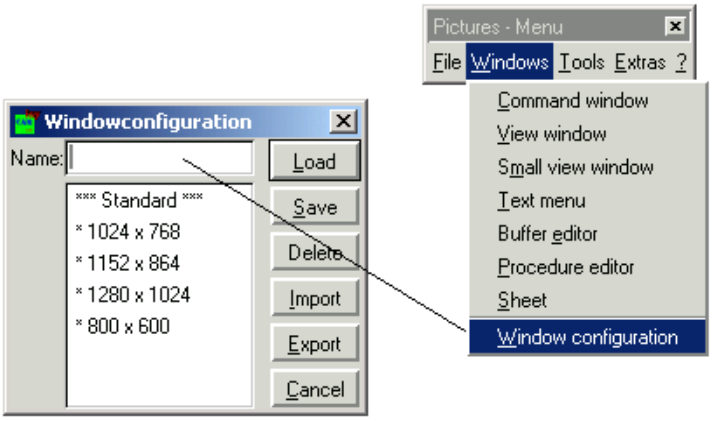
Docked menus can correspondingly also be "undocked" (disconnected). To this end, position the mouse pointer on the double-bars above or left of the appropriate menu. The tool menu can then be moved then with the left mouse button held down.



Saving the Customised User Interface

Customised user interfaces:

After setting the customised user interface, it can be saved with a freely defined name as shown below. Similarly, it is also possible to load configurations already saved.



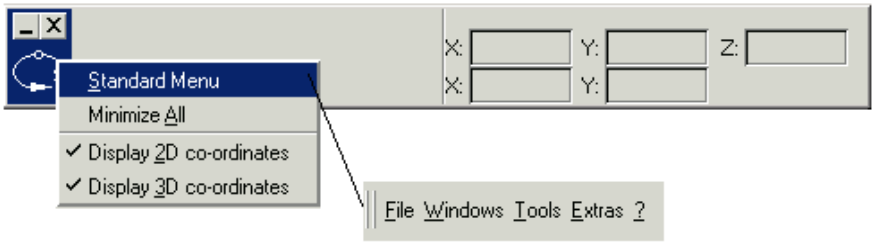
Tips and Hints

Rescuing the menu:

Generation of the standard menu (rescuing from the "no-menu" status): If when adapting the user interface a status has arisen in which there are no longer any menus, there are two possible remedies:

1st method: with a mouse click

Click with the right mouse button on the logo in the header. In the menu which then opens, choose "Standard menu", and generate the standard menu in the header.



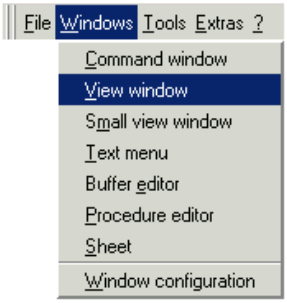
2nd method: using the keyboard

Entering "*stb stdMenu* " in the command line will also generate the standard menu.

Generating a window:

Generating the main view window and further windows

If no main window exists or to open a new view window, select the menu options shown in the screenshot below.



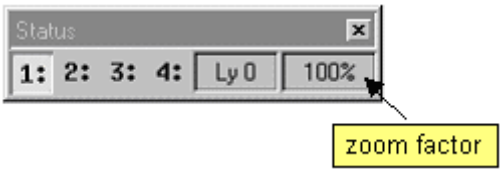
Visual Tools

isyCAM 2.5 (light) provides numerous visual tools for the user to adapt the display on the screen and the display aids flexibly according to his tasks. The most popular visual tools are zooming, setting of 3D views and shading of 3D bodies. Additionally, **isyCAM 2.5 (light)** also provides the possibility of opening several view windows simultaneously. The user aids are complemented by a magnifier, by displaying of the object currently active, by a dynamic preview when selecting points, by colour filters and colour snapping, as well as by a gridline display.

Zooming

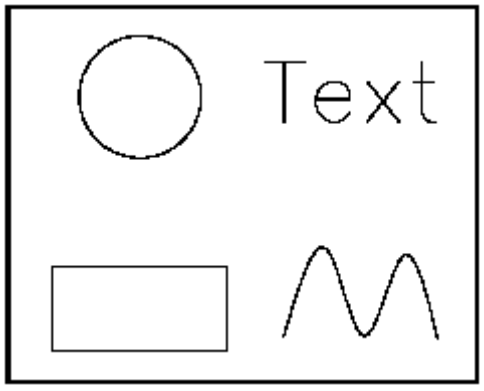
Zoom:

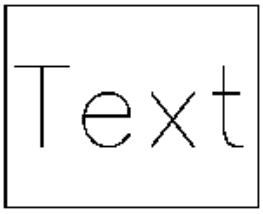
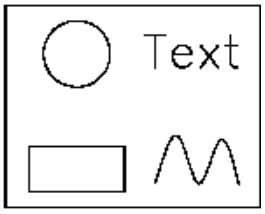
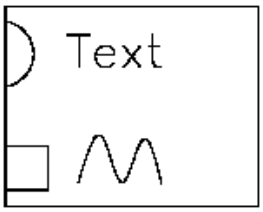
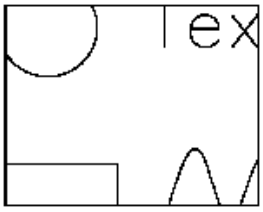
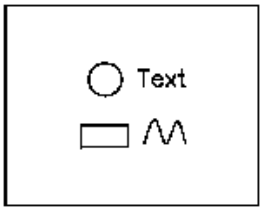
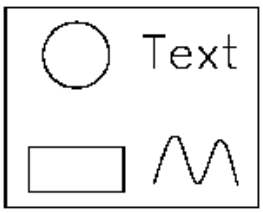
The detail modification used most frequently in **isyCAM 2.5 (light)** pertains to the variable ZOOM. Use the "Zooming" function to enlarge drawing details, for example. The reverse case, the visual "removal" from the drawing, and shifting of the zoomed detail can also be performed using the "Zoom" command. **isyCAM 2.5 (light)** is started such that the standard screen section (0.0 .. 639.511) is displayed centrally in the window at maximum size. This corresponds to a zooming factor of 100%. If the tool menu "Status display" is opened, the zooming factor can be read at any time. Zooming factors other than 100% are displayed in yellow.



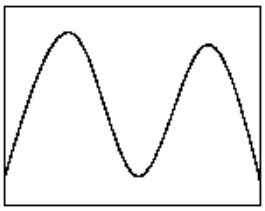
The zooming factor should be observed especially with objects created in freehand drawings (without numerical size specifications). Otherwise, it is possible that the objects will be substantially smaller or larger than intended.

The most important zooming functions are summarised in the tool menu "Zoom". They can also be used when executing selection processes.
All zooming functions are listed with their effects in the table below. All explanations refer to the illustration here on the right.



Function	Button	Alternative Keyboard Input	Action by the User, Remark	Effect
Zoom out detail		**zoom -ir -s102**	After the cross-hair has appeared, select the corner points for the new zoom window; observe the prompts in the command input field	
Zoom All (complete buffer contents)		**Zall** or **zoom -iro all -s102**		
Recall last (previous) zoom section to display		**zoom -l -s102**		Try it yourself!
Move PAN zoom section dynamically		**dynpan -i** during a selection process: key "**P**"	After prompting, position the mouse pointer accordingly and move the section with the left mouse key held down	
Scale with factor Zoom in		**Zoom -ir -f***value* **-s102**	when executing with button *value* =0.5	
Scale with factor Zoom out		**zoom -ir -f***value* **-s102**	when executing with button *value* =2	
Scaling dynamically Dynamic zoom		**dynzoom -i** during a selection process: key "**M**"	After prompting, position the mouse pointer accordingly and change the scale by moving the left mouse button vertically while holding it down.	Try it yourself!
Reset zoom to default value 100%		**zoom=**	The standard screen section (0.0..639.511) is displayed centrally at maximum size (acts exclusively on the main view window)	

Zoom to active object  **zoom -iro * -s102** The active object (for example, a previously selected object) is displayed at maximum size.

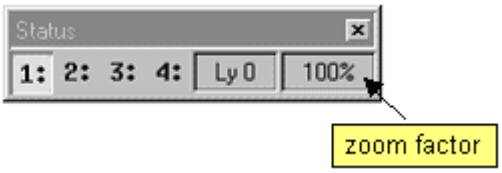


If the execution of a zooming command should not have the desired effect (for example "**zall**" after manipulations, loading of externally converted files, ...), new calculation of the object boxes could be a remedy under certain circumstances. To this end, click on the button 

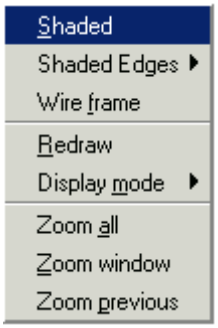
or type the command "**box all**".
Further possibilities for changing the zoom:

- Click with the right mouse button on the display of the zoom factor:

  The following menu appears:



- Click with the right mouse button into the drawing area;
  another menu will appear:



For additional information, please refer to the **isyCAM 2.5 (light)** help. To obtain help, type "**zoom**" in the command line and press <F1>.

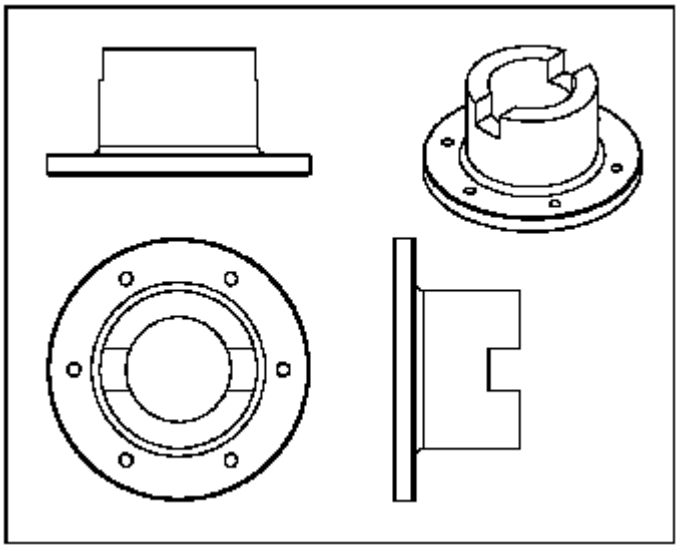
Notes regarding work with several view windows:

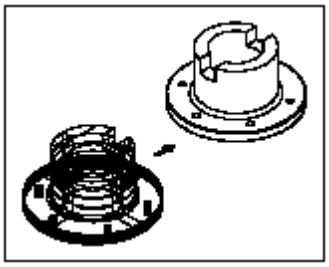
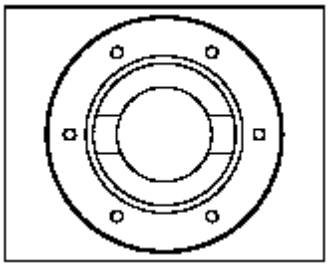
If a tool menu (e.g. "Zoom" or "View") is docked to a view window, the button will only act on this menu. If the tool menu is not docked to any window, it will result in changes in the current view window (marked by a red point in the bottom left corner). Command inputs in the form described above also refer to the current view window.

Views

Selecting a view:

When making designs in the space, the user is assisted by numerous view variations serving for better clarity. It could be imagined that the user moves in the space around the geometry objects, viewing them from the appropriate direction. The flange shown in the following will help make this clearer.

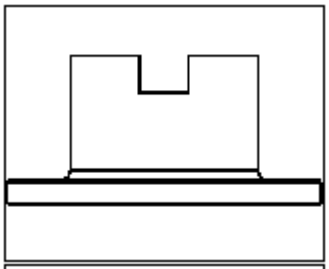


Function	Button	Alternative Keyboard Input	Action by the User, Remark	Effect
Redraw		**redraw -s102**	Execution useful after comprehensive deletion or trimming processes, for example	Try it yourself!
Shade		**shade**	2.5D / 3D surfaces and bodies are displayed shaded; if the function is performed once more, the display is reset to line representation; high-quality images, e.g. for documentation, are created using the renderer	
Hide hidden edges		**hide -t**	For clarification, hidden edges are hidden; if this function is executed once more, the display is reset	
Top view View of the positive X-Y plane from the positive Z direction		**setpp -z**	For further information, type **setpp** and press <F1>	

Side view

View of the Y-Z plane from the positive X direction

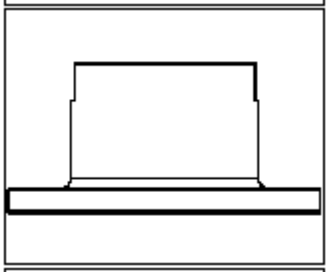
setpp -x



Side view

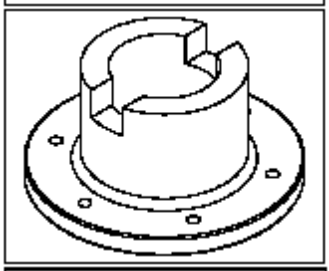
View of the X-Z plane from the positive Y direction

setpp -y



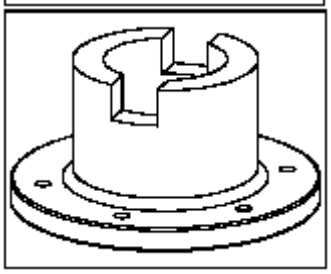
Isometrical view

setpp -i



Diametrical view

setpp -d



View of the current drawing plane

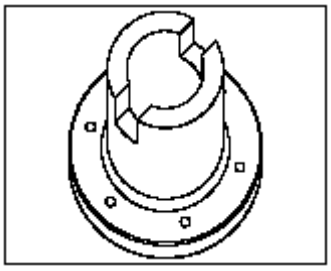
setpp -2

Vertical view of the current drawing plane

Try it yourself!

Military perspective

setpp -c



Rotate view by an axis of coordinate

rview
(for options, see "Help")

After clicking on this button, enter the angle of rotation (observe sign!) and the axis

Try it yourself!

Dynamic rotation of axis

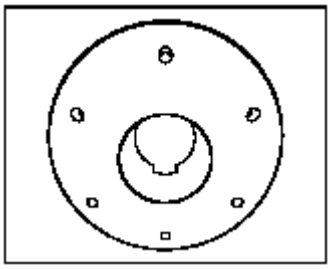
dynrot
When selecting,

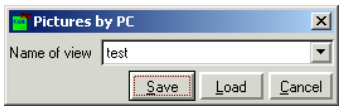
press "**R**".

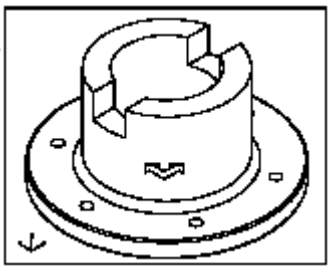
Position the mouse pointer on the appropriate position accordingly and rotate the view by moving the mouse with the left mouse button held down

Try it yourself!

Perspective view		**pview** (for options, see "Help")	After prompting, enter the viewer's location, the viewer's point of view and viewing angle (see selected example)
Save and load set views		**vsave**	Use the menu shown on the right to save and load views under a freely defined name
Hide/unhide axis marking		**toggleaxis**	The three-axis coordinate system serves for orientation in the space, and the two-axis coordinate system specifies the origin of the current drawing plane. If the coordinate system is located in the upper right corner of the screen, the origin cannot be seen in the window section.







Further view functions:

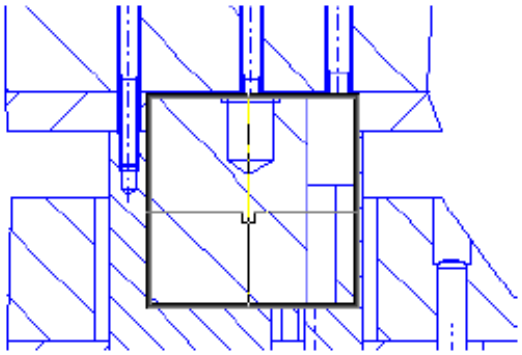
Clicking with the right mouse button into the drawing area opens the following menu:

Here you will also find view functions. You can yourself test the possibilities for representing shaded surfaces, edges and curves.

Magnifier Function / Detail Zoom

Enlargement function:

To make the point selection easier, a magnifier function has been implemented in **isyCAM 2.5 (light)**. It can be activated by holding the CTRL key down during selection processes and pressing either the left or the right mouse button. With the CTRL key held down, each further actuation of the left mouse button enlarges the representation, and each actuation of the right mouse button reduces the representation. The selection tolerance changes with the section displayed.

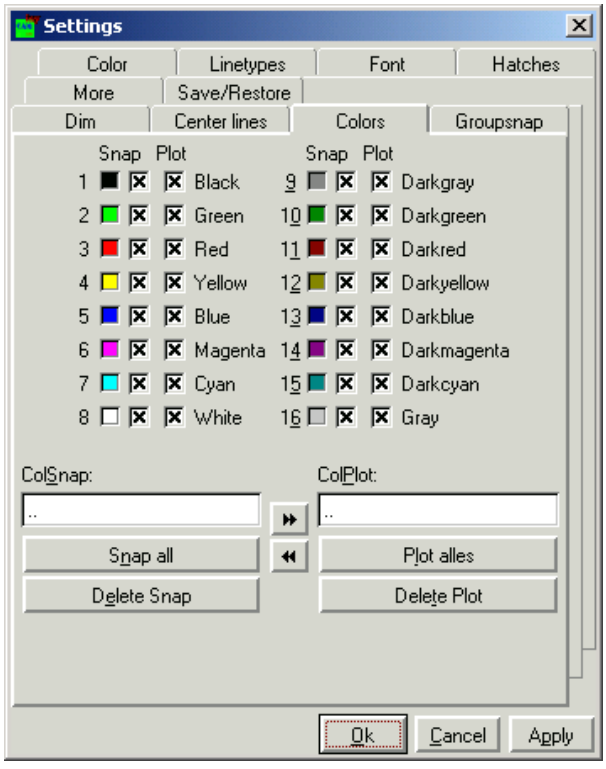


Limiting the View to selected Colours (Variable COLPLOT)

COLPLOT/ COLSNAP:

By setting the variable **COLPLOT**, the colour display can be influenced, and by setting the variable **COLSNAP**, the colour snapping. This is advantageous, for example, if you have assigned various geometry elements different colours when drawing and you want to hide / unhide individual drawing elements or you want to snap only to certain colours.

The variables can be set in the "Settings" menu by ticking. The menu is called via various tool menus and subsequently clicking on the "**G**" button during selection.
The limitations for the colour display do not apply to the shading.



If the variables COLPLOT or COLSNAP are set, the relevant buttons CPL and CSN in the tool menu "Parameters" are displayed in red.

Special Highlighting of the Active Object

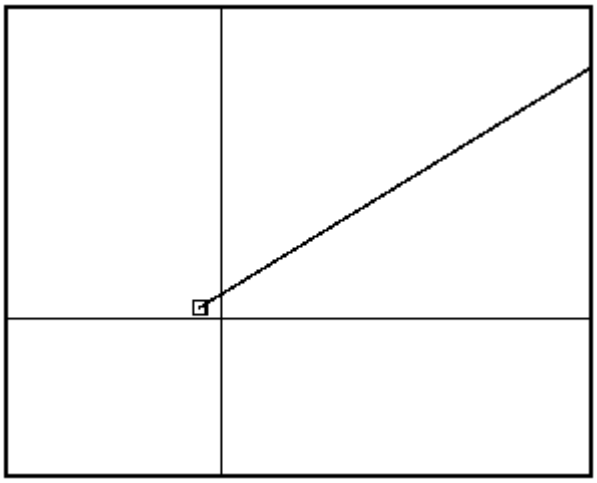
Active object:

With the "ACT" button held down 

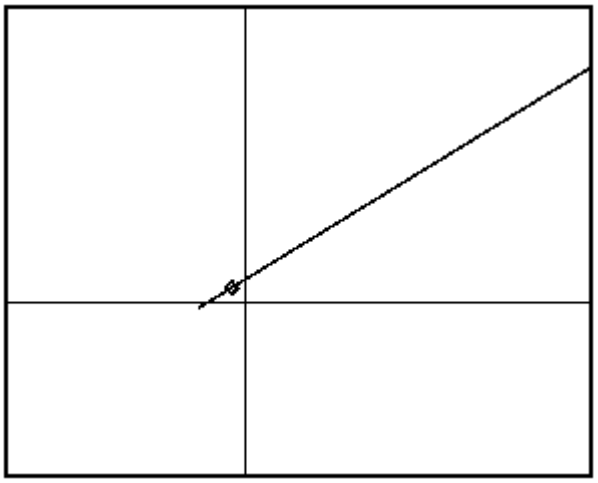
, the object drawn or selected last is highlighted with a special display colour (red-yellow) to indicate to the user which object is affected by his actions. Clicking on this button once more disables the highlighting.

