GALAAD 3 ®

GALAAD 3®

USER'S MANUAL

COPYRIGHT / RESPONSIBILITY

Galaad is supplied as is, and no guarantee is given or implied as to the results obtained by its use, or regarding overall satisfaction with the end product. Galaad is a complex piece of software and has been extensively tested, however, these tests are not guaranteed to have removed all bugs and minor errors from the software. Finally, no guarantee can be given that subsequent versions will necessarily be compatible with existing versions.

Purchasers are solely responsible for their choices, how they set up and use the software for the intended purposes, and whether the results match their expectations. No responsibility will be attributed to Galaad, its authors or distributors, for any damage caused, either directly or indirectly, to the machine tool, any peripheral devices, cutters, computers, stored data, or any other object including hardware and software involved in the usage of Galaad.

Obviously none of the above terms and conditions affect the purchaser's statutory rights under current law or legislation.

Note: in general, and without enumerating them here, are fully recognised all registered trademarks that are mentioned in this manual, with the corresponding rights.

TABLE OF CONTENTS

22 - FIRST AID

23 - TECHNICAL MATTERS

24 - A FEW TIPS FOR BEGINNERS

25 - GLOSSARY OF TERMS

0

0 0 0 0 0

FOREWORD

☐ Brief *Curriculum Vitæ*

Galaad is a CAD/CAM/CNC software suite *(Computer Aided Design / Computer Aided Manufacturing / Computerised Numerical Control),* the point of which is to enable you to design, *i.e.* **draw**, then manufacture, *i.e.* **mill**, parts or objects that you need. In view of the geometrical nature of most components and the associated toolpaths, this requires a classical approach to the design work. **Galaad is a single integrated package that covers all aspects from drawing through to controlling the milling machine**, and not simply a collection of different applications with different interfaces. This integration in a homogeneous set offers many advantages, making it easier both to learn and to use, even for absolute beginners.

It is important to mention here that, in spite of its name, **Galaad 3 is not 3D design software**. Its application domains are **mainly engraving and cutting works** on flat materials. Galaad provides special functions that create 3D wiremeshes, for example cut profiles or curve surfaces; it accepts 3D handling and effects; it can import and mill 3D external files, but it cannot create, handle, import or mill volumes that are made of facetted or NURBS surfaces. Only 3D **vector** trajectories that represent a toolpath are accepted. Therefore **if you already have a 3D CAD system that handles surfaces and volumes, you need an intermediate module that can convert these shapes into vectors which define a final toolpath**. Only then can you mill these 3D vectors using Galaad.

Galaad can drive a number of CNC machines with 2, 3, 4 or 5 axes, either directly or by using an external driver programme. Consequently, Galaad can be considered an open application.

Additional programmes and icons appear when installing Galaad, enabling you to make non-indexed **turning**, automatic **motion programming**, **printed-circuit engraving**, or **3D milling** up to 5 axes. These extra modules are described in the last chapters, but most of this manual is dedicated to the main application "Galaad", *i.e.* the integrated suite for CAD/CAM, design and machining for 2½D workpieces.

❑ Topography of this manual

The heart of Galaad software is its main module, which icon is named "GALAAD (mill)". This module lets you design 2½D workpieces and then mill them on your machine. So, most of this user's manual concerns this design module and its milling partner which is the subcontractor for the operator's manual control and for supervising the automatic process on the machine.

In early chapters, you will learn how to draw different kinds of shapes from this main module, then how to quickly prepare and launch the machining process for what you have drawn, then how to save your works. The purpose is to help you getting started and become familiar with the working methods of the software.

Then come the descriptions of functions that will enhance your design and machining capabilities, allowing you to boost your work, of parameters for engraving, milling or cutting tools, of tool radius compensation and how to mill pockets. This ends with workspace and network settings.

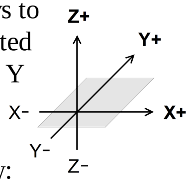
Drawing icons and menu functions are then reviewed one by one with more indications about their use and features.

A big chapter is dedicated to the machine parameters, with all possibilities that some CNC cards or devices offer, along with big tips and little subtleties for fine tuning. This section is probably the most important for an advanced user who needs to optimise his machine functions.

Plasma cutters will find a complementary chapter for their own business.

Lastly, for the unenlightened people who have purchased a limited licence, snubbing such a real marvel that Galaad main design module is, last chapters are devoted to Kay module (3D milling), Gawain module (turning), Kynon module (motion programming), Percival module (printed circuits) and Owein module (3D viewer). Yet, designing workpieces with Galaad does not concern you and therefore you may skip corresponding chapters. However, sections that are dedicated to workpiece origins, milling process or machine parameters remain valid and will at least bring you much useful information, even if they fail to motivate you for upgrading to a higher licence.

Conventions used



On a Cartesian milling machine, there can be several ways to identify axes. An industrial standard more or less accepted consists of defining the X axis as the longest of the machine, Y and Z axes then making a direct orthogonal framework from X. A more classical model, not always compatible with that standard, consists of adopting the operator's point of view: the **X axis increases from left to right**, the **Y axis from front to rear** of the machine, and **Z axis from bottom to top**. However, be aware that confusions often remain frequent for beginners.

Galaad displays directions as cardinal points: west is on the left side (X-), east is on the right side (X+), south is on the front side (Y-), and north is on the rear side (Y+). So the X axis increases from the west to the east and the Y axis from the south to the north. This does not mean that you need a compass for drawing or driving your machine, but the drawing board and the machine bed appear as a geographical map. For Z axis, we can simply talk about *"high"* and *"low"*, which looks quite intuitive. If you wish to comply with the standard that identifies X as the longest axis, then the best solution is to put the operator's station at the appropriate location.

Warning! Do not consider the motion of the machine table if it is mobile, but always **the apparent motion of the tool on the table**, which inverts the direction. When a mobile table moves towards front side, in fact the tool is virtually moving backwards, to *"the north"*.

The distance and speed units are settable. The distances can be displayed in millimetres (mm), centimetres (cm), inches (") or a user-defined unit. Speeds can be displayed in metre-per-minute (m/mn), millimetre-per-minute (mm/mn), millimetre-per-second (mm/s), centimetre-per-second (cm/s), inch-per-minute("/mn), inch-per-second ("/s) or a user-defined unit. Defining a custom unit supposes that the scaling factors on the machine or in the file exchanges are coherent with that unit.

All angles are displayed in degrees, the reference point being the trigonometric zero, direction X+ (3 o'clock in the aviation lingo).

❑ System configuration

For those who are interested in raw details, Galaad 3 was a continuous creation between 1998 and 2020, developed using Borland C++ 5.02 OWL on PCs with various processors and various versions of Microsoft Windows. The compiler has been set for generating **native 32 bits** executable code for Windows. In simple terms, this means that **Galaad 3 will run indistinctly under any version of Windows 32-bits or 64-bits**.

The software was duly tested and validated under all 32-bit kernel systems from Windows 2000 onwards, including server versions, *i.e.* 2000, XP, Vista, Seven, and under 64-bit kernel systems Vista, Seven, Windows 8, Windows 10 and Windows 11. Unfortunately there are no native versions of Galaad for Linux or Mac systems, but the software was tested successfully on Wine and Parallels Desktop emulators, including machine driving. Using Galaad on a Windows tablet is even possible, with an internal option for managing a touch screen.

