LibreCAD for Real Dummies

a free option for Computer Aided Design

Johnny Heikell

2019

Based on LC version 2.2.0rc1

Page intentionally blank

About this tutorial

This tutorial was born out of my attempts to learn – sweating while learning – LibreCAD for domestic purposes. LibreCAD does not have a decent user's manual and the brief tutorials that exist are partly obsolete – geeks do not like to write manuals. The remaining information about LibreCAD is scattered around various web pages, user discussions, and on YouTube. I got a feeling that I would not learn LibreCAD unless I collected and organized basic information in a better way. I am also old-fashioned and prefer a tutorial as a hard copy to skim, laying on the table beside the laptop.

The text is written on LibreOffice Writer and I only talk about Windows and Linux because I have no experience with Apple products. No Mac, no idiotPhone, no nothing.

LibreOffice gave problems that may show up although I try to hide the mess. Some problems could be solved by exporting the document as PDF and using LibO Draw as PDF editor. Scribus <https://www.scribus.net/> would nevertheless have been a better tool for this job. In most cases the screenshots are edited with GIMP

Another IT problem came when I tried to share information on this tutorial on LibreCAD's Forum. I tried to register as Forum contributor, but after CAPTCHA finally accepted that I am not a robot there was no instructions on how to finalize registration. Security is good, too much security – no good.

According to a 2018 decision by the European Court of Justice, borrowing images from the web – even for non-commercial and educational use – is completely forbidden without consent by the creator/copyright holder. To the best of my understanding there are no restrictions on the few that I have borrowed (all from Wikipedia except a screenshot and one image at the very end from <http://www.freedwg.eu/>). Images created by me are in the public domain.

Why the title "LibreCAD for Real Dummies?" Elementary: I consider myself a LibreCAD dummy. Whoever can learn something from me must be a Real Dummy.

I share the tutorial with no guarantee for its correctness – it is after all a diary born while learning. Some of my LibreCAD drawing attempts are clumsy, some are better. I first used LC 2.1.3, then changed to 2.2.0rc1 and that can be a source of inconsistencies. Also, nobody else has checked the text so I'm sure there are both typos and other errors.

Feedback is welcome – perhaps I'll update this text – until the end of this year at librecad<*>heikell.fi, where <*> stands for the at sign. I read what drops in but may not have time to answer and will close the mailbox sooner if spam starts to accumulate.

Johnny Heikell
Espoo in January 2019

Page intentionally blank

Table of Contents

Page intentionally blank

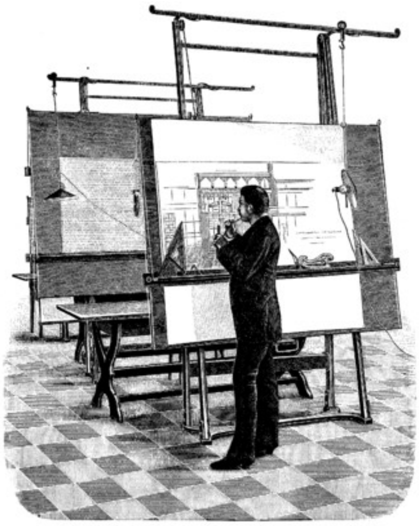
1 CAD basics

This chapter gives a short summary of basic concepts in CAD. You can jump over it if the definitions feel boring and return later for reference. Everything will be discussed later – or so I hope.

But let's start with a nugget of history.

History

Technical and architectural drawing up to the 1970s-90s timeframe was done in the traditional way, with pencil and ink on paper using a drafting table. Since the turn of the century even the smallest engineering offices have used CAD. Today even hobbyists like me have access to them.



Architect at his drawing table in 1893 [Public domain]

In 1957 **Patrick Hanratty** at GE developed PRONTO (Program for Numerical Tooling Operations), the first commercial **CNC** (Computer Numerical Control) programming system. Five years later, Sutherland presented his PhD thesis at MIT titled "Sketchpad, A Man-Machine Graphical Communication System." Among its features was the first GUI using a light pen to manipulate objects displayed on a CRT (Cathode Ray Tube). Hanratty has since been called "The Father of CAD."

CAD software suppliers that are major operators in 2018, emerged in the early 1980s. Examples are Dassault Systèms that today is best known for **DraftSight**, and AutoDesk with **AutoCAD**.

AutoCAD marked a huge milestone in the evolution of CAD. Its developers set out to deliver 80% of the functionality of the other CAD programs of the day, for 20% of their cost. From then on, increasingly advanced 2D (Two-Dimensional) drafting and engineering functionality became more affordable.

AutoCAD was revolutionary in its early days. Today it is considered a rather expensive tool with strong competitors, both lower-cost and free alternatives. Many regard AutoCAD the best CAD solution, but the smart question to ask is: "Which solution satisfies our needs?" Don't pay for something you don't need.

LibreCAD began its life as a fork to QCad and was first given the name CADuntu. The new name LibreCAD has led to suggestions it should be added to the LibreOffice package.

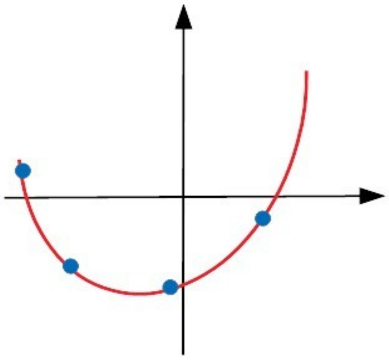
Basic CAD concepts

Entities

The term **entity** refers to a graphical object ("widget") in a CAD system. Typical entities, supported by most CAD systems, are:

- points
- lines
- rectangles and
- circular and elliptical arcs

More complex entities include polylines, texts, dimensions, hatches (closely spaced parallel lines) and splines (curves that connect two or more specific points).



Spline [JH]

Layers

A basic concept in computer aided drafting is the use of **layers** to organize a drawing. Every entity in a drawing is on exactly one layer and a layer can contain numerous entities.

Typically entities with a common "function" or common **attributes** are put on the same layer. For instance, you might want to put all axes in a drawing on a layer named **axes**.

Layers can have their own attributes (color, line width, line style etc...). Each entity can have its own attributes or have its attributes defined by the layer it is placed on. In the latter case you can change e.g. the color of all the entities on the layer by setting the color (red for instance) for the layer in case.

Blocks

A block is **a group of entities**, similar to what you create with the **Group>Ungroup** functions in LibO Impress and PowerPoint. Blocks can be inserted into the same drawing more than once with different attributes, at different locations, and with different scale and rotation angles. Such a block is usually called an **insert**.

Inserts have attributes just like entities and layers. An entity that is part of an insert can have its own attributes or share the attributes of the insert.

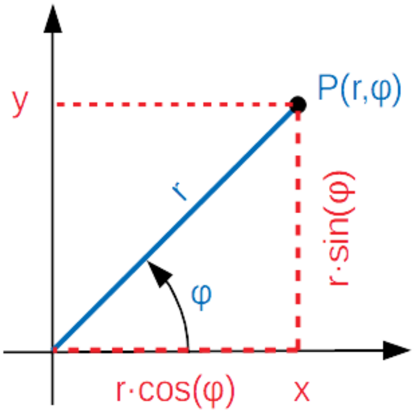
Once created, inserts are still linked to the block they represent. The power of inserts is that you can modify the block once and all inserts will be updated accordingly (similar to using styles in word processing).

Coordinate Systems

In order to get the best out of LibreCAD it is wise to have a good understanding of the coordinate system and how coordinates work.

There are basically two types of coordinates: **Cartesian** and **polar**. The cartesian coordinate system is generally the standard system used in CAD programs. A specific point in a drawing is located by exact distances from the x and y axes – for example, a point in a drawing could be 60,45 ([x,y] note the comma that separates the two numbers).

The **polar coordinate system** uses one distance and one angle to define a point in a drawing. For example, a point in a drawing could be 50<45, meaning 50 units long (r) and at an angle of 45 degrees (φ, note the < sign used for the angle).



Cartesian (x,y) and Polar (r,φ) coordinate systems [JH]

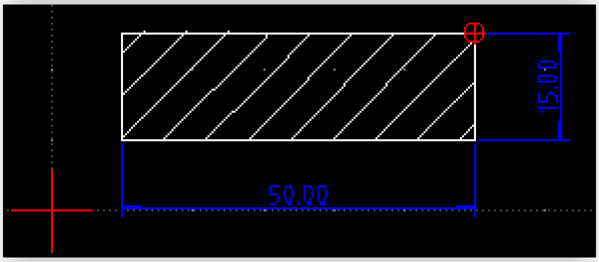
With **absolute coordinates**, the coordinate points are entered in direct relation to the origin 0,0. To do this in LibreCAD you just enter the exact point in whichever units you are using, e.g. 60,45.

In **relative coordinates**, the coordinate points are entered in relation to the previous point entered (not the origin). For example, assume that your first point is 20,45. You then enter the next point relative to this using the @ symbol. Thus @50,50 will put the second point 50 units horizontally along the X axis and 50 units vertically along the Y axis, giving the location 70,95.

Dimensions

Sizes of objects within a drawing are conveyed with the use of dimensions. Dimension distances may be shown with either of two standardized forms, **linear** and **ordinate** dimensions.

Linear dimensions – the usual case – use two parallel lines, called **extension lines** and spaced at the required distance between two given points. A line perpendicular to these extension lines is called a **dimension line**, with arrows at its endpoints. The numerical indication of the distance is placed at the midpoint of the dimension line, adjacent to it or in a gap provided for it.



Object with linear dimensions in blue

Ordinate dimensions use one horizontal and one vertical extension line to establish an origin for the entire view. The origin is identified with 0,0 placed at the ends of these extension lines.

Distances along the x and y axes to other points on a drawing are indicated using additional extension lines with numerical information placed appropriately.

Scales

Drawings are made to scale – it is not often a drawing has the same size as the object it represents (scale 1:1). Typically the drawing is smaller and engineering drawings have scales like 1:2, 1:5, etc. In architecture the scale 1:20 is common for small houses. A road map can have a scale of 1:100.000, meaning 1 cm on the map equals to 1 km in nature. The other extreme is nano-size objects that can have scales like 10.000:1 (also expressed as 1:0.0001).

.dxf and .dwg file formats

DXF and DWG are the major file formats in CAD. Both are **vector graphic file formats**. Vector images maintain the same image quality at practically any scale, making them ideal for design purposes. It is also easy to edit the individual elements that form a vector image, as well as to add and remove elements.

DXF (acronym for **Data eXchange Format**) was developed as a tool to share designs across different CAD and vector based programs. It stores 2D vector images and can be used by almost all CAD software, as well as **CNC** (Computer Numerical Control) and **GIS** (Geographic Information System) software. Every element of the drawing is spelled out in plain text or ASCII format that contains the full range of alphanumeric characters. Due to their file specifications, complex DXF drawings become large – from several to hundreds of MB (megabyte). When transferring such a document, it has either to be split or compressed.

DWG (for **Drawing**) is a proprietary format by AutoCAD-owner Autodesk.Inc. It can handle both 2D and 3D information. The data is binary, i.e., it is a series of 0’s and 1’s. A DWG document is typically 25% smaller that a DXF document.

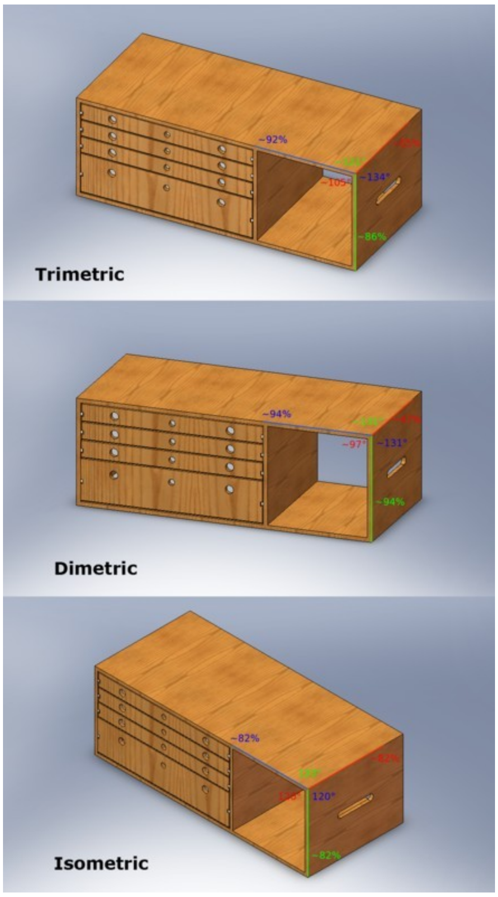
As usual in the IT business, the proprietary DWG format has been reverse-engineered and is released as **OpenDWG** by the **Open Design Alliance** <https://www.opendesign.com/> – not without legal fights with Autodesk, Inc. OpenDWG is released under GPLv2, which Linus “Linux” Torvalds considers the only decent FOSS license (Free and Open Source Software) – “I hate GPLv3.”

The DWG format is still experimental in LibreCAD.

Axonometric projection

Axonometric projection is a collective name for projection methods that aim to create the illusion of a 3D object on a two-dimensional surface.

Axonometric projection has been called technical projection because it is used in technical drawing, including CAD, and more lately in video and **computer gaming**. It is different from **linear projection** used by artists in that it does not have **vanishing points**. Axonometric projection is closer to the parallel projection used in traditional Chinese paintings.



The three types of axonometric projection [Wikipedia, GNU Free Documentation License, Version 1.2]

There are three types of axonometric projection: **isometric** projection, **dimetric** projection, and **trimetric** projection. The image below presents a comparison of the three. The differences may feel subtle but there are basic differences, particularly in the way the three axes of space appear **foreshortened**.

In isometric projection a unit along an axis is equally long for all of the three axes (iso = equal, uniform). That is, a centimeter is a centimeter along x, y, and z axes (but observing trigonometric equations); hence the projection is named "isometric." Dimetric projection, “di” = 2, uses different metrics for axes.

Isometric projection offers advantages in engineering and architecture through the ease of drawing due to fixed 120º (or 30º, if you like) angles, as well as when measures must be taken directly from an image. However, isometric drawings are only visual guides, they should not be used as exact drawings. In addition they result in **perceived distortions** because they are contrary to how human vision works.

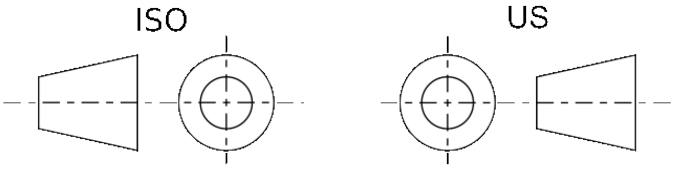
A variant of isometric projection is “military projection,” which uses 90º (45º) angles.

The isometric angle gives problems in computer games and for that reason an angle of approximately 27º is used. To those interested in the subject I can suggest the article at <https://www.compuphase.com/axometr.htm>, which discusses axonometric projection from the computer gaming viewpoint.

Multiview projection

A **multiview projection** is a type of orthographic projection that shows the object as it looks from the front, right, left, top, bottom, or back (e.g.

the **primary views**). There are two main standards in use, the **first-angle** standard used primarily in Europe (ISO standard) and the **third-angle** standard used in the US and Canada:



Symbols used to define whether a projection is either first-angle (left) or third-angle (right). [Wikipedia Commons]

Linear projection

Human vision is best adapted to processing 3D objects in 2D linear projection. LibreCAD is not however intended for linear projection, but it has a basic capability as will be demonstrated with a small test is Chapter 4.

2 Introduction to LibreCAD

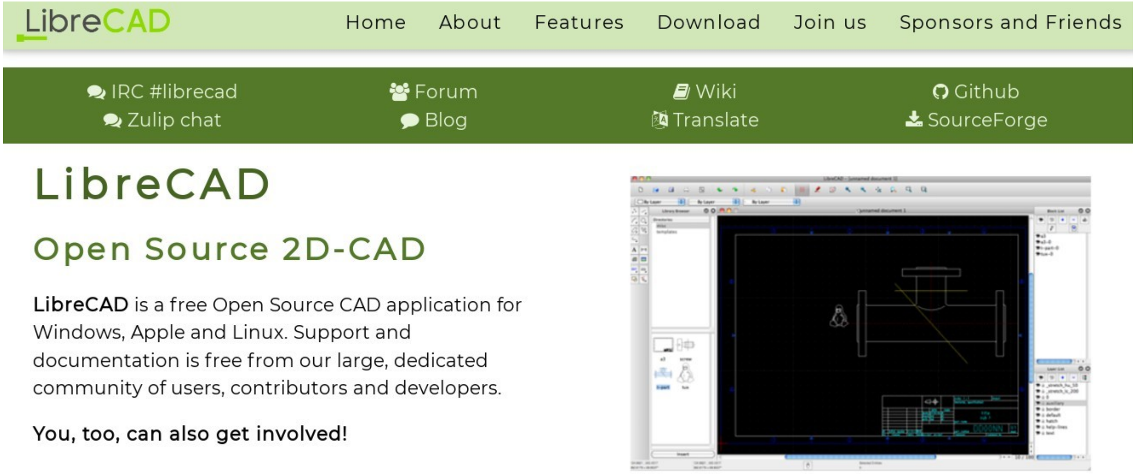
FAQ

What is LibreCAD?

LibreCAD is a free, open source 2D CAD software for Windows, Apple, Linux. It allows isometric **3D** drawings to be made. True 3D systems allow users to rotate and view a drawn object from any angle, even to “walk” into the object – a nice feature in architectural drawing.

Most of the interface and handle concepts are analogous to AutoCAD, making it easier for users to switch to and from this type of commercial CAD applications. LibreCAD uses the AutoCAD’s .**dxf** file format internally to import and save files. It allows export to many other file formats like png, jpeg, and pdf.

LibreCAD is released under the GNU General Public License (**GPLv2**). It can be downloaded e.g. at <http://librecad.org/>:



LibreCAD

Open Source 2D-CAD

LibreCAD is a free Open Source CAD application for Windows, Apple and Linux. Support and documentation is free from our large, dedicated community of users, contributors and developers.

You, too, can also get involved!

LibreCAD home page at <https://librecad.org/>

Who are the competitors?

AutoCAD, as hinted earlier, is the juggernaut in the drafting industry. One could say it is the Rolls Royce of CAD with a Rolls Royce price tag. DraftSight is perhaps the #2 in this field; a limited version is free for private use but for corporate users the price depends on the size of the company.

A list of 2D and 3D CAD alternatives to LibreCAD can be found at: <https://alternativeto.net/software/caduntu/>. However, I have not found any direct competitor to LibreCAD. The main attributes I have been looking for are: 2D, free and open source, no strings attached (not a limited version, no trial time, no registration needed, no piggy-back software installed, etc.), fairly advanced solution, and the question of file formats (particularly **.dxf** and formats suitable for printing – with PDF being important).

The **.dwg** format (for drawing) is important to share drawings among CAD programs. Presently this is only experimental in LibreCAD, but it is reportedly compliant with documents saved in AutoCAD 2007 and older. You can find a small DWG test that I did at the very end of this tutorial.

A free software to look out for is **LibreOffice Draw** (www.libreoffice.org). Today it lacks important features required in technical drawing (e.g. the DXF file format), but with extensions it could become a lightweight CAD alternative – time will tell. Draw already has 3D capabilities beyond those of LibreCAD.

Why is LibreCAD used?

No matter that we live in a three-dimensional (3D) world, the importance of LibreCAD (and similar CAD tools) is that 2D drawings are – and will remain for the foreseeable future – the basic visual planning tool in engineering, architecture, urban planning, etc. CAD is the computerized alternative to producing drawings “by hand” on a drawing board, as told earlier (Chapter 1, section on History).

LibreCAD has many useful features that can help you add more details to your drawings. It is a good, light-weight 2D CAD software with – as said above – no direct competitor. Those who need a more advanced CAD tool have to pay up, but the “mother” of LibreCAD, QCad, is not all that expensive.

How can you start using LibreCAD?

These notes will teach you enough to get going and discuss some general CAD concepts. This tutorial was first based on LibreCAD version 2.1.3 that was released in September 2016. Later I changed to 2.2.0rc1 and did some changes to the notes (don’t know if I did enough). As new versions emerge the notes will become increasingly obsolete.

Additional information sources are listed below under “Where can I find help?”

Who is/are behind LibreCAD?

Web sites typically have a tab called “About” that presents the organization and/or people behind the undertaking. Not so with LibreCAD, the About link only tells a little about the technical background to LibreCAD.

An article on the web site **Little Graphics World** tells that LibreCAD was forked from the Community Edition of **QCAD** in 2010 by Dutchman **Ries van Twisk** <http://libregraphicsworld.org/blog/entry/librecad-vs-

qcad>, and we can still see his name as administrator of the librecad.org web site.

LibreCAD is essentially a one-man show with low transparency and the voluntary developer team has remained small (only four persons contributed with more than 10 changes to version 2.0.8, a single person accounted for some 70% of all changes). This can be a long-term risk factor for LibreCAD.

Are there any problems?

LibreCAD, like all software – and particularly freeware – have bugs. I have met buggy functions, seen it crash and lock-up, had hatches disappear for many reasons, seen it refuse to print, etc. Plus that Selection becomes active all the time and requires involuntarily selected objects to be deselected. Save frequently to avoid losing data when a bug strikes!

LibreCAD does not maintain its own bugzilla, but you can find reported bugs at **Github** <https://github.com/LibreCAD/LibreCAD/issues> and **sourceforge.net**: <https://sourceforge.net/p/librecad/bugs/>. It is worth checking there in case you run into a suspected bug – the bug may already be reported.

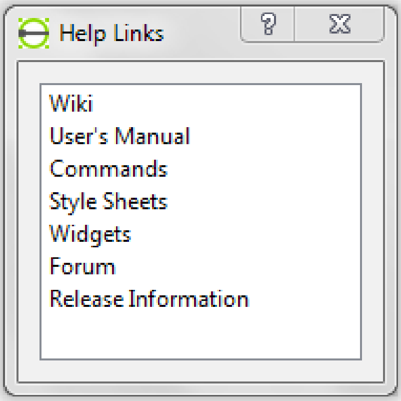
A potential problem is that LibreCAD is essentially a one-man show, as said above. If the King dies or tires, what will be the fate of his brainchild – will anybody to continue the work? In my opinion it would be important to find a governing body for LibreCAD, similar to The Document Foundation that manages LibreOffice.

With the experience I have so far, do I recommend LibreCAD?
For hobbyists: yes. For students: yes, if your professor accepts e.g. wrong line widths in printouts. For professional users: only as a non-critical solution. One reason for the last is that LibreCAD is still somewhat limited and buggy, another is the nagging question: will it survive? A third reason is lack of security features against hackers.

Where can you find help?

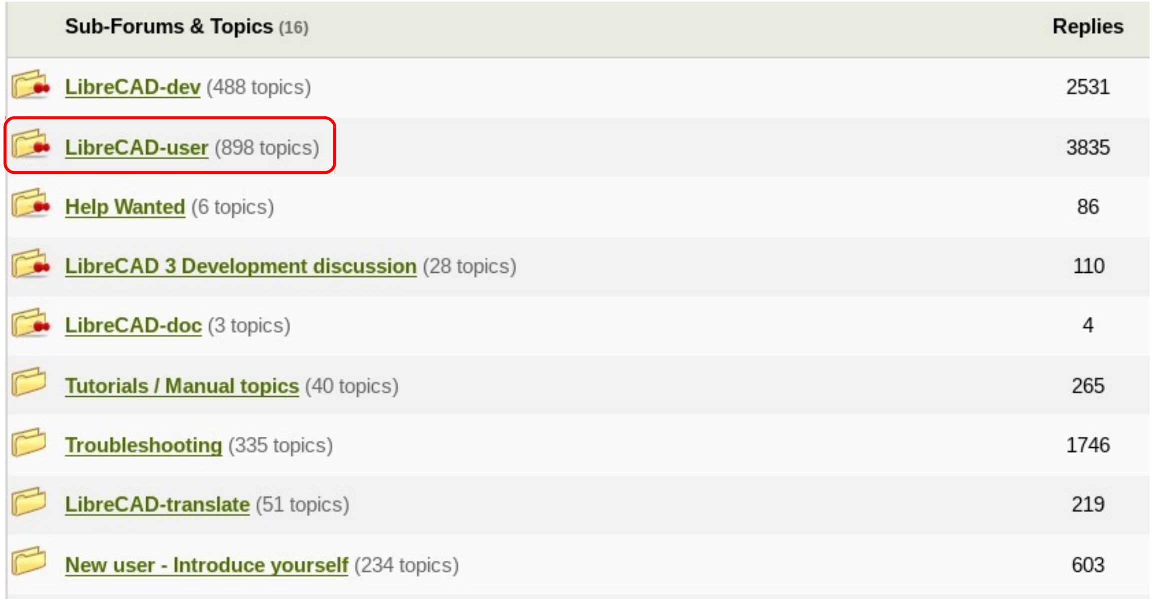
User guides on LibreCAD comes in bits and pieces, scattered around the web.

Under **Help** on the Menu Bar you find the link **Online**. Click on it (you need to be hooked up on the Internet) and the window to the right opens. The links lead to various info pages.



LibreCAD's dashboard has been changed recently and most tutorials, written and videos alike, are a bit outdated. I had to change some images in this reader when I moved from version 2.1.3 to 2.2.0rc1.

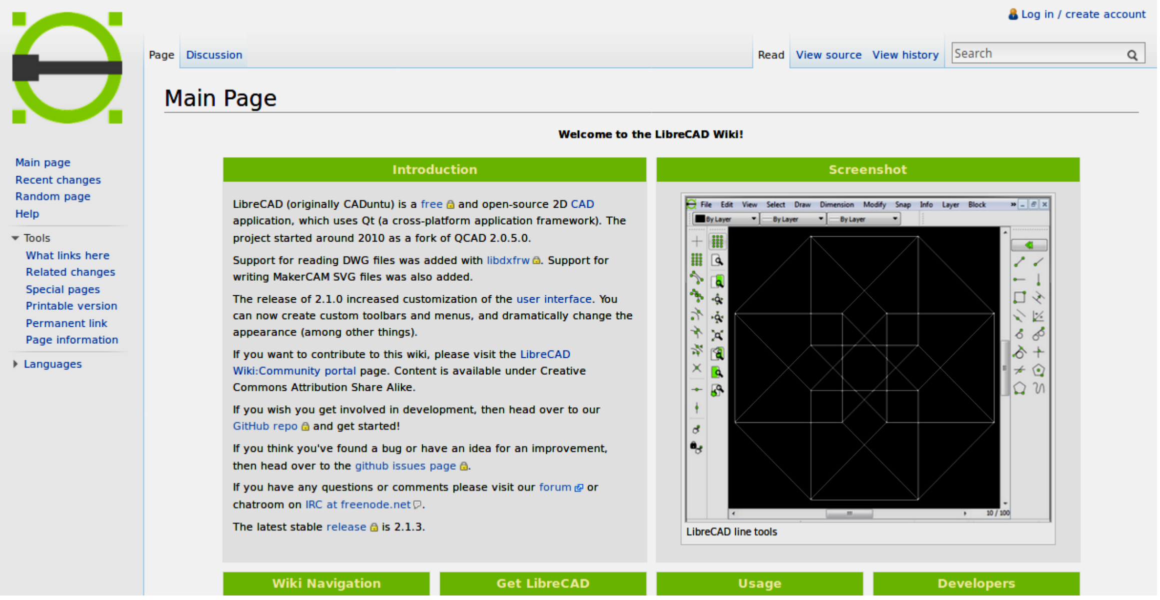
Take however a look at the **Forum** link on LibreCAD's home page. I can especially recommend the **LibreCAD user** discussions. It is very informative to read questions and answers by other LibreCAD users:



Sub-Forums & Topics (16)	Replies
LibreCAD-dev (488 topics)	2531
LibreCAD-user (898 topics)	3835
Help Wanted (6 topics)	86
LibreCAD 3 Development discussion (28 topics)	110
LibreCAD-doc (3 topics)	4
Tutorials / Manual topics (40 topics)	265
Troubleshooting (335 topics)	1746
LibreCAD-translate (51 topics)	219
New user - Introduce yourself (234 topics)	603

LibreCAD forum at <http://forum.librecad.org/>

LibreCAD Wiki, at <https://wiki.librecad.org/index.php?title=Main_Page> and shown below, has some written information. Most information is HTML. If you work off-line – hopefully you do for your own security – you will have to copy-paste whatever information you



Main Page

Welcome to the LibreCAD Wiki!

Introduction

LibreCAD (originally CADuntu) is a free and open-source 2D CAD application, which uses Qt (a cross-platform application framework). The project started around 2010 as a fork of QCAD 2.0.5.0.

Support for reading DWG files was added with libdxfrw. Support for writing MakerCAM SVG files was also added.

The release of 2.1.0 increased customization of the user interface. You can now create custom toolbars and menus, and dramatically change the appearance (among other things).

If you want to contribute to this wiki, please visit the LibreCAD Wiki:Community portal page. Content is available under Creative Commons Attribution Share Alike.

If you wish you get involved in development, then head over to our GitHub repo and get started!

If you think you've found a bug or have an idea for an improvement, then head over to the github issues page.

If you have any questions or comments please visit our forum or chatroom on IRC at freenode.net.

The latest stable release is 2.1.3.

Screenshot

LibreCAD line tools

LibreCAD Wiki at <https://wiki.librecad.org/index.php?title=Main_Page>

may need, e.g. to LibreOffice Writer documents, and save them on your computer. An exception is the "LibreCAD 2.1.3 Manual" at <https://wiki.librecad.org/index.php?title=LibreCAD_Manuals_Summary#LibreCAD_2.1>, which is a

77-page PDF file. It is however a formal, reference-type manual and hard reading for beginners.

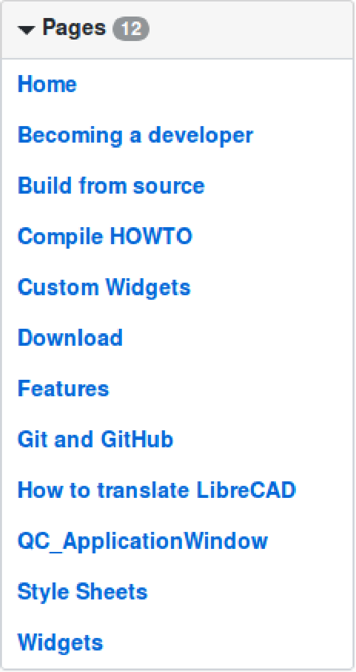
As for videos, try for a start on YouTube the half-hour “Basic Beginners Tutorial” at <https://www.youtube.com/watch?v=74Cb06CLQbo> and the 10-part series of “howtopam” videos (slow-going, poor sound quality, bypasses some design steps but cover lots of details). There are useful videos in other languages as well, particularly in German (e.g. the “Librecad Tutorial deutsch” plus LENatu videos) and Spanish (“Grifo con LibreCAD” covers interesting drawing techniques).

The <http://create-and-make.com/category/design/cad/> web page by **Gary Fox** contains a wealth of material on LibreCAD, including his videos that can also be found on YouTube. The web site has not been updated since 2015 however.

An unusual application of LibreCAD – creating knitting patterns – is presented by **Harry Guetter** in four PDF files available at <https://app.box.com/s/i0cv6gvgbwlspqnd3pux/1/1128879663>. Although a strange subject for engineers, the tutorials are quite informative.

A set of short but more advanced discussions can be found at Github <https://github.com/LibreCAD/LibreCAD/wiki>. The information – 12 chapters in all as seen below – is mainly aimed at developers, but is worth a look by beginners.