Grabbing Objects Intelligently - Support During and After Point Selection

Grabbing an object: During point selection, object grid points lying in the tolerance range (variable TOL) are displayed dynamically. Preference grid points (such as end points, centre points,...) are highlighted with a square, all the other line points with a rhombus.

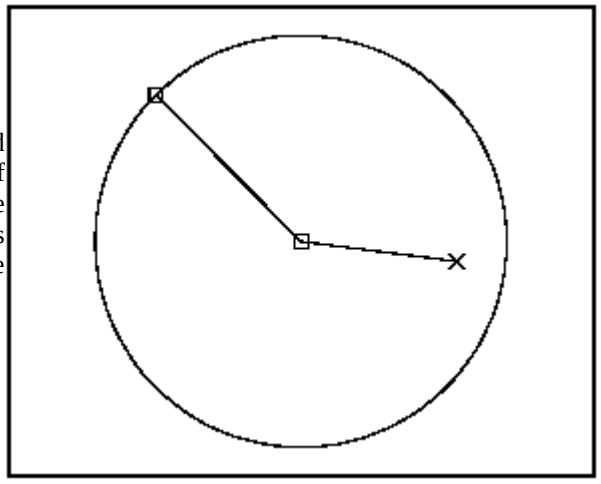


Found preference grid points (here: start or end point of the line) are displayed as a square



The found "any" grid point on the line is displayed as a rhombus.

After performed selection with object snapping, selected points are displayed as a square, and in the case of selection without object snapping, as a cross. In the illustration on the right, when drawing the polygon, it was only snapped to the circle and to the arc centre (preference point). The third point was set without snapping to a grid.



Use the "Settings" menu to limit the object snapping to selected grid points (variable GROUP). It can be suppressed completely by activating the appropriate button in the "Parameters" tool menu.

Snapping Coordinates

Snapping coordinates:

When creating plans and diagrams in the fields of electronics or electrical engineering, structograms and library symbols, for example, the method of drawing with coordinate snapping is often used. When using this method, the computer displays a grid pattern of grid points on the screen which can be used to connect points true to line quickly and simply.

Generally, **isyCAM 2.5 (light)** differentiates the **visible grid** (see **GRID**) and the **snapping function** itself (invisible; **SNAP**). Whereas point markings are merely underlaid to the drawing area by the grid, which can be compared to the use of millimetre paper in conventional drawing, the grid pattern ensures that only the grid point closest to the cross-hair is hit when pressing the mouse button for activation.

Grid pattern snap points can be distributed evenly over the drawing area evenly - in this case we speak of a **continuous grid pattern** - or else created at certain discrete coordinate positions. In the latter case, we will speak of a **discontinuous grid pattern**.

The most important command for handling the coordinate snapping is **snap** (see reference). The procedures GRID3P.PRC, SETSNAP.PRC and RMSNAP.PRC can be mentioned as supplements.

The most important functions of coordinate snapping can be operated easily using the menu displayed on the screen. The appropriate GRID PATTERN menu is called using the sequence

or if you click on the SNAP button in the status bar for a longer period of time, the relevant menu appears as a drop-down menu.

If no grid pattern has even been used so far during the program session, clicking briefly on the above SNAP button 

creates the dialog menu for setting continuous grid patterns.

If a grid pattern has already been generated during the program execution, the snapping can be disabled using the SNAP button (clicked and meanwhile marked in red) in the status bar.

Using the Continuous Grid Pattern

Continuous grid pattern:

To create a continuous grid pattern with a spacing of 50 coordinate units, for example, select the (black) SNAP button 

from the drop-down list. In this case, "**50**" must naturally be entered

as a response to the message **Enter horizontal grid spacing (e.g. 20):**

Because of the default value "<50>", the second input "Enter vertical grid spacing:" need only be acknowledged by pressing <Enter> if we want to have the same spacing in both the horizontal (x) and the vertical (y) directions.

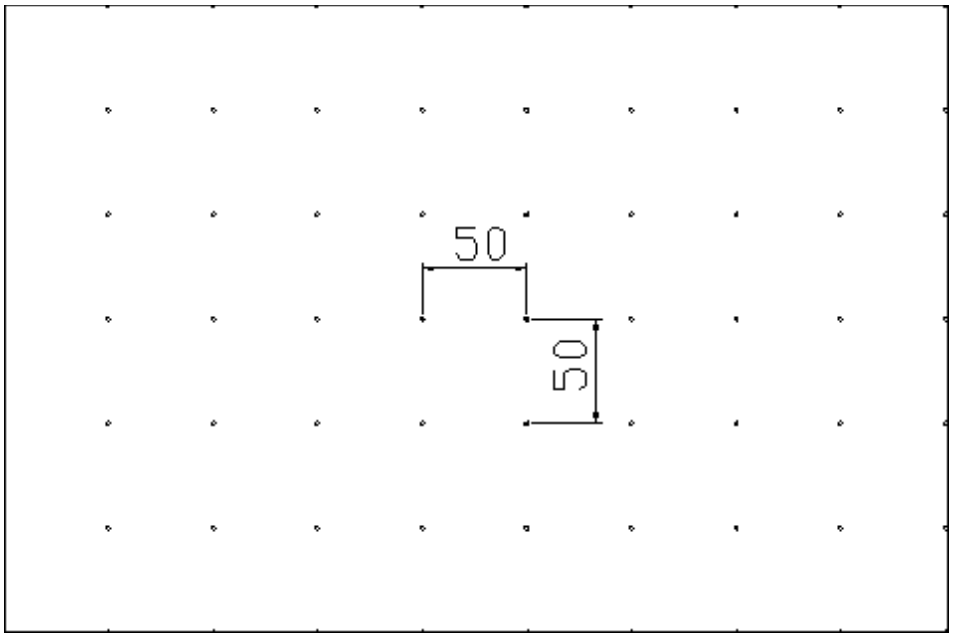
If a grid value other than zero is entered, a white grid (the colour can be changed in the registry) is displayed on the screen. In addition, a clicked SNAP button 

highlighted in red appears in the status display; this SNAP button can be used to disable the grid again (see below SFLAG or GFLAG).
The commands corresponding to the function above could be:

```
grid=50;snap=50
```

or

```
snap x0,50 y0,50 -gsu
```



Grid with setting of the grid origin:

The second method of command input can be used to set the origin of the grid (in this case, the origin of the coordinate system (x0,... y0,..) to any value, if necessary. The options have the following meaning:

- -s (for "snap") Grid pattern function
- -g (for "grid") Visible grid
- -u (for "user") Specification of the coordinates in user units

If after activating the coordinate snapping want to draw a line consisting of several points using the command **poly**, for example,

```
poly *
```

you will see that the cross-hair is only snapped to grid points close to the cross-cursor position whose coordinates are multiples of the grid spacing (here 50; i.e. 50, 100, 150 etc.).

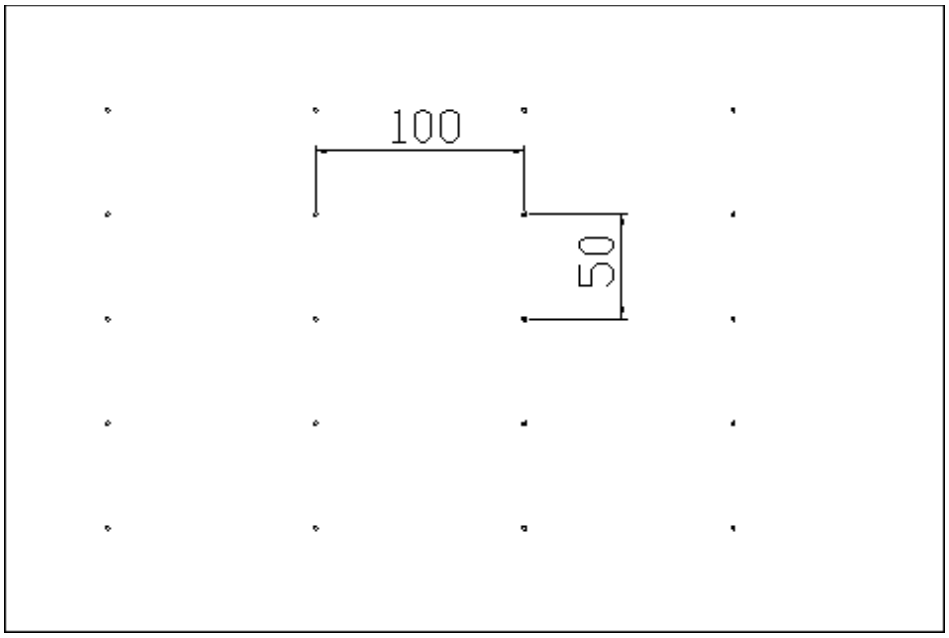
When drawing, observe the running coordinate display. By using the coordinate snapping, it is very easy to create objects with exact dimensions or to place symbols at an exact position.

We mentioned already above that the horizontal and vertical grid spacing can be assigned different values. For example, the two commands below have the effect that the horizontal grid spacing is 100 and the vertical spacing 50.

```
grid=100,50;snap=100,50
```

or

```
snap x0,100 y0,50 -gsu
```

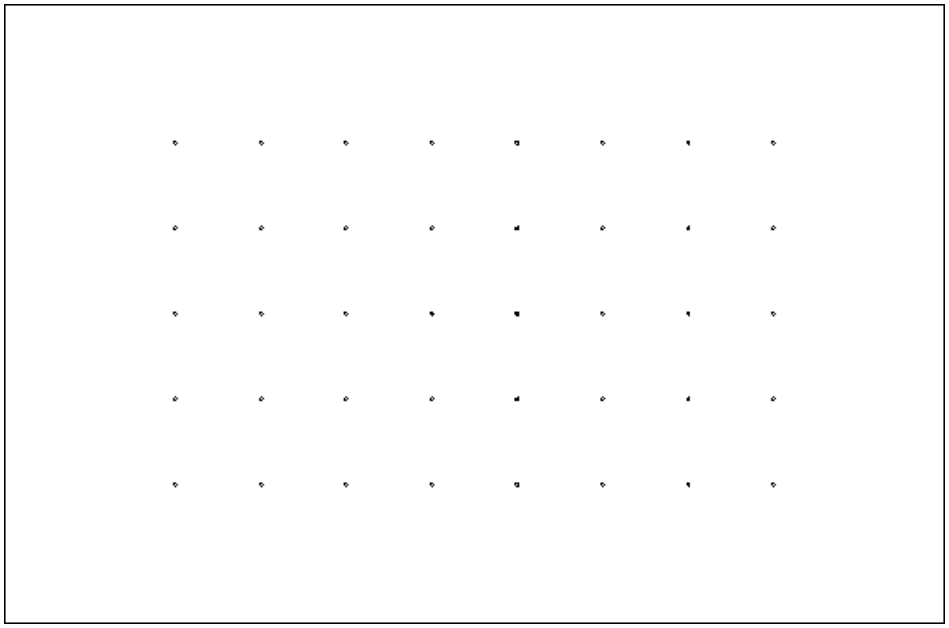


Visible grid:

Similarly, different values can be set for the visible grid (variable **GRID** or option "-g" (for "grid")), irrespectively of the grid spacing (variable **SNAP** or option "-s" (for "snap")). For example, to accelerate building of the display, it is often useful to graduate the visible grid that only serves for orientation less finely than the snapping function (e.g. **GRID=50** contrary to **SNAP=10**).

If you want to occupy only a part of the screen with a continuous grid pattern, with a spacing of 50 coordinate units in a range of x=100 to 500 and y=100 to 400 GDUs, use the command

```
snap x100..500,50 y100..400,50 -gsu
```



Inclined grid patterns:

It is generally also possible to create grid patterns inclined to a horizontal by specifying a parameter "a" (for "angle") with angle specification (0..90).

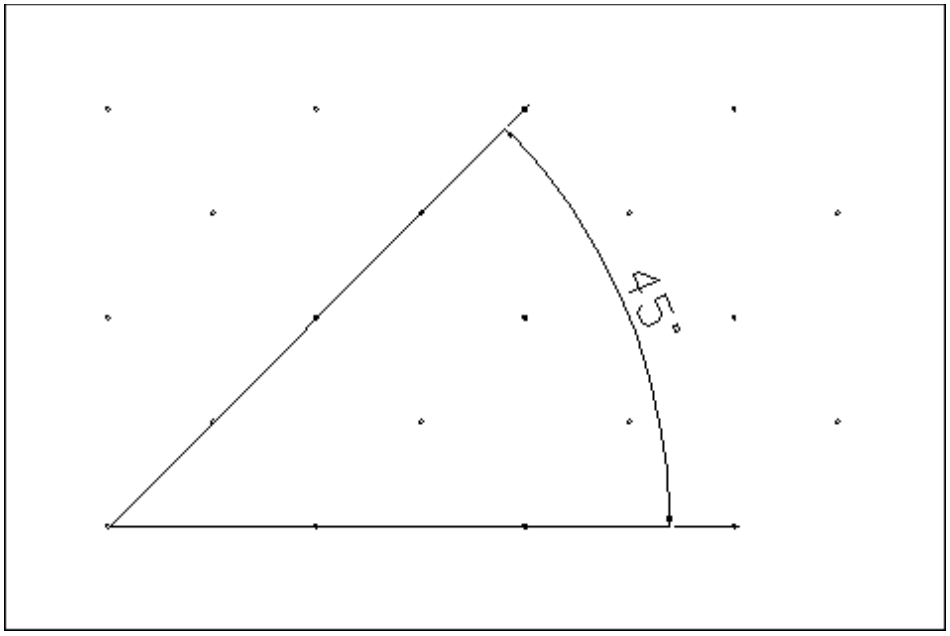
The button 

can be used to rotate an existing grid pattern later by a specified angle.

Similarly, the command

```
snap a45 x0,50 y0,50 -gsu
```

results in a continuous grid pattern with a spacing of 50 coordinate units and with an inclination angle of 45 degrees.



With

```
snap a0 -gs
```

the coordinate window becomes paraxial again.

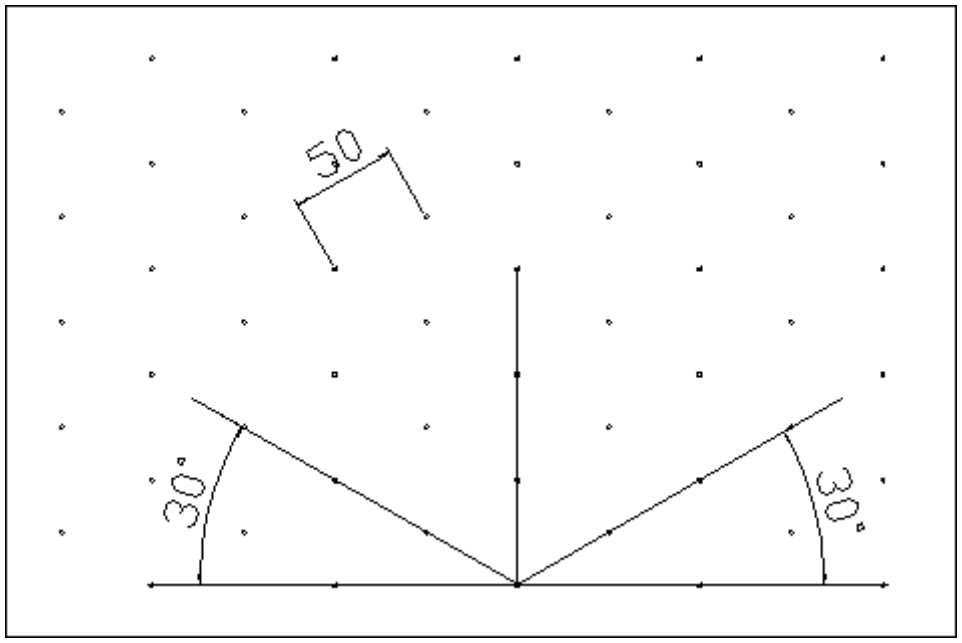
In technical drawing, an isometrical coordinate grid is also often useful. This can be created using the menu option 

.

When you are prompted

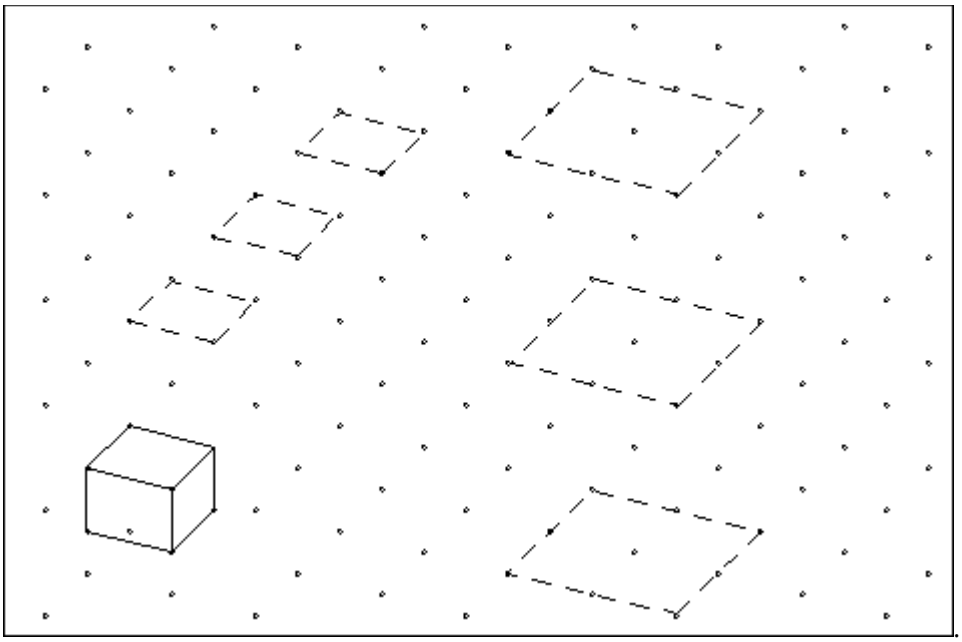
```
Enter grid spacing <20>:
```

enter the desired value, e.g. "50". A grid will result which corresponds to the following illustration. This grid can be used to create isometrical bodies with the angle preference directions (30°, 90° and 150°) by a very simple method.



If certain objects are to be arranged regularly on the drawing area, the procedure GRID3P.PRC may be very useful. By using this procedure, a grid pattern can be defined quickly by three points specified using the cross-hair. To snap to object corners, it is naturally useful to disable a grid pattern (if any) beforehand. You will find this function under the button 

.



Furthermore, it is possible to define a singular grid point. Here is an example:

```
snap x319.5 y255.5 -gsu
```

Deleting a Coordinate Grid

Deleting the coordinate grid:

If you only want to delete the continuous grid pattern, you can do this from the menu using the button 

. Enter "0" (zero) for the value.

The appropriate commands are

```
grid=
snap=
```

If you want to remove all grid pattern points from the screen, i.e. both continuous and discontinuous, enter the command (see Fig. ..., Fig. H)

```
snap -gsc
```

You may naturally also remove the grid or grid pattern separately of each other, e.g. by

```
snap -gc
```

or

```
snap -sc
```

In the practice of drawing, the problem often occurs that you want to deactivate the grid or the grid pattern temporarily, i.e. keeping the numerical values of coordinate snapping (in particular, if the are multi-digit). In such a case, the variables **GFLAG** or **SFLAG** must be set to "OFF".

The activation from the menu is performed by clicking on 

.

In this case, the value is switched from "ON" to "OFF" (or vice versa).