Galaad does not work under Windows TSE systems *(Terminal Server Emulation)* which anyway are not used very often nowadays.

Concerning the hardware, Galaad does not require a very powerful processor for its internal calculations, unless you must handle drawings featuring hundreds of thousands coordinates. Consequently, **there is no need to purchase a supercomputer** for using the software comfortably. Better invest in a good size screen with high resolution for the pleasure of your eyes.

Talking about screens, the absolute minimum graphical resolution for displaying Galaad frame windows is 800×600 pixels (highly condensed mode which will be very uncomfortable for working unless you are a very patient person). This makes it possible to use on a NetBook or a tablet with a small screen for drawing and for driving a machine. Supervising an automatic milling process requires only a very low calculation capability. An old PC can do if you have a workstation for drawing and another workstation elsewhere for driving the machine.

❑ User's licence

As with most software, Galaad is the result of a looooong development work and is therefore not freeware. If you are reading this manual in its paper format, it is likely that you have legally obtained a copy together with a licence. The terms of this licence concern you all the same as you may need to make authorised copies of the software.

Galaad user's licence is always in the form of an electronic key, commonly called a dongle and generally attached to the spring binding of this manual. **That dongle must be plugged into a USB port of your computer**. Then only, Galaad will be able to read and establish your user's rights, and thereby give you full access to all features with no restrictions.

Not having a dongle is in effect not having a user licence, which will prevent Galaad from communicating with the outside world. In this case, it will neither be possible to execute a complete milling process, nor export any drawings to another software application. You will however be able to save and print drawings, and also drive your machine manually, with no time limits. This restricted use lets you copy the software so it can be evaluated by third-parties. Furthermore, a Galaad purchaser can install the application on several computers dedicated to design tasks, without the licence, and one connected to the milling machine, with the licence. A non-licensed version remains no less subject to current legislation on copyright.

For professional licences, the dongle does not need to be plugged in permanently. Galaad just refreshes the licence every month. Then it displays a message asking for a momentary plug of the key into the port, so it can be read and renewed.

You may copy Galaad 3 for yourself or for third parties. This is provided that the contents are not modified in any way, that all copy-rights are acknowledged, and that there is no possibility, even indirectly, of manufacturing the drawings being created with the software without a user licence.

There are three types of user's licence available. The **professional licence** is the main one, and is for those who wish to use the software to design and manufacture objects for commercial gain. There are no restrictions with this licence. The **educational licence,** as its name implies, is restricted to use in a training environment where there is no commercial gain, either directly or indirectly, from any objects manufactured. Finally there is the **hobby licence**, which has milling time restrictions and can be sold only to individuals or associations with no professional activity having any relation to the use of a CNC machine. **Using an educational or hobby licence must have no commercial purpose, even indirect**.

There are no differences in the functionality of the software under the different licences. All offer the same functionalities and features. Only the terms of use are different and so the workpieces created must remain strictly in a non-profit area. Consequently, **using an educational or hobby version for professional or commercial purposes, even indirectly, is a flagrant violation of the licensing agreement.**

Some subsets of Galaad may be sold separately at a lower price, in which case the corresponding licence is valid for these modules only. Conversely the normal user's licence of Galaad covers all modules. This is not a forced sale of useless modules, simply the standard Galaad licence provides access to all features.

If you have any questions relating to the licence that you have been supplied with, please do not hesitate to contact your supplier. Remember that if you have been sold a copy **legally it will come with a dongle.**

Like most protected software, **if you lose that dongle, then you lose your licence**, it is as simple as that. However, in some specific cases, not guaranteed, the lost dongle might be replaced, under conditions that you provide a proof of purchase (*i.e.* the invoice), a written certificate of loss, and the serial number of the lost licence. Nevertheless this replacement cannot be be taken for granted. **Take care of your dongle!** Si you think it could be stolen, put a small USB extension cord with the dongle at its end, eventually glued or locked inside the PC, making it unreachable.

For further information, or in case of difficulty we invite you to visit the official Galaad website: www.galaad.net

Be inquisitive!

Whatever you may get out of this manual, you will no doubt learn mostly by actually using the software. There will always be those complex functions that you will have to refer to the manual to find your way through, however, your best bet, both now and in the future, is probably to follow your instinct.

Galaad is house trained and so will not actually bite when you make a big mistake. At worst, it will send you a disapproving warning, which will no doubt be bearable in view of the good relationship you should have established with it, at least from Galaad's point of view. Take for granted that, whatever you do, including pure provocation, you will really have a hard time offending it. Consequently, if you find a function, an icon or a button and do not know what it is for, well, the best is to simply try it. Of course you can open this user's manual; that works too.

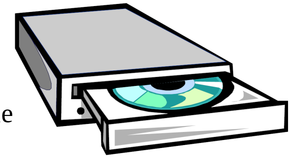
So do not hesitate in learning what Galaad can do. There is no shame in being a beginner and no good reasons for being shy. If you have any doubts about how a function is presumed to work, then the best way to find out is to jump in and try it, even if the end result is quite unexpected. You can always undo an unfortunate operation if things do not seem to go to plan. Trial-and-error is often the best way to learn how software really works. Galaad is no exception.

1

00001

INSTALLATION

Set-up



Galaad 3 is generally supplied on a CD, which contains all the files that will be transferred onto your hard disk. The installation programme will normally start automatically a few seconds after the drive door has been closed.

Under certain circumstances, the *"AutoRun"* may fail to start, depending on your Windows configuration. In this case it is necessary to run the **SETUP.EXE** programme on the CD, using the *"Start"* button and the *"Run"* option. If your CD drive corresponds to the D unit, which generally occurs on a basic PC, type **D:SETUP** and click on OK.

When the SETUP programme is run, be it automatically or manually, it produces a blue window and a series of dialogue boxes to enable the installation to be configured according to your wishes.

The first one asks the usual question regarding where you wish to install Galaad on your hard disk. The default folder is C:\GALAAD, but the choice remains yours.

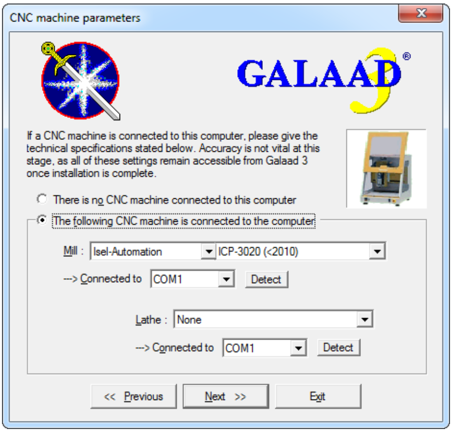
If you wish to install Galaad into another folder, simply type in its name or use the *"Browse"* button, which will open a small window enabling you to browse the directory tree and select a destination folder. It is not necessary to install Galaad in the default directory or even on a local hard disk. What is more **Galaad does not modify any files other than those in its own folder**, except for desktop shortcuts to it. In particular, the innumerable system files associated with the Windows galaxy are not affected by the installation of Galaad onto your hard disk.

Important detail, **Galaad will require full write access in its own installation folder**, even if your drawing files are stored elsewhere. If you intend to use the software in a Windows session with restricted rights (*i.e.* not the system administrator's session), it is very important that the installation directory remains accessible for write-operations in that user's session, otherwise the process will be affected (current drawing and parameters not saved, mainly). The *"C:\Program Files"* tree is indeed not appropriate for receiving the software.