A list of changes that were made to LibreCAD with the introduction of version 2.1.0 can also be found at Github: <https://github.com/LibreCAD/ChangeLogs/blob/master/developers/r-a-v-a-s/2-1/2-1-0/2-1-0_introduction.md>. Take a look at it in case you have an old tutorial that gives instructions that do not work now.



Pages 12

Home

Becoming a developer

Build from source

Compile HOWTO

Custom Widgets

Download

Features

Git and GitHub

How to translate LibreCAD

QC_ApplicationWindow

Style Sheets

Widgets

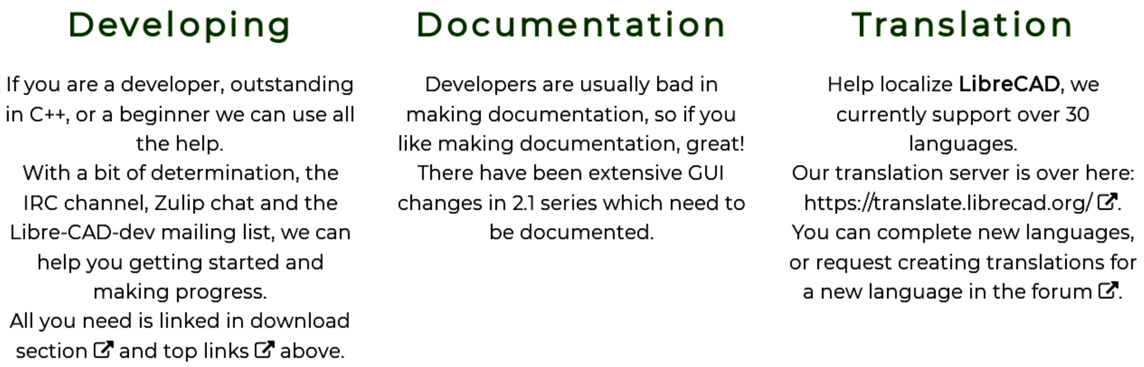
Discussion pages on Github

What can you do to help?

Scroll down on LibreCAD’s Main Page (image above) until you find the headline “Things We Need.” Under it you find a long list of work that needs to be done. You do not have to be a nerd to be of help.

My feeling is that users could be more active in reporting bugs, but then LibreCAD should have a clearly assigned address for bug reports and instructions on how to file them (which data is needed, how to attach examples, etc.). And CAPTCHA should not block those wanting to help.

Of course, LibreCAD does not refuse a tip either.



Developing

If you are a developer, outstanding in C++, or a beginner we can use all the help.
With a bit of determination, the IRC channel, Zulip chat and the Libre-CAD-dev mailing list, we can help you getting started and making progress.
All you need is linked in download section ↗ and top links ↗ above.

Documentation

Developers are usually bad in making documentation, so if you like making documentation, great!
There have been extensive GUI changes in 2.1 series which need to be documented.

Translation

Help localize **LibreCAD**, we currently support over 30 languages.
Our translation server is over here: https://translate.librecad.org/ ↗.
You can complete new languages, or request creating translations for a new language in the forum ↗.

Quest for contributors at <https://librecad.org/#joinus>

Getting familiar with the User Interface

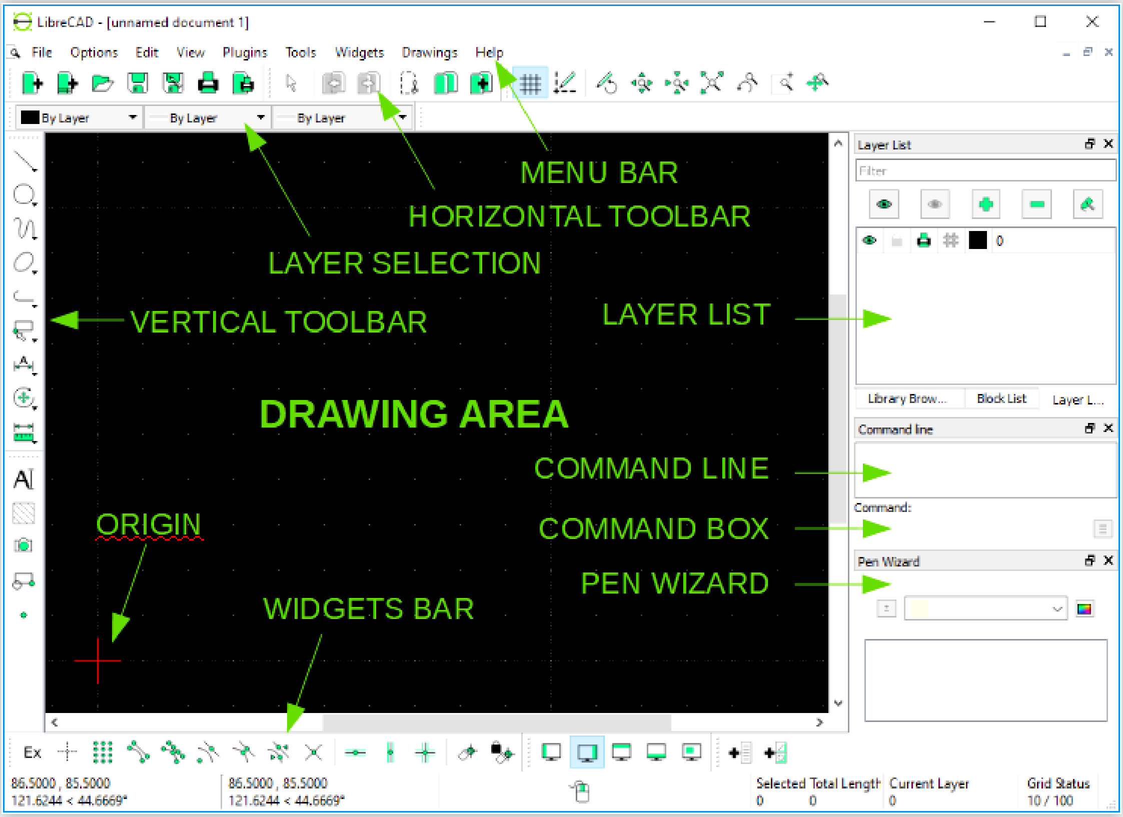
Hopefully you could install LibreCAD without problems. The process is usually plain sailing with Windows, with Linux one never knows how the adventure ends (know it, been there). Linux repositories mostly contain old versions of application software and one has to find other ways to get a fresh release. (As an ordinary computer user I hate the Terminal in Linux. It is a computing dinosaur that should have been declared “for geeks only” in the previous millennium.)

LibreCAD has an interactive **graphical user interface** (GUI), also named **dashboard**. It takes a while to get used to it because quite many of the tools are unfamiliar to those who have not used CAD software before.

Take a look at the image on the next page, it shows LibreCAD’s GUI (v. 2.2.0rc1) in all its beauty: The **Drawing Area** (aka Work Area) and the major toolsets that we work with. There are more, some windows are hidden behind pushbuttons and they in turn can have plenty of options to select between. The biggest difference that I have seen between versions 2.1.3 and 2.2.0 is the Pen Wizard at the bottom right-hand corner. It is possible that the Pen Wizard existed in the background in 2.1.3, but I never noticed, and when I downloaded and installed 2.2.0rc1 in early 2019 it emerged at the top.

LibreCAD GUI toolsets

The GUI is rather peculiar, only the Menu Bar reminds of more familiar software. The Toolbar(s) has (have) only some familiar icons – and many in the difficult green color (is there an Irish and/or Islamic influence behind the design?). There is an informative description of the GUI at <http://forum.librecad.org/GUI-update-td5712246.html#a5712603>. It is from 2015 but still of some use.



Drawing Area

The pitch-black Drawing Area is where we add components ("widgets") to our drawings (in addition to doing it with the Command Line, as we will see later). You can change the black background by clicking (on the Menu Bar) **Options>Application Preferences>Appearance**. On the right-hand side of the pop-up Application Preferences GUI you find Background under the Graphic Colors list. It shows "black." Click on the drop-down menu and select #FFFFFF, then click **OK** and the background shifts to white, or type in #DDDDDD for a light grey background. To return to black you either enter "black" or select the $000000 option.
I don't like black and will use #DCDCDC from now on.

If you look carefully in the lower left-hand corner, you can see the red cross that marks the origin.

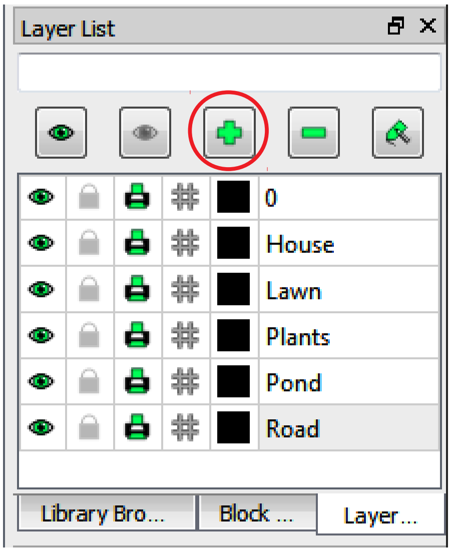
What about the size of the Drawing Area? When you draw something by hand you use a certain paper format – A4, letter, A3, etc. Not so in CAD, the area on and beyond the screen is infinite. We can however define a format for the drawing, and display it on the Drawing Area, but let's return to that later.

Layer List

The Layer List on the right-hand side of the GUI is important. We typically begin a new CAD task by specifying layers for the design (after we have defined the paper size). Layers allow us to "insulate" groups of objects on the drawing from each other (aka **assembly drawing**). We can change attributes of widgets in one layer without having to worry if there will be an influence on widgets in another layer. We can also toggle layers with the eye icon, i.e., make them either visible or hidden.

Layers act like floors when constructing an apartment house: The electrician works on the third floor, the plumber on the fourth, the carpenter on the fifth, and so on, without interfering with each other.

Assume that we are doing the garden design for a house. We could then make layers called House, Lawn, Plants, Pond, and Road. To add a layer, click on the + icon (circled), then type in the name of the layer in the window that pops up. The pop-up window also allows you to specify attributes (line **color**, **width**, and **type**) to each layer. Click [**OK**] and the new layer is echoed to the Layer area, as shown in the figure. Observe however that I have not selected colors or any other attributes for the lines. We'll cover that in the examples.



Layer List

0

House

Lawn

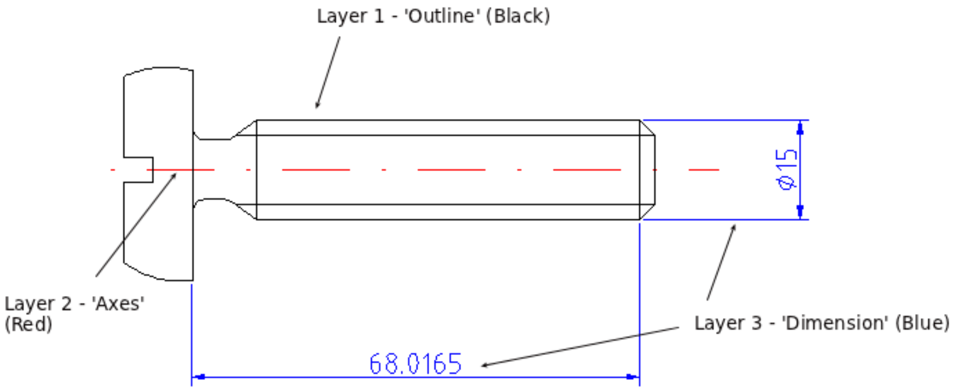
Plants

Pond

Road

A list of layers has been created. Topmost is the default layer "0" into which imported blocks will be directed

Another idea in defining layers can be seen in the drawing of a screw that I found on my laptop at <C:\Program Files (x86)\LibreCAD\resources\library\misc>. The simple screw is divided into three layers, each using separate colors:



Drawing of a screw in three layers colored black, red, and blue.

The garden and screw examples remind that we should plan ahead before defining layers – we can change them as the work progresses but everything related to a layer is erased if the layer has to be deleted. If you need to manufacture a screw the level of detail above is motivated, but not necessarily so if it is a standard screw used as just another part in a drawing of a machinery. In the garden design we most likely need additional layers. For instance, plants not only need to be shown, they should also be specified.

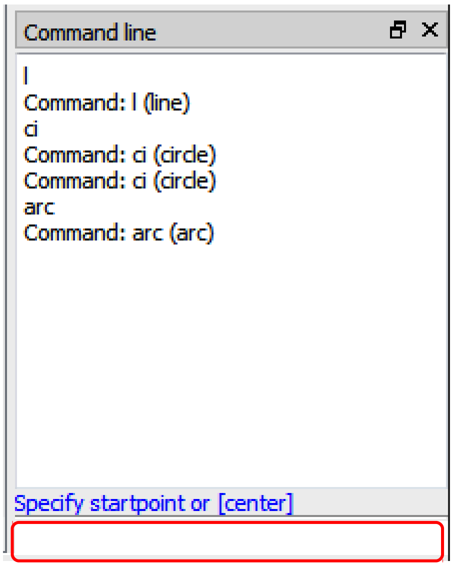
A comment about the screw drawing: We should follow local standards for technical drawing – EN (ISO) standards and their national derivatives in Europe, ANSI standards in the US, GOST in Russia and some Eastern European and Asian countries, etc. That said, I am not going to adhere strictly to ISO standards because this is a tutorial about LibreCAD, not about technical drawing – which I in any case learned half a century ago and have forgotten. Typical line widths in machine drawings according to ISO are:

- 0.5 mm: outlines (solid lines)
- 0.35 mm: hatches (solid lines) and hidden edges (dashed lines)
- 0.25 mm: dimension and leader lines (solid), center and break lines (dash-dot)

Command Line

If you have worked with Linux you are familiar with its Terminal – a **Command Line Interface** (CLI). In Windows it is called the **Command Prompt** – but who does ever need it with Windows?

The **Command Line** (or CL for short and which includes the Command Box) has a similar function in LibreCAD: We can actually draw in LibreCAD using commands that we type into this area. We can use it to draw lines, rectangles, circles and more, as well as to place entities accurately in our drawing.



```
l
Command: l (line)
ci
Command: ci (circle)
Command: ci (circle)
arc
Command: arc (arc)
```

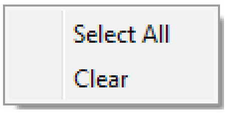
Specify startpoint or [center]

The commands l, ci, and arc have been entered and there are echoes on the Command Line area

Why use the Command Line, why not stick to the mouse and Toolbar icons? The answer is that if you take the trouble of getting proficient in using the Command Line, it can be both faster and a more precise way of drawing. On the other hand, entering position numbers requires fast computation in one’s head to get the snap point location right (good mental training to avoid Alzheimer’s).

To enter a command, **click** inside the Command Box (framed in the screenshot), or press [**Ctrl**]+[**M**] or just [**Space**]. The word Command turns blue to show it is activated and awaiting your command. Check however before you enter a command that you are on the right Layer. The figure above shows a case where three commands have been entered and the Command Line has responded.

If a command you just entered is wrong, you can return it to the Command Box by pressing the keyboard's [**Up**] arrow and correct the command.



You can erase the Command Line by hovering the mouse above it and right-clicking. It opens the pop-up window to the right. Click **Clear** and the contents of the Command Line is erased.

LibreCAD mostly waits for special **input**: you should e.g. select an object or set a point. Look at the blue text above the Command Box, LibreCAD writes down in this area what it wants.

The Command Line's **cache memory** can be erased by entering the command **kill**, or **k** for short. Use it if LibreCAD seems confused about what to do.

The Command Line has a **calculator** funtion. Enter for instance **cal 4 + 9** [**Enter**] and the echo is 4+9 = 13. The calculator can also do more advanced calculations like trigonometric functions and square roots (sin, cos, asin, acos, sqrt, etc.).

For detailed information, read the Wiki tutorial "A short manual for use from (sic) the command line" at <http://wiki.librecad.org/index.php/A_short_manual_for_use_from_the_command_line>. Below however a summary of LibreCAD control commands:

Control commands

The tables below contain the commands that can be used on the Command Box. I have copied the set from <http://wiki.librecad.org/index.php/Commands>.

DRAW		
Action	**Tool**	**Commands**
DrawPoint	Points	po, point
DrawLine	Line – 2 Points	l, li, line
DrawPolyline	Polyline – Polyline	pl, polyline
DrawLineParallel	Line – Parallel	o, pa, offset, parallel
DrawArc3P	Arc – 3 Points	a, ar, arc
DrawCircle	Circle – Center, Point	ci, circle
DrawLineRectangle	Line – Rectangle	rec, rect, rectangle
DrawMText	Tools – MText	mtxt, mtext
DrawText	Tools – Text	txt, text
DrawHatch	Tools – Hatch	ha, hatch
DrawLineFree	Freehand Line	fhl, free
DrawSpline	Spline	spl, spline
DrawSplinePoints	Spline Through Points	stp, spline2

DRAW		
Action	**Tool**	**Commands**
DrawLineOrthogonal	Line – Orthogonal	ortho, perp
DrawLineVertical	Vertical Line	ver, vertical
DrawLineHorizontal	Horizontal Line	hor, horizontal
DrawEllipseInscribed	Inscribed Ellipse	ei, ie
DrawLineBisector	Bisector	bi, bisect
DrawLineTangent1	Tangent (P,C)	tanpc, tangentpc
DrawLinePolygonCorCor	Polygon (Cor, Cor)	poly2, polygon2v
DrawLineParallelThrough	Parallel Through Point	pp, ptp
DrawCircle2P	Circle – 2 Points	c2, circle2
DrawCircle3P	Circle – 3 Points	c3, circle3
DrawCircleTan3	Tangential 3 Circles	ct3, tan3

VIEW		
Action	**Tool**	**Commands**
ZoomRedraw	Redraw	zr, rg, regen, redraw
ZoomWindow	Window Zoom	zw
ZoomAuto	Auto Zoom	za
ZoomPan	Zoom Panning	zp
ZoomPrevious	Previous View	zv

EDIT		
Action	**Tool**	**Commands**
EditUndo	Undo	u, undo, oo
EditRedo	Redo	r, redo, uu
EditKillAllActions	(CL cache memory)	k, kill

DIMENSION		
Action	**Tool**	**Commands**
DimAligned	Aligned	da
DimLinear	Linear	dr
DimLinearHor	Horizontal	dh
DimLinearVer	Vertical	dv
DimLeader	Leader	ld

MODIFY		
Action	**Tool**	**Commands**
ModifyTrim	Trim	tm, trim
ModifyTrim2	Trim Two	t2, tm2
ModifyMove	Move / Copy	mv
ModifyBevel	Bevel	ch, fillet (bug)
ModifyMirror	Mirror	mi
ModifyRotate	Rotate	ro
ModifyScale	Scale	sz
ModifyStretch	Stretch	ss
ModifyDelete	Deleted Selected	er
BlocksExplode	Explode	xp
ModifyCut	Divide	di, div, cut
ModifyEntity	Properties	mp, prop
ModifyRound	Fillet	fi, fillet
ModifyAttributes	Attributes	ma, attr

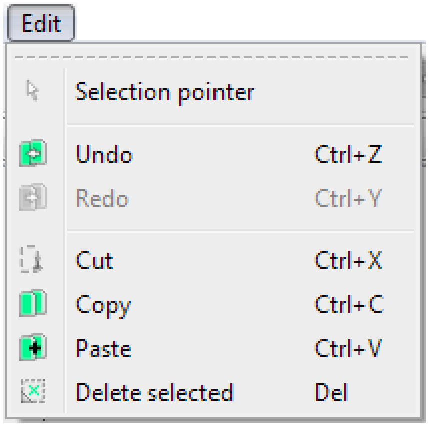
SNAP		
Action	**Tool**	**Commands**
SnapFree	Free Snap	os, sf
SnapGrid	Snap on Grid	sg
SnapEndpoint	Snap on Endpoints	se
SnapIntersection	Snap Intersection	si
SnapCenter	Snap Center	sc
SnapMiddle	Snap Middle	sm
SnapOnEntity	Snap on Entity	np, sn
SnapDist	Snap Distance	sd
RestrictNothing	Restrict Nothing	rn
RestrictOrthogonal	Restrict Orthogonal	rr
RestrictHorizontal	Restrict Horizontal	rh
RestrictVertical	Restrict Vertical	rv
SetRelativeZero	Set Relative Zero Position	rz
SnapFree	Free Snap	os, sf
SnapGrid	Snap on Grid	sg
SnapEndpoint	Snap on Endpoints	se
SnapIntersection	Snap Intersection	si

SNAP		
Action	**Tool**	**Commands**
SnapCenter	Snap Center	sc
SnapMiddle	Snap Middle	sm
SnapOnEntity	Snap on Entity	np, sn
SnapDist	Snap Distance	sd
RestrictNothing	Restrict Nothing	rn
RestrictOrthogonal	Restrict Orthogonal	rr
RestrictHorizontal	Restrict Horizontal	rh

INFO		
Action	**Tool**	**Commands**
InfoDist	Distance Point to Point	dpp, dist
InfoAngle	Angle Between Two Lines	ang, angle
InfoArea	Polygonal Area	ar, area

TOOL		
Action	**Tool**	**Commands**
ToolRegenerateDimensions	dimregen	

Keyboard commands can be quicker than entering commands on the Command Line. They can be found on the Menu Bar under **Edit**, as shown to the right.

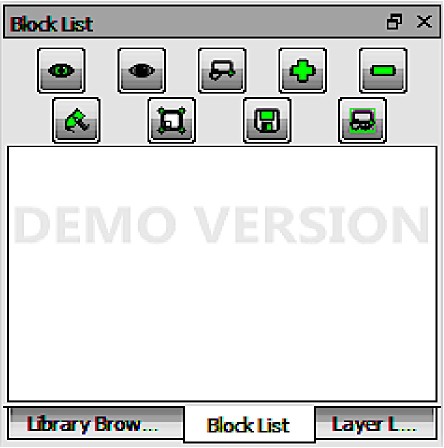


Keyboard shortcut commands

LibreCAD does not have a function that could help you remember all the control commands. It is a lot to keep in mind even if the abbreviations are quite logical. We are faced with the same problem as for instance when using the Terminal in Linux – when needed, you have to dig the commands out of your head. The only suggestion I can make is to have a single-page cheat-sheet of the most important tables beside you when drafting.

Block List

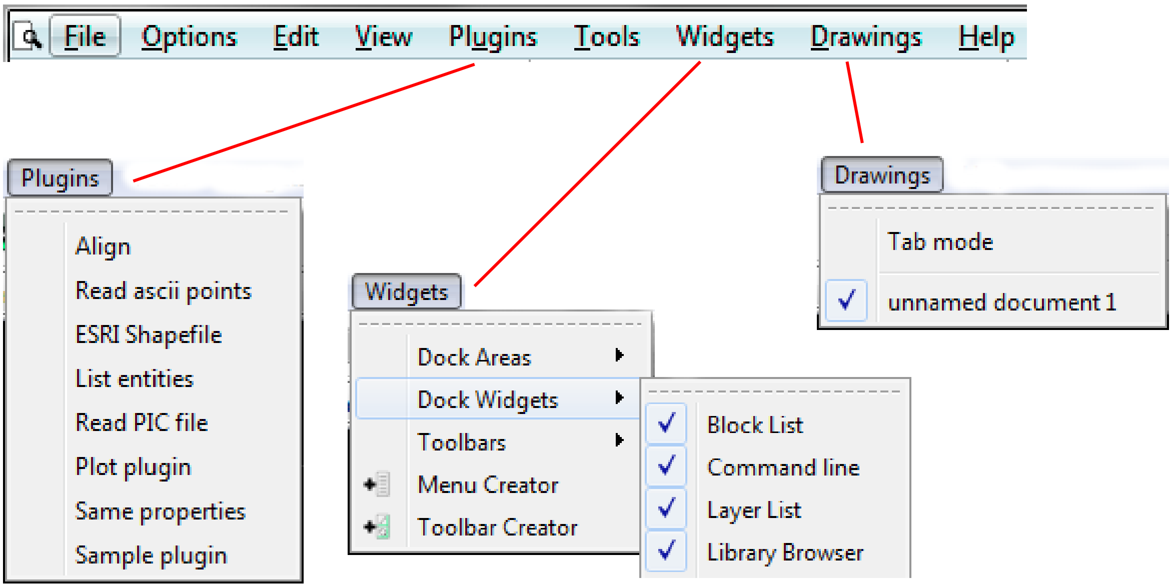
The Layer List shares its screen area with two other functions: The **Library Browser** and the **Block List**. The Library Browser will be mentioned in the section **Preparing for Work**, subsection **Sawing drawings**. The Block List is more complicated and we are not going into the details, but it will be briefly demonstrated in Chapter 3, Exercise #5. Please note that hatches are problematic and usually get lost when you save a block.



Block List

Menu Bar

The Menu Bar looks fairly traditional, although the **Plugins**, **Widgets**, and **Drawings** drop-down menus differ from what we see e.g. in LibreOffice Writer or MS Word. In the figure below I have opened the Dock Widgets drop-down menu (only the top part is shown, the bottom part will be discussed below under the heading “Widgets Bar”). You can see that the ticked items are the Layer List and Command Line, and the buttons between them that allow us to shift the area between the Layer List, Block List, and Library Browser:



The toolset that can be accessed under **Widget>Toolbars** contains groups for the Toolbar, Layer Selection, and Widgets Bar alike, and will be discussed later.

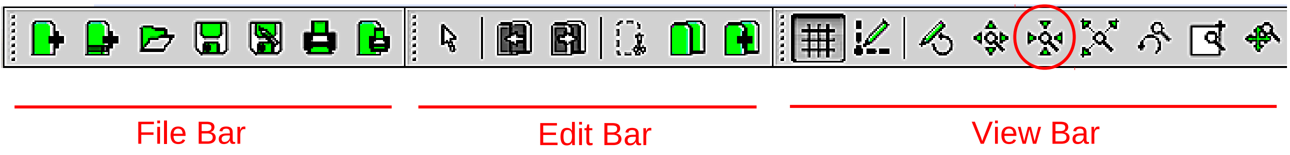
Pay also attention to **Drawings>Tab mode**. In the Tab mode we can open and work on multiple drawings in parallel, similar to having multiple tabs open on a web browser.

Toolbar

The Toolbar is divided into horizontal and vertical parts. The vertical part to the left of the main GUI contains drawing widgets that are quite similar to those you find e.g. in LibreOffice Draw, PowerPoint, and Visio.

The horizontal part of the Toolbar contains three groups. **The File Bar** (left group) has well-known icons related to the document in case: Open, Close, Save, Print, etc. **The Edit Bar** (middle group) are icons related to editing actions: Selection pointer, Undo, Redo, Cut, Copy, and Paste. It is unfortunate that the icons, apart from the Selection pointer, look quite unfamiliar.

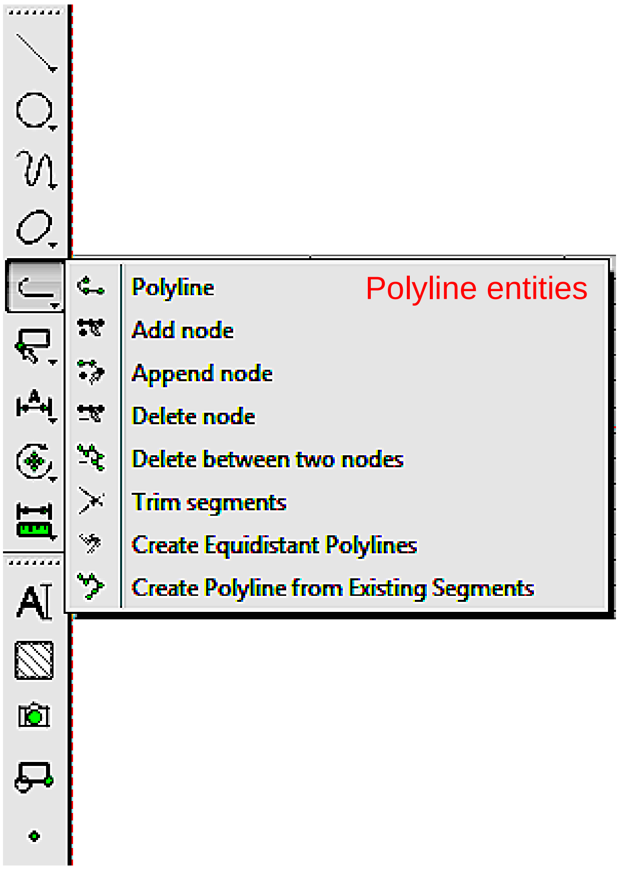
The View Bar (right group) relates to the Drawing Area: Grid, Draft, Redraw, Zoom in, Zoom out, Auto zoom, Previous view, Window Zoom, and Zoom panning. The zoom icon (circled) looks unfamiliar and takes some time to get used to. **Window Zoom** means “fit to current screen” and Zoom panning opens the hand pointer that allows us to pan (move around) the drawing.



The **Draft** button can cause confusion: If something suddenly disappears from view, check that Draft is not activated – if it is, it most likely has caused the disappearance.

The vertical part of the toolbar, shown to the right, has almost the same icons as you find under Tools on the Menu Bar – one or two are missing though.

The top group of nine icons are drawing entities plus Select, Dimension, Modify, and Info (strange icon for Info if you ask me, but it seems to be a piece of a measuring tape – and measure it does).



Each pushbutton acts as a drop-down menu that gives us access to additional control tools, as shown for **Polylines** in the screenshot to the right. If we need more details – e.g. to add dimensions – we can open the **Widget Palette** that will be discussed later.

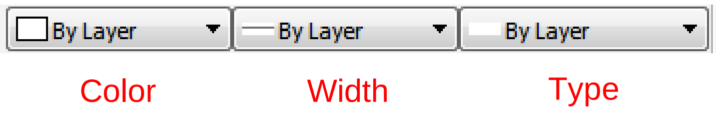
The bottom group of five icons are MText (multiline text), Hatch, Insert Image, Group, and Point. I have used Point only once and think it should be put under Line since a point – if we believe set theory – is a line with zero length and width.

There is a bug in LibreCAD and tool-tips are shown only for the bottom five when you hover the cursor on top of the icons.

We will use most of the icons later when doing exercises, the rest you can find out by experimenting.

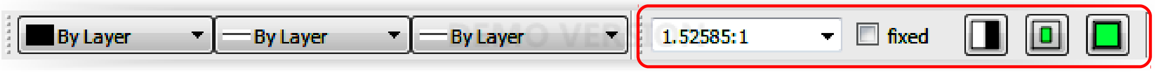
Layer Selection

The Layer Selection area (it goes under the name "Pen" on Widgets>Toolbars) consists of three **drop-down menus** that facilitate speedy selection of layer attributes: Color, Width and Type for widgets. It allows us to **override layer properties** that were defined when we created the layer using the Layer List.



Say, for instance, that we left Color at the default black/white. We can now pick a new color for a line that we are in the process of drawing with the Color drop-down menu. To return to layer definitions, shift back to By Layer on the drop-down menu.

The **Tool Options Bar**, to the right of the Layer Selection area, opens in response to certain user actions. Below I have selected Print Preview for a drawing:

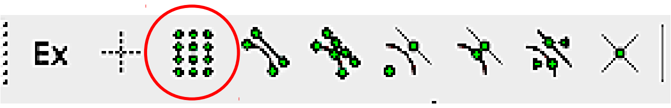


Widgets Bar

Everything on the Widgets Bar can be accessed on the Menu Bar by clicking **Widgets>Toolbars**. The first group in the default setting of the Widgets Bar is **Snap tools**. When you connect lines there will always be a gap or an intersection, it is impossible to hit an exact point or a line by mouse actions – a point has after all **zero dimensions**. The solution is to use tools on Snap tools.