GFLAG=OFF deletes an existing visible grid, whereas **SFLAG=OFF** deactivates the grid function. The relevant commands are:

```
gflag=off
```

or

```
sflag=off
```

The coordinate snapping is reactivated using the original values of **GRID** and **SNAP** by repeated actuation of the button or by entering the commands

```
gflag=on
```

or

```
sflag=on
```

Using the Discontinuous Grid Pattern

Discontinuous grid pattern:

In the practice of drawing, it is often useful to assign selected geometry points (such as centre points, corner points, drillholes etc.) to special discrete grid points. Mostly, this is done in addition to continuous grid patterns. To clarify the principle of functioning, we will set one **single grid point** to the screen centre point using the command:

```
snap x319.5 y255.5 -gsu
```

Such a singular point is naturally more dangerous than useful: If you then try to draw an object (for example, a circle), all object points will coincide in this grid point. No matter where the cross-hair is positioned, you will always only hit the single point x=319.5 y=255.5. It is therefore only useful to complement the grid with further (usually continuous) grid points.

Singular grid points can be deleted via an appropriate coordinate input with the option "-r" (for "remove"). In other words: The command for removing the single grid pattern point above is

```
snap x319.5 y255.5 -gsur
```

With

```
setsnap -x
```

or

```
setsnap -y
```

it is possible, starting out from a grid point already existing and selected interactively using the cross-hair, to enter numerical values for further, singular, horizontally or vertically offset grid points (relative to the starting point). Whether the values are created horizontally "-x" or vertically "-y", depends on which option is selected. The input of a point series is completed by pressing <Enter> and **POLYEND**.

Saving and Loading of Grid Patterns

Commands for saving and loading grid patterns:

If you want to save special grid pattern definitions in a file SNAP.ENV, for example (usually in the subdirectory SETTINGS (see **ENVPATH**)) to be able to use them again later, use the command

```
esave snap -gsu
```

The reverse process, i.e. loading of a saved grid pattern (e.g. in the file SNAP.ENV), is performed with

```
eload snap -gsu
```

With a continuous grid definition with a coordinate spacing of 100 horizontally and 50 vertically, as well as a singular grid point in the centre of the drawing area (x=319.5,y=255.5), the environment text file ("proc settings\SNAP.ENV") contains the following lines, e.g.:

```
Code: @
grid:on
        grid-x:1,0.0000,100.0000
        grid-pt:319.5000
        grid-y:1,0.0000,50.0000
        grid-pt:255.5000
        snap:on
        snap-x:1,0.0000,100.0000
        snap-pt:319.5000
        snap-y:1,0.0000,50.0000
        snap-pt:255.5000
```

Final Remarks Regarding the Coordinate Snapping

Summary: Both the grid and the matrix can be set and deleted using the **snap** command. Thus, the variables **GRID** and **SNAP** are assigned numerical values (or else these values are deleted).

The variables **GFLAG** (for "grid-flag") or **SFLAG** (for "snap-flag") are set automatically to the value "ON" with a value assignment to **GRID** and **SNAP**.

Grid and matrix can be deactivated by

```
set gflag=OFF sflag=OFF
```

keeping the values assigned to the variables **GRID** or **SNAP**. Analogously, activation is performed by

```
set gflag=ON sflag=ON
```

The colour in which the grid is displayed on the screen, is defined in the registry via the parameter "Grid (e.g. white)" (see Configuration.DOC; formerly variable GRIDCOL).

The variable **GRIDMAX** (registry: GridMax (e.g. 100)) can be used to define the maximum number of grid points which can still be displayed on the screen. Default value for **GRIDMAX** is 100, i.e. a maximum of 100 grid points are displayed horizontally one beside the other on the screen. With an assignment

```
grid=10
```

the grid is still displayed beneath the standard screen section (without zooming; after "Zoom Reset"), whereas

```
grid=9
```

suppresses the grid display. This effect can also be traced using the dynamic zooming. If you zoom in, the grid is unhidden, if you zoom out, it is hidden. In other words: The variable **GRIDMAX** (see "Variables") prevents a confusing point representation and excessive screen build-up times in case of insufficient spacing between the grid points. The grid function itself, however, naturally remains active even with the grid display suppressed.

It is now also possible in **isyCAM 2.5 (light)** to change the size of a grid point. The relevant registry input "GridCross 0..3" is mentioned in the file CONFIGURATION.DOC. Assignment of value "3" means maximum size of the grid point, and "0" (default) minimum size of the grid point.

Function Keys and Multi-Key Functions

Preliminary remarks

Operation via keys and using the mouse:

isyCAM 2.5 (light) is operated generally by clicking with the pointing device (mouse, panel pen) on buttons / panel fields or by entering commands via the keyboard. When performing a selection or executing some special commands, various additional functions can be used which can be activated by clicking on a button. These special functions will be introduced in the following.
Furthermore, the creation of multi-key functions (shortkeys / hotkeys) will be described in the following in brief.

Note:

The key labelled with "Shift" is the key which is used, e.g. to create uppercase letters. The key called "Ctrl" can also be labelled with "Strg", depending on the particular keyboard.

see also: Operating Devices

General Function Keys Used During Selection Processes

(after appearance of the graphical cursor, e.g. cross-hair)

Function	**Key / Operation**
Pan (move zoom section using two points)	<P> + moving the mouse with the 1st button held down moves the section
Dynamic zoom	<M> + moving the mouse with the 1st button changes the scale
Dynamic rotation	<R> + moving the mouse with the 1st button changes the viewing angle
Magnifier function – Detail enlargement	<Ctrl> + 1st mouse button
Magnifier function – Detail reduction	<Ctrl> + 2ndmouse button
Quit a selection or cancel -> **POLYEND**	<F6> or 2nd mouse button
Point selection numerically (e.g. centre point of a circle, command **rbarc**)	**<N>,** <X>, <Y>, <Z> (3D), <L>, <W>, <V> (3D)
Select several objects for execution of a command (e.g. moving with window selection, command **mvwin** or simultaneous extraction of several bodies from another, command **bop3d -s**)	Select first object using <Shift> + cross-hair, then select the other objects using the cross-hair; use **POLYEND** to quit selection
Select higher-level macro (e.g. with displacement, rotation, mirroring, multiple deletion ...)	Select first object using <Shift> + cross-hair, select next higher macro by pressing <F10>
Select all surfaces of a body (e.g. conversion of body edges in curves; command **cedges3d**)	Select first one surface, then all surfaces by pressing <F10>

Contour selection - if the contour consists of individual objects (e.g. with extrusion, command **extrude3d** or creation of rotation bodies, command **revolve3d**)	Select first contour using <Shift> + cross-hair, then select the other objects using the cross-hair; use **POLYEND** to quit selection

Special Key Assignments for Individual Commands / Functions

Function	**Key / Operation**
Input of arithmetic expressions (formulas) in numerical input fields of dialog boxes	Press spacebar, then enter formula; press <Enter> to complete your input
Change normal to the surface (command **rfaces3d**)	
Rotate normal to the surface (1 surface)	1st mouse button
Rotate normal to the surface (all surfaces of the body)	<B>
Change both sides / one side of normal to the surface	<S>
Change both sides / one side of normal to the surface (all surfaces)	<Ctrl>+<S>
Change double-side surface of normal to the surface inside / outside (default)	<C>
Select contour when calling **skin3d** and **net3d with option -s** if the (boundary) contour consists **of several partial contours**	Select first partial contour using <Shift> + cross-hair, next partial contours using cross-hair, quit selection of margin curve with **POLYEND**, then select next boundary curve
Tangential contour tracing with edge rounding (command **blnd3d, blnd3d -s**)	Select an edge, then press key <C>

Multi-Key Functions (Shortkeys / Hotkeys)

Defining shortkeys/ hotkeys:

Any user may define multi-key functions, also called shortkeys or hotkeys. Their definitions are stored in the form

```
key = pic:command
```

in a text file with the suffix *.skd in the directory

\Programs\SchottSysteme\IsyCAM\Resources\DE.

The parameter "key" has the form

[ALT+] [CTRL+] [SHIFT+] [APPS +] *letter|digit|function key*

e.g. "ALT+Q" or "APPS+W". In this context, "APPS" stands for the application key on the Windows keyboard (not the Windows key!)

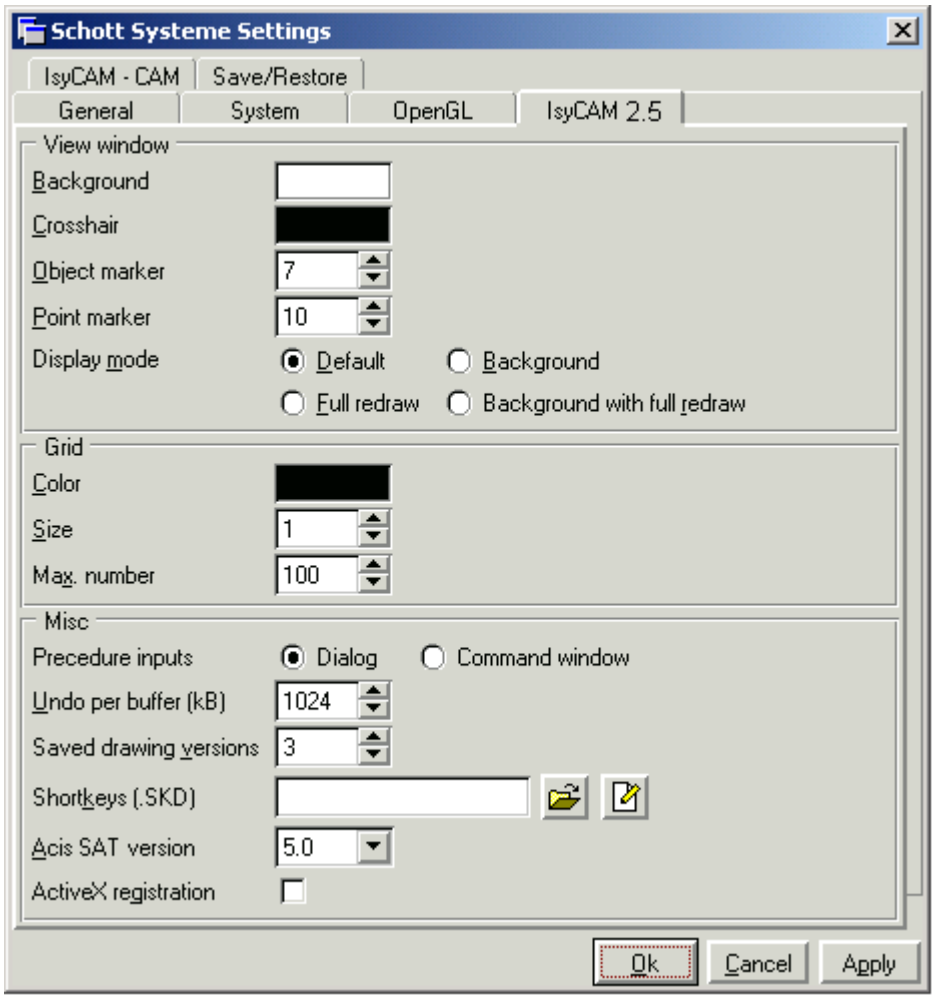
The file "pictures.skd", which already exists, should not be used for user-specific settings, since it is overwritten in the case of updates. The WINDOWS-typical shortkeys defined here should similarly neither be modified, nor otherwise assigned.

Example: File "user.skd"

; user-specific settings
;(comment, beginning with a semicolon)
CTRL+L = pic: label * test
ALT+X = pic: xfind

<Ctrl>+<L> creates the word "Test" as a drawing object; <Alt>+<X> activates your selection.

To activate multi-key functions, specify the file included in SCHOTT SYSTEMS SETTINGS (can be called from the WINDOWS Control Panel, for example).
Subsequently, **isyCAM 2.5 (light)** must be restarted.



Note:

When working with multi-key functions, such as <Ctrl+C>, you should always first press the control key, i.e. <Ctrl>, and then **additionally** the corresponding key, e.g. <C>.

Drawing Parameters

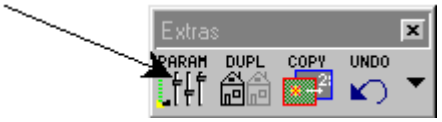
Preliminary remarks

isyCAM 2.5 (light) as an open system provides numerous customised adaptations and settings, ranging from the freely selectable screen layout via user-specific key combinations up to programmability. The present chapter deals with the setting, saving and loading of parameters, such as colours, line types, font type and size, type and size of hatching, settings for dimensioning, etc. We will go into the details of both general and file-specific settings and will demonstrate the relevant procedures taking the example "Setting line widths". Subsequently, we will explain a possibility for automatic loading of defined settings.

Setting, Saving and Loading General Parameters

Editing general parameters:

As a rule, general parameters, such as font or hatching type, colour and size, colours, line types etc., are set before creating a geometry object. One exception is changing of the parameters of an object later. When starting **isyCAM 2.5 (light)**, the settings are loaded from the file **isyCAM.set** (if present; otherwise, from **standard.set)**. The best method for adapting the parameters to your particular requirements is via the "Settings" menu. Please note that this menu provides the option of assigning values to variables interactively.
The relevant fundamentals are to be found in the Section "Important Variables".
The menu is called either by clicking on the appropriate button in the "Tools" tool menu



or by entering

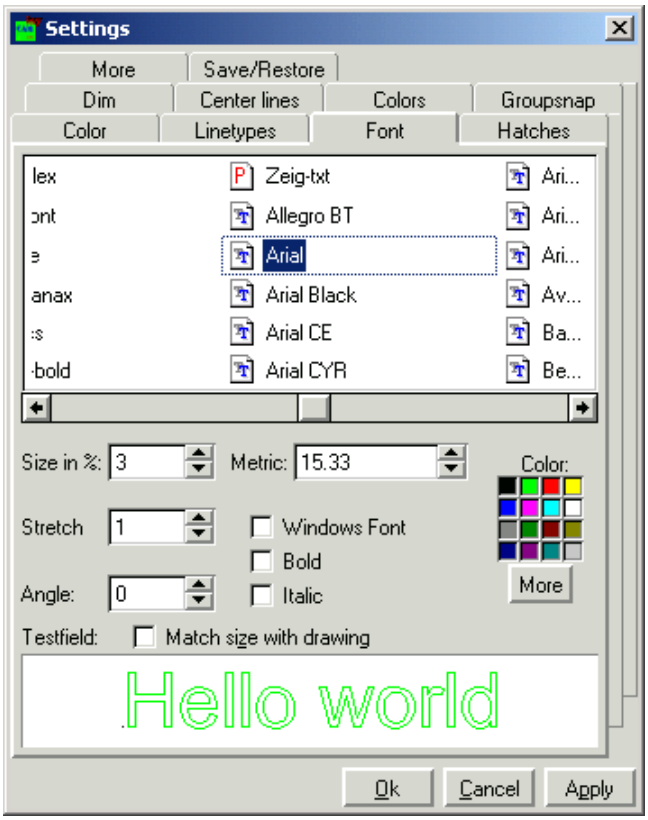
```
settings.
```

After selecting the appropriate tab, you can set the most varied parameters, e.g. parameters for generating and changing objects, such as

- font type, colour and size
- type, colour and size of hatching
- line types
- dimensioning parameters, such as type and size of marking, spacings, number and position of the dimension auxiliary lines etc.
- colour definition via RGB controller, assignment of line widths (line thicknesses)

and parameters determining the behaviour of the system, such as

- group and colour snapping



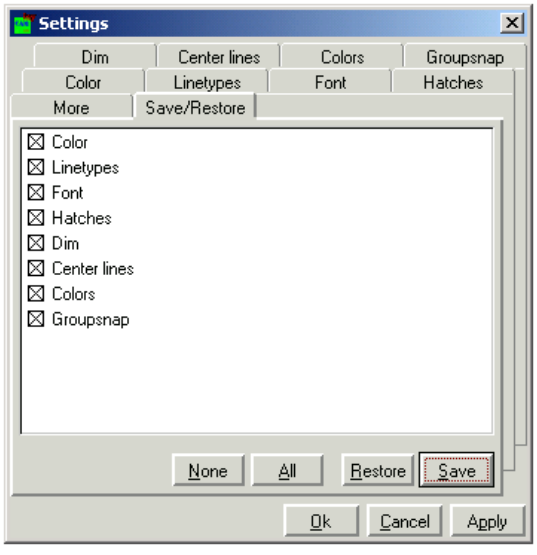
One word regarding the line widths: They can be defined either in mm or in GDU. If specified in mm, the appropriate object is always output in the defined line width without scaling. If, however, you now imagine an A1 drawing printed via an A4 printer, it immediately becomes clear that this is disadvantageous under certain circumstances. Defining in GDU is more advantageous, as in this case the line width is scaled (adapted) accordingly.

Scaling is also possible for hatching and various line types.

Select the desired parameters from the "Save/Load" tab and save them in a file with the extension ***.set** (default: **isyCAM.set**).

The default directory is \SchottSysteme\IsyCAM\Settings (variable SETPATH).

Settings already saved are also loaded using the "Save/Load" tab.



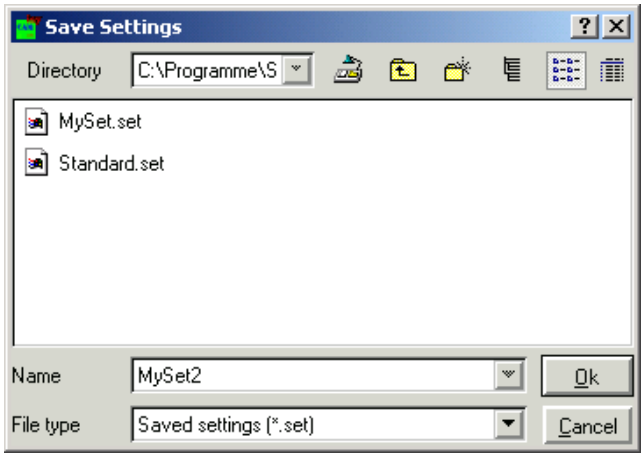
Alternatively, you can also use the command

ssave ***file***

for saving, and the command

sload ***file*** for the loading of settings.

The settings are then saved or loaded as **file.set**. By specifying the option -i, only a certain part of the parameters is taken into account.



Setting, Saving and Loading File-Specific Parameters

Editing file-specific parameters:

In addition to the general parameters, it is also possible to set file-specific parameters. By entering

fsettings

or clicking on the buttons shown below the menu is opened.

The parameters

- colours (RGB controller)
- line widths (positive sign: GDU,
  negative sign: specification in mm)
- Tolerances

can be set in the relevant tabs and be saved or loaded as a file ***.vset**. The default directory is

\SchottSysteme\IsyCAM\Settings
(variable SETPATH).

The commands for loading and saving are

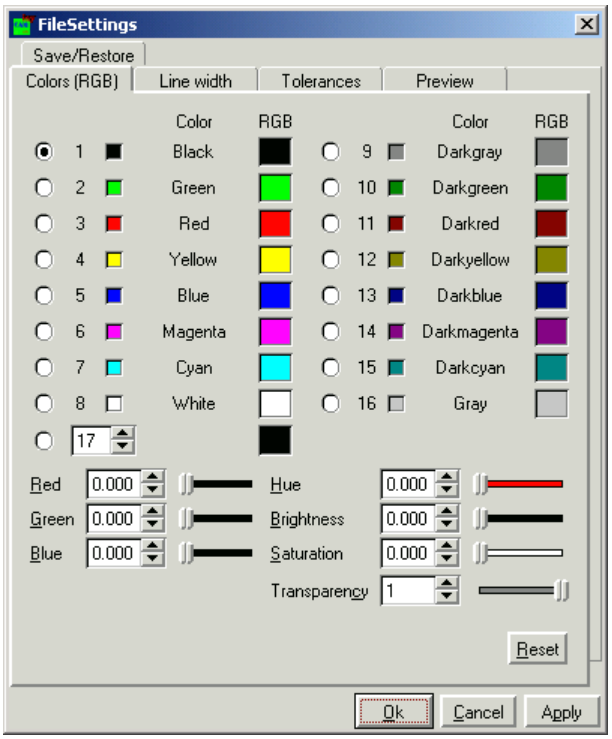
fload *file*

or

fsave *file*.

It is also here possible to select individual parameters using the option -i.

The file-specific parameters are managed in the drawing as global object information (oi).