A standard installation of Galaad 3 requires 100 Mb of space available on your hard disk, plus some for storing your own drawings. The complete installation that is downloadable from Galaad website contains in a compact block the same components as the CD for a given language.

Finally, the SETUP programme performs the little additional Windows tasks, namely the addition of shortcuts for starting Galaad and the association of GAL files with Galaad so that opening them from Windows File Explorer will start Galaad automatically. If you do not require these additional features, uncheck the corresponding boxes at the bottom of the window.

Now click on the *"Next >>"* button.



It is now necessary to tell Galaad the general characteristics of your CNC machine, if one is actually connected to this workstation. If not, do not waste your time going into details, just validate this page as it is.

Please keep in mind that these parameters can be modified after Galaad has been installed. Any error can therefore be fully corrected later.

If you are installing Galaad onto a workstation that is connected to a CNC machine, select the option that will allow you to specify the type of machine and the communications port to which it is connected.

Click on the *"Next >>"* button one last time. A final dialogue box will appear, confirming the folder into which Galaad will be installed and showing the software licensing agreement, which you are, of course, already familiar with, having read the previous chapter.

Now click on the *"Install"* button, which will start the installation process and transfer files from the CD or from the downloaded block to your hard disk. This can take from a few seconds to one minute, depending on the performance of your PC. A small window shows how things are proceeding and a message will confirm when the process is complete.

Following this message you can start Galaad for the first time and thereafter by using the icon that has been placed on the desktop or the shortcut created in the *"Start"* menu. Your installation is now ready for use.

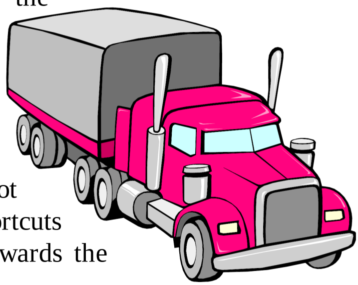
Note: **Galaad requires no drivers**, except for dongle keys dated 2016 and older. On the other hand, some numerical controllers, especially connected to USB ports, may require a driver for being recognised by Windows. These are provided along with the CD, but not installed automatically. When you plug in the machine and Windows asks you where the driver can be, here you can guide the system towards the CD. These drivers are not present in the downloadable installation.

❑ The lay of the land

Remember that the installation does not add any files other than to its own folder, except for the shortcuts on the desktop and in the *"Start"* menu. These are the classic LNK files destroyed by Windows when you delete the shortcuts.

Additionally, and if it can make you feel more comfortable, the only modification to the Windows Registry is the association of GAL (drawing files) and GLI (object libraries) files with Galaad, which allows it to be started by double clicking on *".GAL"* files. This association also adds the Galaad mini icon to GAL files, visible in the File Explorer and Windows dialogue boxes used for file handling, thus simplifying file identification. The above adds a grand total of three little keys to the Registry.

❑ Moving Galaad



If necessary, it is possible to move the installation to another folder with the help of Windows Explorer by simply renaming the target folder or cutting/pasting it to another drive or subdirectory, including via a USB memory stick. No problem with this, that software is easily moved. But do not forget to manually redirect the Windows shortcuts on the desktop and in the *"Start"* menu towards the new target location.

❑ Uninstalling Galaad

It was not felt necessary to create a programme to automatically uninstall Galaad, due to the simplicity of the installation and the absence of any files except those in the folder into which it was installed.

To remove Galaad, **it's a case of deleting its folder** using Windows Explorer or a similar tool. It's that simple. You can then manually delete the icons from the desktop and the shortcuts in the *"Start"* menu, which will remove all trace of Galaad from your hard disk. Never again will you be bothered by Galaad. It's your loss.

If you feel unhappy with the idea of three keys remaining in the Windows Registry, it is pretty easy to delete them: these keys concern the links between *.GAL files (Galaad drawings) and GALAAD.EXE programme; *.GLI files (Galaad libraries) and GALAAD.EXE; and between *.GAW files (Gawain turning drawings) and GAWAIN.EXE. These keys are of no use once Galaad has been removed from your hard disk, they will take up no space or system resources, nor will they interfere with other applications, but you have the right to remove even the smallest dust Galaad may have left on your computer.

On Windows Vista and older, open Windows Explorer and select the *"Folder options"* command (generally found in the *"Tool"* menu, depending on your Windows version), then *"File types"* page. Look for the GAL, GAW and GLI extensions and remove them with the *"Delete"* button. A greyed button means that the key is already erased. The next boot of Windows will clean this up for good.

On more recent Windows versions, clean-up is made automatically when deleting the target application. You have worried for nothing.

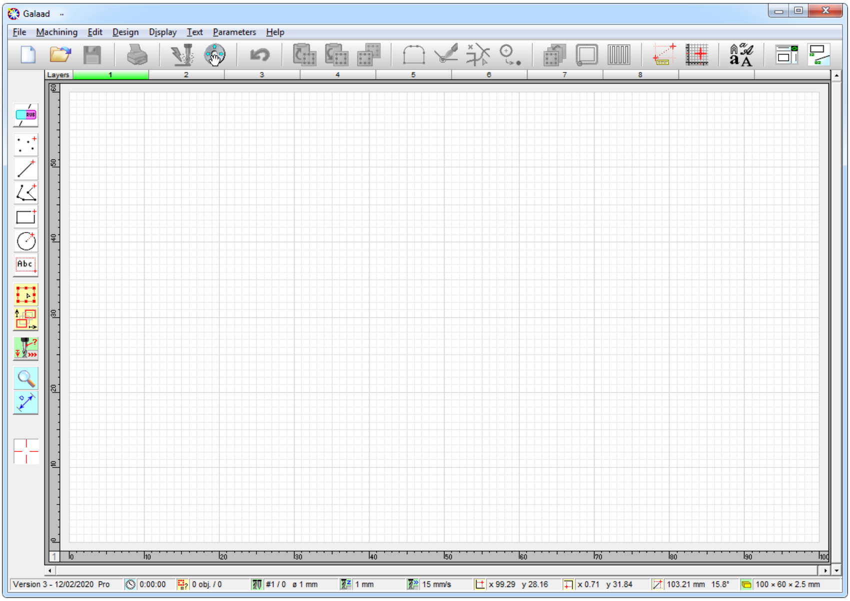
2

00010

LEARNING TO DRAW

Contact

When started, Galaad displays the drawing board of the last work you have made, exactly in the same state as you have left it (we will review that point). You will soon become more familiar with this central window for drawing and controlling.



At first sight it may seem that there is a great deal here, but you will soon get to know your way around. Each icon and each display area has its particular use. In addition, you can easily pick up the key functions available in the menus and drawing icons. All will be clarified later on in this manual when you learn how to restrict the information displayed in the workspace, but for now, let us neither anticipate nor concern ourselves with all that appears on the screen.

Small tip: your Galaad version, more accurately its date of compilation, is displayed at the bottom left corner of the window, at start-up or when no drawing function is in progress.

The working area is divided into five distinct zones.