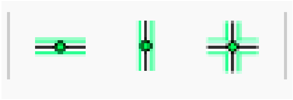
For a more detailed discussion on snapping, read the document "Snapping" that is available at <http://wiki.librecad.org/index.php/Snapping>.

Snap tools allow us to lock (snap) widgets to each other or to certain positions on the screen. The icons on the **Snap Bar** bar (shown below) are, from left to right, Exclusive Snap Mode (Ex), Point Snap, Snap on Grid, Snap on Endpoint, Snap on Entity, Snap Center, Snap Middle, Snap Distance, and Snap Intersection. The alternative that we will use most while drawing is Snap on Grid (circled).

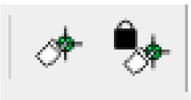


Getting text strings in the right position on a drawing can be challenging. My experience says that the best approach is **deselect** all Snap tools, then drag the text to its location with the mouse - whether we are talking about MText or Text.

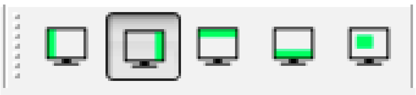
To the right of the Snap Bar, are four more groups of icons, the first two of which are considered part of the Snap Bar. First come three restriction icons: Restrict Horizontal, Restrict Vertical, and Restrict Orthogonal:



Next are two icons that allow us to set and lock widgets relative to the zero position:

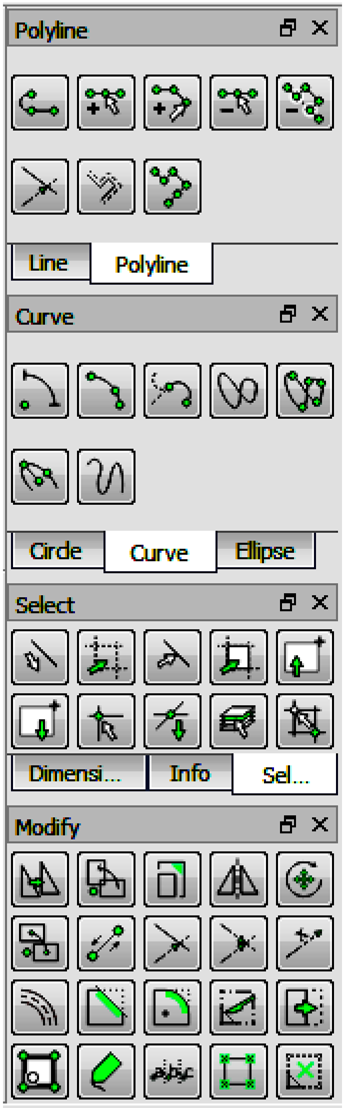


In the third group of five windows only the two leftmost have any noticeable effect, but it is dramatic. You can see that the second icon is active. Click it off and the Layer List and Command Line windows will be closed. Click on the leftmost and the **Widget Palette**, shown above right, is opened (is "docked") to the left of the Drawing Area:

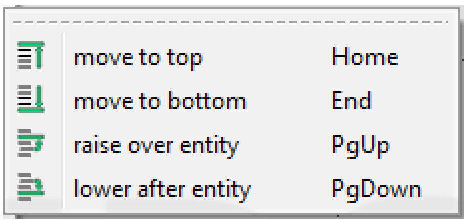


The docking functions can be activated/deactivated by clicking **Widgets> Toolbars>Dock Areas**. The palette contains icons for the vertical Toolbar from Line to Info in a slightly different order.

There is however one detail (at least) that is missing from both the vertical Toolbar and from the Widget Palette: **Order**. Click on the Menu Bar **Tools>Modify>Order** and there you have alternatives by which to select the order by which entities are displayed (image below right).

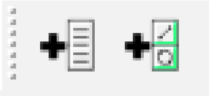


Widget Palette

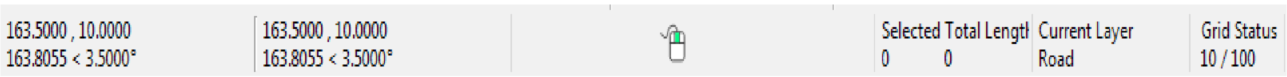


☰	move to top	Home
☰	move to bottom	End
☰	raise over entity	PgUp
☰	lower after entity	PgDown

The last group is the **Creators**, Menu Creator and Toolbar Creator pushbuttons:

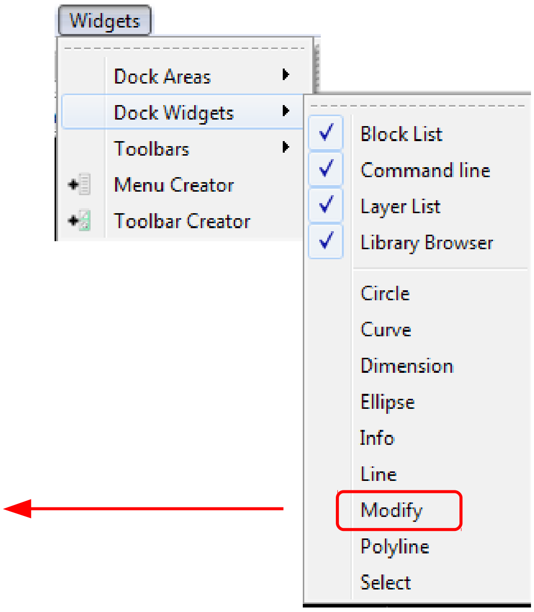


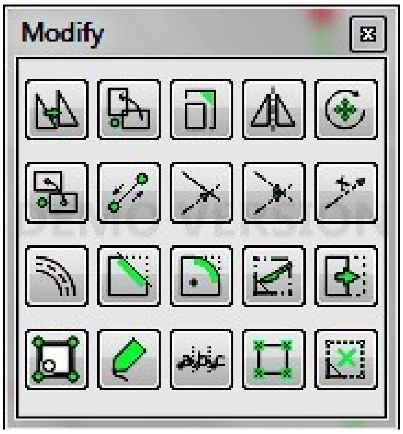
The bottom of the dashboard contains displays for real-time information about the location of the mouse pointer (left) and the widgets we work on (right). In the middle is something that should give information about mouse buttons, but does not work:



```
163.5000 , 10.0000     163.5000 , 10.0000          Selected Total Lengtl  Current Layer  Grid Status
163.8055 < 3.5000°     163.8055 < 3.5000°          0         0            Road           10 / 100
```

The Widget Palette is just one palette that can be docked to the left of the Drawing Area. On the Menu Bar, if we click **Widgets>Dock Widgets**, there is a whole toolset of docking alternatives as shown below to the left. As an example I have opened the **Modify Panelette** (it opens extended along the vertical Toolbar, here I have compressed it).



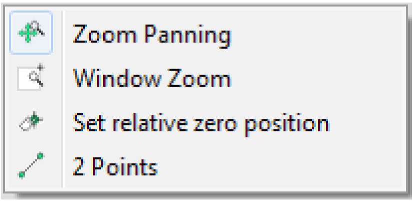


Observe that there are more tool options that can be opened by clicking **Widgets>Toolbars**. Which tools to have active is up to you to decide depending on the current situation.

Copy-pasting parallel lines with the **Line>Parallel** tool will be discussed at the end of Exercise #2 in Chapter 3. **Polylines** are needed e.g. to create a closed frame for hatches, as is also told in Exercise #2.

Mouse actions, zooming and panning

Right-clicking on the mouse opens different selection windows depending on the location of the cursor; for instance, the window below to the right when the cursor is on the Drawing Area. It is a list of the most recently used tools; using it speeds drawing since typically the same tools are used repeatedly.



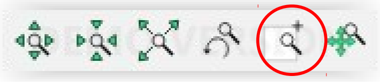
Right clicking when a drawing action is ongoing, ends the action. This can also be achieved by pressing the [**Esc**] button.

Wheel scrolling triggers by default **Zoom in/out** around the mouse pointer. Holding down the [**Ctrl**] key on the keyboard while scrolling triggers vertical panning; horizontal panning is triggered by holding down the [**Shift**] key.

In case I did not say it before: **Panning** = moving the drawing around.

There are still more ways to zoom a drawing:

- by presssing [**Ctrl**]+[**+**] (zoom in) or [**Ctrl**]+[**-**] (zoom out)
- by using zoom buttons on the Toolbar
- by clicking **View** on the Menu Bar and selecting a zoom function (same functions as the zoom buttons)

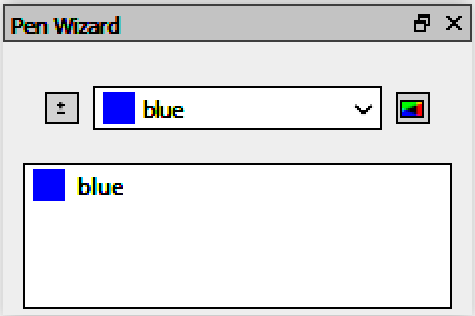


The **Window Zoom** tool (right, circled) allows you to quickly zoom in on a particular area of the drawing: Click on the icon, select the area of interest with the magnifier pointer and LibreCAD zooms in on the area once you release the mouse button.

The **mouse wheel** can also be used for **panning**. Just press the wheel and move the mouse, and the whole scenery moves around (but LibreCAD may zoom at the same time). The alternative is to click the **Zoom Panning** button on the Toolbar to get the hand pointer for panning.

The **Zoom Panning** function to the right of Window Zoom can also be activate in the traditional way: by entering **zp** (Zoom Pan) on the Control box (and click [**Enter**]) and pressing the left mouse button.

Pen Wizard



The Pen Wizard came with LC version 2.2.0 – as far as I know. It is a widget that, according to LC Wiki, allows you to:

- maintain a list of favorite colors
- change the active pen color
- change the color of all selected objects
- select all objects of a specific color

I have not used the wizard and cannot say anything about its utility. It takes a lot of real estate but can be closed by clicking **Widgets>Docked Widgets>Pen Wizard**.

Scaling

If you like me learned technical drawing half a century ago, it was done with ink and paper on a drawing table. One thing to consider before starting to draw was which scale to use. The drawing had to be scaled up or down depending on the size of the object and the size of the paper. And then came the problem of locating the object and its various projections symmetrically on the paper. When you screwed up you had to use a razor blade to scrape off the ink lines.

Those problems are largely gone with CAD (so I have been told): We **draw in scale 1:1** using whatever dimension is needed, millimeter or meter. When drawing on a document template, say A3, it can have a virtual “paper” size in kilometers. The printed format will nevertheless be correct since the format of the hard copy is defined by printer settings.

The other possibility is to draw without a template and then either **scale the drawing** to fit the document size or **scale the document template** to fit the drawing. I try the first approach in Chapter 3, Exercise #2, and the latter in Exercise #6 – concluding that the latter technique is simpler.

Blocks

Blocks work, as said before (Chapter 1), like “group” commands in PowerPoint and LibreOffice Impress and similar programs. The difference is that we can, and are required to, save blocks as separate files. The blocks can then be imported into other drawings. LibreCAD imports them to Level 0 and they should be “moved” to another level with **Modify>Attributes** (or the commands **ma** or **attr**). If a block has to be edited, it must first be “exploded” (“ungrouped” in LibO and M$ Office) using **Modify>Explode** or the command **xp**.

Here are recommendations related to blocks that I have picked from Gary Fox’s videos:

- Blocks can be created and used inside a drawing, but in some cases it may be easier to copy-paste (**Modify>Move/Copy**) the object instead to avoid unnecessary complexity
- Draw blocks in **scale 1:1**. In this way you always know the true size of the object
- An exact **insert point** is needed. One solution is to add to the block, in a separate layer, an insert point of crossing lines. The point (“crosshair”) can then be used to snap the block to the correct position
- Be careful if you modify an inserted block. If you have used more of the same block in the drawing – blocks of a nut for instance – the modification will affect all of the inserted blocks

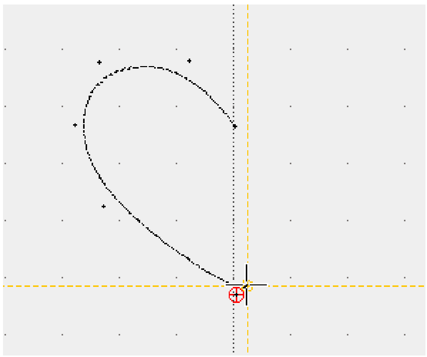
My experience tells that if I create a block with a hatched component, the hatches will be invisible until I explode the block. Another hatches-related bug.

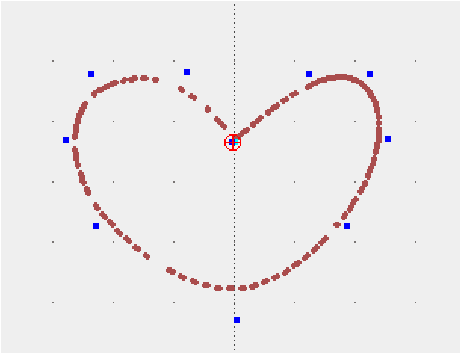
For additional information, see <http://wiki.librecad.org/index.php?title=Blocks>. The videos by Gary Fox are mainly concerned with blocks and worth watching.

Splines

Splines are not a major issue in technical drawing. If you need them, take a look at the tutorial “Spline tutorial 1” by Claudio Guarnieri <http://wiki.librecad.org/ index.php/Spline_tutorial_1>.

Below is a small test that I made by clicking **Toolbar>Curve>Spline**. The left screenshot below shows a semi-finished “heart” that I have drawn by clicking a starting point and five nodes (the small dots can be seen). To the right is the finished drawing (in the Selected mode). The tip of the heart has been rounded since the curve is an approximation of the “waypoints” that I have added by clicking.



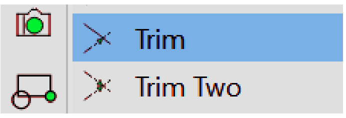


Is **Spline** the correct term in this case when LibreCAD does not attempt to draw the curve through the points? I just wonder. The usual name for points that can be used to manipulate an image is **Handle**. On the other hand, the points above are different from **Points** in a freehand drawing made by e.g. LibreOffice Draw.

Trimming and Lengthening

LibreCAD does not have an eraser tool that would allow us to quickly get rid of a line part, as for instance the part of a line that becomes hidden behind a structure that we add to the drawing. For such occasions we need to “trim” the line.

Trimming (shortening) is done with the **Trim** and **Trim Two** tools available under **Modify** on the Toolbar. The process when cutting off a part of one of two **crossing lines** is briefly stated this (you can draw two crossing lines to see that it works):



- select the line to be trimmed
- select the **Trim** tool
- select the other line (“cutting line”) that now turns grey
- click on the part to be **retained** of the first line, the other part is deleted

The Trim Two tool speeds up the process if the ends of both crossing lines have to be deleted.

An improvement that I hope to see in the future is the ability to trim a part of a line, e.g. the part of a horizontal line lying between two vertical lines.

Shortcuts

LibreCAD, as most application software, offers alternative ways of working, including shortcuts that speed up the process. Here are a few examples, some of them have been mentioned before:

- Enter commands on the Command Box instead of trying to snap widgets with the mouse
- Right-click on the Drawing Area to access recently used tools
- Hover the cursor above an icon to open **tool tips** and see what it contains (does not work with all icons)
- When drawing multiple “2 Points” lines that are not connected, click <u>once</u> on [**Esc**] after finishing a line. The mouse can then be moved to the starting point of the next line
- Offset (“copy-paste”) straight lines with **Modify>Move/Copy** instead of redrawing. The true gain depends on the situation, sometimes redrawing can be faster
- Faster still: Copy-paste lines with the **Lines>Parallel** tool on the Toolbar
- Construct “help lines” on the Construction Layer and draw by snapping to their intersections. Check the videos “LibreCAD Drawing 1,” Parts 1 & 2, by David Brus (Brus’ use of the [**Tab**] key may not work)
- Offset multiple widgets at a fixed angle with the rotation option instead of redrawing (Toolbar: **Modify>Rotate**)

Room for tool improvements

The link for Part Libraries at <wiki.librecad.org>

What I miss in LibreCAD is a decent repository of snap-on drawing symbols and components (“parts” and “blocks” in LibreCAD parlance) for engineering and architectural drawings, the kind you find in e.g. MS Visio and SketchUp. There are the so-called **Part Libraries** at <http://wiki.librecad.org/index.php/Part_Libraries>, but they contain very little and are not well organized. This is an area where LibreCAD’s users really could help out.

You can try to search the Internet for free **DXF templates** or **DWG drawings**, perhaps there is something useful out there that people are willing to share without copyright restrictions and which LibreCAD accepts. One of the very few that I have found is the helicopter drawing mentioned in Chapter 4.

I also miss the ordinary **Group/Ungroup** alternative in e.g. LibreOffice Draw and PowerPoint. Having to create blocks is tedious and quickly clutters the Block List.

There seems to be a bug with the Select function because it is often activated even without a mouse click, resulting in some widget(s) being inadvertently selected. In these cases a quick **Select>Deselect** alternative would help (e.g. [**Ctrl**]+[**D**]).

The ability to **lock** a widget, group or block to the drawing would be very useful. I cannot tell how many widgets or groups I have already lost because they have been inadvertently selected and then deleted with something else.

Interacting with LibreCAD

As a summary we can conclude that there are numerous ways by which to enter data to LibreCAD:

- via icons and drop-down menus on the different bars
- on the pop-up window that opens by right-clicking on the mouse
- by entering commands on the Command Box
- by working in the Drawing Area with the mouse

LibreCAD outputs visual information in several ways:

- on the Layer List
- on the Command Line
- at the bottom of the GUI
- with tool-tips as the mouse is hovered above icons

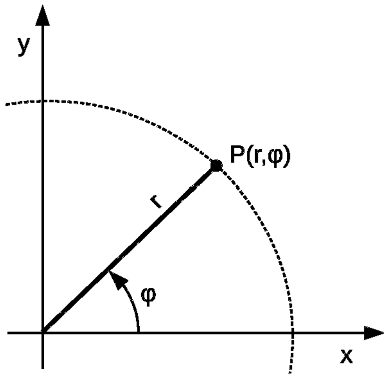
The coordinate system in LibreCAD

The orthogonal case

This case is discussed in more detail in the tutorial “Coordinate system,” available at <http://wiki.librecad.org/index.php/Coordinate_system>.

We had a general discussion on coordinate systems in Chapter 1. Now we shall take a closer look at how LibreCAD’s coordinate system works. It is an important subject, because everything you draw in LibreCAD will be placed accurately based on either the **x/y** or **r/φ coordinates** (Cartesian or Polar coordinate systems).

All x units are measured horizontally and all y units are measured vertically. Coordinates can also be shown as positive (+) or negative (-) values. With polar coordinates angles are measured **counter-clockwise**, starting from the horizontal x direction.



Polar coordinates [JH]

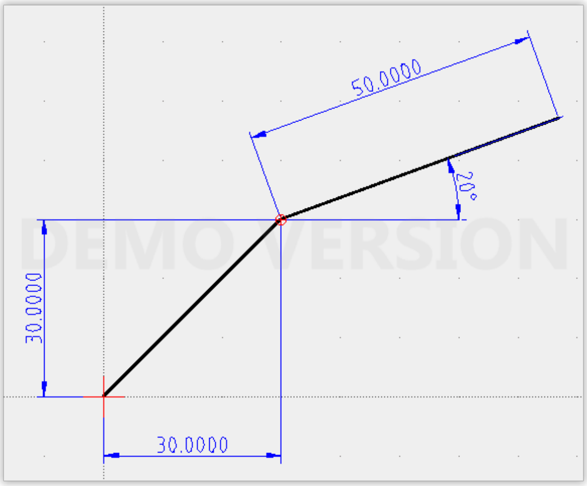
Numeric x and y coordinates are entered as e.g. 40,30 (x=40, y=30). The same point using r/φ coordinates is approximately 50<36.87 (r=50, φ=36.87º). Approximately, because the angle 36.87º is not exact. (You can calculate the angle with the Command Line calculator by entering **cal asin(3/5)*(180/pi)**.)

The absolute **origin** or **zero point** in your drawing is where the x and y axes cross each other and are represented by a **red cross** in LibreCAD. Every entity you draw (line, point, arc, polyline, circle, etc.) is located relative to this origin.

In LibreCAD there is also the option to set the **relative zero point** (small red circle). This relative zero point can be set **temporarily** to a new location in a drawing so that all subsequent x and y coordinates of, for instance, drawn entities or placed blocks will be relative to this temporary zero point.

Combined use of the relative zero point and angle commands (entering commands like @100<30) can in some cases speed up drawing, This is demonstrated in Chapter 3, Exercise #3.

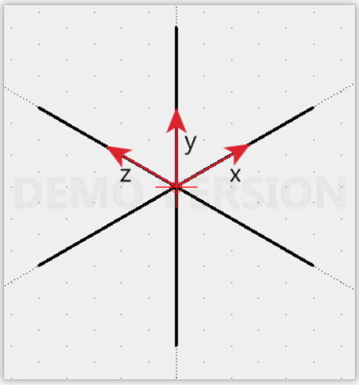
The figure to the right shows a 2 Points line drawn from 0,0 to 30,30; then extended with the relative command @50<20.



The relative zero point can become a pain down there. The simplest way to move it somewhere else (it cannot be deleted) is to click **Set relative zero position** (on the Widgets Bar), move the cursor and click at a location where you want to put the bugger.

The isometric case

The shocking news is that there is no such thing as [x,y,z] dimensions in the 2D LibreCAD. We only have [x,y,0]. The third dimension is pure illusion – but even so a useful one.



We define entities (widgets) in x, y, and z directions although we work only with a two-dimensional plane. This is done by sticking to the rule that entities along the x direction will have angles of 30º or 210º, along the y axis 90º or 270º, and along the z axis 150º or 330º; as is obvious from the image to the right. Elementary, my dear Watson.

We can shift the direction of y and z axes to the traditional “right-hand” definition. It does not really matter because the x,y,z definitions are not used in any way while drawing, they are only mental concepts. But if you are using the drawing in a text on **vector algebra**, the wrong x,y,z (or i,j,k unit vector) definition will cause confusion.

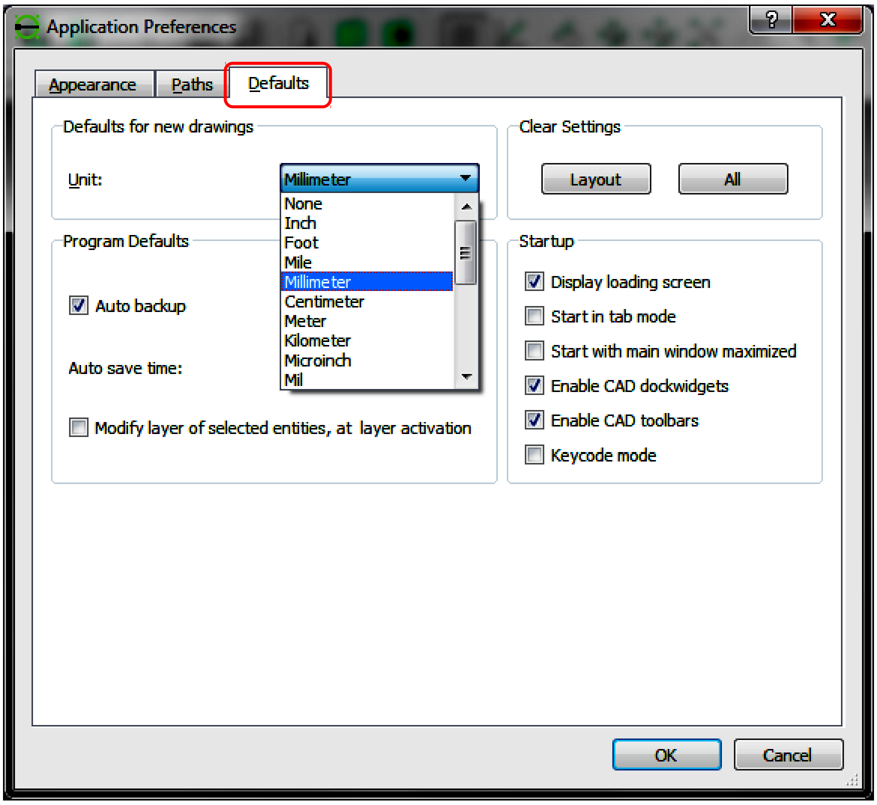
Preparing for work

LibreCAD, when being loaded, by default generates a new unnamed drawing. We name the drawing and select a target folder (address) when we save the work (see below for details).

The first step in preparing for the drawing task is to set necessary attributes – or perhaps start by opening a template for the paper format we are going to use.

Setting units

On the Menu Bar, click **Options>Application preferences** to open this window (shown after I have clicked the **Defaults** tab and the **Unit** drop-down menu):



Application Preferences

Defaults for new drawings

Unit: Millimeter

Program Defaults

☑ Auto backup

Auto save time:

□ Modify layer of selected entities, at layer activation

Clear Settings

Startup

☑ Display loading screen

□ Start in tab mode

□ Start with main window maximized

☑ Enable CAD dockwidgets

☑ Enable CAD toolbars

□ Keycode mode

The first tab, **Appearance**, opens a window that lets us modify to our liking the tools we work with (Graphic View and Graphic Colors) and the languages we prefer to work with (GUI language and Command language). We use English and leave them as they are.

The next tab, **Paths**, is needed when we set up a library of fonts, hatches, part library items, templates, etc. And why would we want to do that? To avoid having to reinvent the wheel. If you for instance create the drawing for a door to an architectural drawing, why not save the little drawing as a component for later use? LibreCAD uses the following **default folders** (system folders):

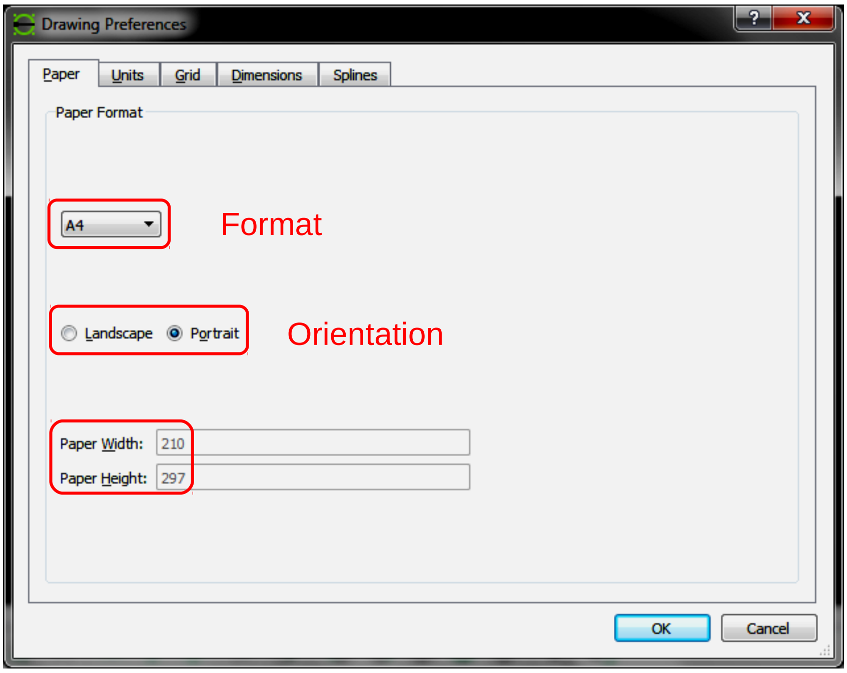
- Under Windows: C:\Program Files (x86)\LibreCAD\resources\library
- Under Linux: usr/share/librecad/library

The third tab on Application Preferences, **Defaults**, is the one that interests us because here we select the units we are going to work with. The default is millimeter, and that is what we are will use, but you can go for other units as well. Check that **Auto backup** is selected. Click [**OK**] to exit.

The attributes that we define on Application Preferences are “global” in the sense that they are saved and remain in effect for future drawings unless we redefine them. Attributes that remain valid only for the current drawing are selected with **Current Drawing Preferences**.

Defining attributes for the current drawing

As said above, selections made under Application Preferences will reflect to your whole application (they are fixed preferences, so to speak). Attributes that should be specific for the current drawing only, have to be made in **Current Drawing Preferences**:



Paper Format

A4

○ Landscape ● Portrait

Paper Width: 210

Paper Height: 297

We find attributes for **Paper Format** under the first tab of the window: The default paper format is **A4**, but there is a whole range of alternatives under the drop-down menu. The A4 format is the basic paper size in Europe and belongs to the ISO family of A documents (cf. Appendix A), while **Letter** is the basic format in the US. Most of us have at home an ordinary A4/Letter printer that allows us to print the work. On the other hand, ink-jet A4/A3 printers for home offices are not too expensive these days (but the ink is).

Below Format are radiobuttons for selecting **Paper Orientation** between **Landscape** and **Portrait**. It would be unusual for an A3 or larger drawing not to be oriented landscape, A4 can be either or.

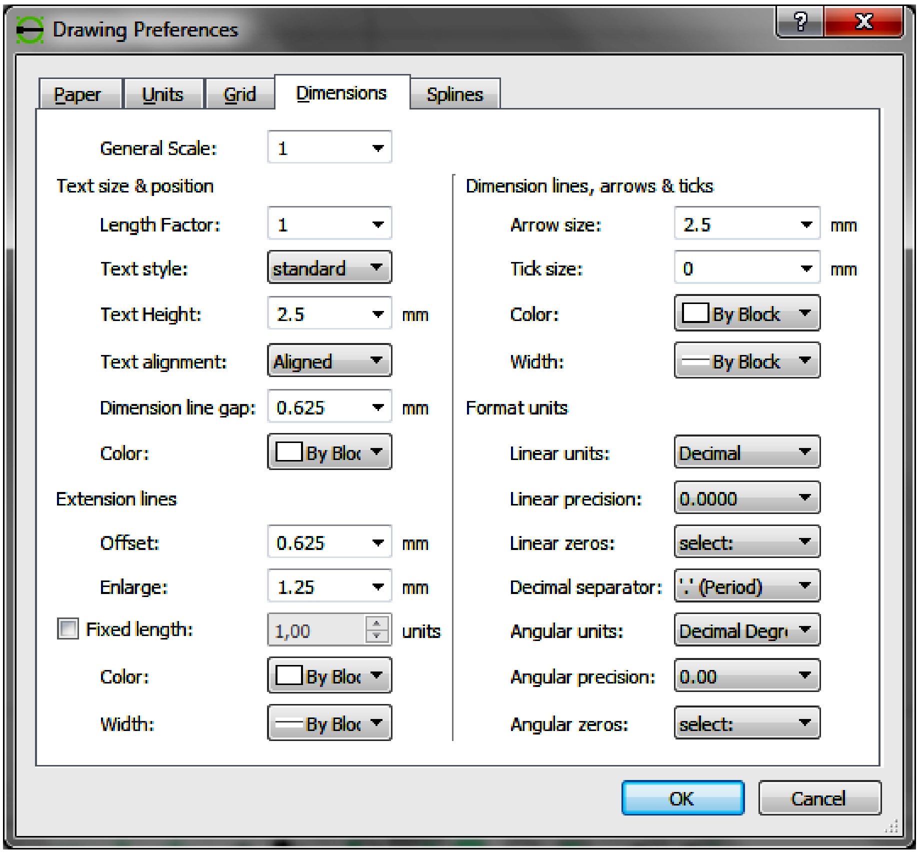
The width and length of the selected paper is given at the bottom of the Paper tab. The measures become important when we specify borders for our drawing. Then we have to enter measures for the drawing format minus the border width. For an A4 portrait format a border 5 mm inside the edges is size 200×287 mm.