Assigning and Managing General and File-Specific Parameters

Parameter management:

As already described above, the general parameters are assigned to the geometry objects directly.
This will be demonstrated taking the following example:
After setting the colour "green" for new objects by clicking on the appropriate button or via the assignment

```
color=2
```

and setting of the line type "dashed" either via the menu or via the assignment

```
linetype=-8.8
```

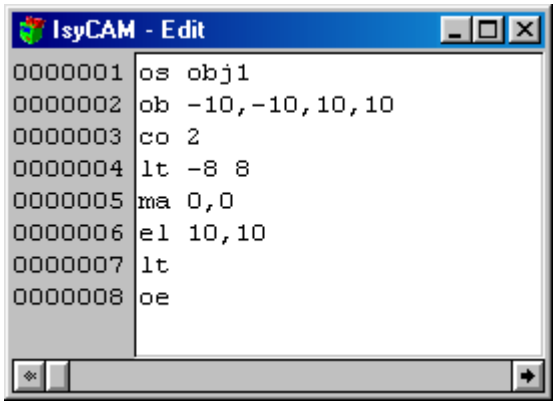
we will draw a circle with radius 10, e.g. by entering

```
arc * -tr10
```

Thereafter, we will open the internal representation of the drawing either by clicking on the relevant button or entering

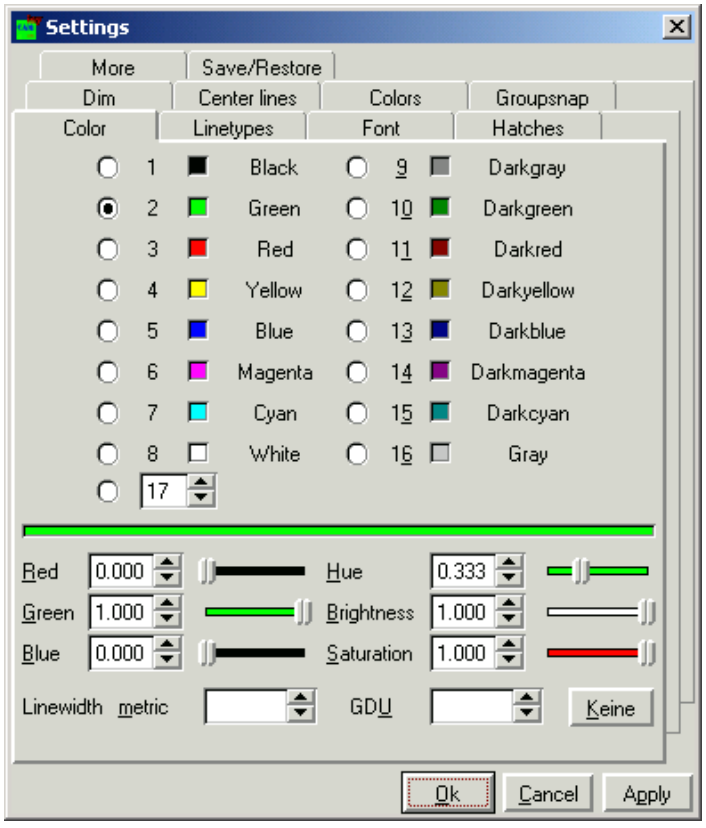
```
edit
```

Object structure: The appropriate assignments can be seen in the object structure.

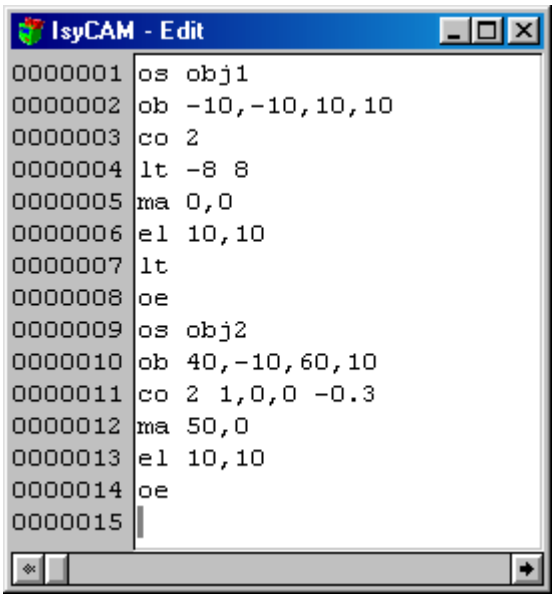


Before we draw another circle with centre 50.0 and radius 10, we will change the colour definition for the colour 2 "green" in the "Settings" menu such that it will consist exclusively of 100 % red and we will define a line width of 0.3mm. The line type will be set to the default value "through".
Then we will draw the circle using the button or the command

```
arc * -tr10 -c50.0
```



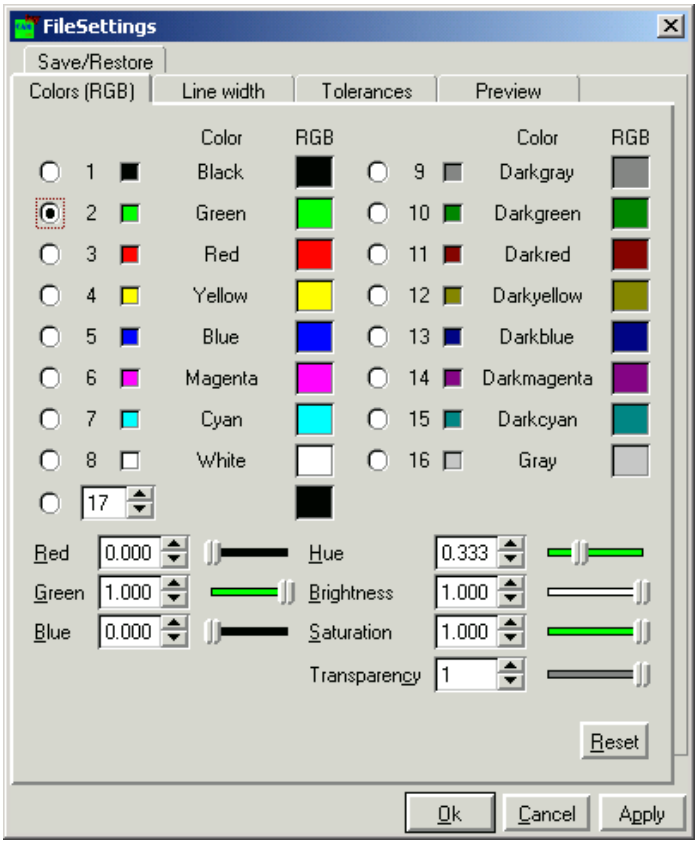
Correspondingly, the second circle (*obj2*) has no parameter for the line type. The parameter for the colour co 2 1,0,0 -0.3 contains the colour number 2 with the defined RGB portions and with the line width. Subject to this definition, the circle is displayed (and printed) in red.



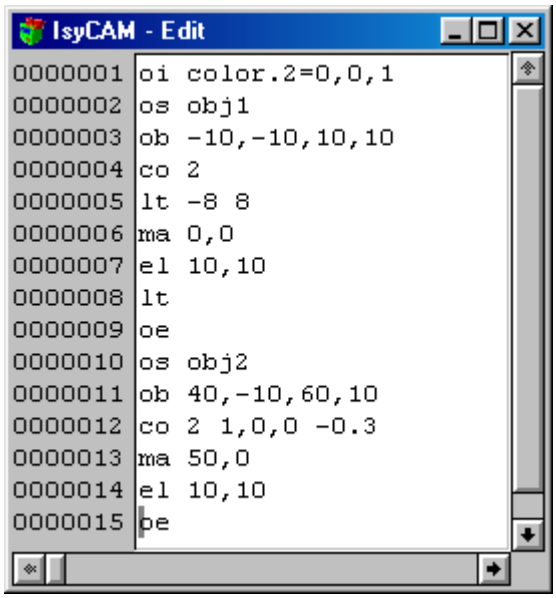
File-specific properties are assigned in a different way. If we change the properties of the colour 2 "green" in the menu after entering

```
fsettings
```

such that it consists of a blue colour portion of 100%, it will be written into the drawing as global object information (*oi*).



All objects in this drawing with the colour 2 (cf. `obj1`), are displayed accordingly, whereas objects with their own specific parameters remain without changes (cf. `obj2`).

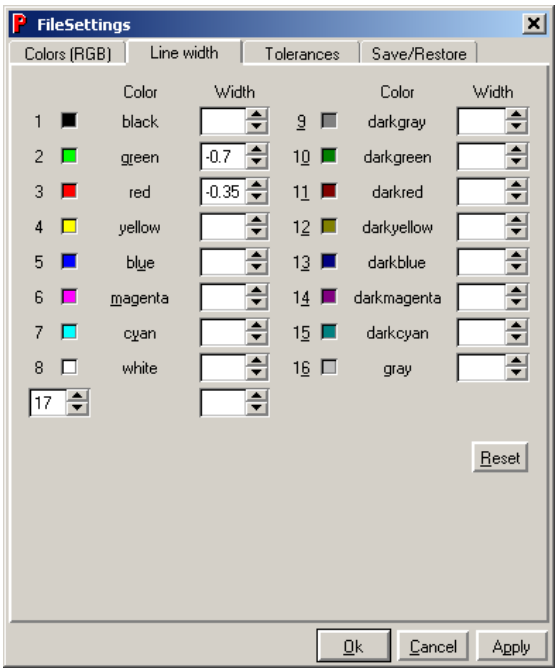


Example: Setting, Saving and Loading Defined Line Widths (Line Thicknesses)

Editing line widths: Subject to the relevant standards (e.g. DIN 15), defined line widths must be used when creating technical drawings, for example, a line width of 0.7mm for thick full lines, depending on the drawing size, and a line width of 0.35mm for thin full lines. This assignment will be explained in brief. Since the line widths are file-specific settings, the relevant menu must be opened after clicking on the buttons described above or after entering

```
fsettings
```

As already described, the line widths in **isyCAM 2.5 (light)** are set via colours. Assuming that thick full lines are to be drawn in green and thin full lines in red, the appropriate values will be entered with negative sign. As explained above, the negative sign means a dimension specification in mm. In addition to the 16 standard colours, it is naturally also possible to define various other colours and to assign them appropriate line widths.



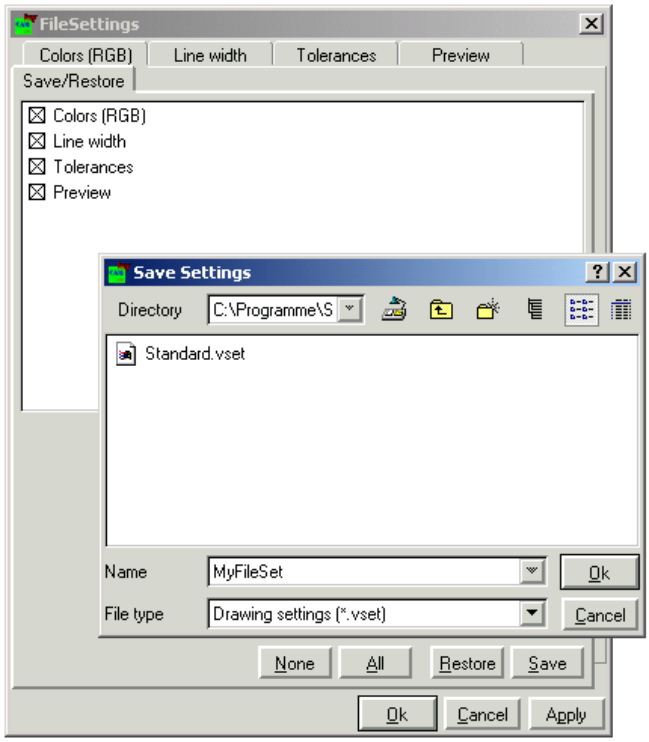
Thereafter, the settings are stored, e.g. in the file **din15.vset**.
Saving can be performed either via the menu or using the command

```
fsave din15.vset
```

Loading of the settings is performed analogously either using the menu or the command

```
fload din15.vset
```

The settings can be loaded prior to or when creating a drawing.



Automated Loading of Predefined Settings

Loading defined settings automatically:

As already described, saved parameter settings can be loaded using the commands **sload** and **fload**.
You can also configure **isyCAM 2.5 (light)** such that the settings are loaded automatically. To this end, enter

```
proc userprofile
```

and edit a procedure that is to be carried out when starting the program and in which user-specific initialisations can be entered. Now enter the appropriate command here, such as:

```
...
  codepage pictures
  options new -m
  trap Error
  trap Inter -i

: ======== User-defined initialisations ==============
  fload din15.vset
: ====================================================
  goto End -f
...
```

Thereafter, save the file **userprofile.prc**, and when **isyCAM 2.5 (light)** is started next time, the settings are loaded automatically.

Summary

By using the appropriate menus of **isyCAM 2.5 (light)**, various general and file-specific parameters can be set according to the particular requirements of the users.
These settings can be saved and also reloaded as necessary. Thus, it is possible to adapt the way of working quickly to different requirements, e.g. depending on drawing formats and output devices.

Numerical Point Selection

Preliminary remarks

When creating a drawing or generating a model, as a rule, the geometry objects are defined by their proportions and their relative position to each other. **isyCAM 2.5 (light)** supports the user with a dialog box when entering these values, thus allowing fast and rational drawing. After explaining some fundamentals, the use of the dialog box will be described in the following.

Fundamentals - Coordinate Systems

Coordinate systems:

To be able to define the position of a point in a drawing, on a model or in the work space of a machine unambiguously, coordinate systems are used. Depending on their intended purpose, these coordinate systems can be defined differently. In most cases, the **orthogonal (Cartesian) coordinate system** and the **polar coordinate system** are used. Both types will be introduced in the following in simplified form; for further information, please refer to the relevant special literature.

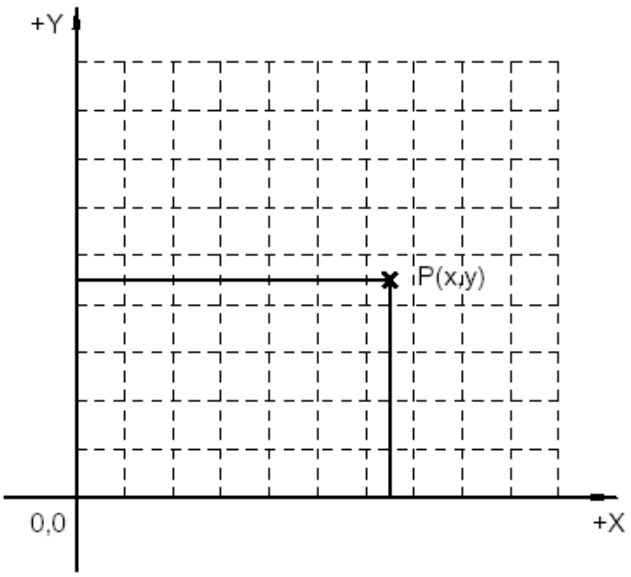
The orthogonal (Cartesian) coordinate system

Cartesian coordinate system:

The position of the points in the plane (2D representation) is described by a pair of coordinates, the x (horizontal direction) and the y coordinates (vertical direction). The point is defined as a pair of values by two numbers x and y delimited by a comma. Decimals are delimited by a point.

Example: 1.234, 5.678

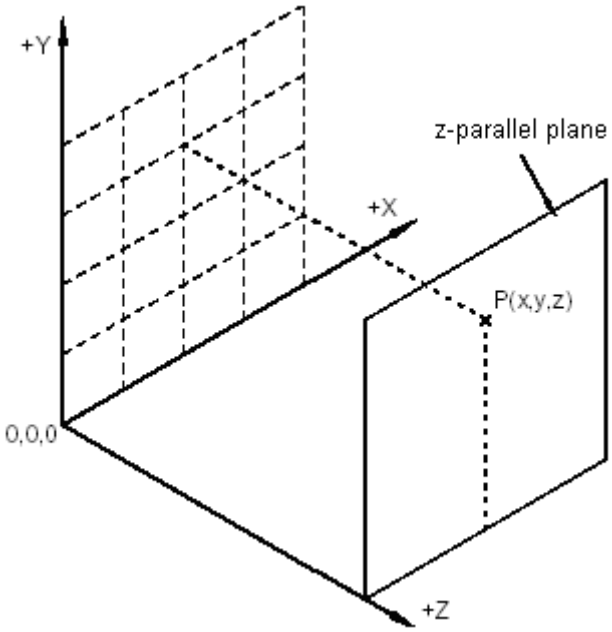
The illustration on the right shows the 1st quadrant (positive x- and y values) of such a two-dimensional coordinate system. The point of intersection of the axes (0,0) is called the origin.



The point position in the three-dimensional space (3D) is defined via a triplet of numbers.
The position is described using three numbers delimited by a comma,

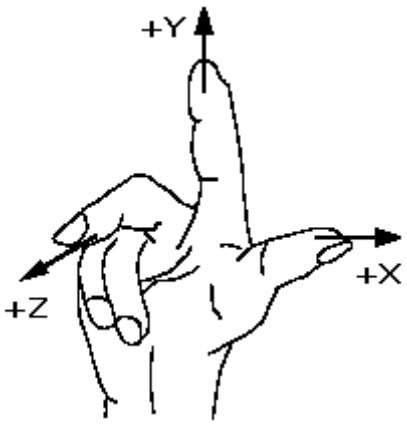
e.g. -45,100.34,0.345

If the x-y plane is displayed on the screen, the z axis protrudes from the screen area to the front.



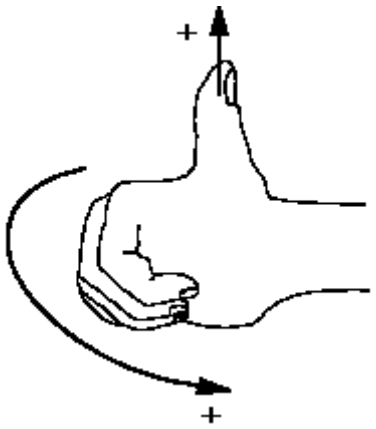
The "right-hand rule" is applicable to three-dimensional Cartesian (orthogonal) coordinate systems.

This rule serves to illustrate how the order and direction of the positive coordinate axes is defined. The term "orthogonal" means that right angles exist between all three coordinate axes.

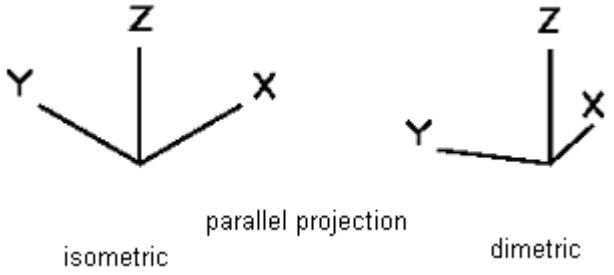


The (mathematically) positive direction of rotation of an axis can be determined according to the illustration on the right:

If the thumb points in the direction of an axis, the 3D direction of rotation is determined by the bent fingers.



Typical 3D views are parallel projections.