- At its centre is your **drawing board**. This modest receptacle shows the result of your creativity, ready to be sent to the CNC to be manufactured with no further ado. It is in this area that your artistic talents will take shape with Galaad's humble contribution. That drawing can be viewed from above in plan, or by using the lateral and 3D views.

- At the very top of the screen is a classic **menu bar**. This provides access to all functions from file handling, through access to the CNC, to manipulation of the display, and is arranged in terms of functionality. Nothing really original here, apparently even in the southern hemisphere the menu bar is at the top.

- Immediately below is the equally classic **tool bar**. Each icon displayed here is a shortcut for a function in one of the menus, which saves having to navigate through the maze of submenus. Just to give Galaad a proper non-conformist feel, some of these icons also have *"fly out"* icons underneath, that pop up when the mouse meets its mother. However, a *"Touch Screen"* mode can be enabled in the workspace parameters.

- On the far left of the screen are the **drawing icons**. There you will find plenty to stimulate your creativity, providing an array of tools for constructing the objects that will allow you to create your drawing. When the mouse pointer is passed over these icons, a group of icons will *"fly out"*, offering a wider range of associated functions.

- Finally, relegated to the very bottom of the screen is the **display zone**. An Aladdin’s cave of information about the current drawing including coordinates, dimensions and angles, packed in higgledy-piggledy.

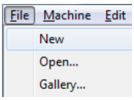
- You may display at the right of the screen a rapid access column that will be explained in detail at chapter 5 *"Advanced design techniques"*.

Note: Please keep in mind that the aim of this manual is not to teach you about the Windows interface, which is supposedly known. However, several little reminders will be given here and there, totally free, but all the same, do not hold out for an advanced technical course on the inner workings of Windows, which are varied, twisted and even sometimes perfectly logical.

❑ Baby steps

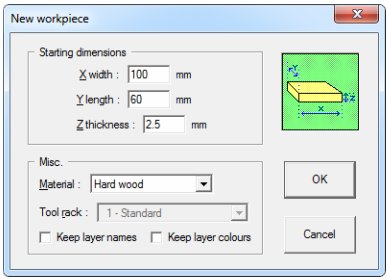
As you will discover, drawing with Galaad is not all that complicated and does not require much aspirin. However, it is necessary to remember that **the object of the exercise is to generate a toolpath for a machine cutter to follow** and not just a pretty drawing to be simply printed. Do not compare this with an image editor like *PaintBrush* or similar for *bitmap* files, which works on a mosaic of lifeless pixels instead of moving coordinates.

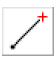
Galaad CAD module is a vector graphic editor, *i.e.* a line is constructed by defining the two points at its extremities then linking them; it is not simply an alignment of black pixels. This calls for greater precision, and for the work to be approached from a graphical point of view as opposed to an artistic one. If you have already used vector drawing software, such as *CorelDraw* or *Adobe Illustrator*, you will have no problem in becoming familiar with Galaad. The *modus operandi* is quite conventional.



Let us begin by opening the *"File"* menu and clicking on the *"New"* command. The current drawing will be removed and replaced by an empty board. Vertigo of the blank page.

Galaad then requests the **overall dimensions** of the new workpiece. Measure your material and enter the dimensions in the appropriate place. These dimensions can be modified at any time by using the following command, *"File / Material dimensions"* in the same menu. Check the values and click on OK. We will see more settings later on.





Let us begin with something simple, like a stupid straight line. Locate the *"**Line**"* icon in the drawing icons on the left hand side of the screen, ignoring the multitude of other icons that fly-out, the basic icon will do fine. Click on it and return to the drawing board.

The cursor will have changed from the oblique white arrow to a red cursor, complete with crosshairs. Move the mouse and the red cursor will follow its movement along with the crosshairs which indicate the current position on the rulers. In addition, the numerical coordinates are indicated at the bottom of the screen and updated when the position settles. Click somewhere on the board and release the mouse button.

This fixes one end of a line and as you move the mouse, you will see a moving line connecting the first point to the cursor. As well as the absolute position of the cursor, its position relative to the start point of the object is also indicated in both Cartesian and polar coordinates at the base of the screen. Position the cursor wherever you wish then click and release the mouse button.

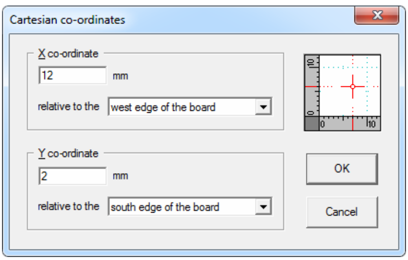
Galaad is immediately ready to repeat the operation for another line. Try again, but this time using another method; press and hold down the mouse button at the starting position, drag the cursor to an end point and release the button. The result is identical: **you may click-release, move and click-release, or click, move and release.** Choose the method that you prefer.

Continue drawing simple lines. The crosshair cursor visible on the board, press the **arrow keys** ←↓↑→ on the keyboard instead of the mouse: the cursor moves by one step on the side rulers.

❑ Simple coordinates

Moving on a little further. Rather than either using the mouse or the spacebar, press Enter and a dialogue box will appear that will allow you to directly enter the numerical coordinates.

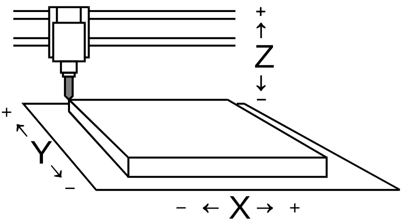
Just enter the value for the X coordinate then press the tab ⇆ key to quickly move onto the Y coordinate value below. Enter the required figure and click on the OK button with the mouse or simply hit Enter key to confirm and finish.



It is worth mentioning that pressing and simultaneously reverses the direction of movement and will allow you to return to the previous entry point. This is not just confined to Galaad but works in all Windows applications. The Enter **key validates the whole dialogue box** as it is.

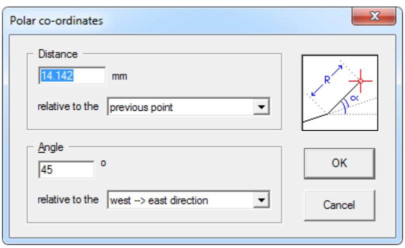
Note: the **orientation system** defines as *"west"* / *"east"* the negative (towards the left) / positive (towards the right) X directions; *"south"* / *"north"* the negative (towards the foreground) / positive (towards the background) Y directions; and *"down"* / *"up"* the negative (towards deep) / positive (towards retraction) Z directions. This corresponds to the **mathematical standard**.

This standard orientation remains valid for machine control. Warning: a greater value for the depth therefore corresponds to a lower Z axis towards negative direction, even if Galaad reads and displays the depth in absolute value.



You will no doubt have noticed that the dialogue box provides drop down boxes for setting the origin of the point entered. Therefore it is possible to define a Cartesian value relative to a point other than the origin of the board, (0, 0), which is, by default, situated at the southwest corner (bottom left). Bear in mind that if you enter a dimension that is relative to a given point located right/above, then it will probably be a **negative value**.

Returning to our cursor, you will now see it has been fixed at the point defined in the dialogue box and that the mouse click has been applied. You now know how to fix something at any **absolute position using Cartesian coordinates**.



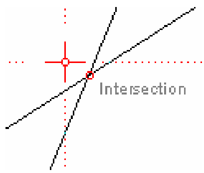
To enter a value in **polar coordinates** (for the second point only), you must press Ctrl Enter simultaneously. This will open a similar dialogue box, but for coordinates in the form (R,θ) as opposed to (X,Y).