Under the next three tabs we can for instance

- change from millimeters to another unit and lower the length Precision from four to two decimals (**Units** tab)
- shift between Orthogonal Grid and Isometric Grid (**Grid** tab)
- switch between **decimal period** and **decimal comma** – important in those countries where comma is used instead of period (**Dimensions** tab)

Attributes on the Dimensions tab become important when we have to **scale a drawing**, because then several attributes have to be changed accordingly. Check what I say about it in Chapter 3, Exercise #4.

The last tab is for selecting the number of line segments per spline patch, with 8 being the default (**Spline** tab).



General Scale: 1

Text size & position
Length Factor: 1
Text style: standard
Text Height: 2.5 mm
Text alignment: Aligned
Dimension line gap: 0.625 mm
Color: □ By Bloc

Extension lines
Offset: 0.625 mm
Enlarge: 1.25 mm
□ Fixed length: 1,00 units
Color: □ By Bloc
Width: By Bloc

Dimension lines, arrows & ticks
Arrow size: 2.5 mm
Tick size: 0 mm
Color: □ By Block
Width: By Block

Format units
Linear units: Decimal
Linear precision: 0.0000
Linear zeros: select:
Decimal separator: '.' (Period)
Angular units: Decimal Degr
Angular precision: 0.00
Angular zeros: select:

The all-important Dimensions tab

Changing the type of dimensions

A Dimension widget consists of **Extension Lines** that stretch out from the object to the **Dimension Line** with **Arrows** and a numeric **Label**. Dimensions are added to the drawing by selecting **Dimensions** on the Toolbar, or **Edit>Dimensions** on the Menu bar, or with one of the

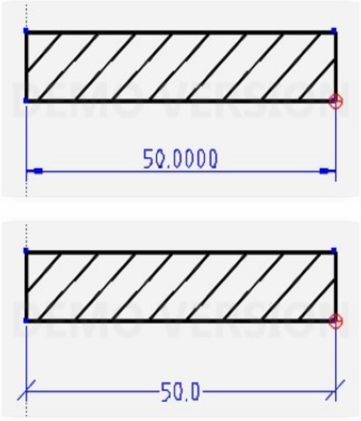
Dimensions commands on the Command Box (**da**, **dr**, **dh**,**dv**, **ld** – three dimension types lack a command),

Adding dimensions to a drawing can be done by clicking **Dimension** on the Toolbar and selecting the type of dimensions we want (usually **Horizontal** or **Vertical**). Then

- click on the location (e.g. a corner) from where to define the dimension
- move the mouse pointer to the other end and click
- drag down the dimension with its arrow to the wanted position
- click to lock the position

The default dimension type is the one used in **engineering**, with an arrow and the dimension above the arrow. In **architecture** the arrow heads are changed for a slash (“tick”), sometimes with the dimension located within a gap of the arrow. The image to the right shows the default dimension (top) and the alternative for architectural drawings (bottom). The latter is achieved by going to **Options>Current Drawing Preferences >Dimensions** and making the following changes

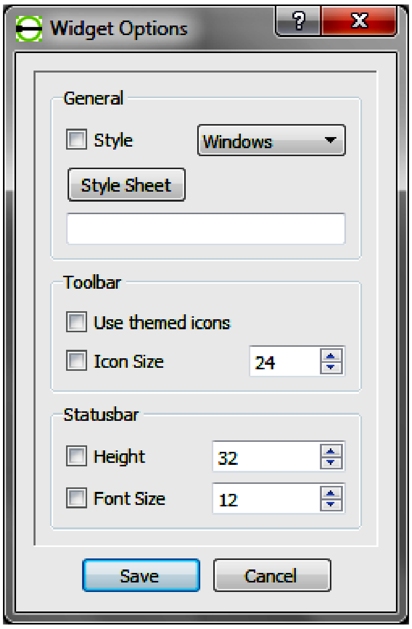
- Text alignment: Horizontal
- Arrow size: 0 mm
- Tick size: >0 (2 mm used in the figure)
- Linear precision: 0.0



Dimensions styles in engineering (top) and architecture (bottom)

Widget & Device Options

Click **Options>Widget Options** to open the window to the right. It lets us change widget attributes, although the result is more a visual preference than of practical importance.



General

☐ Style Windows

Toolbar

☐ Use themed icons

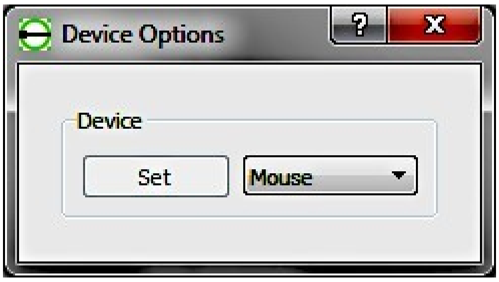
☐ Icon Size 24

Statusbar

☐ Height 32

☐ Font Size 12

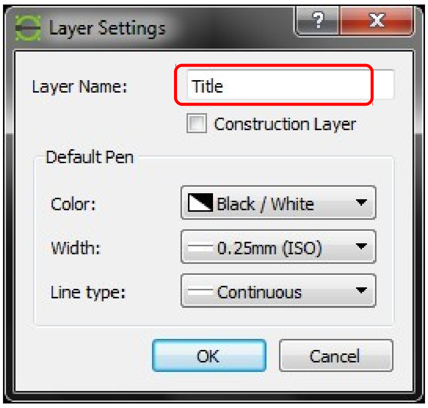
The next option in the list is **Device Options**. It pops up the small window below that allows us to select the device we are working with, should it not be a mouse.



Device

Mouse

Using the Construction Layer



Layer Settings

Layer Name: Title

□ Construction Layer

Default Pen

Color: Black / White

Width: 0.25mm (ISO)

Line type: Continuous

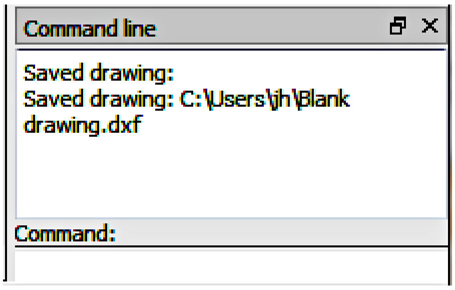
The **Layer Settings** window that opens when you create a new layer has a box **Construction Layer** that by default is not ticked. If you tick it, you will create a layer of **help lines** of infinite length that do not show up on a print.

Help lines are handy because with them we can outline the drawing and speed up the drawing process by snapping the lines we draw to the help lines and their intersections.

An example of the use of the Construction Layer is described in the video “LibreCAD Drawing 1 Part 1” by David Brus, which you can find on YouTube. A lesson of the video is that you need a good mental picture of the planned drawing to place help lines right. The alternative is to make a sketch on a paper that outlines the drawing and position of help lines.

Saving drawings

Where do our drawings go unless we specify an address? Click **File>Save** on the Menu Bar, name the empty drawing **Blank drawing**, and click **Save**. Congratulations, you just created your first drawing! On Windows you can find it under C:\Users\your_name (for me that is C:\Users\jh), on Linux in the folder home/your_name. The save action is also echoed to the Command Line (image right).



Command line

```
Saved drawing:
Saved drawing: C:\Users\jh\Blank
drawing.dxf
```

Command:

Blank_drawing.dxf has been saved

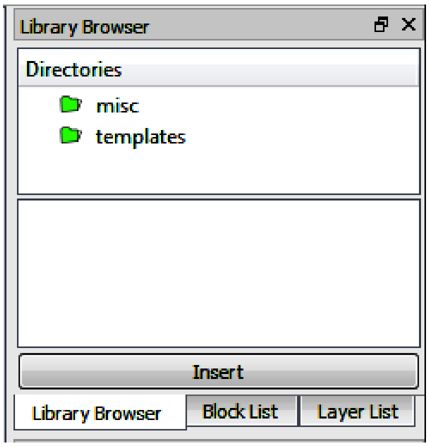
In the normal case we would however have created a folder for our drawings, say My_LibCAD_drawings, and save our drawings there. In addition My_LibCAD_drawings could have sub-folders called My_Templates and My_Blocks, into which we put templates and blocks that we create.

Templates and block can also be stored in the **system folders** mentioned earlier:

- Under Windows: C:\Program Files (x86)\LibreCAD\resources\library
- Under Linux: usr/share/librecad/library

The system folders have two pre-installed sub-folders, called “misc” and “templates,” so you could create a third called “blocks.” Administrator rights are required to fool around in the system folder.

The advantage of using the system folder is that you can access its contents through the **Library Browser** (as shown below).



For a more detailed discussion on templates, check the document "LibreCAD Quick Start Guide/Creating Your Own Templates" that is available at <http://wiki.librecad.org/index.php/LibreCAD_Quick_Start_Guide/Creating_Your_Own_Template>.

Printing drawings

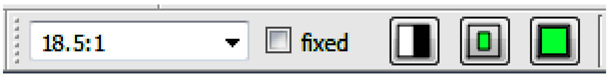
The first thing to think about is **printer margins**. Most printers cannot print on the entire sheet of paper. There's a "margin" around the outer edge that will vary depending on the printer. It's usually said to be around ¼", say 7 mm in civilized metric units. A printer can therefore print on an area no greater than about 283×196 mm instead of the full 297×210 mm of the A4 format – and the 283×196 mm assumes that the area will be accurately aligned on the paper.

In short, we either plan for using a smaller area of the paper or let the printer cut off what it cannot print.

There is an instructive short blog on printing PDF documents at <https://askleo.com/why_dont_pdfs_print_at_the_same_size_as_the_original/>. It's main message is that if we want a PDF document to be printed at 100% scale, we should change the **Print scaling** from the default **Fit to Printable Area** to **None**. My experience is that even so may shrink an A4 printout by 7 mm in both directions (shrunk by 3-4%). However, on some versions of **Acrobat Reader** you have the **Custom Scale** on the **Print** window. When I Custom Scale to 104% the printout produced by my old Brother HL-1430 printer is almost exact.

The latest version of Acrobat Reader lacks the Custom Scale feature (so I uninstalled the 300 MB Reader). I'll discuss it in Example #6.

Then to printing LibreCAD drawings: Do not go straight for the **Print** icon on the Toolbar (or **File>Print** on the Menu Bar). Select instead **File>Print Preview** (or click **Print Preview** on the Toolbar) and the **Tools Options Bar** opens next to the Layer Selection area (under the horizontal Toolbar). The five control items on the Tools Options Bar are in this case (left to right):



- **Scale**, drop-down menu that allows you to change the scale of the drawing to match that of the paper (and the scale of the printout should match the scale stated on the Title Block)
- **Fixed**, fixes our selections so they cannot be accidentally changed
- **Toggle Black/White mode**, switches the drawing between color and black/white
- **Center to page**, puts the drawing in the middle of the Drawing Area
- **Fit to page**, does what it says

When you are happy with the preview, click **File>Export>Export as PDF** (do not use **Export as image** because the PNG alternative gives poor resolution). Name the file and save it to wherever you want. Open it with a PDF reader and you should be all set to print it (Firefox is not bad as a reader and LibreOffice Draw lets you edit PDF files).

Should there be a scaling problem with a LibreCAD widget, you can look at the video “How to print with LibreCAD!!” on YouTube. It may offer advice on your problem,

Another YouTube video tutorial dealing with misaligned printer outputs is #6 by **Gary Fox**, “Blocks, Borders, and more advanced printing.” It presents a method for checking one’s printer with the help of parallel lines.

Page intentionally blank

3 Exercises

Observe that I use the following abbreviations in the exercises:

- CB = Command Box (commands to be followed by [**Enter**])
- Clk = Click (left-click)
- CL = Command Line
- DA = Drawing Area
- MB = Menu Bar
- TB = Toolbar
- ** = Marks beginning of a comment (not entered on the CB)

And remember that this chapter contains experimental exercises – not skilled demonstrations – that I did while learning LibreCAD.

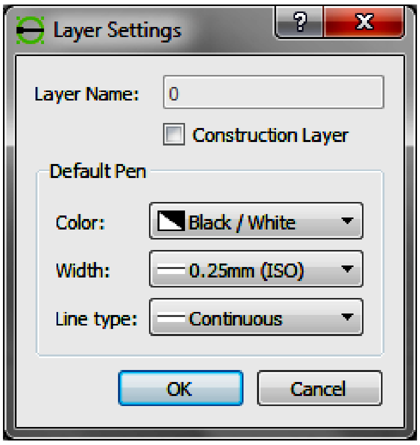
Exercise #1, A4 drawing template

I assume that you like me eventually have to print hard copies of your drawings. A printed drawing needs a Title Block with information like part name, material, thickness, scale, and also a part or drawing number for filing purposes. We can make the title block as simple or complex as we want. Here we go for the former, then print the drawing and save it as a template for future use.

This is my first ever LibreCAD drawing and there could be smarter ways to perform the task – try to find out.

Preparations

The drawing will consist of two parts: the border line and the title block (including text). Should we go for separate layers or not? I would vote for a single layer (the default 0) since the drawing will be used as a template in the future and then multiple layers are needless clutter. The default Continuous **Line type** is the only sensible alternative. There is also no need for a Construction layer. We do need text in the title block, but that should not be a problem – I hope.



Layer Name: 0

☐ Construction Layer

Default Pen

Color: Black / White

Width: 0.25mm (ISO)

Line type: Continuous

We have to select line **Color** and **Width**, but in this case the default Black/ White (B/W) is the right choice. I don't know which line width the standards ask for, but 0.25 mm feels OK for all lines. Then check Application Preferences. The most important attribute I see is **Unit**: it should be in millimeters. Change if needed, leave the rest as they are.

And finally to Current Drawing Preferences. Here is more to check and do:

- **Paper:** Select **A4** Format, Orientation **Landscape**, and pay attention to the dimensions: Width 297 mm, Height 210 mm
- **Units:** Main drawing unit **Millimeter**, Format **Decimal**, precision **0.000** (precision machinery today go to a thousandth of a millimeter or less – in construction engineering an inch is a precise measure)
- **Grid:** Default selections are OK (**Grid on**, **Orthogonal**, etc.)
- **Dimensions:** Change **Text Height** to 2 mm, leave the rest as they are
- **Splines:** No actions

With that we should be all set to go.

Draw the border

My suggestion is to draw the border 10 mm inside the A4 paper edge (better safe than sorry – know it, been there), which means that the border is a rectangle of size 277×190 mm and we start with the lower left-hand corner at the origin:

```
Clk TB: Line>Rectangle ** or on the CB: rec
CB: 0,0 ** lower left corner at the origin
CB: 277,190 ** end point, upper right corner
```

The rectangle is now displayed on the Drawing Area with the red circle of the relative zero point at the end point. To fill the entire DA:

```
CB: za
```

The border is ready but I suggest that to add 2 mm ticks at 5 cm spacing around the bottom and left side of the border (x and y axes, so to speak) to be able check the accuracy of the printout with a ruler. We need six ticks at the bottom – starting at 0,0 – and four along the left side. I am not going to repeat commands for them all, but here are the two at 0,0:

Tick at the bottom:

```
Clk TB:  Line>2 Points ** or CB: l,li,line
CB: 0,0
CB: 0,-2
```

See below for use of the Move/Copy tool

Press [**Esc**] or right-click on the DA to end the task.

Tick to the left:

```
Clk TB: Line>2 Points ** or CB: li
CB: 0,0
CB: -2,0
```

Press [**Esc**] or right-click on the DA to end the task.

CB commands for a 2-point line are **l, li**, or **line**. I do not like "l" because it can easily be mixed with 1 (one), therefore I will use li.

Did a tick go wrong? If so, Clk TB: **Selection pointer**, Clk on the tick in case and press [**Del**]. That deletes it and you can make a new attempt.

NOTE: I am not going to repeat the “Press [**Esc**] or right-click on the DA to end the task” any more.

To highlight the ticks, we can add lines outside them (starting at -2,-2):

```
Clk TB: Line>2 Points ** or CB: li
CB: -2,-2
CB: 277,-2
```

and

```
Clk TB: Line>2 Points ** or CB: li
CB: -2,-2
CB: -2,190
```

The line around the “paper” is ready and we can turn our attention to the title block. I will use only the CB: **li** command (no Clk **TB**).

Draw the title block

I am aiming at a title block of 110x40 mm (relatively small one) located in the lower right-hand corner of the border area, with 10 mm high text cells. The first line will be vertical, starting at 277-110 = 167 mm (i.e., start at 167,0 and end at 167,40). The next will be horizontal and start at 167,40 and end at 277,40:

```
CB: li
CB: 167,0
CB: 167,40 ** vertical line finished
CB: 277,40 ** horizontal line finished
```

We have now drawn the top and left hand borders of the title bar, the bottom and right-hand limits are formed by the page border.

Next we draw a vertical line 30 mm from the left-hand border (at 167+30 = 197 mm):

```
CB: li
CB: 197,0
CB: 197,40
```

And another vertical line 30 mm from the right-hand limit (at 277-30 = 247 mm):

```
CB: li
CB: 247,0
CB: 247,40
```

We still need three horizontal lines: one starting at 167,10 the next at 197,30 and the last at 247,20. All end at the right-hand border:

```
CB: li
CB: 167,10
CB: 277,10
```

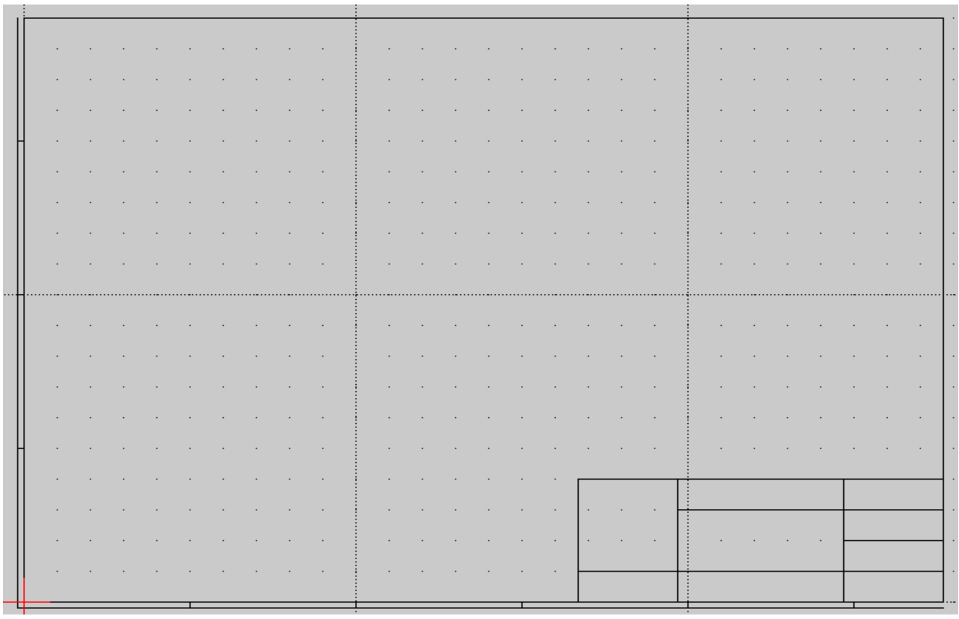
```
CB: li
CB: 197,30
CB: 197,30
```

```
CB: li
CB: 247,20
CB: 277,20
```

And the title bar is ready as far as lines are concerned.

It may be time to save the beauty now, just in case. I name it “A4_L-template” (L for Landscape) and save it in my Template folder.

And here is what it looks on the Drawing Area:



Are the dimensions right; for instance, is the rectangle really 277 mm wide? We can measure the width e.g. by first selecting **Snap on grid** and **Snap on Entity**, then Clk TB: **Info>Distance Point to Point**. Put the cursor exactly on the left border, Clk, move precisely horizontally to the right border and Clk. You can now read the distance on the Command Line (and also see the error you made in the horizontal movement).

I have to admit that I begin to like the Command Line. Entering the commands were at least in this case fast and painless, and they only required me to do some arithmetic in my head.

Add text to the title block

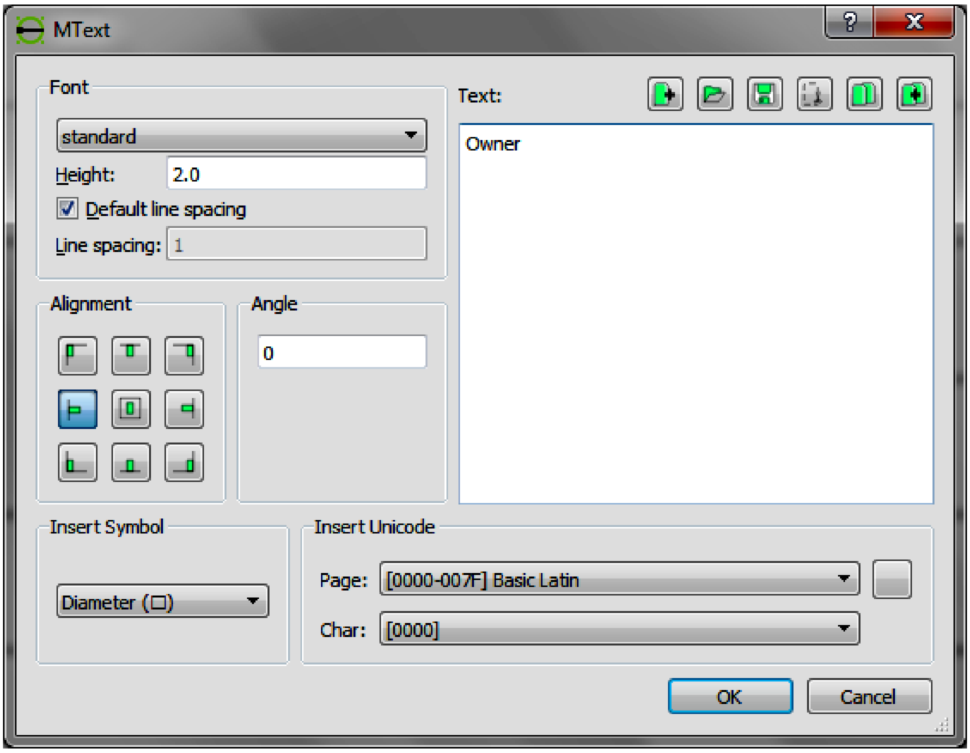
The text we are adding now is the clarification for each cell in the title block. The specific details will be added later while drawing.

Above under Preparations I selected Text Height to 2 mm, and this was with the foreknowledge that I was going to have 10 mm high cells in the title block (apart for two larger ones).

The method for inserting the text into the cells is the same in each case. This one is for the large cell in the upper left-hand corner, which is intended for the logo of the firm that has produced the drawing – the legal owner of the drawing. You start by **deselecting all Snap buttons**. Then:

```
Clk TB: Mtext ** or Text for a single line
```

and the **MText window** opens (with MText you can write multiple lines):



Font: standard

Height: 2.0

☑ Default line spacing

Line spacing: 1

Angle: 0

Text: Owner

Insert Symbol: Diameter (⌀)

Page: [0000-007F] Basic Latin

Char: [0000]

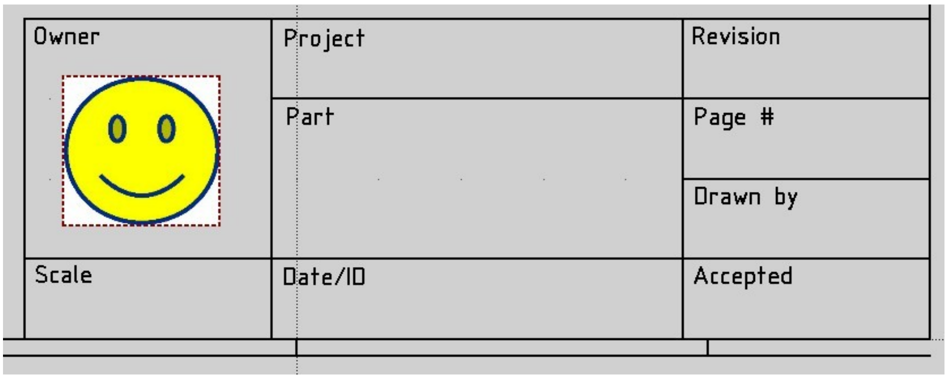
Regardless of the earlier 2 mm text specification, the default on the MText window is 2.5 mm and we have to change it.

What the “standard” **Font** means is an open question, let’s leave it at that.

The **Alignment** selections do not matter when no snap mode is selected.

Next write the text (above: Owner) and Clk [**OK**]. Bring the cursor to the point where you want to drop the text and leftclick. The text snaps to the right of the cursor (you will have to do it a couple of times to drop the text in the right place).

Repeat for all title block cells and by the time you are finished the result should be something like this:

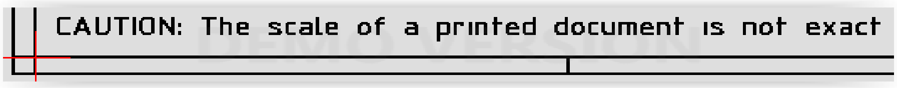


Owner	Project	Revision
	Part	Page #
		Drawn by
Scale	Date/ID	Accepted

Smiley is the “company logo,” a PNG image that I imported using Clk TB: **Insert Image** and scaled down to a suitable size (in this case to 0.06). The problem is to drag the image to its final position as LibreCAD’s drag-

and-drop function is a bit odd. I found out it could be done by drag-and-drop and then entering CB: **za** to see where the image ended up.

I also added the following note about printouts to the lower left-hand corner:



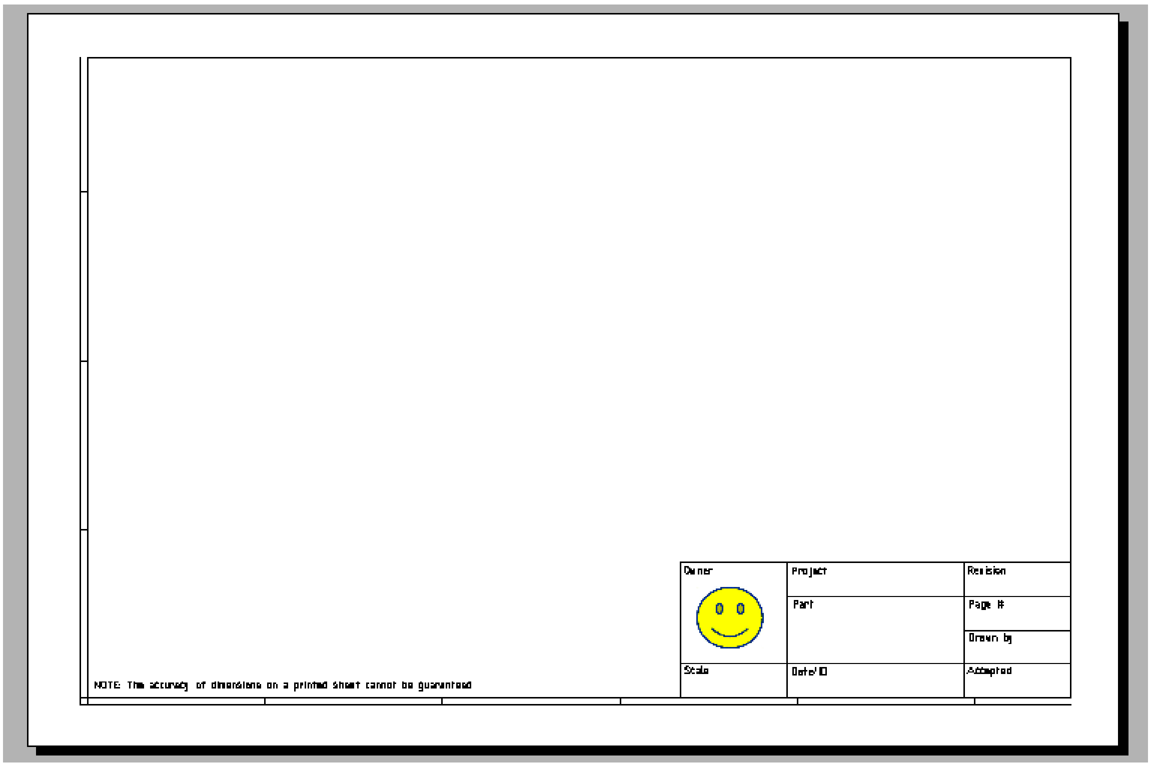
CAUTION: The scale of a printed document is not exact

Printing the template

It is time to make a hard copy of the template to see that everything is OK.

```
Clk: File>Print Preview
```

On the Drawing Area the A4 template looks as shown below – promising if you ask me, since lines show up and the borders look symmetric in Print Preview:



NOTE: The accuracy of dimensions on a printed sheet cannot be guaranteed

The next step is to export the drawing (template) as a PDF file and print it.

Here it is said

NOTE: Drawing the ticks is simplified if you first draw the ones at (0,0) and copy-paste them using

```
Clk TB: Modify>Move/Copy
```

You specify the reference point (0,0) and the target point (50,0 for horizontal ticks), and finally, on the Move/Copy Options window, the number of copies to make. The process will be demonstrated in the later exercises.

Exercise #2, cut cone with thread

The cut cone is a popular drawing exercise. In all its simplicity it gives insight into projections, dimensioning, helplines, scaling, and more. We’ll now draw a cut cone that is 40 mm high, has a base diameter of 30 mm, and top diameter of 20 mm.

To make the exercise a bit more challenging we can add a ϕ8 mm thread through it and partly cut up the drawing to expose the inside – that will need hatches.

The cone is rather small so we scale up the drawing to the **ratio 2:1**, which still allows us to draw on the template from Exercise #1. The next scale offered by ISO, 5:1, is too big for the A4 format.

NOTE 1: Drawing on a fixed paper format is “unorthodox” in CAD work, as said earlier, but let’s do it and see what the result will be.

NOTE 2: Scaling is done clumsily here; check Exercises #5 for a better solution. The other alternative method is discussed in Exercise #6.

Preparations

Begin by opening the A4 template. Save it e.g. as “A4_L-cut-cone” to avoid messing with the template.

Scaling means that once the cut cone is ready, I’ll have to move it outside the A4 template, **scale down the template** to 1:2 (i.e. 0.5), and then move the drawing back again.

The next step is to create layers on the Layers List. We can expect to need the following layers in addition to the template’s Layer 0.

- **Construction Layer** for help lines: Color: Dark Green, Line Width: 0.5 mm, Line type: Continuous
- **Edges:** Color: B/W, Line Width: 0.5 mm, Line type: Continuous
- **Dimensions:** Color: Blue, Line Width: 0.25 mm, Line type: Continuous
- **Center/symmetry lines**: Color: Red, Line Width: 0.18 mm, Line type: Dash-Dot
- **Hatches:** Color: Dark Grey, Line Width: 0.35 mm, Line type: Dash

I tried with a 0.18 mm green dotted line for the Construction Layer but found it to be virtually invisible on my 15.6” laptop screen. There is a problem here: Drawing standards may require one thing, what our computers can display is another.

Next the Application Preferences. There is not really anything to do there, the defaults are OK provided the units are in **millimeters**.