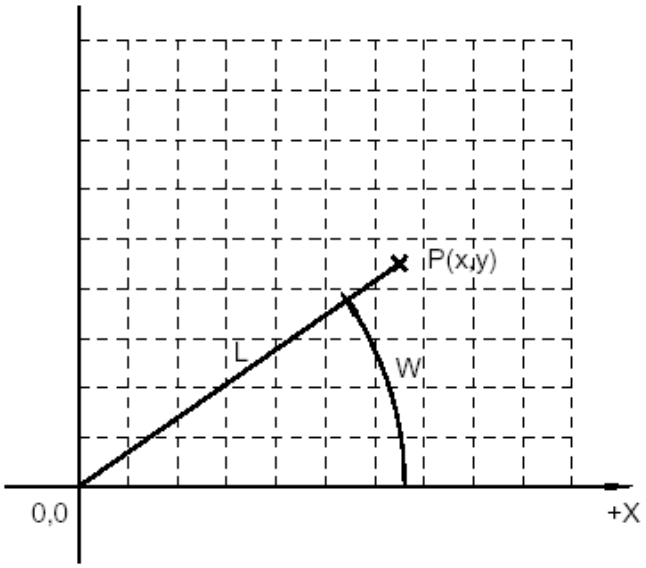
The polar coordinate system

The polar coordinate system:

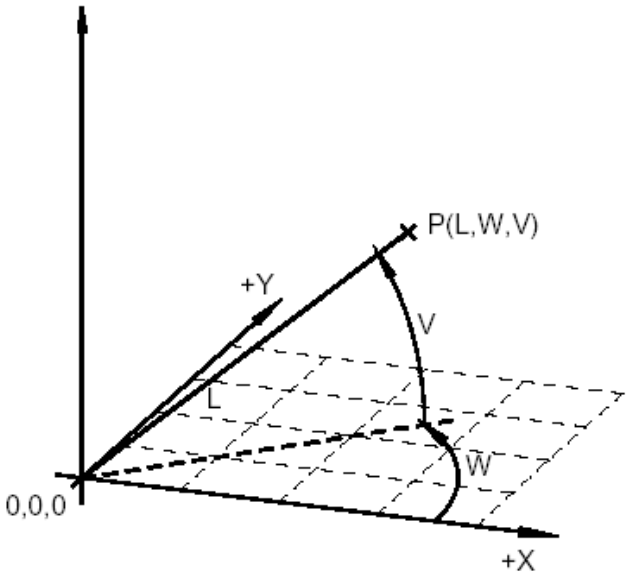
Instead of the Cartesian coordinates x and y, the position of a point in the plane may also be determined by

- its distance from the origin (in the dialog box labelled "L"), and
- the direction in which it lies (viewed from the origin). This direction is defined as an angle to the positive x axis. The angle is measured in the anticlockwise direction. In the dialog box, this angle is labelled with "W".

The position of each point is defined by a pair (L,W) of numbers.



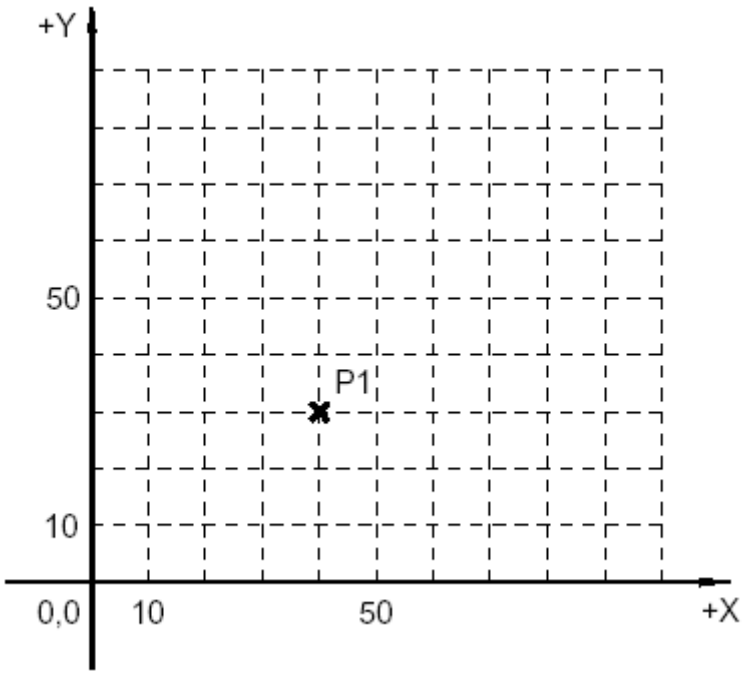
The position of points in the three-dimensional space is determined additionally by a second angle to the x-y plane. In the dialog box, this "vertical" angle is labelled with "V".



Fundamentals - Absolute and Relative Coordinates

Absolute coordinates

Absolute coordinates: When specifying absolute coordinates, the reference point is always the origin of the coordinate system. This applies both to orthogonal Cartesian and to polar coordinate systems.

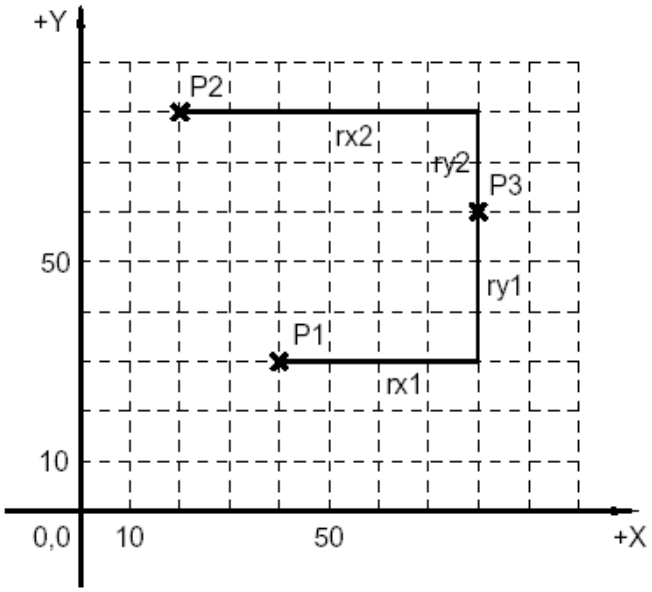


In the diagram above, the point P1 has the (absolute) coordinates 40,30.

Relative coordinates

Relative coordinates:

When specifying relative coordinates, it is not the coordinate origin, but some other point which is selected as the reference point.



In the diagram above, the point P3 has the absolute coordinates 80,60. With reference to point P1, its relative coordinates are 40,30 and, with reference to point P2, 60,-20.

Note:

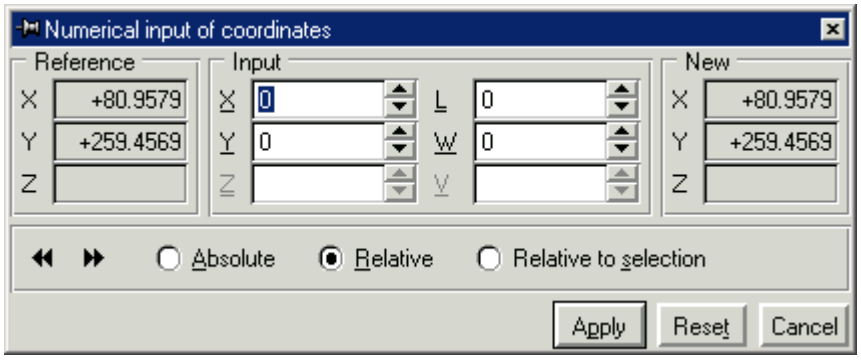
Please note that also with relative coordinate specifications, positive and negative values are defined.

The Dialog Box "Numerical Coordinate Specifications"

Coordinate input via the keyboard:

The dialog box can be called whenever the user is expected to select a point. As a rule, this is the case when the cross-hair appears.
Pressing the key "N" opens the dialog box. Alternatively, you may also use the keys "X", "Y", "L", "W", "Z" and "V"; the latter only with 3D points. In this case, the cursor is immediately positioned in the appropriate input field.

The dialog box is divided into several fields. The three columns "Reference", "Input" and "New" are immediately conspicuous. The modes "Absolute", "Relative" and "Relative to selection" can be selected beneath.



The numerical values are entered by the user in the column "Input" as Cartesian (X,Y) or polar (L,W) coordinates. This can be performed either by scrolling, direct input or as a formula after pressing the spacebar (e.g. "5/9", "3*pi" or "are(30)", cf. command CAL). Cartesian and polar coordinates are converted mutually.

The column "New" always displays the absolute Cartesian coordinates of the point to be selected in accordance with the inputs. You may follow it conveniently by changing the input values.

The column "Reference" displays the absolute Cartesian coordinates of the reference point when entering relative coordinates.

In the input mode "Absolute", this column is without meaning; the point is selected at a defined distance to the coordinate origin.

The coordinates of the reference point are displayed in the input mode "Relative" in the "Reference" column. As a rule, this is the point selected last. It is, however, also possible to select the previous points by browsing.

The reference point is defined in the input mode "Relative to selection" by selection using the cross-hair. The column "Reference" displays the coordinates of the cross-hair. This can be followed easily by moving the mouse. Pay attention also to the changes in the column "New".

The individual modes may also be activated by pressing the underlined letter - "B" for "Absolute", etc.

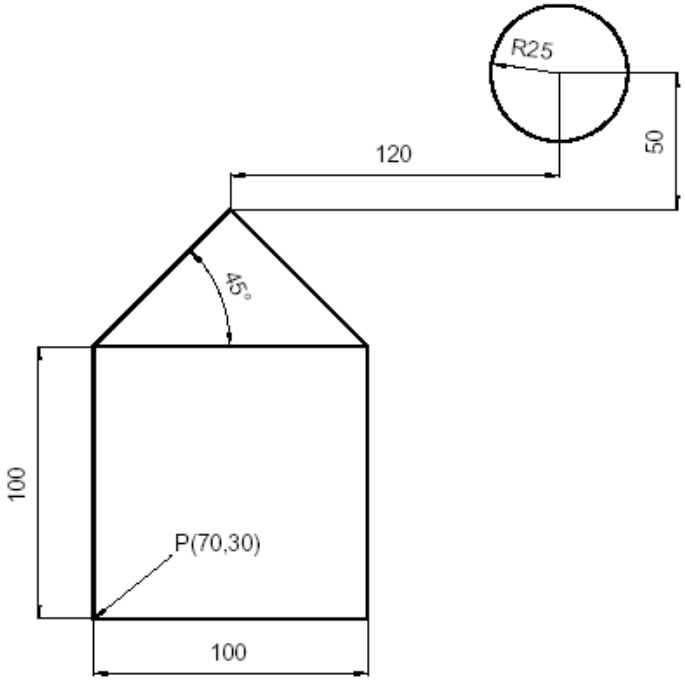
By clicking on the button "Accept" in the modes "Absolute" and "Relative", the coordinates are accepted according to your inputs. You may reset all input values to "0" by "Reset" or else cancel the process of selection with "Cancel".

Upon completion of the drawing process, the dialog box is hidden.

Numerical Point Selection - A Simple Example

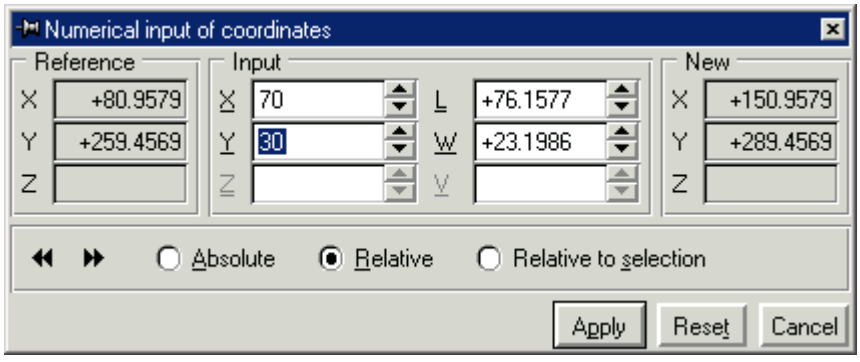
Example:

Let's do a short exercise for numerical point selection using the drawing shown on the left:

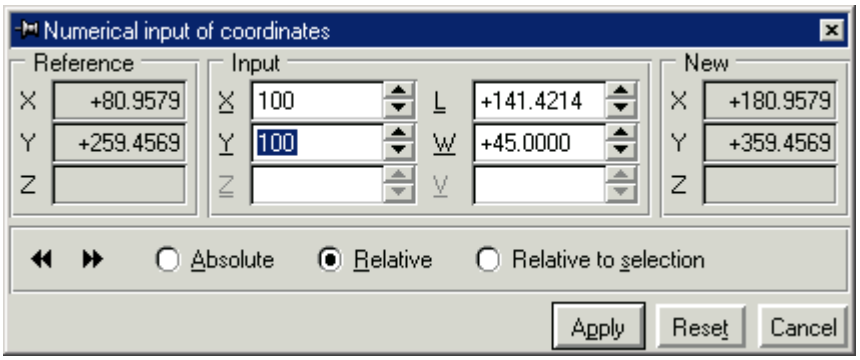


The following steps of work would be possible:

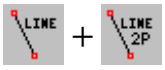
1. Drawing the square
After you have been prompted to "Select the first corner", click on "**X**" to open the dialog box. Enter the values 70 for X and 30 for Y as indicated on the drawing. Then click on "Absolute" to accept the values.



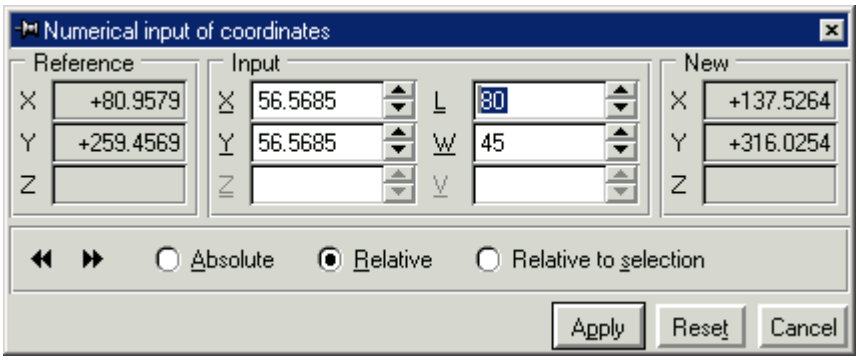
When you are prompted to "Select the second corner of the section", enter the values 100 for X and 100 for Y. After selecting the mode "Relative", click on "Accept" to select the second corner point; thereafter, the dialog box is hidden.



It is obvious that the previously selected point 70,30 is now suggested as the reference point. The "new" point has the absolute coordinates 170,130.

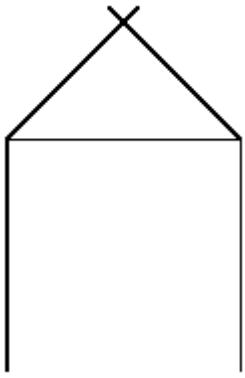
2. Drawing the "inclined roof surfaces" 

First, we will draw the left oblique line. After you have been prompted to "Select the first point", select the upper left corner of the square. The direction of the second point is known. It is specified by the angle of 45°. Correspondingly, we will open the dialog box by pressing the key "W" and will enter here the value "45". The length "L" is initially assumed with "80". Then, we will activate the mode "Relative" and "accept" the values.



Please note that the first point of the line is now suggested as the reference point.

The second inclined roof surface is drawn accordingly, but with the difference that the upper right point of the square is selected here as the first point on the line and 135 is entered for the angle.

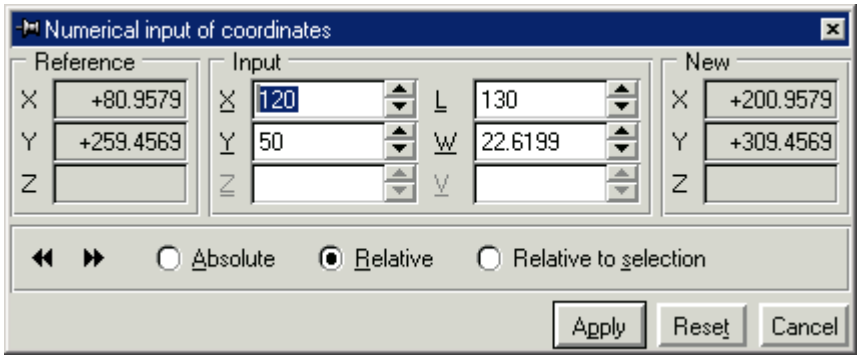


3. Trimming 

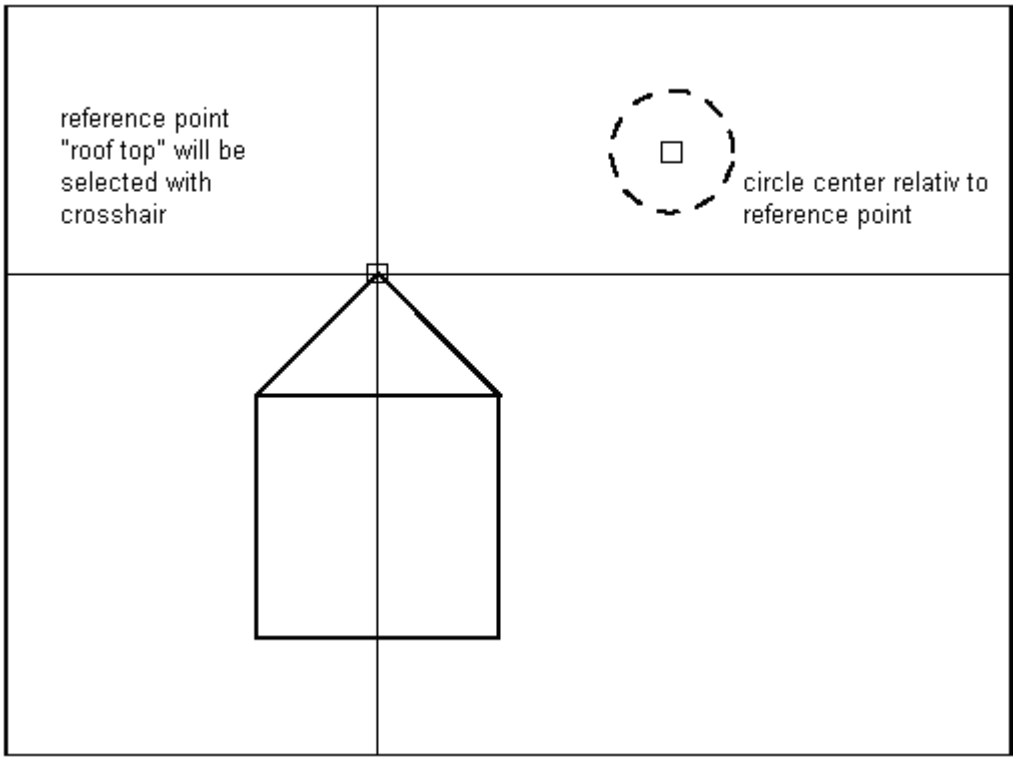
Since the length of the "inclined roof surface" was only estimated, they must still both be trimmed. After clicking on the button "Trim two elements", the part of each "oblique line" beneath the joint point of intersection is selected. Thereafter, quit the function with **POLYEND** (key "F6" or second mouse button).

4. Drawing the "sun" 

The "sun" is drawn according to the drawing as a circle with radius 25. The position of the centre point, which can be determined conventionally only by auxiliary constructions, is problematical.
The numerical point selection, however, will solve this problem easily. After calling the circle function and selecting the radius 25, the dialog box is displayed by pressing the key "X". Enter the values "120" for X and "50" for Y, and select the mode "Relative to selection". Thereafter, select the tip of the roof as reference point (if necessary, move the dialog box beforehand). The circle will then be drawn relative to it.



Please note that the appropriate position of the cross-hair is displayed in the window "Reference", and the new point position in the window "New".



Summary

2D and 3D points can be selected quickly and efficiently using the numerical point selection. To do so, both Cartesian and polar coordinates can be used. Auxiliary constructions when drawing are therefore to a large degree no longer necessary.

Setting the Line Widths

Line widths

Line widths and colours:

Each drawing colour in **isyCAM 2.5 (light)** can be assigned a freely selectable line width. Line widths become visible as such only when printing or plotting (using suitable devices); they do not appear on the screen.

Line widths can be assigned to the colours of a drawing using various methods:

- using the **FileSettings dialog box** (effect: entire drawing)
- directly **when printing** (effect: entire drawing)
- as the "current" colour **(COLOR) in the Settings dialog** (effect: generated objects)

Note:

Drawing colours are represented by integer numbers whose actual colour representation on the screen or on a suitable printer is effected by RGB colour assignments.

When using the methods 1 and 2, the colour numbers ("**co**" drawing commands with one parameter only) of all objects of a drawing are interpreted with certain line widths or RGB values. The appropriate assignment is defined at the beginning of the drawing using "**oi**" drawing elements.

When using method 3, the current colour (variable: **COLOR**) are assigned an integer colour number, three RGB values and a line width and is written to the drawing file directly after each appropriate "**co**" drawing command. All newly created objects are generated using these values contained in the variable **COLOR**. These fixed assignments after the "**co**" drawing commands prevail over the interactive assignments of the "**oi**" drawing commands, i.e. line widths which can be defined using the "**co**" drawing command cannot be interpreted otherwise when working with "**oi**" assignments.