As for Cartesian coordinates, the numerical values of the current position of the cursor are shown when the dialogue box is first opened.

At this stage of the apprenticeship, you know how to use the drawing cursor and define points numerically. Practice drawing by using some of the other drawing icons, for example rectangles and circles. However, do not spend too long at this early stage and stick to the white drawing icons as the others (yellow, green and blue) are not directly involved with the drawing process as you will see a little later.

❑ Snapping to positions

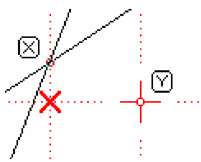
If you already have something visible on the board, perhaps a simple line, you will have noticed that locating the cursor in the neighbourhood of an existing object pops up a small red point. A short tip about the logical position will also appear.



By pressing the **space bar** Space on the keyboard, you automatically snap the cursor to the small red point, *i.e.* you directly validate the corresponding position. This is very useful when dealing with a lot of pointing tasks, particularly on polyline vertices, intersections, arc centres, *etc.* If there is no small red point, then the validated point will simply correspond to the current cursor position.

The **middle button** (or wheel button) of a three button mouse has the same function, rather than requiring a keyboard input even though the space bar is not too difficult to find. To perform a snap operation that would correspond to a click using the right mouse button, press simultaneously the ⇧ key (Caps arrow key). This is also valid for the mouse middle button (or wheel button).

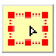
A little bit more complicated, but worth mentioning, you may undertake a **two-stroke snapping**, *i.e.* snap to the X coordinate of a small red point then move somewhere else on the board to snap another Y coordinate or the coordinate of the current cursor position.



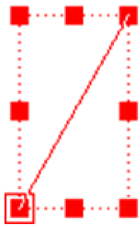
You only have to locate the cursor close to the first concerned point so that it is highlighted, then **press the X or Y key to temporarily store the X or Y coordinate**. A vertical or horizontal red axis appears across the board, but nothing is stored at this moment. You may redo the same operation using the same key to validate another point in case of error, or even cancel it by making an ordinary point. Move the cursor and highlight a new red point at another location and press the other Y or X key, the one you have not used yet. Galaad will immediately display in a dialogue box the position of the point that corresponds to this couple of X and Y coordinates that were temporarily stored, a position that you can then validate.

Handling objects

Let us stop scribbling now and see what can be done with objects that have been previously created.



Move the cursor to the left and click on the yellow selection icon, ignoring the others that fly out. The cursor returns to a white arrow and the last object drawn is framed by a matrix of eight red blocks. Try clicking on other objects here and there on the board: the red frame moves from one to another, the outline of the object also turns red for ease of identification. The drawn item within the red frame is then said to be a **selected object**, an expression which will be used very often in this manual.



This point is fundamental. With Galaad, it is necessary to **first select an object** and **then specify the action** that will be applied to it. With some other CAD software, exactly the opposite is true, and the action is specified first, then the target object must be plotted. Each approach has its advantages and disadvantages, which we are not going to discuss here. However, for many drawing functions, Galaad also accepts the opposite method when nothing is already selected.

An object is not modified by being selected, though its colour temporarily changes to enhance it. But it is possible to manipulate it with the help of the numerous tools that Galaad offers. Let the fun begin.

The first and most obvious thing that can be done to a selected object is to **delete** it. What could be easier? Press the Del key or use *"Edit / Delete"*. The object and the selection matrix disappear. You can **undo** this deletion using the ← key (Backspace) that is generally located just above the Enter key on an ordinary keyboard, or by calling *"Edit / Undo"* or the corresponding icon of the top toolbar.

The second and no less obvious thing is that the object can be **repositioned** at another location. Simply move the cursor to inside the selection frame, click and hold down the left mouse button and move the mouse. The selection frame follows your movements, and so does the selected object itself. Release the mouse button to place the object in a new position. The crosshairs show the position on the rulers and the coordinates in the display zone are updated during this operation.

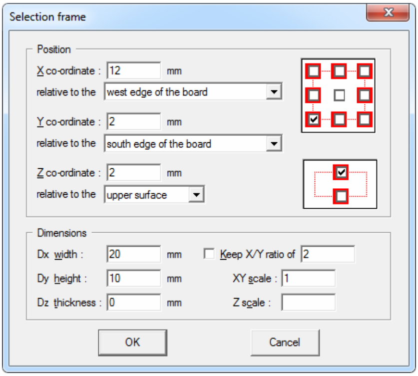
Now try clicking on one of the **red median blocks** of the selection frame with the left mouse button and whilst keeping it held down, move the cursor then release the mouse button. The frame is either enlarged or reduced in size, depending on the movement, and the dimensions of the object are also changed similarly. When using the **diagonal blocks** (*i.e.* the corners), you will see that all the dimensions change but the aspect ratio is maintained. This allows you to vary the overall size of an object without changing its general appearance.

Snapping to positions *(bis repetita):* when you move, enlarge or reduce a selected object, the presence of another object in the vicinity shows orange axes for Cartesian alignment, near its borders or centre. To automatically align the selected object on an orange axis, just **press the** Space **bar** on the keyboard without releasing the mouse button. The X or Y keys make a partial snap so you may align only one axis when both are displayed. Once an axis has been snapped, the object moves only along the other axis until you release the mouse or snap to another position.

You will no doubt remember that the Enter key opens a dialogue box to allow a position to be defined from the keyboard. This little feature is also available for all drawing and object manipulation functions in Galaad. Now press the Enter key.

This time a different dialogue box appears in the middle of the screen. You can define the **position** of your object in the upper part and its **dimensions** in the lower part.

By default the XY position of your object, shown in the dialogue box, refers to the southwest corner of the selection frame, but you can also use any of the other reference points by clicking on the **red blocks**.

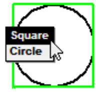


You will quickly realise that Galaad does not really like it if your objects extend beyond the edge of the board, especially when entering a coordinate from the keyboard. Since your board represents the raw material to be machined, it seems somewhat logical that creating toolpaths outside of this workpiece makes no sense.

At this stage of the proceedings, you know how to **draw** basic objects, **snap** to existing points, enter **dimensions** for their positions, **select** objects, **delete** and **manipulate** them.

Now we are going to select several objects simultaneously so that they can be manipulated as a group. There are several ways of doing this. The first consists of clicking on our selection icon (if the crosshair cursor is currently displayed) then defining a rectangular area of the board. To use this method, press and **hold down** the left mouse button, drag the cursor a little way eastbound and release the button: a red rectangle has framed the area. **All objects fully inside this area have been selected**. If instead you move the mouse westbound after clicking, the rectangle is green and **all objects at least partly inside the area** will be selected. If you only caught one or none at all, try again. You now have several objects selected that can be manipulated as if they were one: position, size, delete, *etc.*

Another way is to start by selecting one or more objects, then **press and hold down** ⇧ on the keyboard and select some other objects. Unlike earlier, the new objects are selected without deselecting the ones already selected. You can continue like this until all objects are selected, including selections from zones. Reciprocally, **if you click on one object already selected among others with this key ⇧ pressed, it will be removed from the selection**.