Drawing Preferences. Check the following:

- **Paper:** A4 / Landscape

- **Units:** Main Drawing Unit: Millimeter, Length Format: Decimal, Length Precision: 0.000, Angle Format: Decimal Degrees, Angle Precision: 0.00
- **Grid:** Show Grid: Orthogonal Grid
- **Dimensions:** General Scale: 1, Text Height: 3.5 mm, Decimal Separator: Comma
- **Splines:** No changes

Comments to Drawing Preferences>Dimensions:

1. I have no idea what General Scale means and it is best to leave it at the default value
2. According to requirements that I have seen, the height of general text should be 3.5 mm, but LibreCAD does not offer this option. Let’s see how it works out
3. I live in a country where decimal comma is used and want to check how it works in LibreCAD

Working with the Construction Layer

The Construction Layer used to be called Help Layer for a reason – with it we can create help (or guidance) lines that do not show up when printing the drawing. The lines are helpful because once they are in the right position, you can snap your drawing lines to them.

The first Construction line we have to draw in this exercise is the horizontal **center line** for the cone. It determines where the other horizontal lines have to be.

The drawing will be small compared to the A4 template (height 30 mm vs. 190 mm) and we don’t have to worry about where we locate the drawing. Say we put the center line at 110 mm.

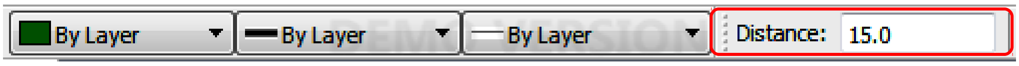
Make sure the **Construction Layer** is selected on the Layer List, then draw the line:

```
Clk TB: Line>Horizontal ** or CB: hor, horizontal
CB: 0,110
```

The x value (0) is irrelevant since the Construction line is infinitely long, but LibreCAD insists on a value. The green line should now show up across the DA.

Then we need two offset lines, one 15 mm above the center line, the other 15 mm below. These are the maximum vertical extensions of the cone. We start with the one below (this is the method used by David Brus in his videos):

```
Clk TB: Modify>Offset
Clk on the green center line
Clk [Enter]
CB: 15 ** or use Distance window (see below)
Clk on the DA below the center line
Clk [Esc]
Clk with Selection Pointer on the offseted line
```



By Layer | By Layer | By Layer | Distance: 15.0

The new line shows up 15 mm below the first one (the center line), or at x = 110 – 15 = 95 mm.

We could repeat this process and click above the center line to create the second line. However, I am not sure this is the fastest method so let's do the second offset line by creating a new line above the center line (I include the [**Enter**] and [**Esc**] operations for fair comparison with Brus's method):

```
CB: hor
Clk [Enter]
CB: 0,125
Clk [Enter]
Clk [Esc]
```

That's it. I prefer this method even if I have to compute 110 + 15 mm in my head (check however at the end of this exercise for a discussion on the **Parallel** tool).

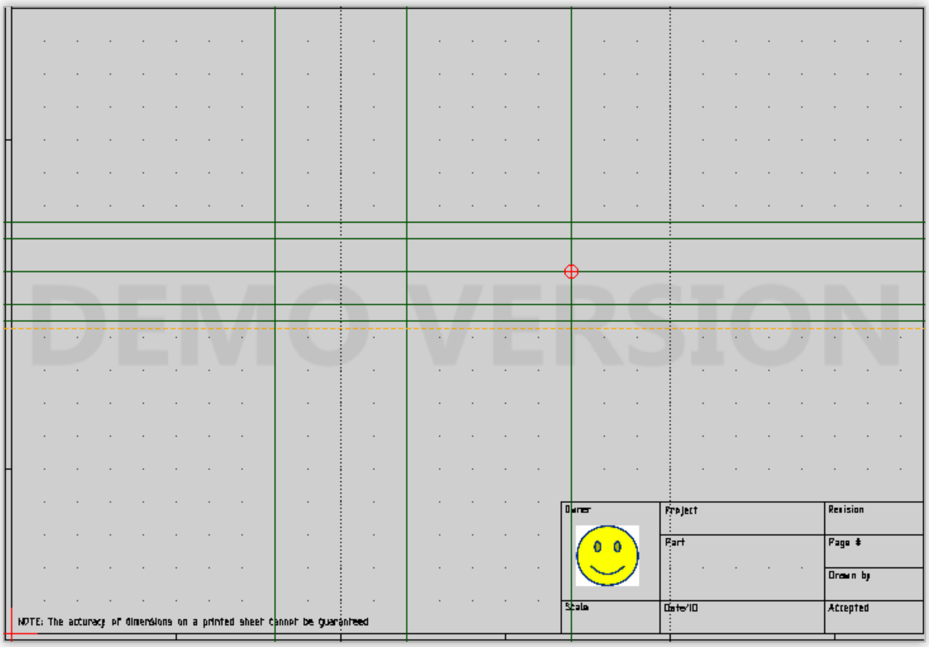
The top of the cone should have a diameter of 20 mm, so we need two more horizontal lines, one 10 mm above and the other 10 mm below the center line:

```
CB: hor
CB: 0,100 ** below center line
CB: 0,120 ** above center line
```

We also need a few vertical help lines: One at each end of the cone and at least one that points out the center of the projected cone. The height (/length) of the cone should be 40 mm, that makes two vertical lines 40 mm apart, and assume that the center of the projection is 50 mm from the side view. Then let's say we draw the first vertical line at x = 80 mm and use the (irrelevant) y value 110:

```
CB: ver
CB: 80,110
CB: 120,110
CB: 170,110
```

The DA now looks as shown on the next page. Not that the red circle of the relative coordinate is located at the center of the circular projection (you can see the magnified center part on the next drawing):


DEMO VERSION
Owner
Project
Revision
Part
Page #
Drawn by
Scala
Date/ID
Accepted
NOTE: The accuracy of dimensions on a printed sheet cannot be guaranteed

Draw the outer edges

It is time to zoom in on the small area that we will work with. We select to the **Edges Layer** and pick **Snap Intersection**.

```
Clk TB: Polyline>Polyline ** or CB: pl
```

Next, go a full circle around the edges of the cone, clicking at each corner. Clk [**Esc**] to end the session. Click off the Construction Layer to check that the cone is OK, then click back on.

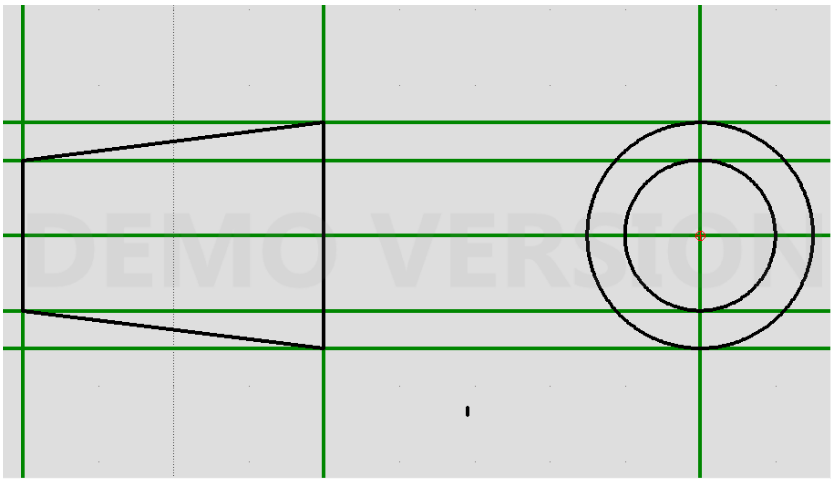
Then to the circular projection (two circles, at 20 mm and 30 mm):

```
Clk TB: Circle>Center Point ** or CB: ci
```

Put the cursor at the center point of projection, click and drag out to the first horizontal line (20 mm diameter). Click to snap the circle, Clk [**Esc**].

Repeat for the 30 mm diameter circle.

The drawing now looks as shown below (ISO projection):



Adding details

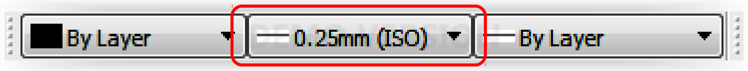
We can limit screen clutter by deleting some help lines. Click off all except the horizontal and vertical center lines.

We need however two new help lines for the ϕ8 mm thread: One at 8 mm diameter, one at the borehole edge – say ϕ6 mm. That means horizontal lines at 110-4 = 106 mm and 110-3 = 107 mm. We draw them only below the center line because we do not cut the upper half of the cone. Remember to shift to **Construction Layer**:

```
CB: hor
CB: 0,106
CB: 0,107
```

Next we'll draw the outline of the thread. Shift to **Edges Layer**, click off Snap Intersection and click on **Snap on Grid** and **Snap on Entity**. We can now draw a Polyline along the edges of the thread (see screenshot below).

Decrease the **line width** to 0.25 mm as shown below and draw a line along the 107 mm thread line from end to end of the cone. This is the "bottom" of the thread.



Then to the circular projection. Click off Snap on Entity and click on **Snap Intersection** and increase **line width** to 0.5 mm. Draw the ϕ6 mm hole as outlined above.

Decrease the **line width** to 0.25 mm, then:

```
Clk TB: Curve>Center, Point, Angles
```

Put the cursor to the center of the circular projection and draw the ϕ8 mm thread outline, manipulate to leave an open sector in the circle.

Hatches can only be added to closed surfaces. Since our cone has been draw in bits and pieces, we have to add a polygon on top of the existing lines. We do that in the Edges Layer with 0.25 mm line width (as above). Click **Snap on Grid** and **Snap to Endpoints**, and

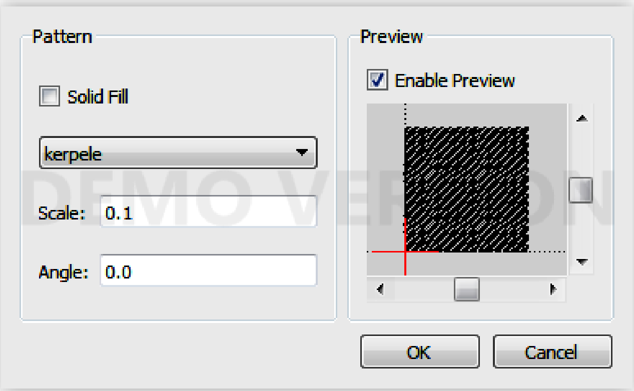
```
Clk TB: Polyline>Polyline ** or CB: pl
```

Draw the polyline by clicking on each corner of what is to become a closed surface. Finish by clicking [**Esc**].

Click on the drawn polyline (now a closed polygon), then

```
Clk TB: Hatch ** or CB: ha
```

and the window to the right pops up. Select "kerpele" from the **drop-down menu** and decrease the **Scale** to 0.1 as shown in the figure on the next page. Enable also the Preview.



Pattern

□ Solid Fill

kerpele

Scale: 0.1

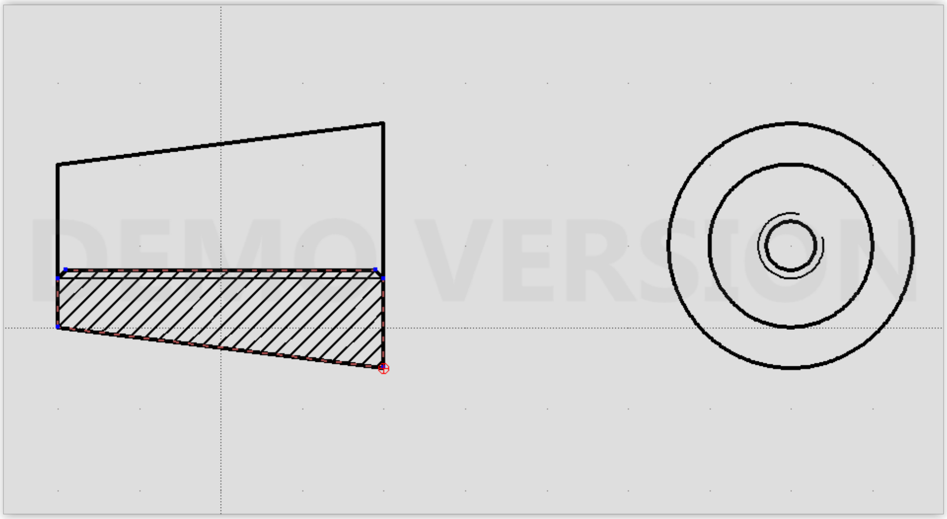
Angle: 0.0

Preview

☑ Enable Preview

The Preview window is small and it is difficult to see details at this Scale, but go ahead and Clk [**OK**] and the selected area will be hatched.

The drawing looks as shown below when I have clicked off the Construction Layer:



We have still to add the center/symmetry lines before we can a) add dimensions and b) fill in the title block.

Center line for the cone to the left. Remember: shift to **Symmetry_line Layer** and the **hor** and **ver** commands for the Construction Layer have to give way for **li**:

```
CB: li
CB: 70,110
CB: 130,110
```

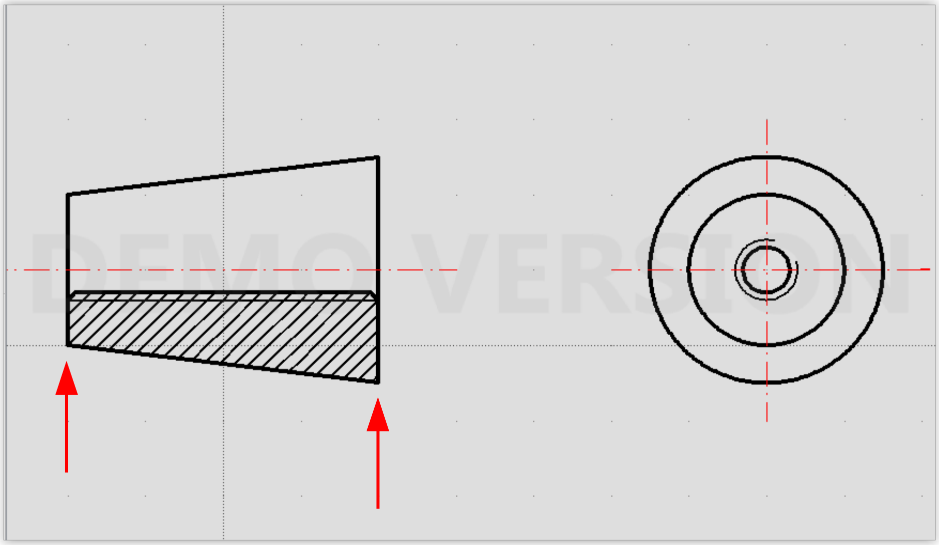
Horizontal symmetry line for the projection:

```
CB: li
CB: 150,110
CB: 190,110
```

Vertical symmetry line:

```
CB: li
CB: 170,90
CB: 170,130
```

The drawing now looks like this (arrows added):



Dimensions

We add dimensions to the side view. Activate the **Dimensions Layer**, **Snap on Grid**, and **Snap on Endpoints**. Then:

```
Clk TB: Dimension>Horizontal ** or CB: dh
```

Click at the lower left-hand corner of the cone (red arrow in figure above). Move to the lower right-hand corner and click. Then drag the cursor down and you can see the dimension with its arrow following. Click to lock it when you are satisfied with its position. Clk [**Esc**] to end the task.

The dimension should now be exactly 40,0000 millimeters. Too many decimals for good engineering practice, but let it be for now. If the dimension is not exactly 40 mm you have probably selected the wrong snap tools.

We can repeat this exercise for the cone diameters using the same settings.

The **leader** for the ϕ8 mm thread is initiated with the following command:

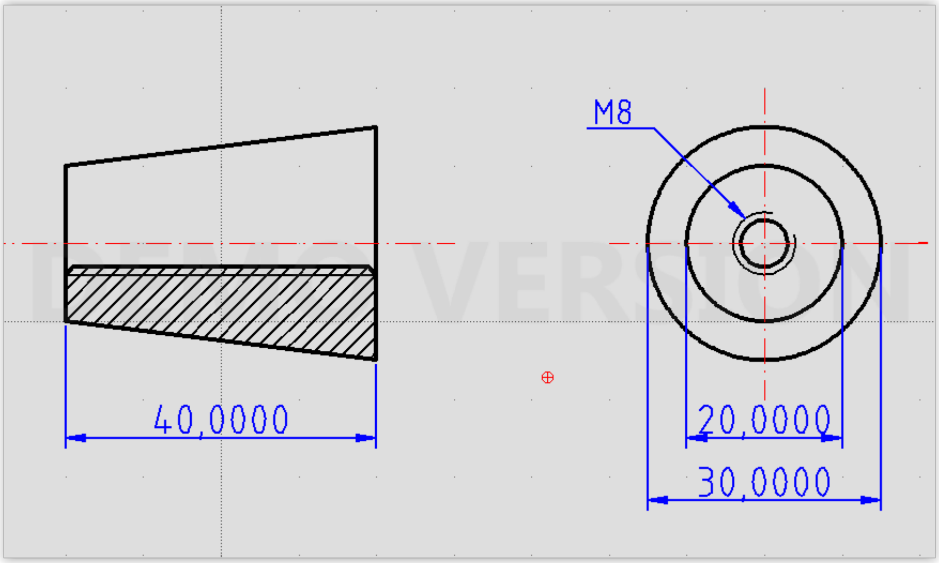
```
Clk TB: Dimension>Leader ** or CB: ld
```

You start from the thread and finish with the horizontal part for the text. LibreCAD has a limitation however: there is no way to automatically generate the leader text, it has to be done separately and snapped above the horizontal leader line. **Click off all snap tools**, then

```
Clk TB: MText ** alt CB: mtxt
```

Write the text as outlined in the previous exercise, Clk [**OK**] and move the cursor above the leader. Left-click to snap it to the drawing and Clk [**Esc**] to finish.

The drawing is now as ready as we want it to be (below), after this we could turn our attention to the Title bar but I prefer to increase the **Scale** of the drawing first.



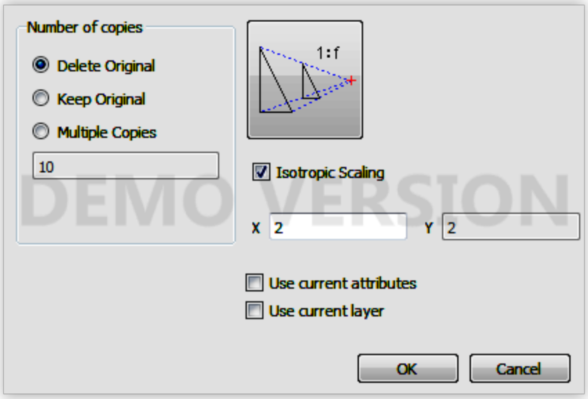
Changing scale from 1:1 to 2:1

Before attempting the scaling operation it is wise to save a copy of the drawing under another name – just in case Murphy strikes.

The big problem in the scaling process is to select the reference point so that the drawing ends up in the right position – I had to try several times to get it right. Here are the commands I gave after I had **highlighted** the drawing using the Selection Pointer:

```
CB: sz ** or Clk: Modify>Scale
CB:130,90
```

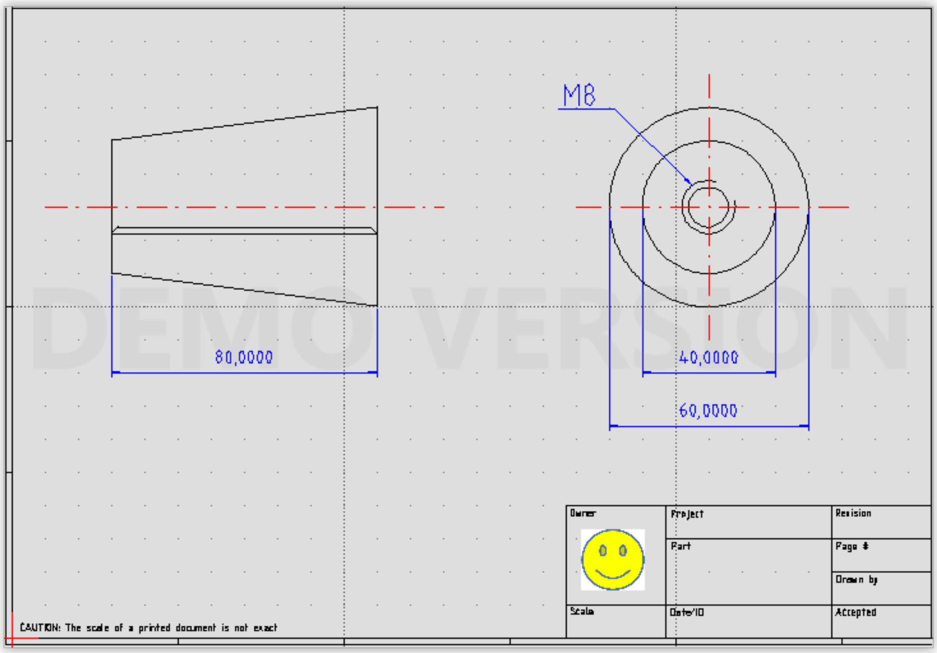
The Scaling Options window to the right pops up. Select the shown alternatives and Clk [**OK**].


Number of copies
● Delete Original
○ Keep Original
○ Multiple Copies
10
☑ Isotropic Scaling
X 2 Y 2
□ Use current attributes
□ Use current layer

The drawing stays highlighted (selected) also after scaling. To deselect,

```
Clk TB: Select>Deselect all
```

And below is the drawing in its present state:



There are (at least) three details wrong with the scaled drawing:

- The hatches have disappeared
- The dimensions have been doubled
- The leader text height has been scaled up

The doubled dimensions can be corrected by changing **Length Factor** under Drawing Preferences>Dimensions form 1 to **0.5**. At first the Caution in the lower left-hand corner also disappeared but then I noticed I had drawn it in Edges Layer instead of the 0 Layer. When I changed it the text strip stayed there.

A check with Github, sourceforge.org, and LibreCAD's discussion forum revealed that other people have had problems with hatches as well, but the only formally reported problem is hatches disappearing when moving a drawing (has occurred to me as well). Obviously this is a bug and we can only redraw the hatches in the new scale.

Let's however try the other scaling option: scaling down the paper. Begin by zooming out the DA enough to have room to pull out the drawing from the A4 template and save it temporarily. Then **select** (highlight) the drawing with the Selection Pointer and

```
Clk TB: Modify>Move>Copy
Clk on the drawing (e.g. in the middle)
```

Drag the drawing to the side of the template

Left-click to snap the drawing to the position

Select "Delete Original" and Clk [**OK**]

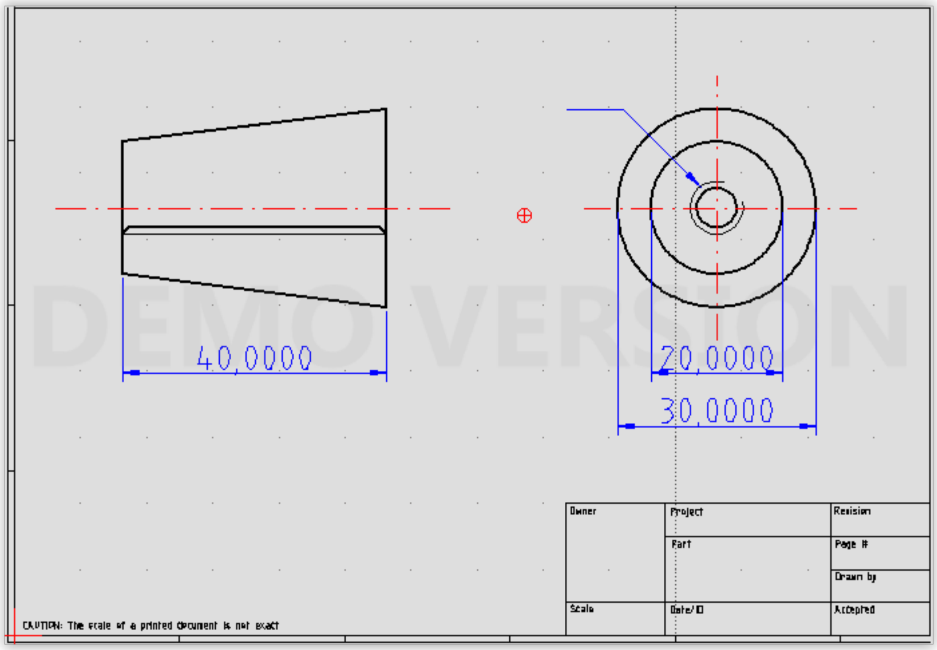
```
Clk TB: Select>Deselect all
```

The drawing is now outside the template. Next, **select** the template and

```
Clk TB: Modify>Scale
CB: 0,0 ** and press [Enter]
```

Select “Delete Original” & Scale to 0.5, Clk [**OK**]

The template shrinks to half the size. Then repeat the above “recipe” and move back the drawing on the template, after which the drawing looks as shown below:



There are problems this time as well:

- The hatches are missing
- The leader text is missing
- The “logo” is missing (it had jumped far from the drawing)
- The text height of dimensions are scaled up
- Entering CB: **za** revealed small “leftovers” to the side of the drawing

Question: Does one need to move the drawing outside the template? You should try to find out.

We could fix the text height by changing **General Scale** on Drawing Preferences>Dimensions, from 1 to **0.5**, but the rest of the problems would have to be corrected manually. My conclusion is that it is better to stick to the first method and scale down the drawing. Then to work and do the necessary corrections – this time drawing Hatches in the Hatches Layer.

Let’s add text to the Title bar before looking at the final drawing.

Completing the Title bar

The Title block is just another drawing entity, not a macro where we could easily fill in empty fields. We therefore have to create and insert text strings one by one into the title bar cells.

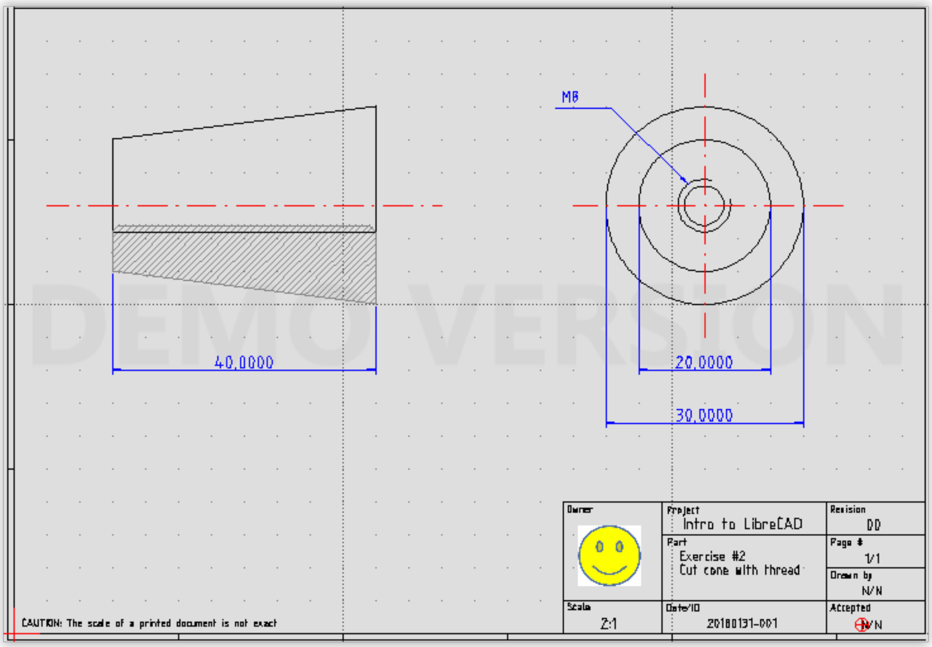
We do not have a Layer for text so we can use the Edges Layer to get a black text. That is what I have done. However, the text becomes quite fat so one should perhaps change the line width to 0.35 mm.

The process of adding test was outlined in Exercise #1, but to repeat the essentials:

```
Clk TB: MText ** or Clk MB: Tools>Text
```

Write the text, Clk [**OK**], drag the text to the right location, left-click.

The drawing is now finished and looks like this (I forgot to switch off the grid):



Summary of observations form Exercise #2

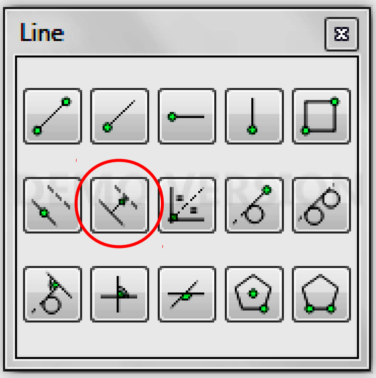
This was a rewarding exercise and here are the main “lessons learned”:

- I had defined the text height for Dimensions to 3.5 mm but LibreCAD delivered only 3 mm
- Decimal commas worked as ordered
- Having different colors for different Layers helped me see that I was working on the right layer – one easily forgets to switch to the right layer
- I made the mistake of drawing Hatches in the Edges Layer, although I had defined a Hatches Layer – an example of what was said above
- Only dark colors have sufficient contrast on my light grey Background, but even so I prefer it to a black background
- The importance of the CL cannot be stressed enough, entering commands on the CB is mostly to prefer over using the TB
- Finding the right Snap tool for the current situation requires trial-and-error but experience helps
- Zoom in and out as needed while drawing – it is frustrating to try to make changes to an item that can hardly be seen on the screen
- The Draft Mode button causes items to “disappear” when pressed

- Two-point lines (CB: **li**) was more suitable for drawing than Horizontal (CB: **hor**) and Vertical (CB: **ver**) lines – for now at least
- LibreCAD's scaling function did not work to satisfaction as applied here and there is a bug that prevents Hatches from being scaled
- Scaling a drawing gives problems, one should stick to the rule of drawing in scale 1:1 and later worry about the document size

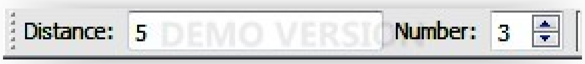
Speeding up: Copy-pasting with the Parallel button

METHOD 1, the complicated way:



An alternative to copy-paste using Modify>Move/Copy is to activate the **Line Panelette**, by clicking on the Menu Bar **Widgets>Dock Widgets>Line**. It opens the window shown to the right (I have compressed it here).

The **Parallel** pushbutton we are interested in is circled. Click on it to open the Tool Options Bar shown below:


Distance: 5 Number: 3

Assume that you have drawn a vertical line. Click on it, then click on the Parallel button and insert the **Distance** (mm) and **Number** on the Tool Options Bar as shown above. When you now hover the cursor on either side of the drawn line, you will see three parallel lines emerge to the side of the original line. Left-click to snap the lines to the drawing. You have then four vertical lines with 5 mm separation.

If you have also drawn a horizontal line, you can just move the cursor above or below it to snap the same three lines there.

METHOD 2, the simple way:

There is no need to open the Line Panelette. You can do exactly the same operation by Clk TB: **Line>Parallel** and then proceed as told above. The pushbuttons are much clearer on the panelette however.

Exercise #3, 3D object without isometric projection

The task is to draw a simple 3D object – a wooden pen holder – without the help of LibreCAD’s isometric projection. The task is influenced by the Spanish video “Librecad proyección isométrica con líneas curvas” that you can find on YouTube. The video however deals with a much more complex 3D object.

Let’s say that the pen holder is 7×7 cm in diameter, 10 cm high, and has a material thickness of 8 mm. We cut it to show the interior but do not bother with a paper format although A4 Portrait would be the natural choice when drawn in scale 1:1.