Defining line widths using the "FileSettings" dialog box 

The settings made in the "FileSettings" dialog box are stored at the beginning of the drawing (VEC file) using "**oi**" drawing commands, i.e. each drawing may contain its own user-specific settings with reference to RGB assignment, line widths and representation accuracies. User-specific default settings, however, need not be made individually for each drawing, but can be loaded from a file (stored beforehand) (e.g. STANDARD.VSET).

Example:

The colour "2" will be assigned a line width of 0.3 mm: -> The appropriate command line is stored at the beginning of the drawing: "oi linewidth.2=0.3" (EDITOR).

Calling via the menu: 

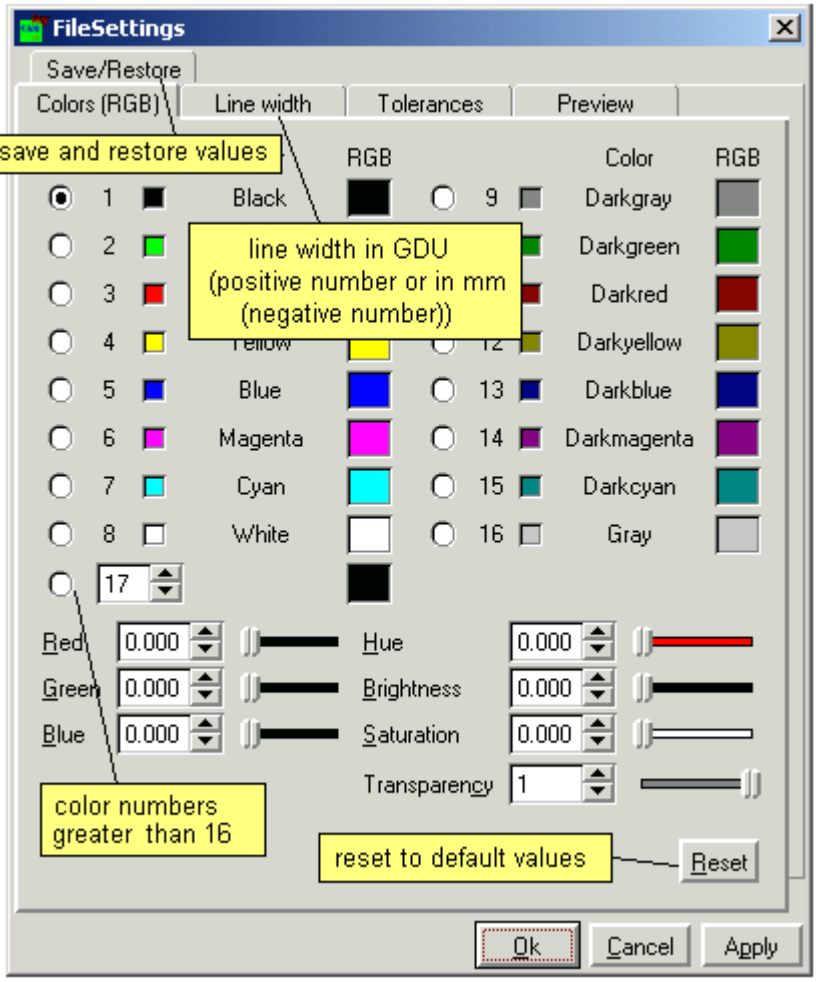
or using the command: **`fsettings`**

Procedure:

- Call the function (FILE-SETTINGS).
- Assign the colours line widths.
- Save the values in your drawing.
- Saving values possible (see above).

Note:

The line widths are saved for a specific drawing. The relevant information is to be found at the beginning of the buffer.



Defining line widths immediately before printing

Line widths when printing:

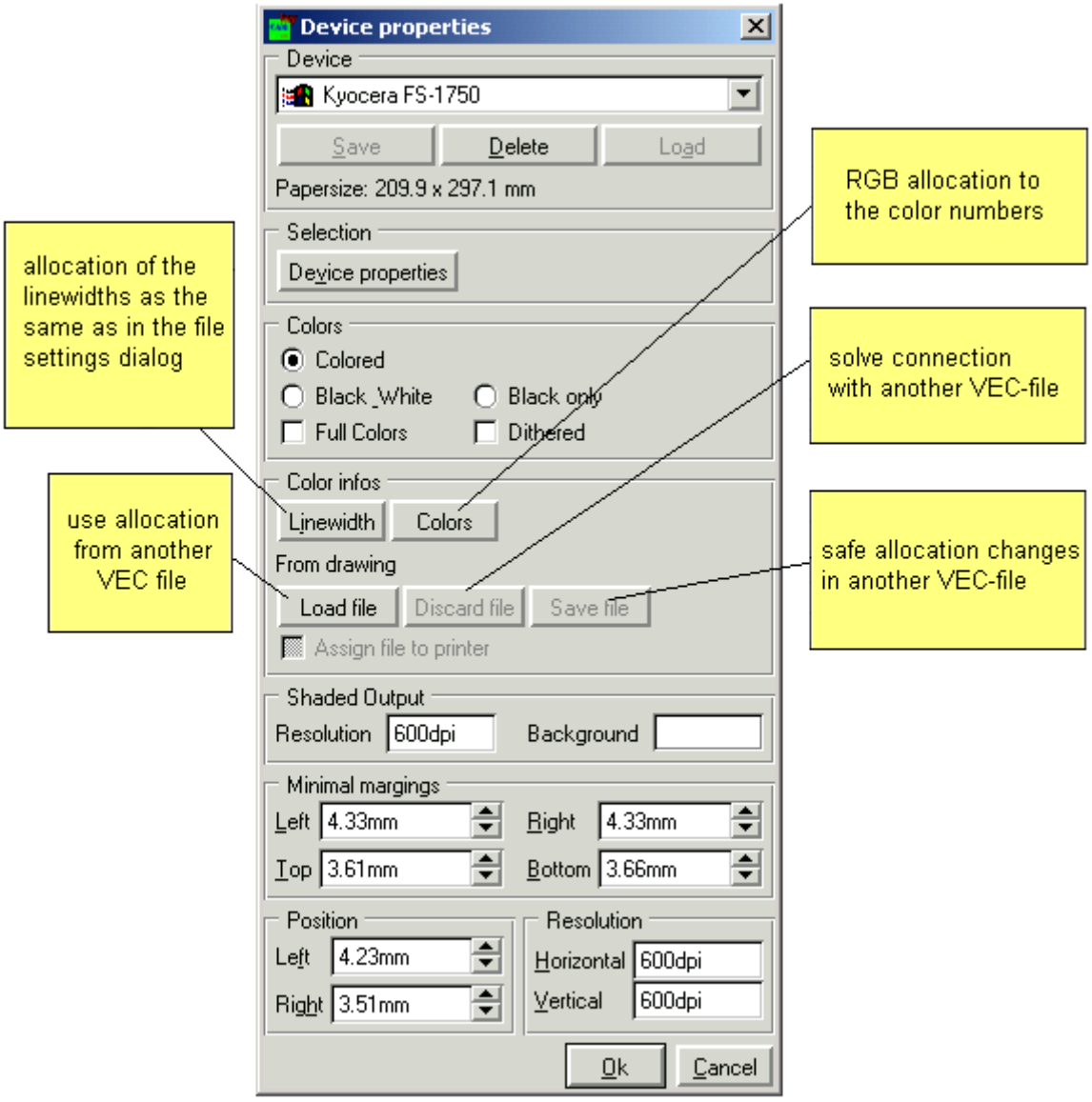
Calling the dialog menu via: Datei + Drucken ... + Eigenschaften ...

or using the command: **output**

Procedure:

- Call the printing menu.
- Dialog box: Activate properties.
- Edit the colour assignment.
- "ok" and "output"

Notes: The line widths can be loaded from another drawing (VEC file) without modifying the drawing to be output.



The desired line widths can be assigned to the **isyCAM 2.5 (light)** colour numbers by pressing the key "Line thicknesses". The relevant definitions are also pasted at the beginning of the drawing using "**oi**" commands, similarly to the "FileSettings" dialog.

By pressing the key "Load file", the desired line width interpretation can be loaded from another drawing (VEC file) (e.g. default specifications).

The key "Save file" only comes into effect if another line width file has been loaded beforehand. This key can be used to save any modifications to the line width interpretation to the file from which they have originally been loaded; to do so, use the "FileSettings" dialog (key: "Line thicknesses").

By pressing the key "Remove file", the line width interpretations of another VEC file loaded beforehand are undone, i.e. the drawing to be output is now output again without external interpretation of the line widths.

Defining line widths using the variable: COLOR ("Settings" dialog)

Line widths using the "Settings" dialog:

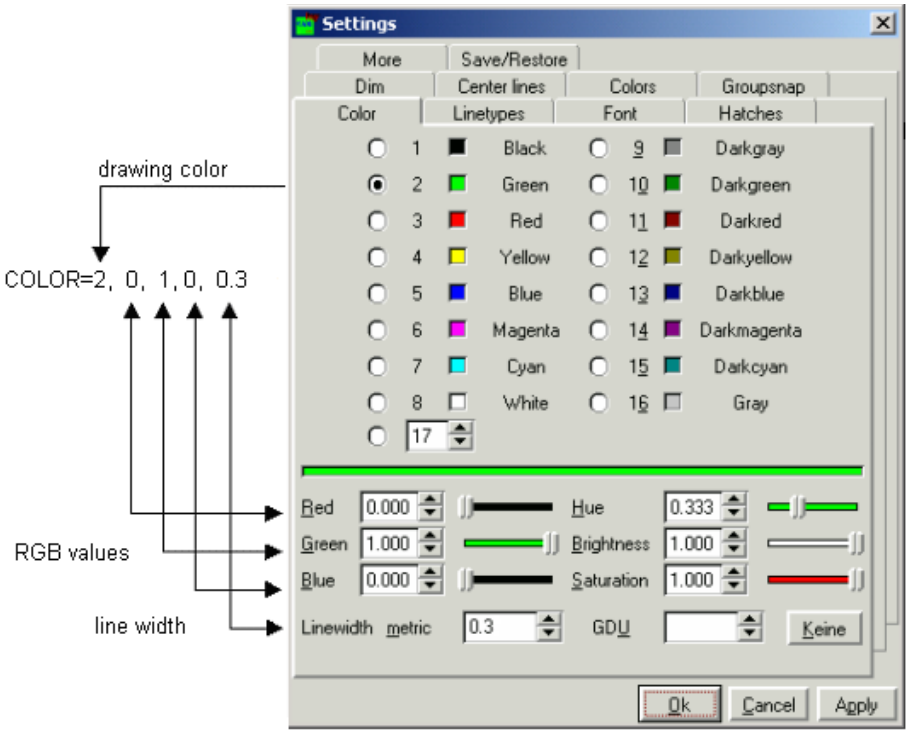
Calling the dialog menu via: 

or using the command: `settings`

Procedure:

- "Settings" dialog: Call "Colour".
- Select the number of the current colour.
- If necessary, assign other RGB values.
- "Accept" the settings into the variable COLOR.

Notes: All new drawing objects will be generated with these settings. These assignments cannot be changed using the "File Settings" dialog!



In addition to the integer **isyCAM 2.5 (light)** colour number, the variable **COLOR** also contains RGB values and a line width. In the drawing file, this supplementary information is written directly after the "**co**" drawing command. Thus, drawing objects with the same colour numbers may contain different colours and line thicknesses, irrespectively of the interpretation assignments made in the "FileSettings" dialog ("**oi**" drawing commands at the beginning of the file).

Important Variables

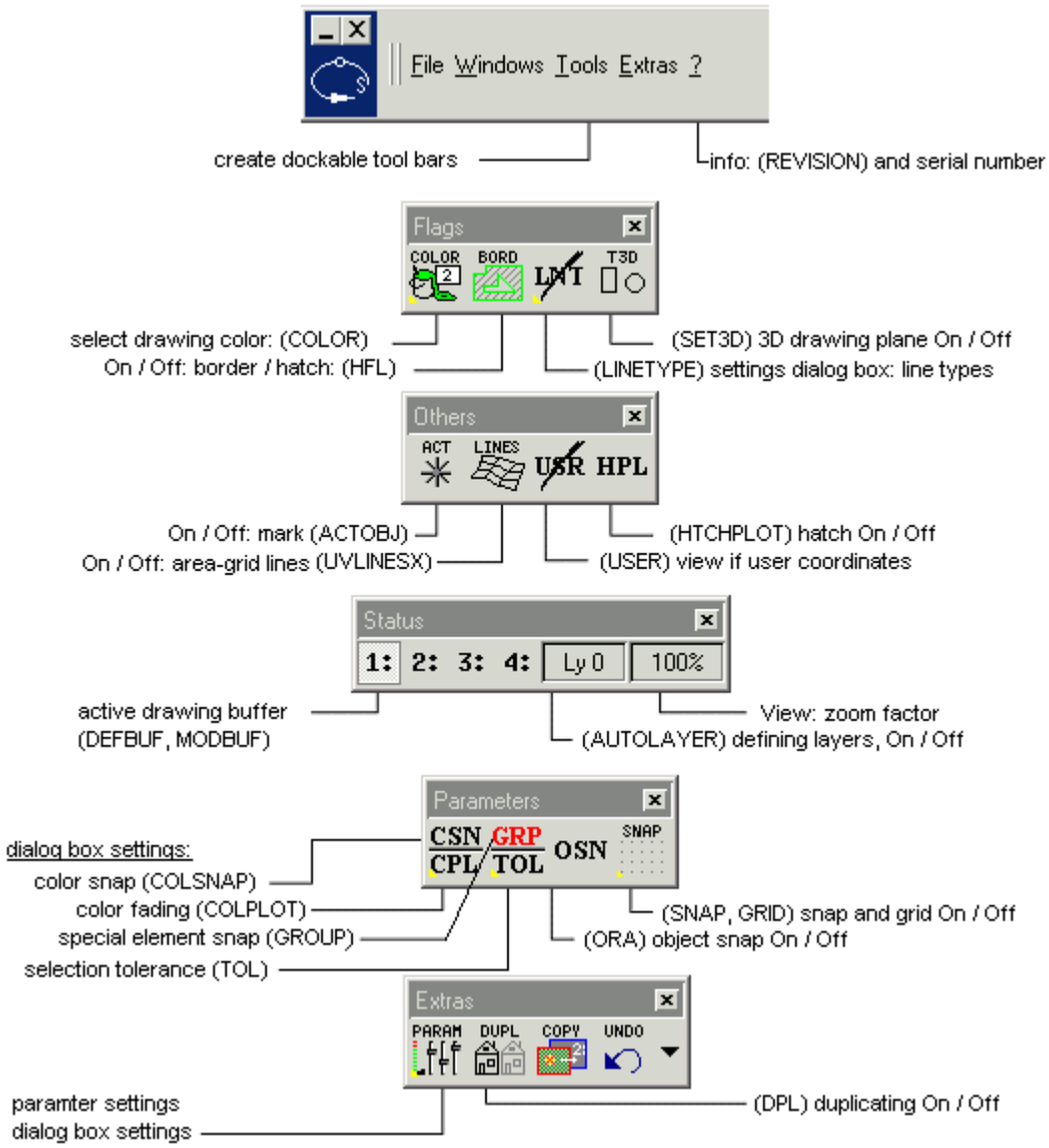
Variables in isy:

Variables serve to control commands and procedures of isyCAM 2.5(light). With different variable assignments, the same functions may show a significantly different behaviour! There are different types of variables:

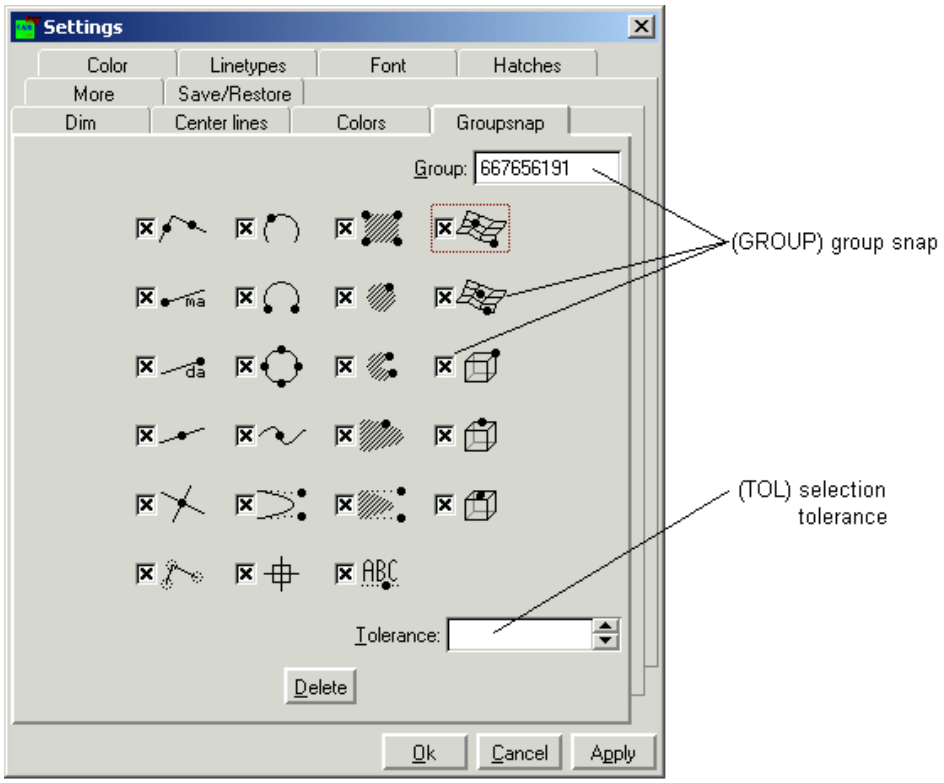
Types:

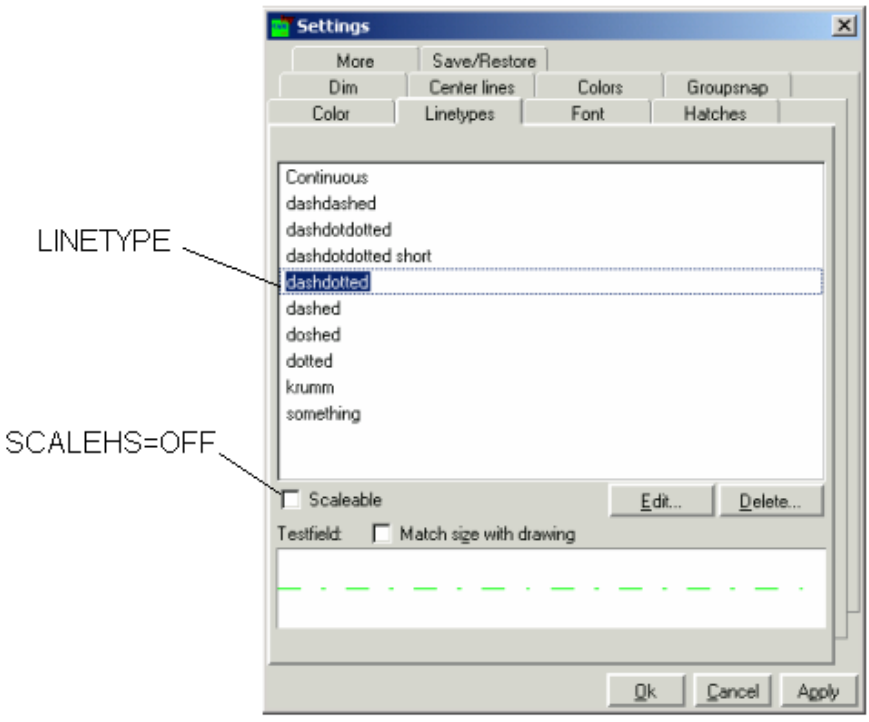
C (Checked)	Assignment according to a prescribed syntax
F (Free)	freely available and can be modified freely
R (Read only)	can only be read

System variables of PICTURES by PC are predefined and can, in addition to the direct input in the command box, also be set using buttons or dialog boxes, whereby the prescribed syntax must be observed when using the direct input. It is recommended to dock the tool boxes using the variable buttons in the "Global" bar of isyCAM 2.5(light):



Examples of parameter settings using the "Settings" dialog box:





In addition to the above mentioned assignments via a dialog box, variables can always be polled or reassigned using the "**set**" command. In the case of value assignments using the "**set**" command, always observe the correct, variable-specific syntax!
For example:

1.) Interrogation: set coloro Display: "COLOR=2"

2.) Assign value: set color=3 The variable COLOR now has the value: "3"

3.) Check: set color Display: "COLOR=3"

It is also possible to define new user-specific variables (e.g. "set today=Monday"). The variable "today" is there of the type "F" and has the text "Monday" as the value.