On the other hand, when you click a location where several objects could be selected (or deselected), Galaad will pop up a small menu to help you do your shopping. Let us mention that there are many other selection methods, particularly with an automatic filter, and also a locking function that prevents selection. That will be described later on.

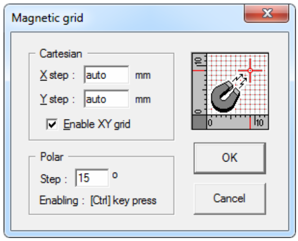
Tip: when one or several objects are selected, pressing **cursor arrows** ← ↓ ↑ → on the keyboard directly move the selection frame – including the objects – by one step on the rulers (even if these are not displayed), as when moving the drawing cross cursor. This may eventually help you adjust objects more accurately than with the mouse and leads us to the functions of the magnetic grid.

❑ The magnetic grid

As you will certainly have noticed, both the drawing cursor and selected objects can only be moved in discrete steps that correspond to the **graduations of the rulers** along the edges. This is a common feature of vector drawing software, which provides a grid of invisible points that are not linked to the graphical resolution of the display and cannot be addressed directly by pixel coordinates. Although this grid is invisible, its influence on the drawing is no less effective.

The default value for the graduation is 1 mm unless your board is very big or very small. At first sight it may appear that it is not possible to construct or position the drawing cursor or an object anywhere other than on a grid point. However, please note that **numerical values entered from the keyboard are never rounded**. The smallest step is 1/1000th, which should be sufficient for most needs. Galaad considers that if you have entered a numerical value for a position, then this is where you wish that point to be and the magnetic grid is not applied, hence it is possible to select any position on the board even when the grid is active. The entered position remains unchanged.

If entering a numerical position is not suitable for a particular situation, then simply change the grid step size by using the function *"Design / Magnetic Grid / Set"* from the menu bar (a quick shortcut also exists among the command icons of the top bar).

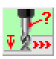


A small dialogue box allows you to independently set the X & Y steps, *i.e.* the values to which positions are rounded up/down. If you leave a **value empty** (or *"auto"*), then the rounding will be to the smallest graduation on the ruler that varies with the zoom. This is probably the best mode of operation because you can zoom in/out and stop worrying about the grid. The polar grid rounds the slope angle of the line that is currently under construction, when pressing Ctrl key.

You can also completely disable the grid and work directly at a pixel resolution, but this is not recommended for precision work, especially since the step will vary along with the zoom. This grid is there for your convenience and using it will certainly make life easier. Do not neglect it.

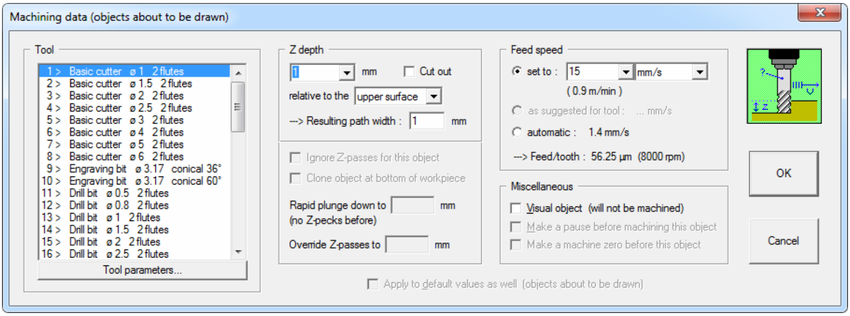
❑ *De profundis*

Your artistic creations must not be made without forgetting that the ultimate aim, even if a long way away, is for Galaad to drive a CNC milling machine. This naturally requires additional information, namely the machining depth, feedrate and details of the cutter tool to be used. Those three define the extra dimensions that Galaad needs to manage, compared to a classical 2D CAD system that is presumed to talk to a printer. They are all reachable simultaneously.



Click on the green icon on the left of the board, again without worrying about the other fly-out icons.

A new dialogue box will then pop up, which will let you enter the **cutter tool** that will be used to manufacture the objects that will be drawn hereafter, and their milling **depth** along with their **feedrate**.



Enter a new machining depth for your chosen objects. Note that you can select the *"Cut out"* option to link depth to actual board thickness. Further options are also accessible from this dialogue box, that we will see later on.

If you are not yet familiar with calculating feedrates, then let Galaad do it for you automatically. By default, the software calculates an approximate value and takes into account the hardness of the workpiece, the physical characteristics of the cutter and the depth of every stage. Over time you will gradually learn to estimate feed speeds without running the risk of breaking cutters, often caused by going too fast. You will soon develop a feel for the correct values required. Conversion of the entered feedrate into classical units like m/min or mm/s is displayed below. This may help you feel comfortable with both these units which are the most commonly used.

When you validate this dialogue box by clicking on OK (or pressing Enter), the **selected objects** immediately get the new values indicated. If no objects were selected, then your change did not go to the bin and will affect the **next objects** that you are about to create with drawing icons and until you call up this dialogue box again.

Important corollary: **you can have as many different depths and speeds as there are objects on the board**. These two parameters are completely independent of each other and not connected with either a given tool, colour or milling pass as with some other 2½D CAD/CAM software. In addition, you can have as many objects on the board as you like, limited only by the memory size of your computer, which can be huge. The practical limit is likely to be your creativity. Let us also mention that you can even draw 3D objects, which indeed have a varying depth, as we will soon see. But let us not anticipate because it might then become a bit more complicated, and the purpose of this chapter is just help you get started.

You now know how to **draw** objects, **delete** them, **reposition** them, **enlarge**/**reduce** them, and finally precisely define their **milling parameters**. So you are ready to use the milling machine that has been impatiently fidgeting and pulling on its cable. One last effort, a minor detour via the zoom, and we will be there.

❑ Zoom

It is useful to be able to enlarge part of the board to check or adjust the objects. A set of functions is going to help you there, for now we will limit our review to the first one: you may have seen it, this blue icon which looks familiar. Click on it and then you can **define an area** of the board to enlarge, or a **simple point**, around which Galaad will apply a magnification factor of 2. In the latter case, it is sufficient to click and release on the same point of the board.

Small tip: it is possible to perform a **fast zoom** using the ⇞ (or PgUp) key. Galaad will automatically enlarge the zone around the mouse cursor without aborting the drawing operation that is in progress. This can be reiterated, and you can zoom out using ⇟ (or PgDn).

If you have a wheel mouse, which is probable nowadays, a **backward or forward rotation of the mouse wheel** calls the same zoom functions without touching the keyboard. This is often helpful to ease snap operations in crowded zones: several fast zooms magnify the location where a snap point is available, the snap is performed then a zoom out is made if necessary to enter the next point of the object under construction.

The zoomed window can be moved with the mouse middle button (or wheel button): just click, hold down and drag the view in a given direction. This possibility remains valid **even while drawing**, when the red crosshair cursor is active.

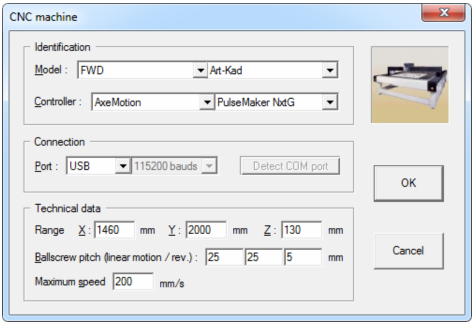
3

0 0 0 1 1

LEARNING TO MILL

Roadworthiness inspection

Before firing up your machine, it may be worth casting an eye over the settings within Galaad. There will inevitably be some misunderstandings, for example in cases where parameters are out of range. Proceed immediately to the menu and select the function *"Parameters / Machine / Basic data"*. This should open a dialogue box for a model which displays a summary of the key technical characteristics of your machine.