Preparations

My guess is that three layers will suffice:

- **Edges:** 0.5 mm, Black, Continuous
- **Hatches:** 0.35 mm, Black, Continuous
- **Dimensions:** 0.18 mm, Blue, Continuous

We’ll use the following preferences:

Application Preferences

Dimensions: millimeter

Current Drawing Preferences

Paper Format: A4, Portrait (I stick to drawing on “paper”)
Units: Default values
Grid: Orthogonal Grid

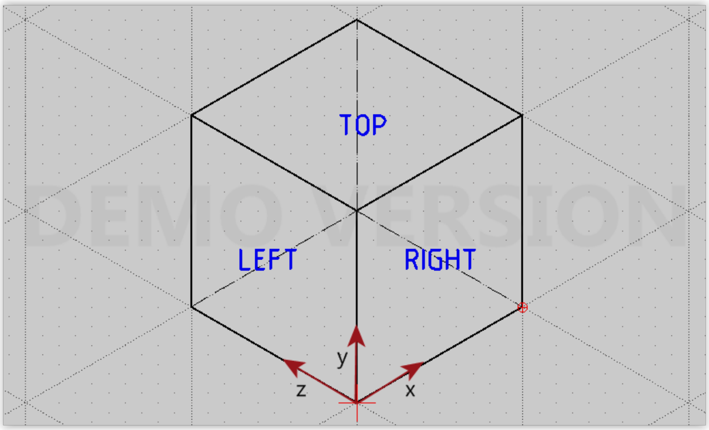
Dimensions:

Arrow Size: 0
Tick Size: 2
Linear Precision: 0.0
Angular Precision: Default

The size of an A4 Portrait sheet is 210×297 mm. If we put the lower left-hand corner at the origin, we can use 90,100 as the starting point of our drawing and it should sit quite in the middle of the printed paper sheet.

When needed, I will refer to coordinate axes and isometric drawing sides according to the figure to the right.

That should be about it.



Draw outer edges

Recall that the “horizontal” lines must be at a 30º and 150º angles (or get a 180º shift when needed), vertical lines are at 90º and 270º. We start with the right side and mainly draw by shifting the reference point using the @ prefix:

```
Clk Layer List: Edges
CB: li ** or Clk TB: Line > 2 Points
CB: 90,100 ** starting point
CB: @100<90 ** vertical line up
CB: @70<30 ** “horizontal” line on top
CB: @100<270 ** vertical line down
CB: @70<210 ** “horizontal” line to close side
```

We continue without a brake with the left side (comments not needed):

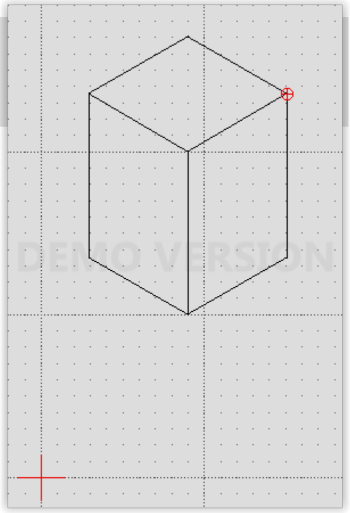
```
CB: @70<150
CB: @100<90
CB: @70<330
```

The left side is now closed, but we continue without interrupting drawing (do not click [**Esc**]). We have to move the relative coordinate point from its location at 90,200 to one of the upper corners in order to draw the back lines of the top:

```
Clk: Snap intersection ** icon on Widgets Bar
Move cursor to selected corner, Clk
```

The red circle moves to the corner (in my case the left one) and we continue:

```
CB: @70<30
CB: @70<330
Clk [Esc]
```



The outer edges of the pen holder are now ready, as shown to the right.

If you had problems moving the relative coordinate point, the reason is most likely that you did not put the cursor close enough to the corner (happened to me).

But somehow I am getting the feeling that my estimate about the location on the paper may not be right. Never mind.

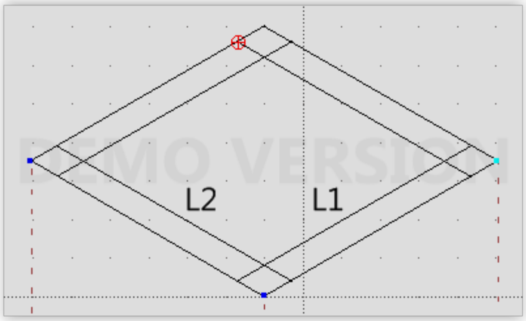
Draw edges for the hole

The wooden block needs to be hollowed out if it is to be pen holder. We need on the top a rectangular shape, 8 mm inside the outer edges of the top surface. We therefore copy each outer edge line and move the copy 8 mm inside, then trim the extra line lengths.

I start with the edge of the left front side and activate the whole set of Snap tools: **Snap Grid**, **Snap to Entity**, **Snap Intersection** (overkill perhaps). Then

```
Clk on line to Move/Copy
CB: mv ** or Clk TB: Modify>Move/Copy
Move cursor to end of selected line, Clk
CB: @8<30
Select Keep Original, Clk [OK]
```

The first line is now copied and moved. Repeat the process for the other three lines. The top of the pen holder now looks as shown to the right.



We have to trim the end of the lines. The process was explained in Chapter 2, but let’s take Trim Two for lines marked L1 and L2:

```
CB: t2 ** or Clk TB: Modify>Trim Two
Clk: L1
Clk: L2
```

The end of both lines at the front corner disappear. Repeat the process for the other corners and the top surface is ready.

Cut and hatches

We’ll cut through the front corner of both left and right sides in order to show the interior of the pen holder. We can for instance cut off half of both sides.

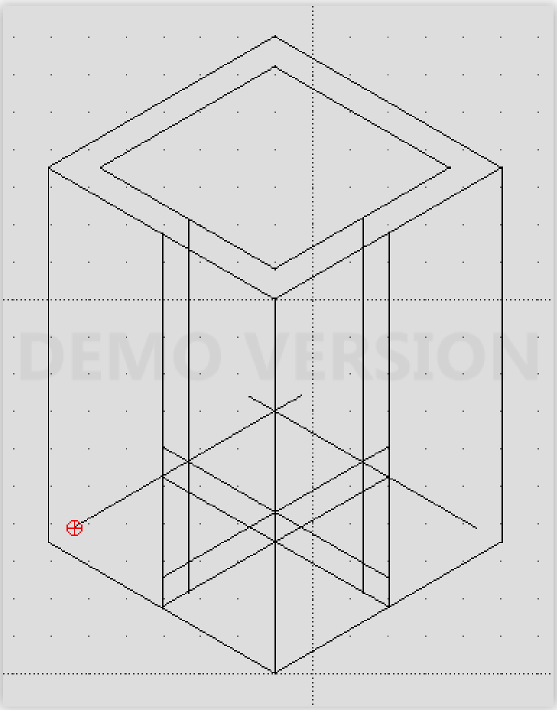
Move first a copy of the vertical front line to the left and right, halfway along the side:

```
Clk on the vertical line
CB: mv
Move cursor to the lower front corner, Clk
CB: @35<150 ** Scaling Window pops up
Select Keep Original, Clk [OK]
```

That moved to the left, so repeat to the right. Then move a copy of both the new lines 8 mm in, to form the edge of the inner surface.

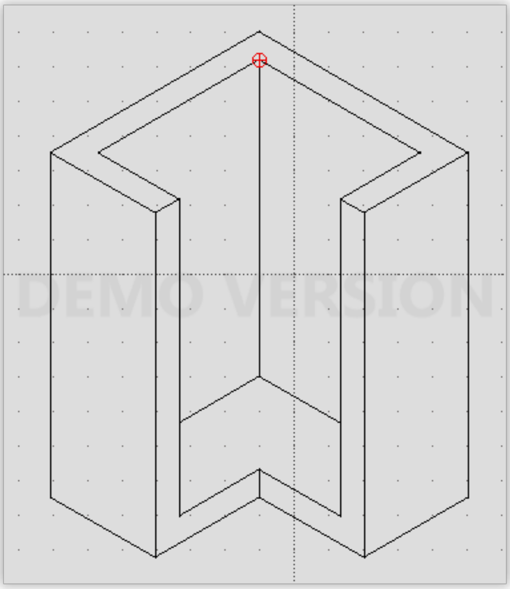
Next move a copy of both lower “horizontal” lines 35 mm in, then a copy 8 mm up. These are the outlines for the bottom of the pen holder.

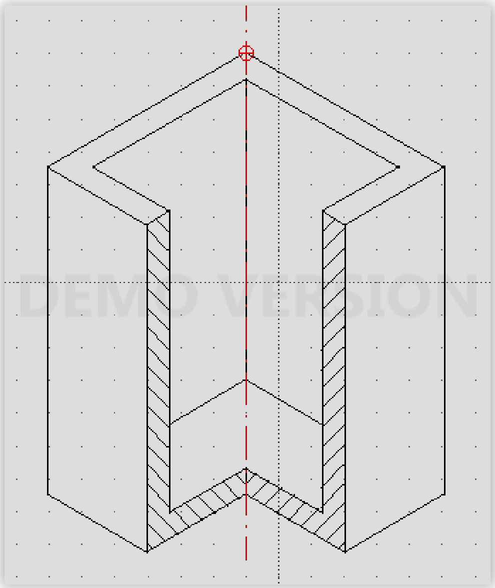
Before starting to clean unnecessary line parts we can outline the “floor” of the pen holder by moving a copy of the internal bottom lines 27 mm (35-8 mm) back.



The drawing now looks as shown to the right. There are plenty of lines to trim, and a few to add, before we can add hatches.

We have discussed both trimming and hatches before so I don't go into the details. The cleaned drawing with addition of some missing lines is shown to the left below. At this stage it occurred to me that the drawing is in fact symmetric, so I added a **Symmetry Layer** with 0.25 mm red Dash-Dot (small) line. With that change and addition of hatches, the drawing is shown below to the right.

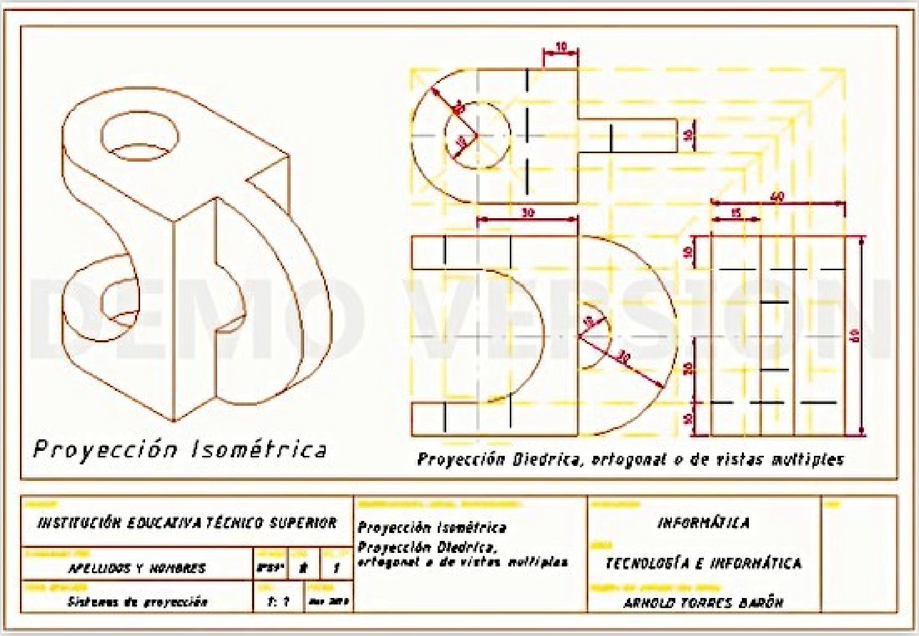




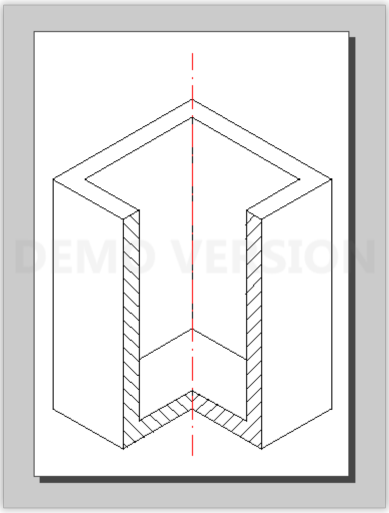
The question of dimensions

Should we add dimensions to the drawing or not? The answer is no, it is an isometric drawing on an orthogonal grid. The dimensions are up to interpretation due to the twisted nature of the drawing.

Isometric drawings on orthogonal grids are for information only. Look below at the screenshot of the earlier-mentioned Spanish video. The accurate orthogonal projections are to the right, the dimensionless isometric drawing to the left:

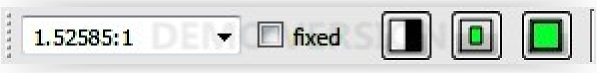


Printing



How well did my guess about the location of the drawing on an A4 Portrait paper go? Not too bad if we can believe the Print Preview shown to the right. But the drawing is far bigger than it should be.

LibreCAD adjusts the drawing to the paper format. This is revealed in the Print Preview window next to the Layer Selection area:


1.52585:1 □ fixed

Change the scale to 1:1, click the **Center to Page** button, and the printout should be quite OK – even if it doesn't work on my Win7 laptop. Nothing is sent to the printer – a bug.

We have to export the drawing to PDF for printing, but remember to click the **Toggle Black/White Mode** button while in Print Preview to change all colors to black. Other colors do not show up well on a B/W printer.

Discussion

This exercise was before all about working with relative coordinates, angles, copying and moving lines, and trimming. The work with relative coordinates using angles is fast and accurate, but requires that we have a good mental picture of how to proceed.

Exercise #4, rack in isometric projection

The task is to draw in isometric projection a rack (an IKEA bookshelf or whatever) that is 600 mm high, 400 mm deep, and 1800 mm long; with bottom, top, and an internal shelf, and divided into two equal vertical parts. The material thickness is 20 mm, with a 2 mm back plane. The drawing should be on our A4 template and show all hidden lines. Hidden lines will make the drawing messy but it is good exercise to do them.

For alternative examples of isometric drawings, see “LibreCAD users Manual” <http://wiki.librecad.org/index.php/LibreCAD_users_Manual>. The tutorial is written for an older version of LibreCAD and needs guesswork. The tutorial “Drawing a little mechanical part using LibreCAD” by Claudio Guarnieri at <http://forum.librecad.org/file/n5708338/mechanics_tutorial.pdf> looks interesting, but his explanations are sketchy and the good attempt is of no use.

Preparations I: Scaling up the Template

The first preparation is to scale up our A4 template to match the up-and-coming drawing. The length of the rack is 1800 mm, the width of the A4 sheet is 297 mm, so it should be OK to scale up by 10:1.

Scaling was discussed in Exercise #2, but let’s do it again since repetition is the mother of all learning. Begin by opening the template and saving it under another name (e.g. isometric_rack.dxf), then:

```
Clk TB: Select>Select All
Clk TB: Modify>Scale ** or CB: sz
CB: 0,0 ** lower left corner at 0,0
Select: Delete Original, Isotropic Scaling, X 10
Clk: [OK], [Esc]
```

You can now measure the width of the A4 drawing area to see that it is 2800 mm, which leaves 500 mm (minus one end) on each side of the rack free.

What about the height of the drawing? To compute along the y-x-z edges:

Error

Height = 600+1800×sin(30º)+400×sin(~~60º~~)≈1846 mm

The available space is 1900 mm. It is going to be tight, there will be only 27 mm free space to and bottom. We’ll have to move the CAUTION text in the lower left-hand corner of the drawing to above the Title bar (deactivate all Snap tools to be able to move the text to an exact position).

Preparations II: Layers and Preferences

The next important preparation is to change from Orthogonal to **Isometric Grid** in **Options>Current Drawing Preferences>Grid**. Instead of the two-dimensional (2D) x/y grid we have had before, we have now a three-dimensional (3D) **x/y/z grid** that is turned to 30º viewing angle(s). Please do repeat the discussion on coordinate systems in Chapter 2.

The coordinate angles are fixed, we cannot change them as with 3D CAD programs.

The template is on Layer 0. As for other layers, my guess is that we need five (a bit uncertainty involved):

- **Construction:** 0.5 mm, Dark Green, Continuous
- **Edges:** 0.5 mm, Black, Continuous
- **Hidden edges:** 0.35 mm, Black, Dashes (small)
- **Text:** 0.35 mm, Black, Continuous
- **Dimensions:** 0.18 mm, Blue, Continuous

For the remaining preferences we use (Exercise #5 shows a better way of mastering the scaling issue):

Application Preferences

Dimensions: Millimeter

Current Drawing Preferences

Paper Format: A4, Landscape (I still draw on “paper”)
Units: Millimeter

Dimensions:

Text Height: 20 (Scaled up 10:1)
Dimension line gap: 6.25 (Scaled up 10:1)
Arrow Size: 25 (Scaled up 10:1)
Linear Precision: 0.0
Angular Precision: Default

Drawing outer edges

My plan is to draw the entire front, then move a copy of it by 388 mm along the z axis, leaving 2 mm for the back plan that has to be nailed on. After that we can draw the interior of the rack. Here we go folks, starting in the **Edges Layer**:

```
CB: li
CB: 700,30 ** starting point
CB: @600<90 ** up along the y axis
CB: @1800<30 ** along the x axis
CB: @600<270 ** down along the y axis
CB: @1800<210 ** back to the starting point
```

There will be four openings at the front, each 870×270 mm in size. One way of drawing them would be by adding help lines (Construction Layer) and drawing using them as guidance lines. I’ll however copy and move the existing lines, then trim the extra lengths – it makes a less cluttered work space.

Let’s work vertically with the lower horizontal (x axis) line. It’s first copy has to be moved up by 20 mm, then by 270+20 = 290 mm, etc. Here is how to copy and move the first:

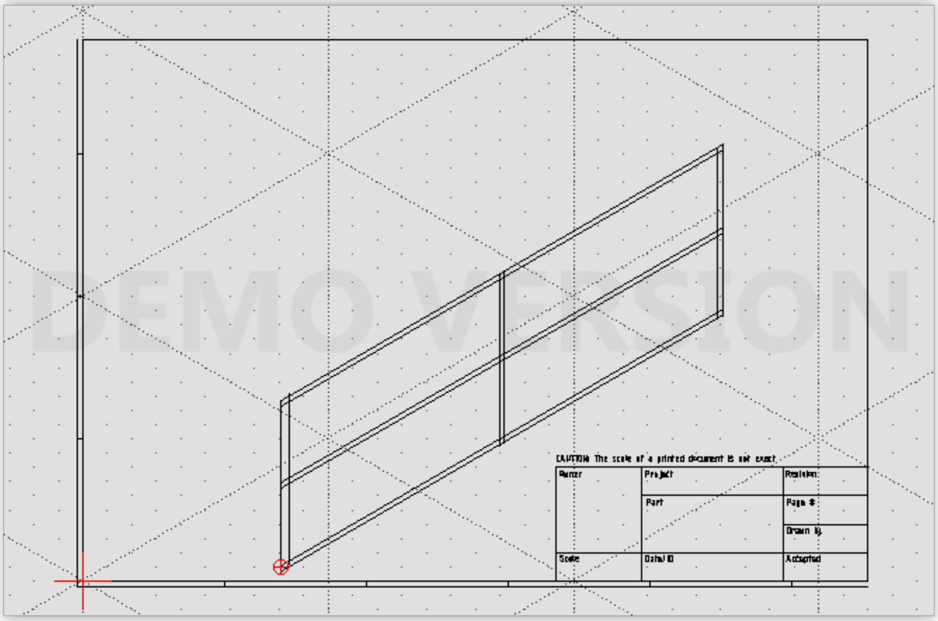
```
Clk on the line
CB: mv
CB: 700,30 ** reference point
CB: 700,50 ** target point
Select: Keep Original
Clk [OK]
```

Continue copying and moving the rest.

Next we move the vertical lines with the help of the leftmost. This time we use polar coordinates to move at an angle of 30º – the distances would not be right without involving trigonometry. Her is the first one:

```
Clk on the line
CB: mv
CB: 700,30 ** reference point
CB: @20<30 ** target point
Select: Keep Original
Clk [OK]
```

Check with CB: **za** that you have no extra lines floating around (I had). The drawing now looks as show below.



Before continuing, zoom in and check that all lines end as they should. After that there is the task of trimming line ends – we did that earlier, I am not going through it here. However, think of how IKEA would build this thing. I would say that the ends are full size (600×400 mm), It means that both their edges will show on the top of the rack. Think through the rest as well – no need to delete lines that later will have to be redrawn.

But trimming the center is challenging since two lines will have to be cut in the middle…. Beats me.

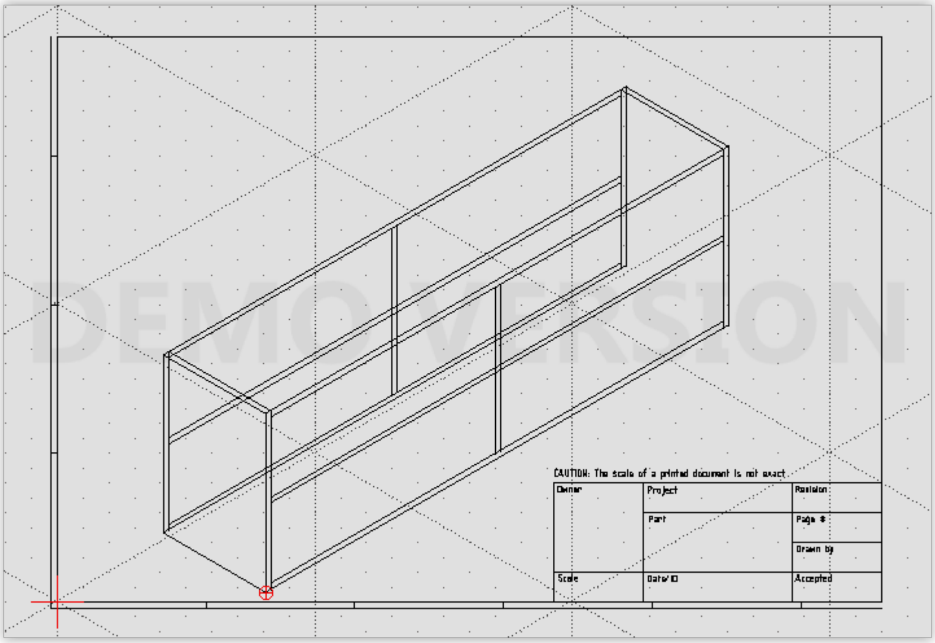
With the trimming done the task will be to move a copy of the front drawing 398 mm along the z axis. Start by clicking off the **0 Layer**, then

```
Select the drawing
CB: mv
CB: 700,30 ** reference point
CB: @398<150 ** target point on z axis
Select: Keep Original
Clk [OK]
```

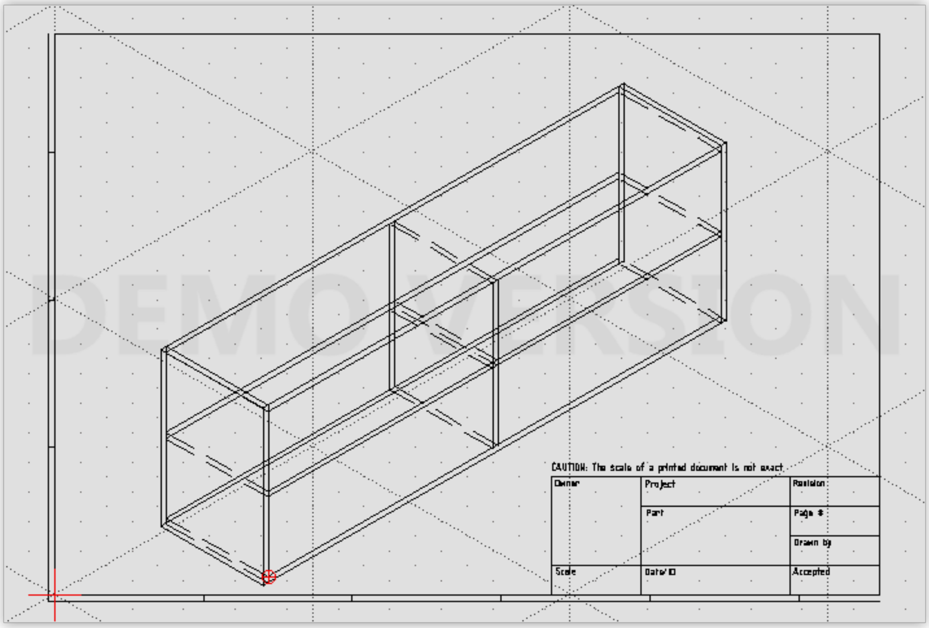
The error

And the drawing shows that I have made a **rookie error** in computing the height. I should have used 400×sin(30º) instead of 400×sin(60º). The drawing is now too close to the bottom of the A4 template. But we can see what the dimensions require and later make the decision if the drawing has to be moved up.

Add the three missing outer edges and the drawing looks like this (**0 Layer** clicked back on):



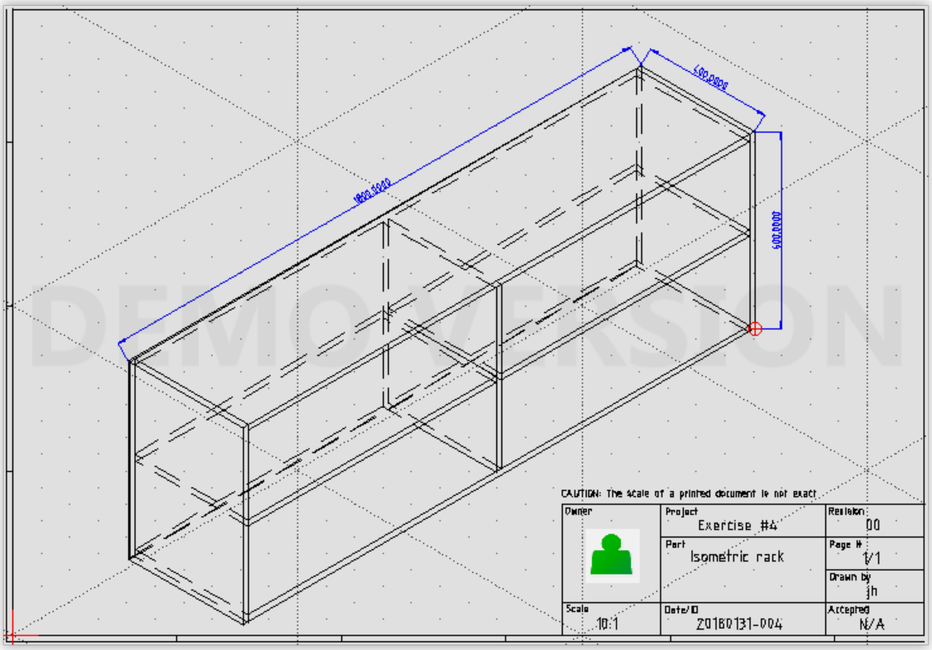
My suggestion is to now draw the remaining lines as Hidden lines (Dashes). Once they are in place there are line ends by which to delete and redraw what now are shown as visual lines. Here the dots are in place:



Then to the big cleanup: Redraw hidden edges as Hidden lines, add Edges on top of the four dotted lines that partly show, and trim whatever lines need be trimmed.

There is also the 2 mm back plane to keep in mind, even if it hardly can be seen. It can be drawn by moving the rear edges and hidden lines 2 mm along the z axis (CB: @2>150), then extending the four z-directed edges of the rack by 2 mm.

We now have this final drawing:



Error #2

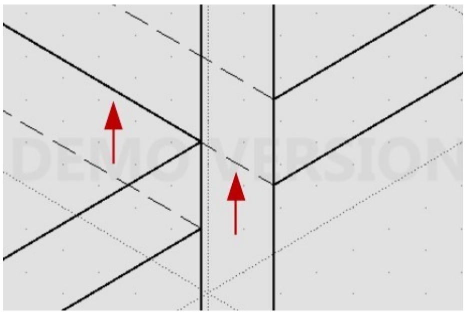
It turns out that my "rookie error" wasn't too bad after all, because I had made a **second error** in forgetting the height of Dimensions. The drawing is placed quite nicely after all.

Comments

There were lots of lines to be changed from Edges to Hidden lines (and the other way). This process could be sped up if LibreCAD allowed shifting widgets from one layer to another.

It was difficult to snap lines to the correct point. I missed a Snap alternative that would allow snapping to the end of a line, Snap to Intersection did not seem to do the trick. (Did I try Snap on Endpoints? Cannot remember.)

There will be overlapping lines with the geometry we have (equally tick parts). This can be seen particularly well at the center, where the middle upright cuts through the internal shelf (shown right).



At the start of this task I scaled up some dimensions. Actually, I scaled them up only when I added dimensions to the drawing and noticed that everything became too small – that is the trial-and-error approach at its best. I should also have changed the **Extension lines>Offset** to 6.25 as there is no obvious gap between the edges and the extension lines.

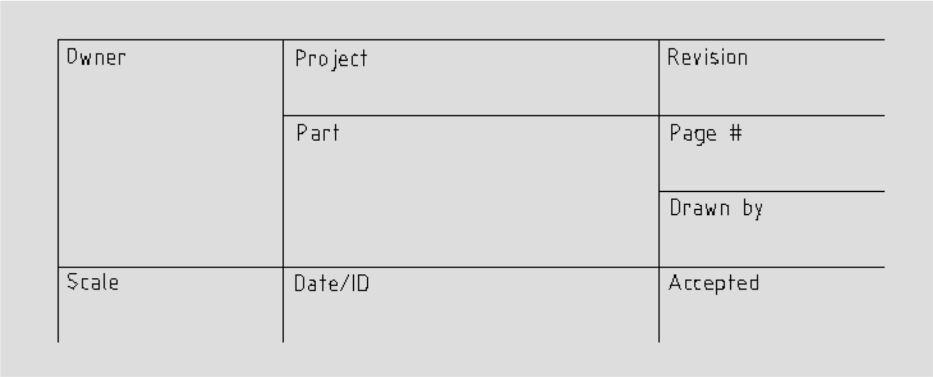
Exercise #5, flange with holes

The aim of this task is threefold: First, to look at how to work with blocks by creating a block from the Title bar of our A4 Landscape template bar and import it to an A4 Portrait template. Second, to take a simpler approach to scaling than before. Third, to draw a flange with holes using LibreCAD's rotate function.

After the two, not too successful scaling attempts in Exercise #2 and a slightly better attempt in Exercise #4, it is time to try something different: Scale the document template to fit the drawing but leave other scale definitions in their default values until we see what needs to be done.
(I am still drawing on "paper.")

Create a block

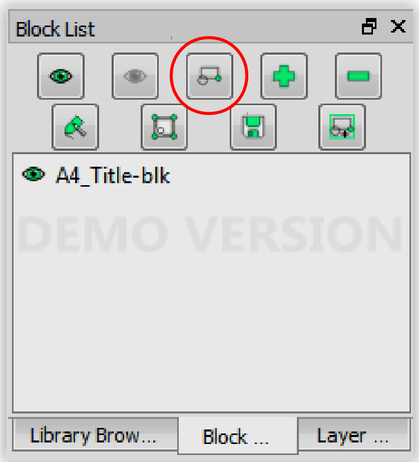
Open the A4_L-template and save it under another name, e.g. as A4_Title-blk, then delete all lines that do not belong to the title block. What remains is shown below:

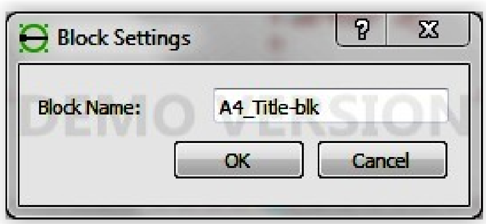


Recall that the insertion (reference) point is located at 277,0 – the lower right-hand corner of the A4_L-template's drawing area.