Interrogation: set today Display: "TODAY=Monday"

In the case of system variables, an empty assignment will delete the contents:

e.g. "set zoom=" ➔ Reset

User-specific variables are deleted with an empty assignment:

e.g. "set today=" ➔ The variable "TODAY" previously defined is removed again.

Overview of the Most Important System Variables

General:

Name:	**Type:**	**Contents / Meaning:**
ACTOBJ	F	contains the name of the "active" drawing object (the object last processed by the system) => ACTOBJ=*#:objectname* (# is the number of the drawing buffer*)*
ARR	C	limiting symbol for dimensioning ARR=length,type length: Length in GDU's type: type of the symbol: 1: arrow, 2: circle, 3: line
AUTOCAT	C	contains object names for automatic linking. => "SET AUTOCAT=PART" ; All objects generated from now on will be assigned to the macro object "PART". Up to and including "SET AUTOCAT="
AUTOLAYER	C	Layer no. in the range 1..1000; last parameter of the "ob" drawing command
BOX	F	Object box: Coordinates of the enclosing rectangle of a 2D object or, in the case of 3D objects, of an enclosing cuboid 2D: BOX=Xmin,Ymin..Xmax,Ymax 3D: BOX=Xmin,Ymin,Zmin..Xmax,Ymax,Zmax 2D+3D: BOX=Xmin,Ymin..Xmax,Ymax Xmin,Ymin,Zmin..Xmax,Ymax,Zmax
CELL	C	Designation of the current SHEET cell, (e.g. CELL=EA1)
CHAMFER	C	2D chamfer when trimming: CHAMFER=dx, dy (corner distance)
COLOR	C	Curves, lines and circles are created in the COLOUR (1..32).
COLPLOT	C	Colour filter: Only the specified colours appear on the screen, => for example: COLPLOT=2,3..5,9,12.. ; all colours visible: COLPLOT=
COLSNAP	C	Colour snapping: Only objects with the colours specified under COLSNAP can be snapped. Syntax as with COLPLOT

DEFBUF	C	contains the designation of the drawing buffer that incorporates parameters for certain drawing functions without their own buffer specification. Values => for example: 1,2,3,4 hide, lib
DIGITS	C	0..8 (default: 8), drawing and buffer-specific, number of decimals in the drawing geometry
DIML1	C	Protrusion of the dimension auxiliary lines over the dimension line in GDUs
DIML2	C	Distance of the dimension auxiliary lines from the body edges in GDUs
DPL ?*	Flag	DPL= duplication OFF DPL=* duplication ON for modifications to the geometry
FILLET	C	Radius for 2D fillets (for example: when trimming and with rectangle)
GETSCALE	C	Scaling factor when accepting library symbols
GFLAG	C	ON: The visible grid (GRID) is unhidden. OFF: The visible grid (GRID) is hidden.
GRID	C	Point spacing in the continuous visible grid: => GRID=dx,dy
GROUP ?*	C	The number which limits the snapping to certain points; this number can also be a summation of the individual snapping types. In the case of "GROUP=", the snapping is not limited, i.e. the snapping is carried out to all points. 1 Straight lines and polygon definitions (internally: connection from **ma**/**da** to **da**) 2 Circles, ellipses and arcs (internally: **el** and **es**) 4 Starting and circle centre points (internally: **ma**) 8 Corner points (internally: **da**) 16 Corner points of a hatching (internally: **hp** and **ha**) 32 Hatched circles and hatched ellipses (internally: **he**) 64 End points of circle arc or ellipse arc (internally: **es**) 128 End points of hatched arcs (internally: **he**) 256 Centre points of straight lines (internally: connection from **ma**/**da** to **da**) 512 If this value is added to the group number, the snapping is performed preferably to corner points, end points of arcs and centre points if these lie within <TOL>. Otherwise, the snapping is performed as usual to lines and arcs. 1024 Intermediate points of Bezier curves (internally: **bz**) 2048 Bezier curves (internally: **bz**) 4096 Hatched Bezier curves (internally: **hb**) 8192 Reference points (internally: **rp**) 32768 Points of a 3D mesh net (internally: **mp**)

		65536 Edges of a 3D mesh net (internally: **me**) 131072 Arc apexes at 0°, 90°, 180° and 270° 262144 Intermediate points of Bezier surfaces 524288 Intermediate points of hatched Bezier curves (internally: **hb**) 8388608 Points of intersection 16777216 3D edges of ACIS objects 33554432 3D surfaces of ACIS objects 67108864 3D corner points of ACIS objects
HATCH1	C	Hatching definition for objects created with hatching HATCH1=type[-angle]/size/color type: Type of hatching (e.g. dirct) angle: Angle (can optionally be appended to *type* using "-") size: Size of hatching color: Colour of hatching
HFL ?*	F Flag	Representation of closed 2D geometries: HFL= (BORD), bordering by margin (without hatching) HFL=-h (HTCH), hatching without bordering by a margin HFL=-hb (BD&HT), hatching with bordering by a margin
HTCHPLOT	C	ON: Hatchings are displayed in the screen window OFF: All hatchings are hidden (not deleted !)
LINETYPE	C	Free line types for curves, lines and circles 1. type: LINETYPE=<pattern>,<length> Pattern: Bit pattern as a number Length: Segment length in GDUs for example, pattern: 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 = 2175= dotted line 2. type (beginning with "-"): LINETYPE=-<Length1>,<Gap1>[,<Length2>,<Gap2>,..] Length1: Length of the 1st line in GDUs Gap1: Length of the 1st gap in GDUs
LSPACE	C	Line spacing for justifying as a block as a percentage of the font height
MODBUF	C	contains the designation of the drawing buffer to which certain drawing functions without their own buffer specification refer. Values => for example: 1,2,3,4 hide, lib
ORA	F	<object>, default= `all`, when selecting snapping to <object>; If the variable is empty (set ora=), object snapping is generally not performed.
P2D	C	P2D=X,Y 2D coordinates (e.g. after taking the points)

REVISION	R	Number: Release of isyCAM 2.5(light)
ROUND	C	The dimensioning is performed with an accuracy of <ROUND> decimals, whereby trailing zeros are suppressed.
SFLAG	C	ON: Snapping to the points defined via GRID. OFF: The snapping is disabled. GRID is nevertheless preserved.
SNAP	C	Point spacing of the continuous grid: => SNAP=dx,dy
TEXTPOS	F	contains the current number of the line from the file last edited with "**text**". TEXTPOS=#:line #=buffer memory, line=line number
TOL ?*	F	Tolerance range for selection in thousands of the visible window (if not set, the value "20" is used)
TXTSPEC	C	Fonts and texts are created using the font TXTSPEC. TXTSPEC =type[,tts]/size[,rw]/color type: font type of isyCAM 2.5(light) tts: if necessary, Windows font: - tts=d (Windows full font) - tts=b (bold) - tts=i (italic) - tts=b,i (bold-italic) size: size (as a % of the standard screen section) rw: factor for the relative font width color: font colour *Note:* When using true-type fonts under Windows95, there are some restrictions: Fonts rotated by a free angle cannot be displayed as full fonts.
UNIT	C	Mode of counting of the coordinates (USER zero = 0.0 GDUs): e.g..: UNIT=10 => 1 unit = 10 GDUs (USER is adapted)
USER	C	Mode of counting of the coordinates in the GDU range (default: 0.0..639.511): e.g..: USER=-319.5,-511..319.5,511 (UNIT is adapted) => USER zero = 319.5,255.5 GDUs, UNIT=1,0.5
ZOOM	C	Drawing section displayed in the current window. e.g..: ZOOM=202.6,295.6..230.7,336.9 (reset with "set ZOOM=")

(The variables marked with ?* are only evaluated by functions which are called via buttons, menus or dialog boxes. When calling commands directly, the contents of the variable can be used as a parameter of appropriate options if necessary.)

Configuration Settings in the Windows Registry

Settings in the Control Panel:

Most of the settings mentioned below can be made via

START →
Settings→
Control Panel →
Schott Systeme Settings

or

START →
Programs →
Schott Systeme →
Settings

The appearance and the behaviour of
PICTURES by PC can be influenced by means of entries in the registry

All entries must be made beneath

HKEY_CURRENT_USER\Software\Schott Systeme GmbH\IsyCAM

or

HKEY_LOCAL_MACHINE\Software\Schott Systeme GmbH\IsyCAM

The entries made in HKEY_CURRENT_USER prevail.

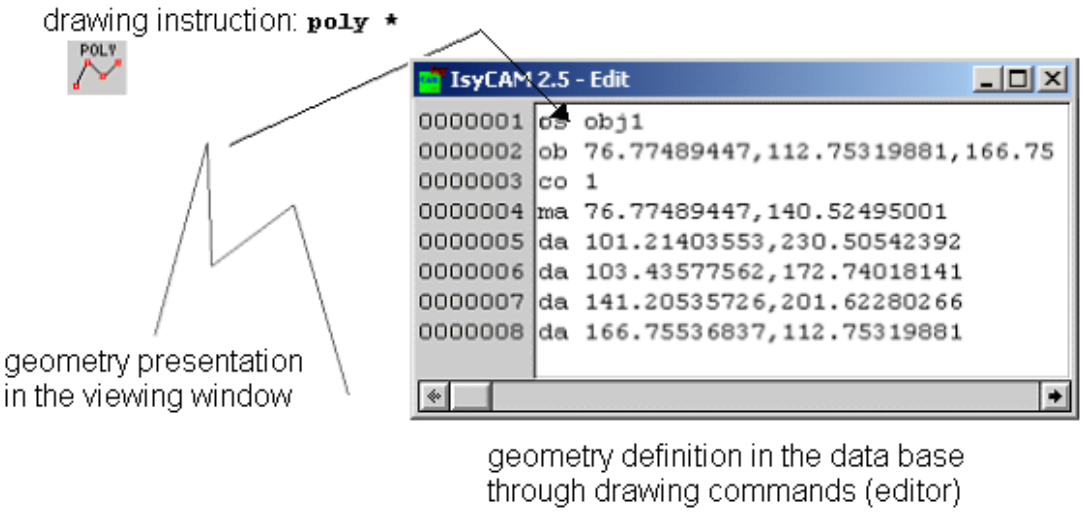
Key	Value	Default	Meaning
Configuration	CmmdPrompt	yes	Defines whether prompting messages are displayed in the command box when selecting points
Configuration/Colors	ViewBack	black	Background color for the view windows
Configuration/Colors	CrossHair	White	Colour for the cross-hair
Configuration/Colors	Grid	White	Colour for the visible grid (GRID)
Configuration	GridCross	1	Size of the grid points (1..7)
Configuration	GridMax	100	Maximum number of grid points (20..300)
Configuration	UndoSize	1,024	Size of information stored for "Undo" (in kB). (64..16,384)
Configuration	ViewMarkerSize	7	Size of the object snapping markers when selecting points (7..31)
Configuration	AcisSaveVersion	5.0	ACIS version for which the SAT/SAB files are generated without specifying "-v"
Configuration	PointMarkerSize	10	Size of markers for special points in CAM etc. (command mark - 5..30)

Configuration	SaveVersions	0	Number of old versions to be preserved with the command **save** (0..9)
Configuration	UserProfile	UserProfile	Initialisation procedure carried out after "profile.prc"
Configuration/GUI	FlatTools	Yes	The tool buttons are displayed as "flat" icons (not 3D)

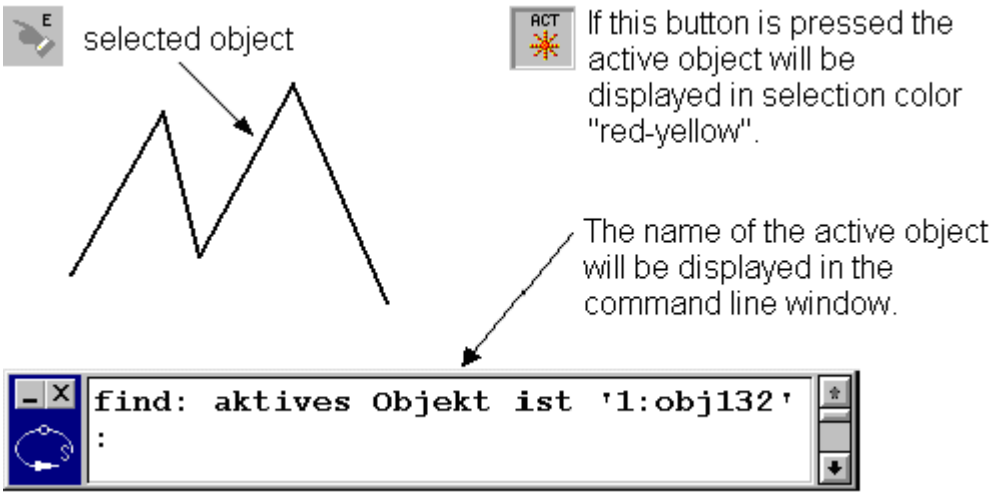
Object and Data Structure

What are drawing objects?

Drawing objects (briefly: objects) are geometry units which can be selected and whose definitions are generated in the database by means of drawing commands which are embedded between the drawing commands "os" (object start) and "oe" (object end). For example, each drawing command creates an object:



Unless otherwise stated, the object name is generated by the system itself when creating the geometry (obj1, obj2, obj3, ...), and is to be found in the database (editor) in a separate line as a parameter after the "os" drawing command. This name can generally be used for addressing the geometry defined in the object.



After selecting an object using the cross-hair, the object name is displayed in the command window, and the geometry is marked in the view window provided that marking of the active object was generally enabled. The variable "ACTOBJ" contains the object name of the "active" or "current" object (geometry created or selected last) after the designation of the appropriate drawing buffer (e.g. "1:obj1").

All buttons with a red-yellow, star-like marking always refer to this "active object".

"red-yellow, star-like" marking

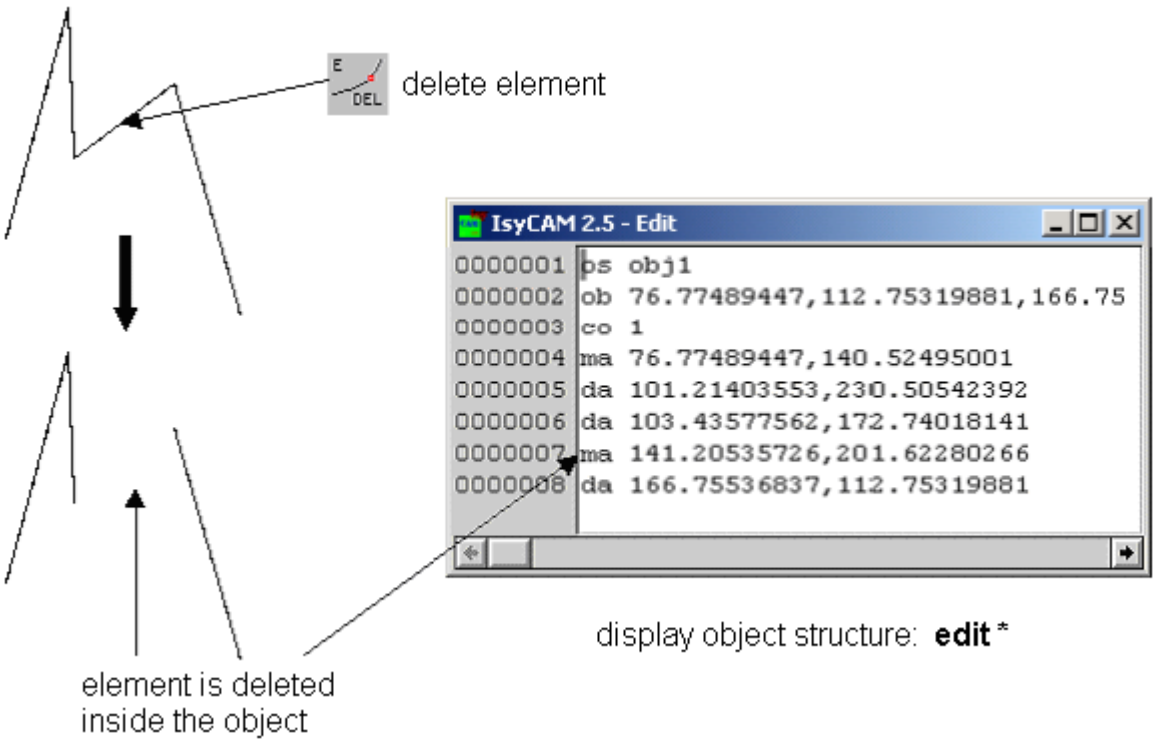
When entering the command, you can use the character: "*****" as a representative identifier for the "current object". For example, it is possible to use the command "**del** * " 

to delete the "current" object immediately. Or: With "**edit *** ", it is possible to jump to the "current" object in the textual database immediately.

What are elements?

An element is the smallest geometry unit within an object. Unlike objects, elements can be selected. Individual elements can only be handled using special manipulation operations. For example, the command 

("**trimx** –*md* ") can be used to delete an individual element. When doing so, geometry-determining drawing commands within an object are modified or deleted.



CAUTION:

A complete object should never be removed using the "Delete element" option! If the whole geometry of an object is deleted by deleting its elements, a formal object without graphical contents is kept. These residual objects occupy unnecessary memory space and could have negative effects on further drawing due to non-visible grid points. Such inadvertently created "data leftovers" can naturally be removed from the entire drawing (after "Select All"):

1.) 

2.) 
3.) 

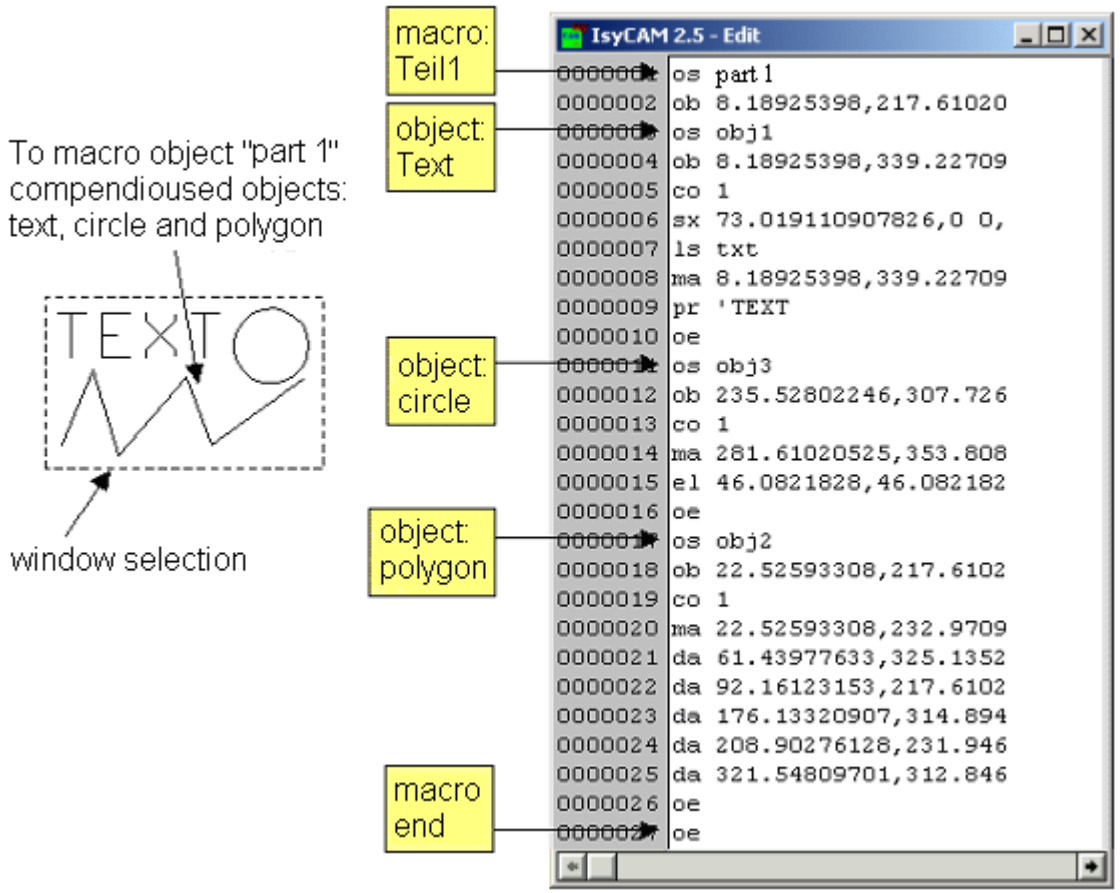
4.) 