If all is well, you will find that these are the same as you used when defining the CNC during the installation of Galaad. If this is not the case, then it is not too late to change any of the parameters displayed.

It is assumed that you know the model of your machine, or at least the type of controller it uses. If not, then it is time to contact your distributor after first of all carefully checking that the model number is not marked somewhere, possibly in the most inaccessible of places. **Entering the wrong machine type is not a serious problem**. Neither your machine nor your computer will suffer. In the worst case, it will not respond and simply ignore you. If in doubt, then give it a try. Daring!

An important detail, you have probably already installed a cable between your computer and your CNC. If not, Galaad could find it difficult to control the machine. Progress in wireless information technology is certainly rapid, but it is unfortunately still necessary to have physical connection between most machines and Galaad. And whilst on the subject, certain old machines require a special asymmetrical cable (with hardware handshake loops at machine end) of which the correct end must be connected to the PC and the other to the machine.

To send information through this cable, Galaad must at least know which **communication port** to use. Even if fashion is in the USB, still many machines receive their instructions down a serial RS-232 link. It is important to tell the software to which port the cable is connected. On an old computer, the machine should be connected on COM1 or COM2 serial port in most cases. On a recent computer, it is more probable that you will use a USB-Serial converter, which will be seen from software applications as a classical serial port, but with its number above COM2. In its drawn-down list, **Galaad displays an asterisk beside the existing COM port numbers**, real or virtual. It is up to you to try, and remember that it will not cause collateral damage to the computer nor installed applications, even those already running.

You will notice that the dialogue box for configuring the machine connection allows you to tinker with the **baudrate** of the connection. If you have chosen an existing model from the list of known machines, it is best not to meddle with it. Using a higher communication speed will not make your CNC run any faster.

Possibly you have **no machine** connected to your PC. In this case, select *"No machine"* as model number and *"None"* as communication port. The machine will be virtual but you can follow the logical machining process on the screen all the same.

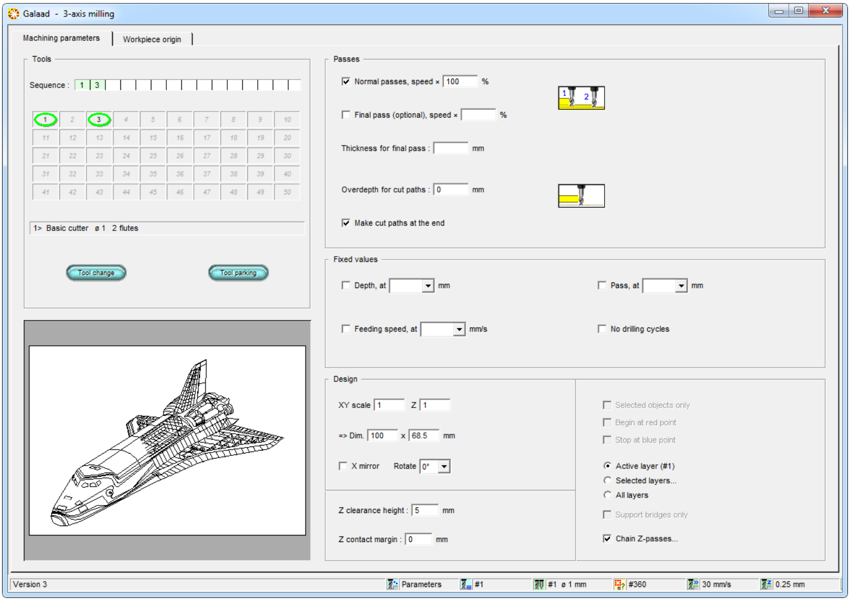
Validate this dialogue box by clicking on OK if you have changed anything, otherwise still click OK just in case.

❑ Guided tour of the launch pad

Your board contains the wonders of your creative genius; the parameters of your machine are correct; the launch window is clear and your seat belt is fastened. There is no longer any reason to delay the milling. Authorisation for ignition and blast off is found in *"Machining / Standard machining on 3 axes"*. Clear for lift off.

A minor digression, already: check that you have not forgotten to turn your machine on. The old adage *"it might work better if you turn it on"* sometimes makes miracles. Fortunately, polite machines are in the habit of letting you know when they are powered up. If nothing is illuminated as such, then you have Galaad's permission to remove your seat belt and investigate. Incidentally, certain commands also require the **drives to be powered** and the **safety covers to be closed** so check these as well.

In the meantime Galaad has collected a large amount of information and displayed it in the window shown below. Do not panic and do feel free to experiment, as you cannot do any harm, yet!



At this stage we are not going to examine all the options available in this window. You will have already noticed that your drawing is also displayed here. The only machining parameter that interests us at the moment is the tool sequence, which is displayed above the drawing.

If you have drawn objects with several different tools, the corresponding blocks in the tool matrix will be highlighted. In the absence of any defined sequence, the boxes are a uniform yellow and await your selection. Simply click on the tools required in the order that you wish them to be used. When a tool is selected for use it is circled in green and those yet to be selected have a red cross through them. Once a tool has finished its work, it is crossed out in yellow.

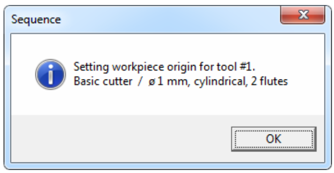
This function is indeed important: **it is up to you to specify the sequence in which tools are used and objects machined**. Lacking a user defined sequence, indicated by all the boxes remaining yellow, the tools will be used in their numerical order, though these numbers are non-significant.

A summary of the tool characteristics is shown below the tool selection matrix. When the cursor passes over a tool that has been used in your drawing, its parameters are shown, whether or not it is selected for use. The rest of the time it shows the characteristics of the first tool to be used.

It is possible that you have only used one tool. In this case, the tool sequence cannot be used to control the order in which the objects will be machined, so move on to the next phase. Mount the appropriate tool into the spindle and click on the *"Workpiece origin"* tab, or press Enter.

❑ Workpiece origin

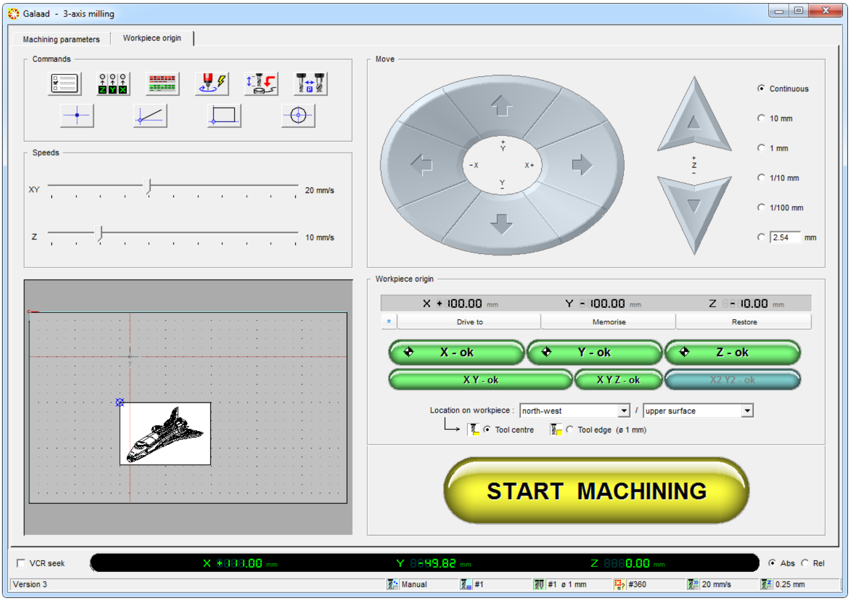
The machining parameters page disappears and is replaced by the page for setting the workpiece origin, which is also packed with control options. A small message box appears from nowhere, to remember which cutter tool is to be used in the process. Click on OK, you have no choice.