Select the entire title block including reference lines. Then click **Create a block** on the **Block List** (shown right), or the same icon on the Toolbar. The Control List asks for the reference point. Select 277,0 as it defines the corner that is later used to snap the block to other drawings.

The **Block Settings** window (below) opens and the block can be named – I call it A4_Title-blk, shown below and also on the Block List.





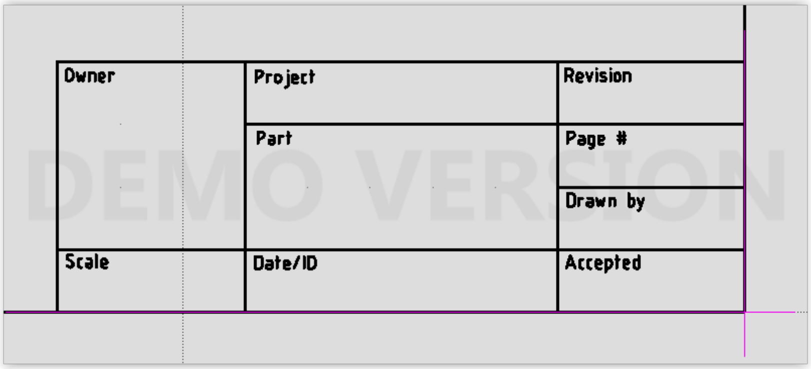
Block Name: A4_Title-blk

Next click **Save the active block to a file** button on the Block List (circled). Define the folder where to save the block and click [**OK**]. Done.

Add the block to A4 Portrait

The next step is to create the 0.25 mm frame for an A4 Portrait format drawing. It is the same procedure as in Exercise #1, only this time the width is 190 mm and height 277 mm. When the A4 Portrait frame is ready, it is time to import the saved Title block.

Select **Snap on Grid**, **Snap on Entity**, and **Snap Intersection** (that should be enough). On the Menu Bar, click **File>Import>Block**; select the block and click [**Open**]. The block opens and floats on top of the A4 frame. Position it accurately with the mouse and left-click to snap it on. It now sits down in the corner:



Preparations

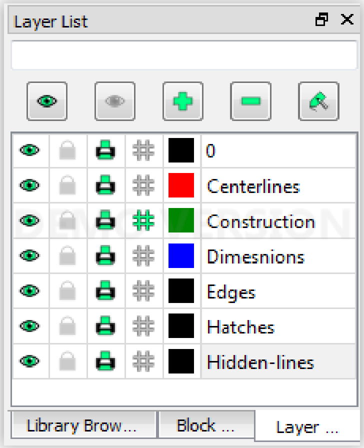
The flange was not specified above, but let’s assume it has an outer diameter of 200 mm and the plate is welded to a 50 mm stub tube with 100 mm internal and 110 mm external diameter. The tube end has a 2.5 mm 45º chamfer, other edges are 90º. The plate is 10 mm thick and has six evenly spaced ø10 mm holes for bolts.

We created the A4 Portrait template but the flange is too big for A4 size paper and the printed drawing must be in scale 1:2; which means we must scale up the A4 template 2:1 in order to draw in scale 1:1. (Better still, use A3 format or even better than that: add the template only when the drawing is ready.)

We scale up the template the same way it was done in Exercise #2: **Select** the drawing using the Selection Pointer; then:

```
CB: sz ** or Clk: Modify>Scale
CB: 0,0
```

On the Scaling Options window, select **Delete Original**, **Isotropic scaling**, and **X 2**, then Clk [**OK**]. Check that everything is OK by measuring the width of the frame (should be 380 mm) and enter CB: **za** to see there is no crap floating around. Done.



Layer List

⬛	0
🟥	Centerlines
🟩	Construction
🟦	Dimesnions
⬛	Edges
⬛	Hatches
⬛	Hidden-lines

Layers: We define for a start the layers shown to the right (same widths and types as before).

Applications Settings and **Current Drawing Preferences:** We stick to default settings, but check that the following are valid for the latter:

Paper Format:

A4, Portrait

Units:

Main drawing unit: Millimeter
Length Format: Decimal / Precision: 0.0000
Angle Format: Decimal Degrees / Precision: 0.00

Dimensions:

General Scale: 1
Test Length Factor: 1
Text Height: 2.5
Dimension line gap: 0.625
Extension lines Offset: 0.625
Extension lines Enlarge: 1.25
Arrow Size: 2.5
Linear units: Decimal / Precision: 0.0000
Angular units: Decimal Degrees / Precision: 0.00

Drawing edges

We'll draw edges using help lines (Construction layer) and there will be quite a number of those.

Vertical help lines will be distributed symmetrically around the center line. We should position the center line slightly to the left of the A4 center in order to leave space for dimensions to the right. Half of the width is 190 mm and we can start at 170 mm. Here the locations are, counted from the origin and all at y = 0:

170,0	Center line
120,0 & 220,0	Tube, internal walls
117.5,0 & 222.5,0	Chamfer
115,0 & 225,0	Tube, external wall
90,0 & 250,0	Hole centers
70,0 & 270,0	Flange, outer reach

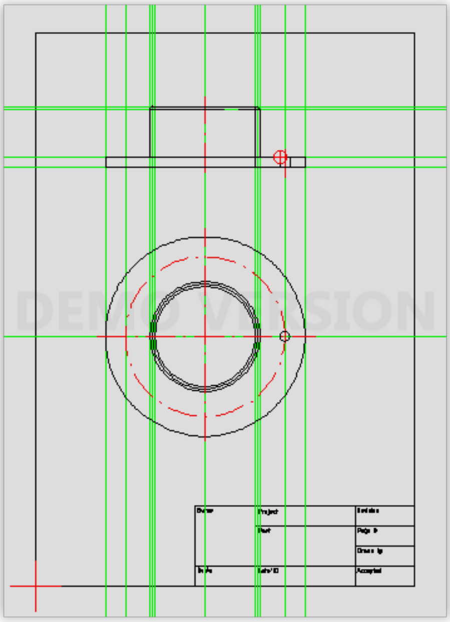
Horizontal help lines have to be distributed to divide the distance between the Title block and the top of the paper, 474 mm, minus the flange diameter (200 mm) and flange height (10+50 mm) in three parts – not forgetting space for dimensions. It makes 70 mm for each gap.

- The center of the flange plate will be at 80 mm (for the Title block) + 70 mm + 100 mm = 250 mm (help line at 0,250).
- The lower surface of the flange disk will be at 250 mm + 100 mm + 70 mm = 420 mm (help line at 0,420)
- The upper flange surface is at 420 mm + 10 mm = 430 mm (help line at 0,430)
- The end of the tube is at 430 mm + 50 mm = 480 mm (help line at 0,480)
- The chamfer is at 480 mm – 2.5 mm = 477.5 mm (help line at 0,477.5)

Those lines should suffice.

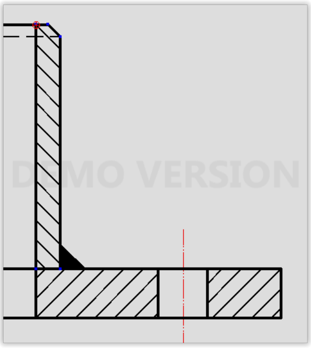
It may be necessary to turn off the grid to see all help lines:
Click **Options>Current Drawing Preferences** and tick off **Show Grid**.

With the help lines in place, we can draw edges, hidden lines, and center lines. The result is shown to the right. A few hints:



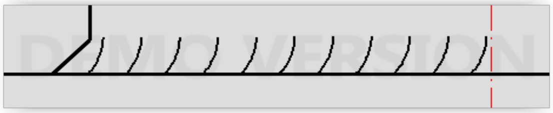
- Click on **Snap on Grid**, **Snap on Entity**, and **Snap Intersection**
- Draw all circles as **Center, Point** and snap them to respective help lines
- Draw the bolt hole as **Center, Point** and enter CB: **5** (radius)
- The projection is drawn as **2 Points** lines, snapped to the intersections
- Starting points of bolt hole edges on the projection are, respectively, 245,420 and 255,420 and drawn 10 mm up
- Linear symmetry lines are **2 Points** lines, drawn on top of the help lines

Next the details: Start with the projection where hatches have to be added to the cut half and the welding has to be shown.



The hatches, as said before, require a closed frame. In this case we need two frames for the flange disk, on for the tube, and one for the welding. All shown to the right.

We have also to mark the weld seam on the uncut side – for instance as shown below. The marks are arcs with a radius of 10 mm and having an angle from 150º to 180º.

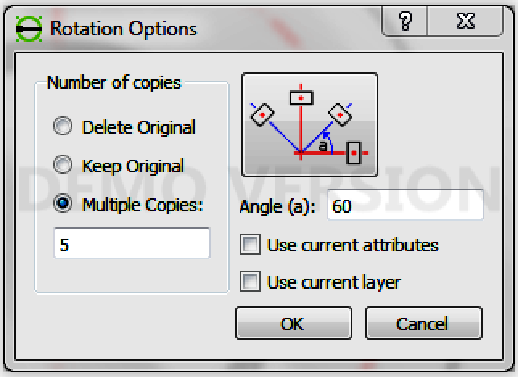


Then to the flange plate and adding bolt holes. You can first add a short vertical center line to the hole, then

Select hole + its horizontal center line

```
CB: ro  ** or Clk TB: Modify>rotate
CB: 170,250 ** rotation center
CB: 250,250 ** center of hole
CB: @80<60 ** rotation radius & angle
```

On the **Rotation Options** window, select as shown to the right. Then click [**OK**] and the five additional holes will be drawn, as shown in the figure on the next side.



Rotation Options

Number of copies

○ Delete Original

○ Keep Original

● Multiple Copies: 5

Angle (a): 60

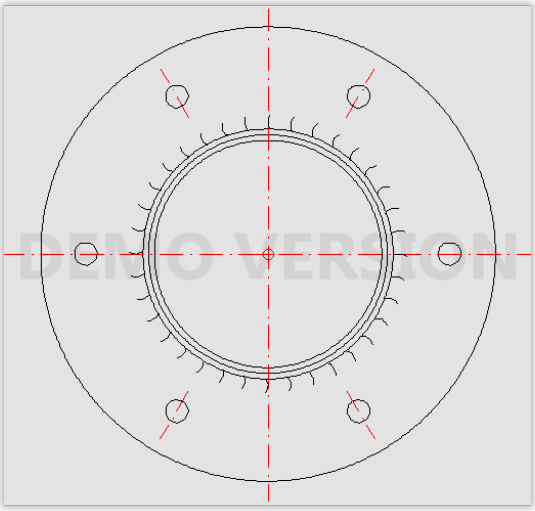
□ Use current attributes

□ Use current layer

We still have to add weld marks around the tube because the tube end is pointing in our direction and the fillet weld can be seen.

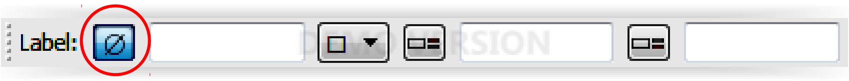
What I did was to create a small arc just outside the tube in the y=0 direction (arc center 228,250), then rotated as above but with 10º separation. The result is shown to the right (sorry, the circle for bolt holes suddenly dropped off).

We have to add dimensions and fill in the Title block. In line with common practice, each measure has to be stated just once. Only the first angle for the bolt holes should be given.



All six holes are created

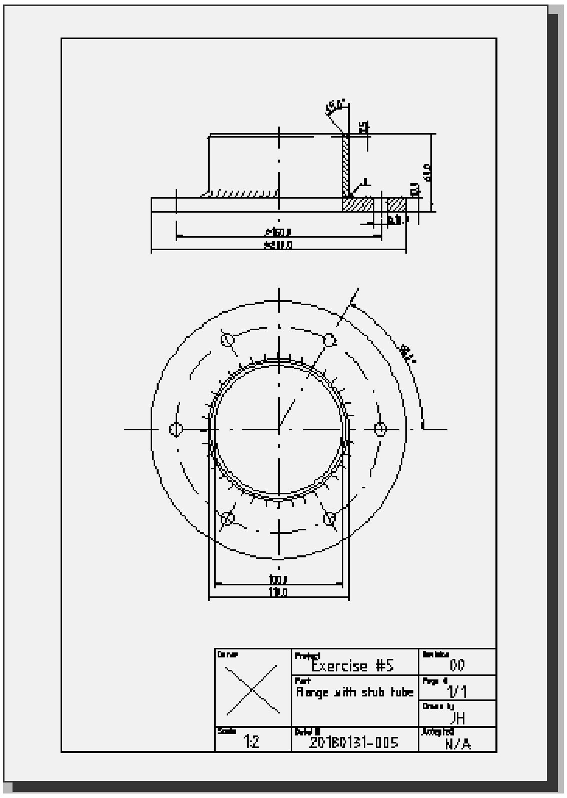
The symbol for diameter can be added to dimensions on the **Tool Options Bar** (pushbutton circled in image below):



When you insert dimensions it immediately becomes clear that the font is too small and the precision for length and angles too high. To correct this, do the following changes in **Current Drawing Preferences>Dimensions**:

General Scale: 2
Linear precision: 0.0
Angular precision: 0.0

The simple act of changing General Scale takes care of the dimension scaling. What I did with individual settings in earlier exercises was a true “overkill.”



To the right is finally our beauty contestant. It is a PNG image with poor resolution because LibO Writer has something against PDF images.

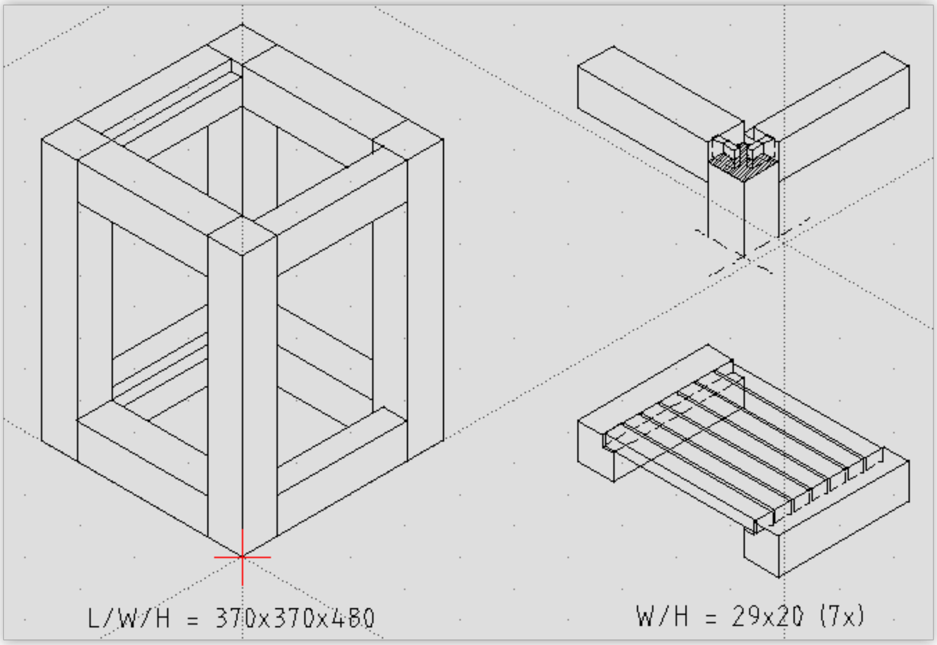
Font size for the Title block can be set in the Text (or MText) window. Here I have used sizes 6 and 8.

Exercise #6, forgetting the document size

As said in Exercise #2, adjusting the drawing to a fixed paper format is an “unorthodox” way of drawing. This is however what I have done in the previous exercises – old habits from the paper era I guess. So let’s do it differently by drawing without worrying about the document size and add the document template only at the end.

The drawing

The drawing I am going to use is the outline for a small table that I made for a woodwork project. There is no need to discuss the details of how it was drawn – we have discussed everything before. Here is the table drawing, ready to be “put on paper” (the frame around the drawing was put in by Gadwin, which I used to take the screenshot):



As you can see from the dimensions, the size of the table is too big to fit in scale 1:1 on either A4 or A3 document size – but a hobbyist hardly has a bigger printer. I decided to go for the smaller alternative.

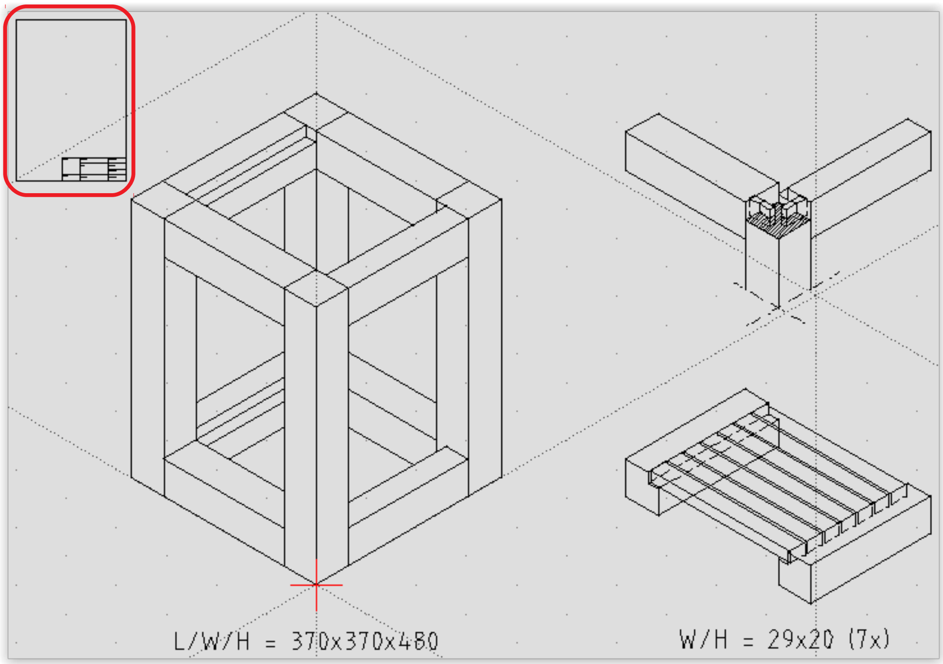
First import the A4 Landscape template:

```
Clk File: Import>Block
```

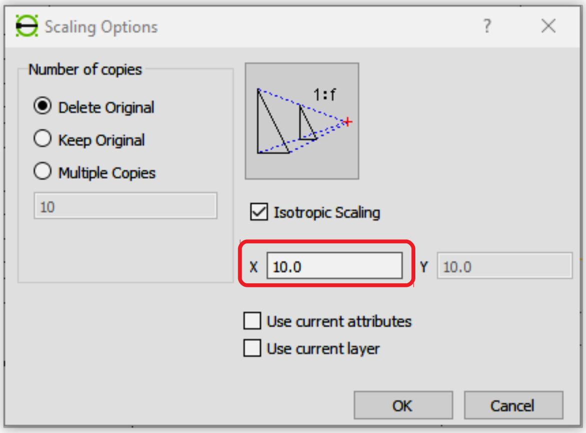
Find the template wherever it is saved and Clk **OK**. Adjust the location close to the drawing and leftclick to fix it. Clk **Esc** to end the process. We have now a screen picture similar to the one on the next page. The A4 template (shown below as Portrait, which I tested first, later changed) has minimal size compared to the drawing.

The next step is to increase the size of the template to fit around the drawing. Select the template and

```
Clk TB: Modify>Scale
```



Select a **Reference point**, e.g. the lower right-hand corner of the template, and the Scaling Options window opens. I estimated that the template had to be scaled by a factor of 10, as shown in the image to the right (5, the next ISO scaling option would obviously be too small).


Scaling Options

Number of copies

● Delete Original

○ Keep Original

○ Multiple Copies

10

☑ Isotropic Scaling

X 10.0 Y 10.0

□ Use current attributes

□ Use current layer

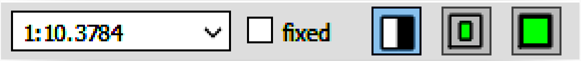
Clk **OK** and the size of the template increases and you can drag it to its location around the drawing. The result is shown in the screenshot on next page (I had to readjust the location – with Move/Copy – for the best fit).

The result is not optimal. It it will be too small when printed; one should rather use scale 1:5 on A3 Landscape instead.

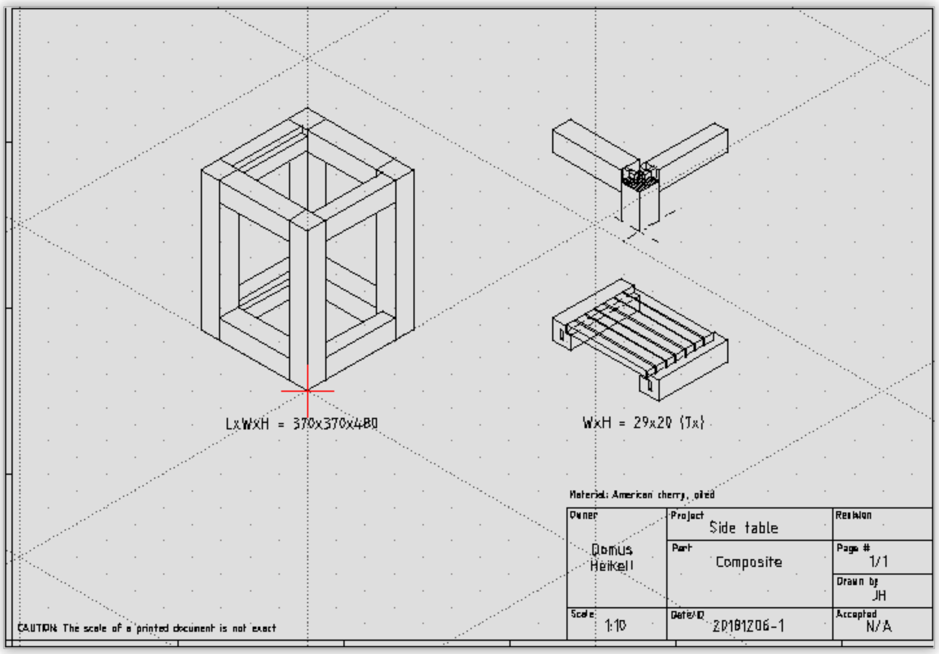
Printing

Printing and related issues were discussed in the last section of Chapter 2. However, and as I tell below, importing and scaling up the A4 template brought new problems.

The first step is to Clk **File>Print preview** and the Tools Options bar shows the scale to be 1:10.3784. It is smaller than the targeted 1:10 and it also shows as

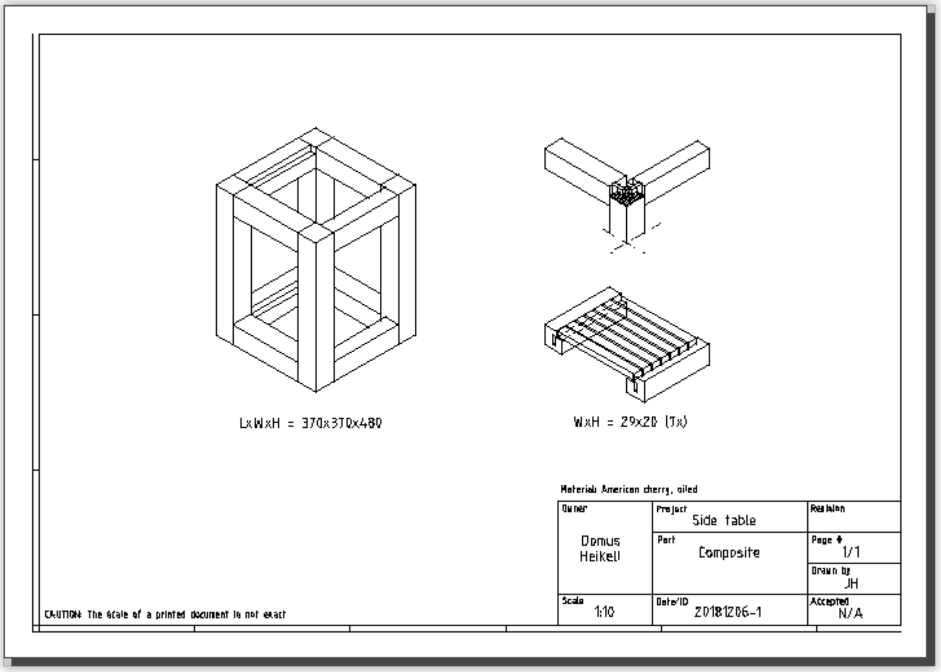

1:10.3784 □ fixed

wide margins on the printing area. Change the value to 1: 10 and the result is closer to what we want.



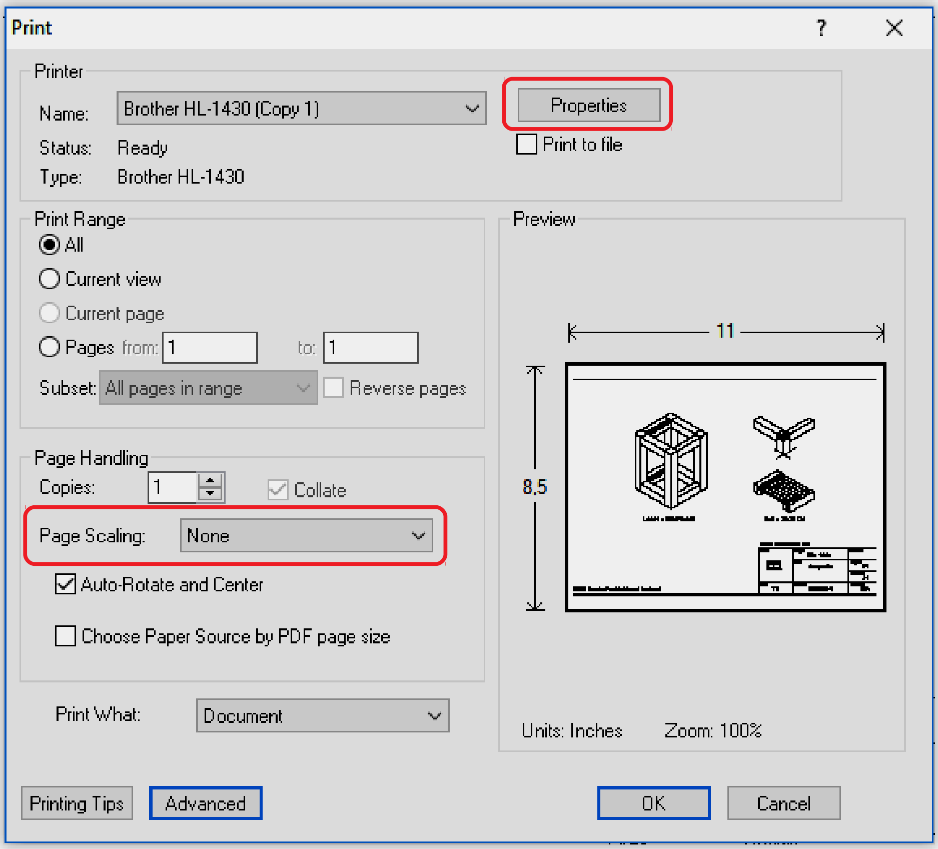
The A4 Landscape template is now in position, but it will be difficult to see all details of the plotted drawing in A4 format.

Then Clk **Center to page** (middle pushbutton) to center the drawing on the print area. The drawing is now ready for printing:



Clk **File>Export>Export as PDF** and save the drawing somewhere.

Open the PDF file with e.g. Adobe Acrobat. For me at least as everything is in order. Clk **File>Print Setup** and check that the paper size is correct (A4 Landscape), then Clk **File>Print** and Acrobat's print screen opens:



Print

Printer

Name: Brother HL-1430 (Copy 1)

Status: Ready

Type: Brother HL-1430

□ Print to file

Print Range

● All

○ Current view

○ Current page

○ Pages from: 1 to: 1

Subset: All pages in range

□ Reverse pages

Page Handling

Copies: 1

☑ Collate

Page Scaling: None

☑ Auto-Rotate and Center

□ Choose Paper Source by PDF page size

Print What: Document

Preview

Units: Inches Zoom: 100%

Page Scaling is set to **None**, because this version of Adobe Acrobat does not have the more advanced Custom Scale option that I mentioned in Chapter 2. Next, Clk **Properties>Layout> Advanced** and check that the paper size still is A4 (for me it tends to jump to Letter, as you can see from the 8.5”×11” measures above).

Clk **OK** to print.

Then comes a problem: The drawing is OK but the A4 frame (template) is printed as a weak shadow. The A4 frame (template) looks OK on Acrobat, but when I open the PDF file with GIMP it is hardly noticeable.

After a while (a long one) I noticed that all lines and texts that do not show up are defined as Layer 0, which has not been defined and has a line width of 0.00 mm. The text that I added to the header was also in Layer 0 because I had not changed the layer after importing the template – still difficult to remember to check the Layer List every time.

Lessons learned: LibreCAD puts the imported block into Layer 0 (which I originally used to draw the template).

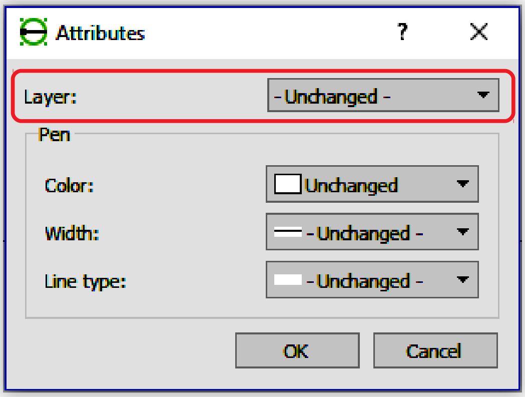
We could solve the problem by changing Layer 0 to 0.25 mm, but what if we have multiple blocks that need different line types and widths? The answer is that we have to redefine the imported block under another layer. In this case I added a layer called Template with the same line attributes as the A4 template has (continuous line, 0.25 mm wide). Then to move the block from Level 0 to Level Template:

```
Select the A4 block
```

```
Clk TB:  Modify>Attributes
```

The Attributes window shown to the right opens and the Layer can be changed from Unchanged to Template.

You can check that the block has moved to the new layer by clicking on the relevant eye icon on the Layer List.


Attributes

Layer: - Unchanged -

Pen

Color: Unchanged

Width: - Unchanged -

Line type: - Unchanged -

Discussion

The motivation for this example was to find out how to draw without worrying about the size of the final hard copy – the preferable method when working with CAD. The main findings are:

- This method is more flexible than drawing with the template in place
- Blocks are imported to Layer 0 and must be redefined into another layer with the right dimensions
- For a drawing that consists of multiple objects, as is with projections from three angles, it will most likely be necessary to rearrange the parts to better fit the printout
- It is necessary to check and recheck that the paper size is correct before clicking **OK** to print

4 Additional tips

This chapter covers various tips that I have come across, or ideas learned by experience, presented in no particular order and followed by a discussion on export and import. LibreCAD's user forum with questions people have asked has contributed to the tips, but unfortunately, many good hints are lost as people provide only partial answers. Additional tips can be found e.g. in the LibreCAD tutorial "GUI update" at <http://forum.librecad.org/GUI-update-td5712246.html#a5712603>.