5.) 

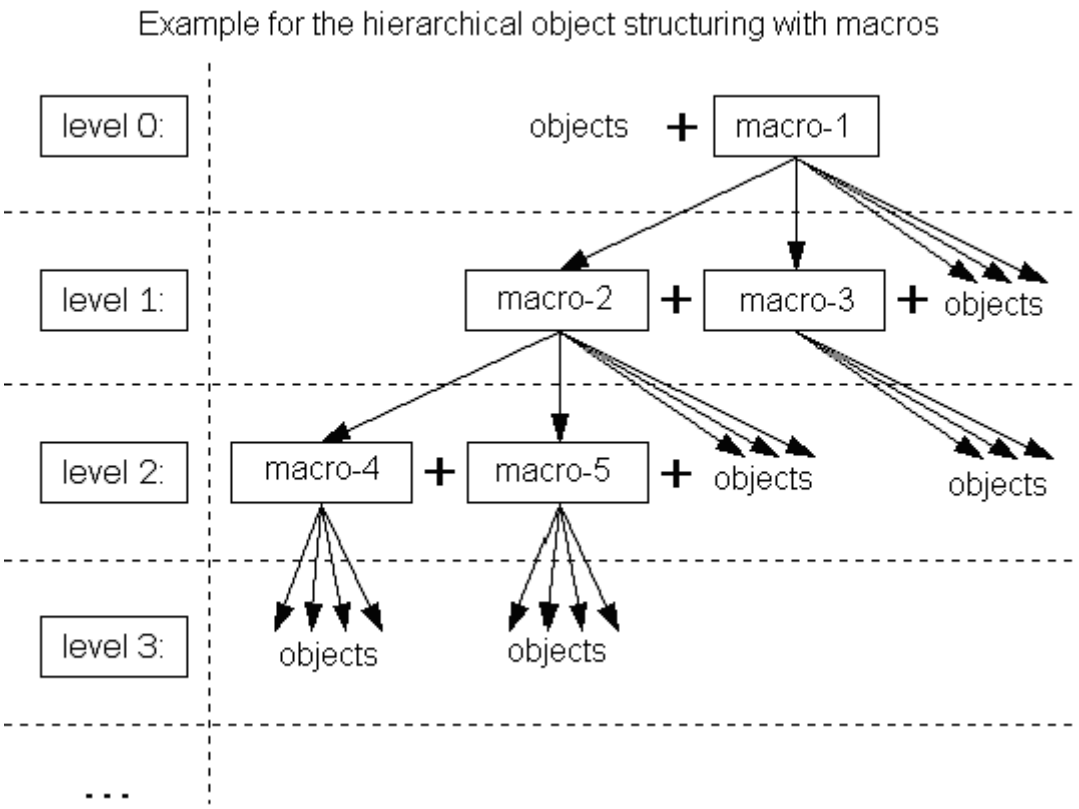
oder 

What are Macro Objects (or Object Groups)?

Object groups: Several drawing objects can be grouped into so-called macro objects (or envelope objects), i.e. into logical object groups. In this case, the macro object within the data structure encloses the summarised (grouped) drawing objects with a freely assigned name (os name ...oe).



It is possible to address either the entire macro object, i.e. the whole group, or each individual object separately. The simplest way of creating macro objects is by using the command: "**macro**", whereby the macro name can also be specified: e.g. "**macro** *part*". The objects to be summarised in the macro object "*part*" can be selected separately by clicking on them or by selecting from the window.



Macro objects, in turn, may also incorporate other macro objects (nesting). This makes it easier to structure drawings hierarchically (similar to a reverse tree structure).

Summary of the Most Important Object Structure operators

Button	Command operator	Effect
	macro [<*name*>]	A macro object with the specified name is created for which the appropriate objects must be selected.
	**name** * [<*new name*>]	Renaming of the current object: The current object contains the name to be entered directly (observe the notation).
	**box** *	The object dimensions of the current object are recalculated. (update in database (ob) and variable: BOX)
	**edit** *	Editing of the current object in the database: The cursor is located in the first line of the current object. The drawing commands and the geometry data can be verified and changed directly as necessary.
	**set** *macro* =<*name*>	Defining the macro name: The variable MACRO is assigned a name. All buttons in conjunction with the macro structuring will refer to this macro name!
	**set** *macro=$actobj[3,]*	The variable MACRO is assigned the name of the current object (without leading designation of the drawing buffer).
	**macro** *$macro*	Linking: A macro object with the name stored in the variable MACRO is created. The appropriate objects are selected either directly or from the window.
	**cat** * *$macro -i*	The current object is added to the current macro object. (The macro name is included in the variable MACRO).
	**rmmac** *$macro*	Unlinking: The envelope object of the current macro is removed, and the contents of the variable MACRO are deleted. The logical object group is dissolved.
	**exmac**	The selected objects are removed from the macro.
	**mkobj** *$macro*	The current macro is restructured: The internal structure (nesting) of the macro is lost. Objects that can be selected separately are created from elements between split points ("ma" drawing commands) within objects of the macro. All objects within the current macro, which follow after this operator, have the same *level*. The macro name and the macro envelope are kept.
	**obrename** *–n obj**	All objects *obj** in the drawing buffer are renumbered. Thus, errors caused by names occurring twice are avoided. (Object names occurring several times may arise with "**load** *–a*" or "**trans**", for example.)

		A drop-down menu with further structuring options is offered in conjunction with *Layers*.
	**pack** *$macro -o*	All subobjects of the current macro are packed in a single object which can be addressed as an object using the current macro name.
	**pack** *$macro -m*	All redundant "ma" drawing commands within the current macro (positionings, reference points).
	**pack** *$macro -ercst*	"Empty" objects, repeated colours and unnecessary drawing commands within the current macro are removed.
		A drop-down menu with further structuring operators is offered. (These are listed below.)
	**level** *	Structure analysis of the current object: displays the nesting depth and - where appropriate - the higher-level macro name.
	**plotopt**	The objects selected one after the other are arranged in the memory (drawing buffer) in the order as they have been selected.
	**box** *all*	The object dimensions of all objects in the drawing buffer are recalculated. (update in database (ob) and variable: BOX)
	**cover** * <*new name*>	An envelope object (macro) with the name you have entered is created around the current object. Thereafter, the command "**box** * " is required to calculate the object box of the new envelope object. (This envelope object first only contains the current object. Further objects can be linked to this newly created macro.)
	cover *all test*	An envelope object (macro) with the name "test" is created around the current object. To calculate the object box for this envelope object, use the operator: "**box** *all*".
	**trans** * <*bf:[obj]*>	The current object is **moved** (not copied!) in the memory to a position after another object *obj*. If, for example, only a different buffer designation is specified, the appropriate object will disappear from the current drawing buffer and is appended to the end of another drawing buffer in the rear, involving, however, the risk of assigning multiple object names, since an object may already exist with this name in the other drawing buffer.
	**rmma** *	The split points ("ma" commands following after "da" drawing commands with the same coordinates) are removed from the current object.

	**rmda** *	Points occurring twice in succession ("da" commands following after "da" drawing commands with the same coordinates) are removed from the current object.
	**mkobj** *	The current object is restructured: The internal structure (nesting) of the current object is lost. Objects that can be selected separately are created from elements between split points ("ma" drawing commands). All objects within the current object, which follow after this operator, have the same *level*. The current object, however, remains envelope object for the newly structured objects.
	**obrename**	To avoid multiple object designations, all objects in the entire drawing buffer can be renumbered. (Object names occurring several times may arise with "**load** –*a*" or "**trans**", for example.)
	**obreport** *[Buffer]*	A list of names is created for all objects in the first sheet cells of the *buffer*.
	**pack** * *-o*	All subobjects of the current object are packed in a single object which can be addressed as an object using the current object name.
	**pack** * *-m*	All redundant "ma" drawing commands within the current object (positionings, reference points) are removed.
	**pack** * *-c*	Repeated colours are removed from the current object. Only the first "co" drawing command is preserved.
	pack * *-x*	Irrelevant colour commands are removed from the current object. Only the first new "co" drawing command is preserved in each case.
	**pack** * *-erts*	Unnecessary residual files are removed from the current object.
	**set** *autocat = $actobj*	Begin of the automatic linking of newly created objects under the specified macro name contained in the variable **AUTOCAT**. In this case, the macro name is created from the name of the current object.
	**set** *autocat =*	Completion of the automatic linking of objects if the variable **AUTOCAT** is deleted.

Note:

Structuring of drawing objects into layers is naturally also possible, but does not provide the advantages of a hierarchical summary of objects in macro objects. The layers are managed using the basic routine LAYERS.BIX. If - contrary to our recommendation - you use layers and macro objects simultaneously, all objects of a macro object should be assigned to a layer; otherwise, you could easily lose track of things. It is also possible to use colours together with colour filters and colour snapping (COLPLOT and COLSNAP) for structuring drawing objects (similarly to layers).

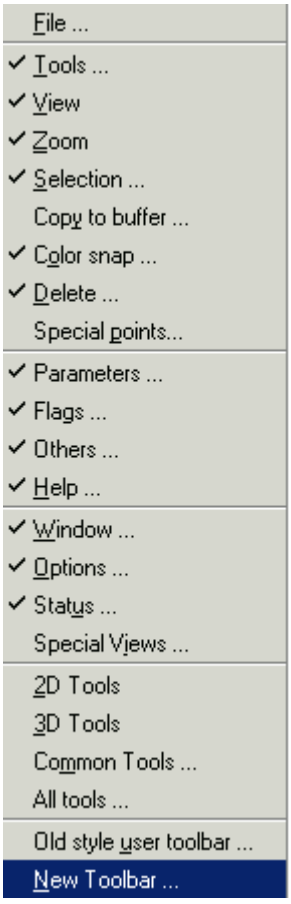
Creating Customised Toolbars

What are buttons and toolbars ?

User-defined buttons and toolbars:

isyCAM 2.5(light) is operated mainly via buttons. These buttons, which are selected by clicking on them with the mouse, are assigned commands or command sequences which are triggered when releasing the mouse button. Usually, several buttons are summarised in groups called toolbars. The most common toolbars in isyCAM 2.5(light) can be called via the textual menu "Tools". All standard tools which are not ticked can be selected using the bar, appear on the screen and can be docked to the screen margin as usual. If the names of the toolbars are known, they can naturally also be called via a command. For example, the command "stb psspec" (observe exact notation) creates the toolbar for engineering help with "special points".

Please note the two new menu options "User toolbar" and New toolbar" shown in the menu on the right. The "New toolbar" can be used to compile user-specific menus in a very straightforward manner and to save them for later use under the "User toolbar". In this case, we speak of "dynamic toolbars". In the following, we will show how efficient toolbars can be created from already existing buttons.



Defining your own toolbar:

If you select "New toolbar" from the menu above, a dialog box appears where you can enter the title for our user-specific toolbar.

We will enter, for example, "MyToolbar". This is, on the one hand, the header for our

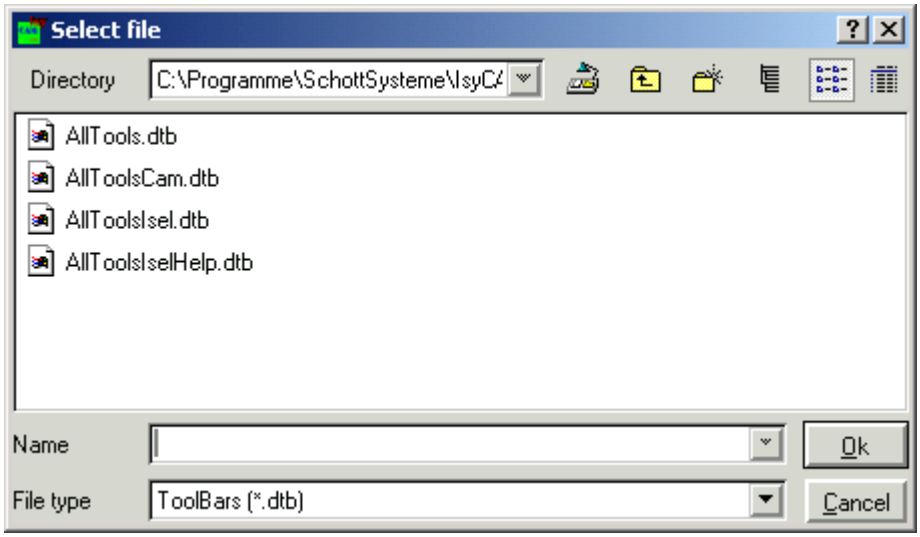
toolbar and, on the other hand, the file name with which we will save the toolbar if necessary. (Note: Since the title must be legible even in case of limited space conditions, it should be as short as possible.) After confirming the title with "Ok", a relatively simple "empty toolbar" appears, to which we now want to assign the desired button functions. To do so, let's first browse in the isy button environment and go to the toolbar that contains the desired function. This function should now be visible in the lower button area on the screen. In our case (see below), we have chosen some buttons from "Tools". Now we will paste the first desired button (for example "Undo") to our empty toolbar using "Drag & Drop". This is done in the following manner:

Position the mouse on the desired button, press first the <Strg> (or <Ctrl>) key and then on the first mouse button, hold down both and drag the desired button to the empty toolbar. A black line will appear in the empty toolbar in the place of pasting. If we now release the button/key, a duplicate of the button is generated in our customised toolbar. Similarly, we could continue add further buttons (e.g. "Parameters" and "Copy of current object to alternative buffer").

To ensure that the customised toolbar is maintained after the end of the program, it should be saved. To do so, point with the mouse on the toolbar, call the context menu using the right mouse button and there use the menu option "Save". Dynamic toolbars can naturally be used as normal and can be docked to the screen margin, for example.

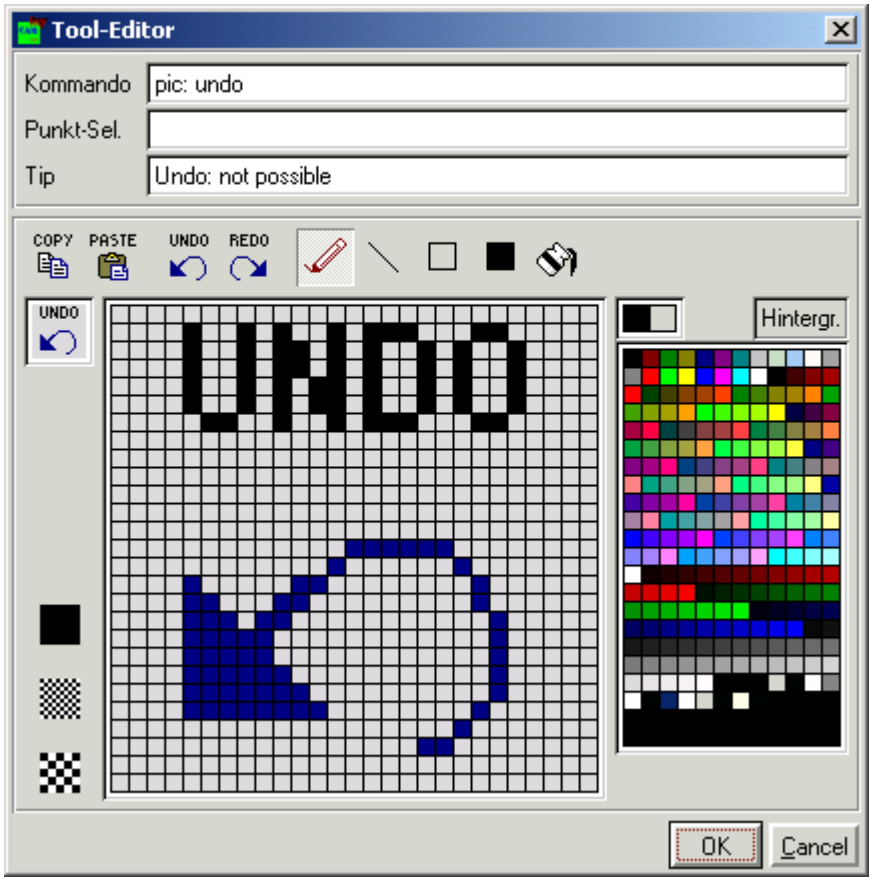
If you remove the toolbar from the screen using the (x) symbol, it can be reloaded via the "Tools" menu and the "User toolbar" (from the directory "....isyCAM\Resources\DE) by clicking on the name (Extend *.dtb) provided it was saved beforehand.

Saving toolbars:



Designing buttons:

A customised, dynamic toolbar saved beforehand is naturally preserved for the next session at the position at which isyCAM 2.5 (light) is quit. We have only used copies of existing buttons for our customised toolbar to date. An experienced isy user may naturally also modify buttons or even create new buttons. To this end, place the mouse pointer without your toolbar on the button you wish to be modified, activate the context menu by clicking on the right mouse button and select there "Edit". Subsequently, a bitmap tool appears with which you may design the graphics of the button and edit the assigned commands or user help functions.



You may naturally also try the graphical tools and modify the graphics. This is perfectly safe as you are always working on the copy and only fix the modified status with "Save". The graphical tools are self-explaining to a large degree and are similar to those of the dialog editor which is described further below in detail.

You may naturally also modify the message (tip). This is the text which is displayed if the mouse pointer dwells on the button for a short moment (without clicking on it).

If you acknowledge all your changes with "OK" and save the toolbar from the context menu, the dynamic toolbar can also be used in the next program session.

To call a toolbar as a Pop-up", the command for the higher-level button is

```
bix: PopUserToolBar "Name of the toolbar"
```

i.e for example

```
bix: PopUserToolBar "MyTools"
```

Index

2

2D tools 23

A

Absolute coordinates 59
Active object 34
All tools 23
Autostart 5

B

Backslash 12

C

Calling the program 6
Cartesian coordinate system 56
colours 49
COLPLOT 33
COLSNAP 33
Command box 9, 20
Command input 9
Command stack 12
Common tools 23
Continuous grid pattern 36
Control Panel 6

D

Defining a toolbar 87
Delete element 18
Delete the active object 18
Deleting the coordinate grid 41
Delimiter 11
Designing buttons 88
Discontinuous grid pattern 41

E

Element 17, 80
Enlargement function 32

F

File 21
Flags 22
fload file 49
fsave file 49
fsettings 49

G

Generating a window 26
Grabbing an object 34
Grid origin 37
GRID PATTERN menu 35
grid points 34

H

Header 20
Hierarchy levels 9
Hotkeys 45

I

Inclined grid patterns 39
Input via keyboard 9
Installing isyCAD/CAM 2.5 5

L

License code 5
Line width 53, 66
Line widths when printing 68

M

Manual 5
Menu technique 9
Multi-key function 12

N

Number formats 11
Numerical coordinate specifications 60

O

Object 17, 79
Object name 79
Online help 7
Optical grid 38

P

Parameters 22
Point selection 14
Polar coordinate system 58
Program icon 6

R

Relative coordinates 60
Remote 5, 6, 7
RemoteTools 5, 6
Rescuing menus 26

S

Saving toolbars 88
Scope of supply 5
Selecting a view 29
Selection 22
SET-UP 5
Shortkeys 45
Slash 12
sload file 49
SNAP.ENV 42
Snapping coordinates 35
Software 5
Special points 22
ssave file 49
Status display 23
System requirements 6

T

Tolerances 49
Tool menus 24

Tools 21, 23

U

UNDO 18

V

Variable window cursor 16
Variables 71

View 21

W

Window 23

Z

Zoom 21, 27
Zooming factor 27