Miscellaneous tips

Rounding corners

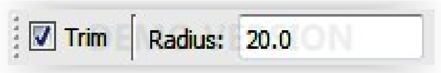
Say that you have drawn a rectangle and want to round a corner or two. LibreCAD developers have camouflaged the necessary tool by trading the word "round" for the engineering-inspired "fillet." Thus you have to

```
Clk TB: Modify>Fillet
```

or by

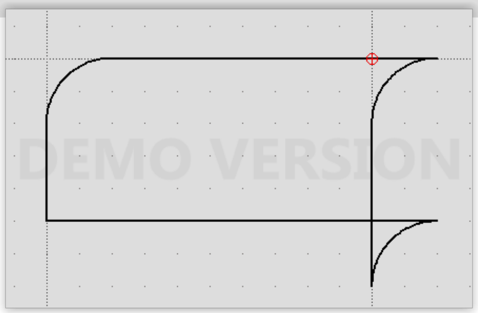
```
CB: fi, fillet
```

Define the radius on the **Tool Options Bar**:



☑ Trim Radius: 20.0

Play around with the rectangle and you can do all sorts of roundings.

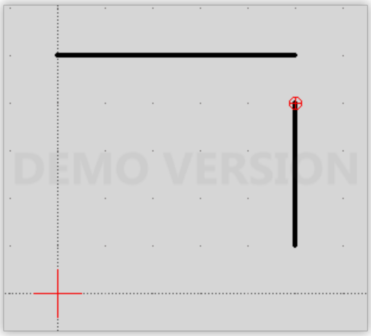


Lengthening a line

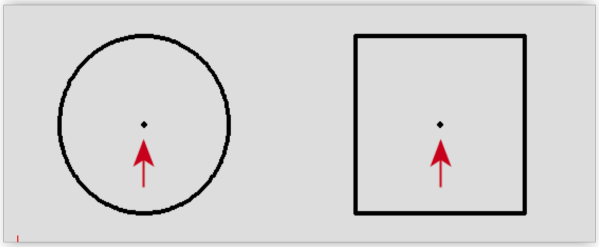
If a line is too short we can lengthen it by

```
Clk TB: Modify>Lengthen ** or CB: le
```

Then add the distance by which to lengthen on the Command Bar, and finally click on the line to be lengthened. The additional part shoots out in the direction the line was drawn. It will be shortened from the tail end in case you enter a negative number. The latter case is shown in the image right: I entered -10 on the CB instead of 10 and now the vertical line is short in both ends.



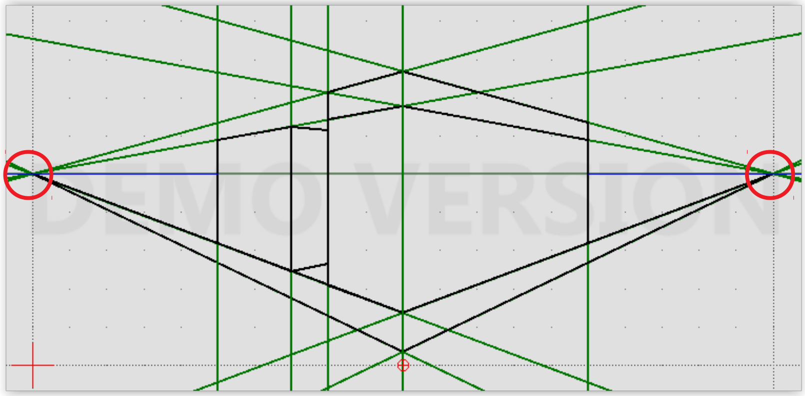
Add center points



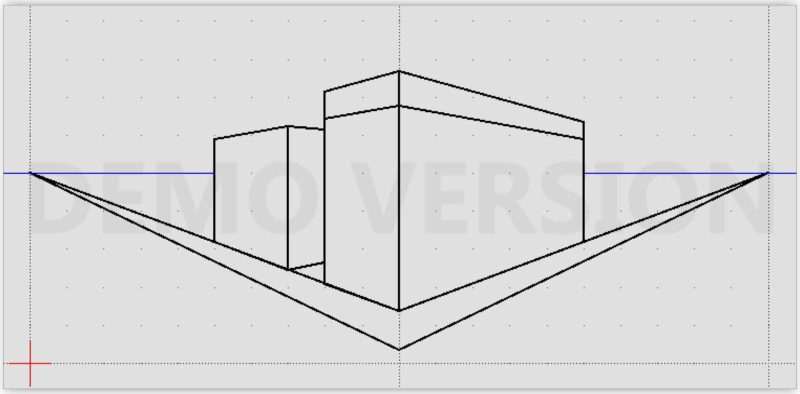
LibreCAD does not mark the center point of circles and quadrants but they can be added quite easily: For a circle, activate **Snap Center**, Clk TB: **Points** (or CB: **po**) and click at the center of the circle. The process is the same for a quadrant, but activate instead **Snap on Endpoints**.

Linear projections with LibreCAD

Working with **linear projections** means that we have to set **vanishing points** along a **horizon** and then define lines on the Construction Layer (help lines) to pass through the vanishing points at a certain angle. The **Relative Angle** function (Clk TB: **Line>2 Points**) offers a way to do that at least on a basic level, as this small test shows:



Horizon with vanishing points



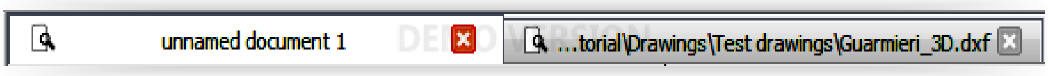
Horizon

A single vanishing point on the Drawing Area allows us to draw the interior of a room or a similar object. More advanced linear projections with multiple vanishing points would require better ability control the interaction between vanishing points and help lines – particularly when the vanishing points are located far outside the Drawing Area.

A discussion on drawing with a single vanishing point can be found e.g. at <https://design.tutsplus.com/articles/technical-drawing-for-beginners-one-point-perspective--vector-21839>.

Tab Mode & copying into another document

On the Menu Bar, click **Drawings>Tab mode**. It opens a **Tab bar** on top of the Drawing Area, into which we can open another drawing:



LibreCAD lacks the clipboard of Windows and Linux but in the Tab mode you can copy a part of a drawing (Source) into the other (Target) as long as both drawings are open. The following process copy-pastes not only the object but also the Layer of the source:

```
Select Source document
Select (highlight) the object to copy
Clk [Ctrl]+[C]
Select reference point (e.g. CB: 0,0)
Switch to Target document
Clk [Ctrl]+[V]
Select reference point (e.g. CB: 10,10)
```

That’s it – remember that I do not mention [**Enter**] and [**Esc**] clicks.

On LibreCAD’s discussion forum is a different explanation for the process, but it has not worked for me (or I have not understood the brief explanation).

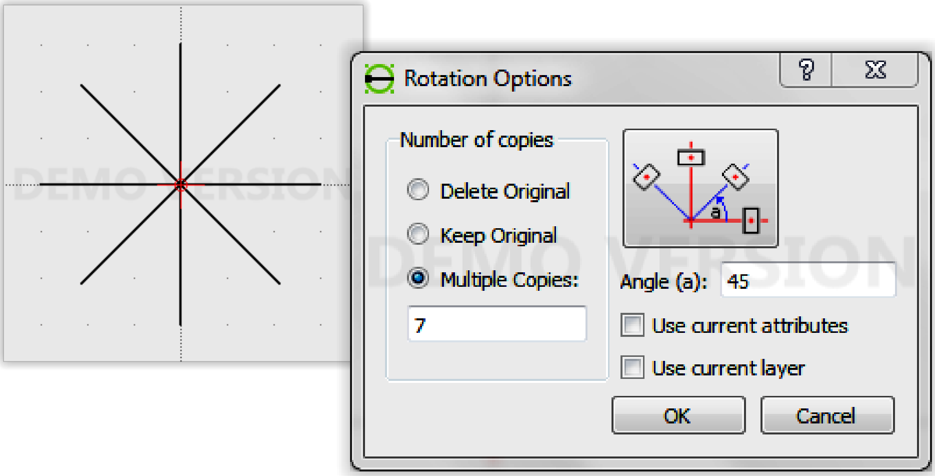
Create a fan of lines

We did it in Exercise #5, but let’s repeat: Assume that you have drawn a line from the origin [0,0] to @30<45 and want to create a fan of symmetrical lines (spokes) around the origin (shown below left). Here is the recipe:

```
Clk TB: Modify>Rotate ** or CB rotate, ro
CB: 0,0 ** select rotation center
CB: @30<45 ** point to rotate (end of line)
CB: @30<90 ** target point (end of next line)
Select as below on the Rotation Options window
Clk [OK]
```

The number of spokes to select will always be odd for evenly spaced spokes since the first spoke has been drawn before we use the Rotation Options window.

It is a bit tedious but it works – you can try to find an easier way.



Rotation Options

Number of copies

○ Delete Original

○ Keep Original

● Multiple Copies:

7

Angle (a): 45

□ Use current attributes

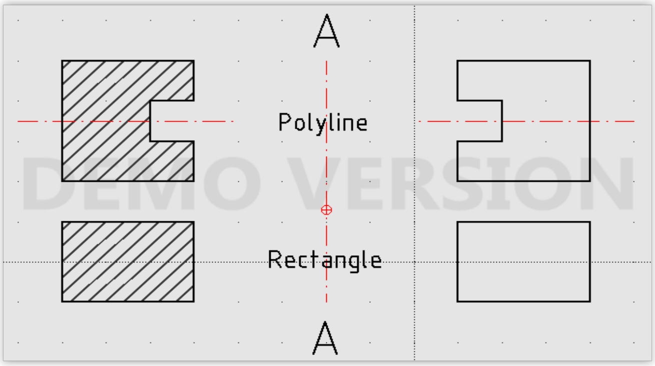
□ Use current layer

Mirror objects

Assume we have objects like the one to the left on the picture below – one drawn as a polyline, the other as a rectangle – and want to create a copy of them (mirror it) with respect to a certain line (A-A in the image).

Start by selecting **Snap on Grid** (if you use the mouse). Then

```
Select object
Clk TB: Modify>Mirror ** or CB: mi
Clk at start of mirror line ** or enter
    value on CB
Clk at finish of the line ** or enter value
    on CB
Select Keep Original on the Mirror Options
    window
Clk [OK]
```



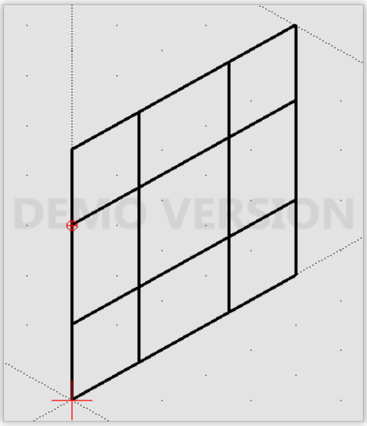
The objects snap to the side of the mirror line. Mirroring works, but in both cases the hatches are lost – same bug as with scaling in Exercise #2.

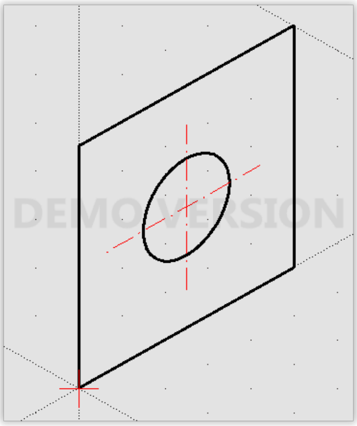
Isometric hole

Assume that you have to draw a ϕ20 mm hole at the center of a 50×50 isometric plate as shown in the figure to the right. To do it, you have to inscribe the hole in a 20×20 mm rectangle. This can be done by copy-pasting each edge line of the plate by 15 mm inward to form the rectangle, as shown below to the left, then

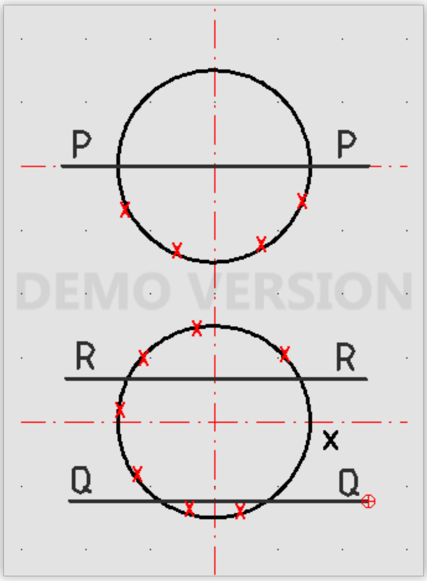
```
Clk TB: Ellipse>Ellipse Inscribed
    ** or CB: ei, ie
Clk on each of the four lines outlining the hole
```

The hole is ready but center lines should be added. They can be drawn from the middle of the plate sides by activating **Snap Middle**, then shortened by Clk **TB: Modify>Lengthen** and CB: **-10**, and finally clicking on all line ends.





Creating semi-circles and arcs



LibreCAD’s arc tools under **TB: Curve** can be a pain. An alternative way to create semi-circles and arcs – at least if they are non-critical – is to cut them out (trim) from circles.

In the top picture to the right is a circle with the line P-P cutting exactly through the center of the circle. By using P-P as the limiting line we can delete the half that we do not want and the result is an exact semi-circle with defined radius.

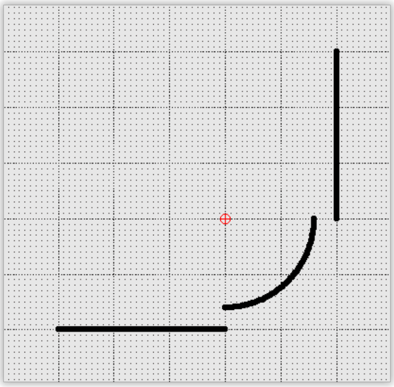
In the lower case we want to create the arc marked x. The solution is trivial: Use first e.g. line R-R to trim the top part of the circle, then line Q-Q to trim the rest. This is the method I used in Exercise #5 to create small arcs to mark the weld seam.

Lost widget

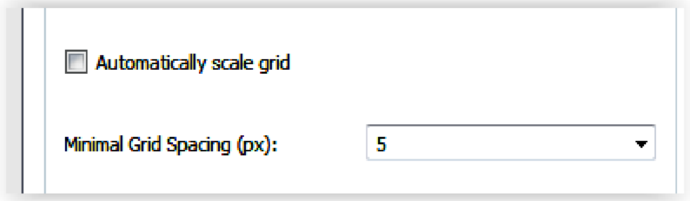
It is not uncommon to suddenly find that a widget has, for one or another reason, jumped outside the Drawing Window. How can it be found, particularly if it is small compared to the drawing?

One way is to enter **za** (Zoom All) on the Control Box. LibreCAD now displays the entire drawing – all widgets there are. The lost widget is somewhere close to the margin and can be either dragged back or deleted and redrawn. If the widget is too small to be seen, you can paint (select) the area with the Selection Pointer and Click [**Delete**]. Most likely you get rid of the strayed item.

Grid settings



Grid settings often feel too coarse when working with the mouse, as for instance when drawing with an arc tool (Clk **TB: Curve>Center, Points, Angle** shown to the right). If so, zoom in sufficiently and change default grid settings on **Options>Application Preferences> Appearance** as shown below (tick off **Automatically scale grid** and set **Minimal Grid Spacing** to a value lower than 10 px). The background now looks like millimeter paper.

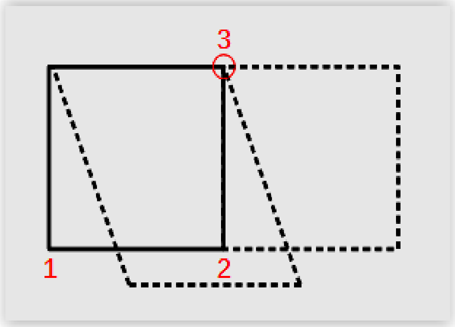


□ Automatically scale grid

Minimal Grid Spacing (px): 5

NOTE: The default setting is normally better because it can be difficult to snap accurately with Minimal Grid Spacing set to 5 px.

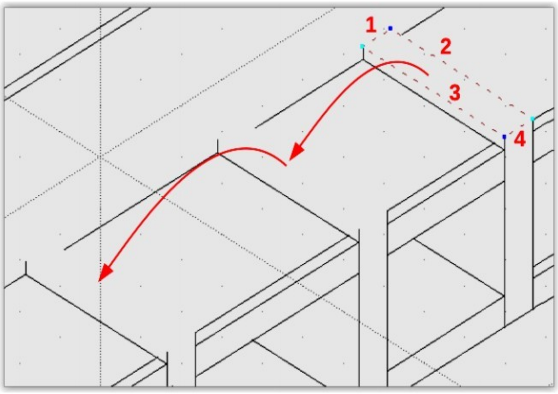
Stretch



The Stretch tool (CB: **ss** or Clk **TB: Modify>Stretch**) can be used to stretch or deform an object. Shown to the right is a rectangle. With Stretch activated, click on the corners in the order 1-2-3. The last corner becomes the center of movement and allows you to either stretch the rectangle in one direction or to move corners 1 and 2 to form a parallelogram (the drawing is done with LibreOffice Draw).

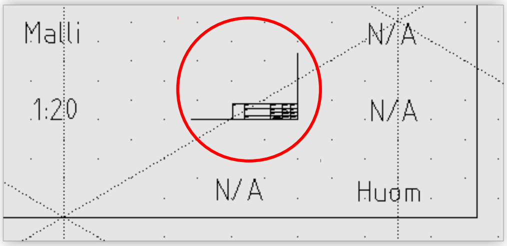
Drag-and-drop widgets

It saves time to copy-paste multiple widgets in one go. To the right is a case where I have to add four lines to the top of a book-case leg. I have drawn the four lines to one leg (1-2-3-4) and selected them.



The next step is to CB: **mv**, or Clk TB: **Modify>Move/Copy**, then left-click on one point as the reference point. After this the set of lines can be dragged and dropped to the next leg. A new reference point has to be selected before the drag-and-drop process can continue – a detail LibreCAD developers could improve.

Scale of an imported block



A block is imported in the scale it has been saved in. You are in for a surprise if your new drawing is in a different scale – as I was when I imported a scale 1:1 title block to a scale 1:20 drawing (circled, right). The original title block had been inadvertently deleted (why am I not able to lock it?).

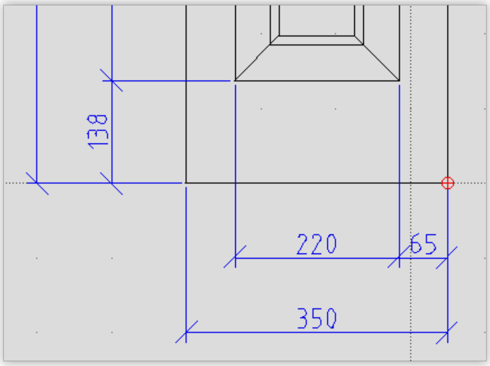
The only solution I have found is to create and save as many scaled-up blocks as I need of a certain type.

Remember to zoom and measure

Zoom in and check that you really have been able to **snap** a widget – for instance the end of the line – in the right place if you are snapping with the mouse. It can easily be a millimeter or two off without showing in a small scale.

Use the **Info>Distance Point to Point** tool (or one of the other in the set) liberally. Use it to check that help lines (Construction Layer) are right, that the size of a rectangle is right, etc. It only takes a typo to run into trouble.

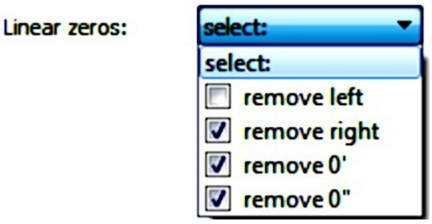
Dimension problem



I had drawn a small door (lower part shown to the right) and began to add dimensions. The first one I added was 138 vertical, which gave no problems, but after that every dimension came out a factor 10 smaller (22, 35, etc.). Restarting

LibreCAD did not change the situation. What finally helped was to change Length Precision under **Options>Current Drawing Performances>Units** from 0 to 0.0. But as you can see, the dimensions are still given in full millimeters. I have later changed Units to 0 with no problems.

I’m not sure about this behavior (have not tested enough) but I think it has to do with the **Linear zeros** tool under Options>Current Drawing>Performance>Units, which is shown to the right. Tick off the **remove right** and LibreCAD adds a decimal zero.



Linear zeros: select:

select:

- ☐ remove left
- ☑ remove right
- ☑ remove 0'
- ☑ remove 0"

Exploding a rectangle

Using the rectangle command (CB: **rec** or Clk **TB: Line>Rectangle**) is a quick way of drawing when we need that kind of objects. However, rectangles created in this way enclose a surface and individual sides and must be exploded (CB: **xp** or Clk **TB: Modify>Explode**) before modifications can be done.

Lockup with leaders

A drawing suddenly locked completely, the only sign of life was that a **leader** blinked when I left-clicked on the drawing. Restarting LibreCAD did not help, nor did rebooting Windows.

I have asked for the ability to lock components and blocks, but not for LibreCAD to lock everything by itself. A bug.

Since there was a slight sign of life in the leader, I deleted it together with another ledger on the drawing. Bingo! Problem solved but now I have to draw leaders with CB: **li** and LibreCAD does not allow me to add arrow heads to lines.

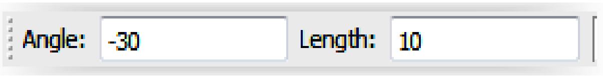
Modifying text

A string of text that is inserted to a drawing can be modified by Clk TB: **Modify>Properties** and left-clicking on the text in case. This opens the **Text** window with the text that can now be modified.

This is a text

This is a modified text

Keep an eye on the Tool Options Bar



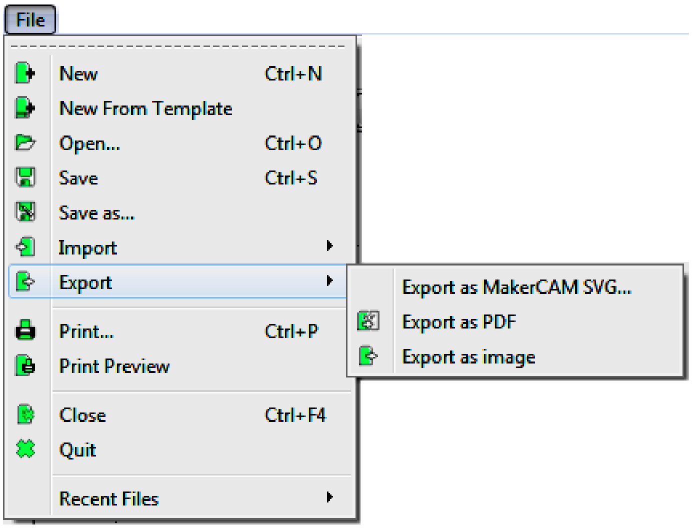
Angle: -30 Length: 10

The **Tool Options Bar** that opens with certain functions to the right of the Layer Selection area is easily forgotten. For instance, if you want to draw a line with an angle relative to another line (Clk **TB: Line>Relative Angle**), the Tool Options Bar will open as shown above. You have to select the angle (and optional length) here before entering the reference point on the Command Box. Forgetting the Tool Options Bar leads nowhere (know it, done it).

Import-export

This subject is not my cup of tea, as I guess our Brexit friends would say, but I’ll try to come up with something. Check however the LibreCAd document “Export” at <http://wiki.librecad.org/index.php/Export>.

Export



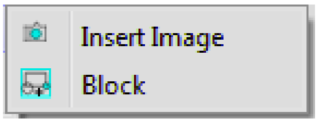
Images that we want to save have mainly to be exported via **File>Export**. We met this function already in Exercise #1, when I exported the A4-template to whatever folder of my choice.

In the typical case we export a drawing as PDF for the simple reason that this function works in LibreCAD. However, look under **Export as image**, there are plenty of alternatives to choose between. They may or may not work, you never know.

I have never tried the Export as **MakerCAM SVG…** alternative – don’t know what it does – but feel free to go ahead.

The LibreCAD forum has a discussion on exporting from LibreCAD to Blender (2D to 3D). It may or may not be possible with the SVG image format <http://forum.librecad.org/Import-to-Blender-2D-to-3D-td5715460.html>.

Import

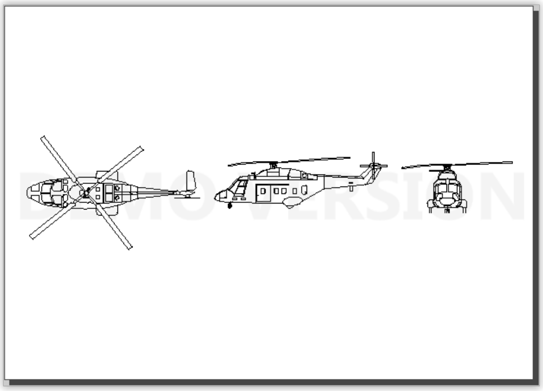


I used **File>Import>Insert Image** in Exercise #1 to add a PNG “logo” to the title block. The import went OK, but the imported Smiley had to be down-scaled to fit into the cell of the title block.

The second time **File>Import** came to use was in Exercise #5 (and again in Exercise #6), when I imported the title block to the drawing. There is no problem snapping a block to a new drawing if the reference point of the block is selected well (remember however that preferably both should have the same scale).

Importing a DWG drawing

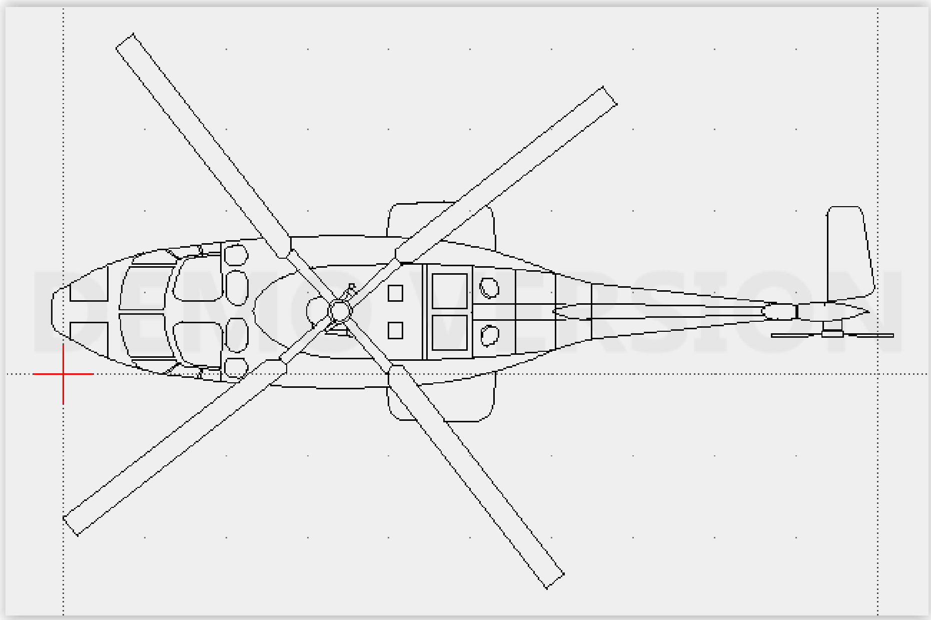
There are some web sites that announce free DWG drawings, but in reality almost all require a fee or have other copyright restrictions. The one that seems to be truly free is <http://www.freedwg.eu/>. From them I downloaded a Zip file with a helicopter drawing, opened it and imported the DWG file to LibreCAD without incidents.



The figure to the right shows a screenshot of the drawing as DXF in LibreCAD Print Preview mode.

I could save the drawing as DXF2007, PNG, PDF, and SVG. All succeeded except for SVG, which came out as a messy blob.

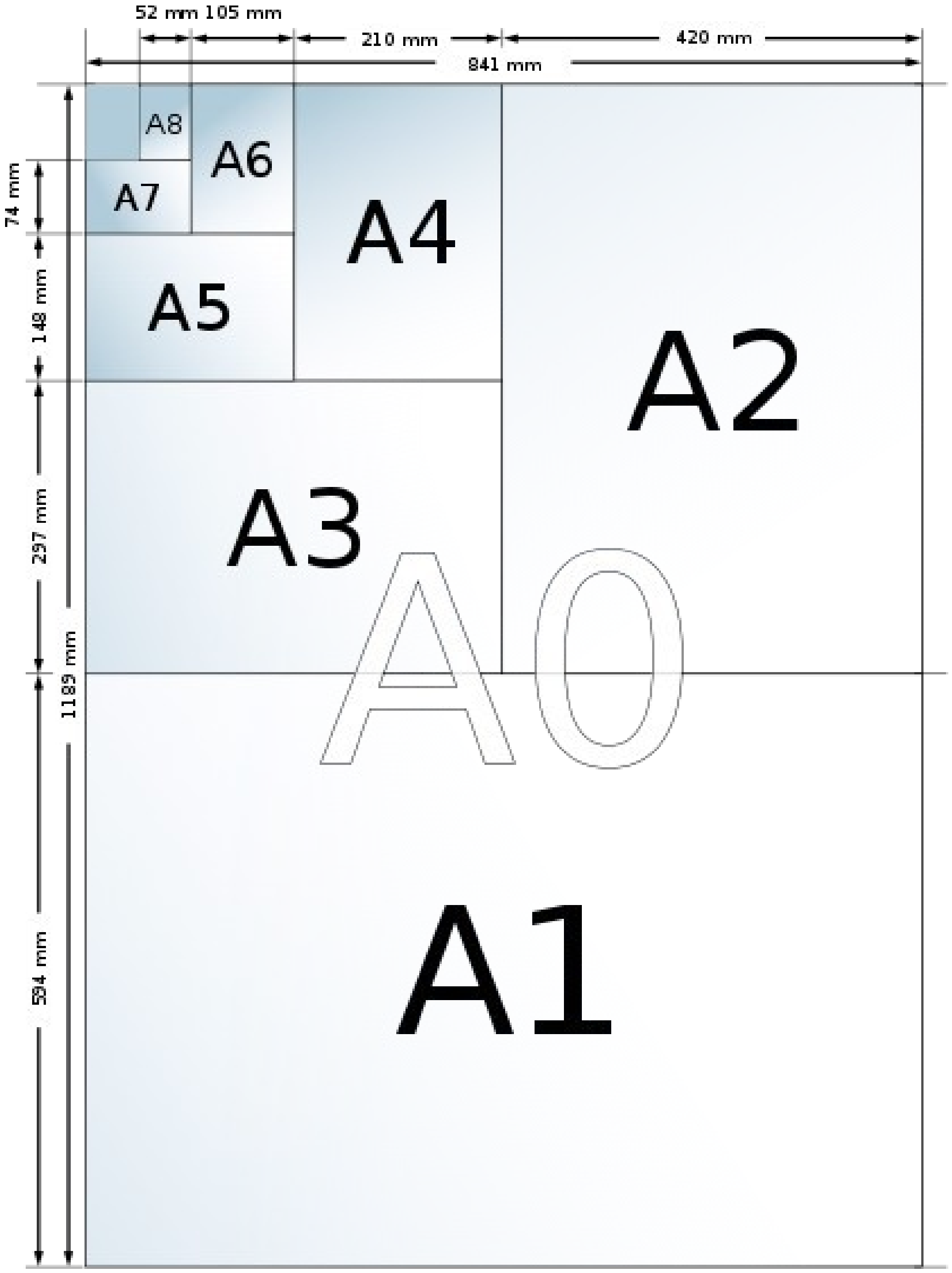
Next I exploded the DXF2007 drawing and created separate blocks of each view (top, side, front). You can see a screenshot of the top view below; it has pretty good resolution despite its size of only 69 kB.



As a summary, this small DWG import experiment went smoothly with the exception of saving as SVG.

Appendix A

The ISO family of A document sizes (image source: Wikipedia):



52 mm
105 mm
210 mm
420 mm
841 mm
74 mm
148 mm
297 mm
594 mm
1189 mm
A8
A7
A6
A5
A4
A3
A2
A1
A0