Communication between your favourite software and your favourite CNC is opened upon accepting the above message. This initialisation may take several seconds depending on the machine and if it has just been rudely awakened.

If this initialisation fails, a small message box will appear informing you on the nature of the problem. Galaad will spend up to ten seconds trying to initialise communications with the CNC, and in case of failure offers to retry. But first of all attempt to establish why it did not work, using the following check-list: Is it switched on? Is the cable connected correctly between the computer and CNC? Are the machine parameters set correctly in Galaad?

Assuming that all is well and communications are successfully established with the CNC, Galaad may warn you that it needs to make the machine perform a short reset so as to find the zero point of each axis. Let us return to the workpiece origin.



It is now necessary to tell Galaad where to find the workpiece on the bed of the machine. You can see it, certainly, but Galaad cannot. It only knows the dimensions and the toolpaths that it must follow. Therefore we must **give a reference point** XYZ and **tell it precisely where to find the workpiece in relation to this point.**

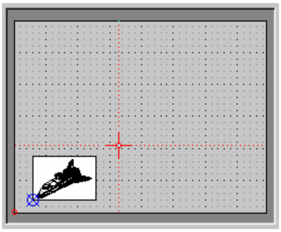
In fact, do not forget to mount the workpiece, but if you just want to have a dummy run without breaking anything, then that is fine.

The process consists of driving the machine manually, one axis at a time, until the tool is situated at an edge or on the surface work, as explained below. Use the buttons arranged in a circle, situated at the top right side of the control window, for X and Y motion, and the triangular buttons for Z (up and down) movement. When you press a button, the movement is continuous until the button is released. You can use the small arrows ← ↓ ↑ → and + - keys on the keyboard to produce the same movements.

Important: the **right mouse button moves axes at slow speed** (or the Ctrl key). Very helpful for finishing the workpiece origin approach on a given axis. If you need to make a movement of a fixed distance, use the **radio-**

buttons on the right hand side to select the distance. The movement stops when you release the button or when the distance has been covered.

Secondary, but convenient: the **mouse wheel** moves by increments of 1/100th of millimetre (or 1/1000th of inch depending on the unit) the last axis that moved, provided that this occurred less than one minute ago. Very useful for fine approaches, especially for Z.

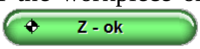


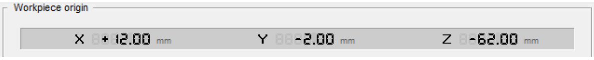
In the preview window, a cursor with crosshairs moves at the same time as the axes. If you do not have a machine connected, it is the only thing that moves. You can also set a position by **double-clicking directly** in the preview window. The coordinates of the position will be shown by the LED display immediately below.

You can also enter a **numerical position** by clicking on these LED displays, or by pressing one of the X, Y or Z keys on the keyboard.

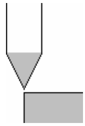
However, the aim is more than just to make the machine move along its axes as, although Galaad may always know where the cutter head is, it still has to be told where to find on the machine flatbed the workpiece that is going to be milled.

Move the X and Y axes so that the spindle is positioned above the workpiece, somewhere towards the middle. Then use the Z down button to lower the cutter until it is a small distance above the surface of the workpiece, say 1 or 2 mm. Go slowly, cutters are expensive and accidents easily happen. Next, use the right mouse button or select the radio button for 1/10th mm steps and carefully lower the cutter to the position where it is just gently touching the top surface of the workpiece, but not actually cutting into it. Eventually perform the final approach with the mouse wheel. A thin sheet of paper placed under the tool helps a lot: when you can no longer move it without tearing it, then you are done.

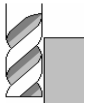
You have now found the Z value of the workpiece origin. Click on the green button to **validate the Z position**: 



The Z position is then uploaded to the workpiece origin box. Now re-select continuous movement and lift the cutter a small amount, if Galaad did not already do it, then bring the cutter to the left hand side (west) of the workpiece.



When close to the edge, reset the step size to 1/10th mm, or use slow motion with the right mouse button, and carefully position the cutter so that the point is directly above the edge of the workpiece. It often helps to lower the cutter to improve accuracy.

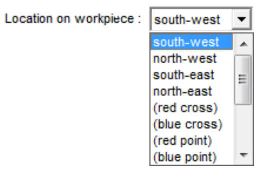


If using a cylindrical or hemispherical cutter, it is easier to find the edge of the workpiece with the side of the tool. Galaad knows the diameter of the tool and can automatically correct the position.

In the latter case, do not forget to select the option *"Tool edge"* instead of *"Tool centre"*, situated just above the large yellow start button. Galaad needs to know which method to use in order to make the correct adjustment. The tool profile is not necessary.

You have now found the X value of the workpiece origin. Click on the corresponding green button: 

Finally, repeat the above operation but for the Y axis, and using the south edge of the workpiece, again confirming the position with the appropriate button. With a conical cutter, it is possible to save time and find the X and Y positions simultaneously at the southwest corner of the work, then confirm both positions simultaneously. It is possible to do all three axes together, but **it is recommended that the Z axis is set towards the middle of the work** rather than at a corner, as it is more accurate.

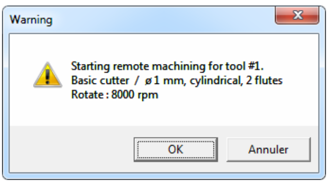


Without going into detail, note that you can also use several other positions on the board, besides the southwest, for the reference point. Simply tell Galaad by selecting one from the combo box below the green buttons.

When you confirm the position of the workpiece origin, indeed Galaad will know the coordinates and therefore the position of the origin, but it is still necessary to tell it where this is in relation to the workpiece, *e.g.* at the southwest corner. Generally either the southwest or the northwest corners are used, but there may be times when another position would be more appropriate. The important point is that the origin used when drawing is consistent with the origin set on the machine. If not, Galaad is going to mill by the side of the workpiece, eventually on the clamps with a very unpleasant noise of tortured metal. Likewise, if you have approached the workpiece top surface and validated *"Z - ok"* whilst indicating that it was the machine bed, then Galaad will mill the air above the workpiece. Much more serious is the case where you indicate that you have approached the top surface whilst the tool was actually touching the machine bed. In that case – unfortunately not so rare – you are going to mill deep in the machine bed through the workpiece and probably pronounce big swearwords against a stupid piece of software which just applies your guidelines.

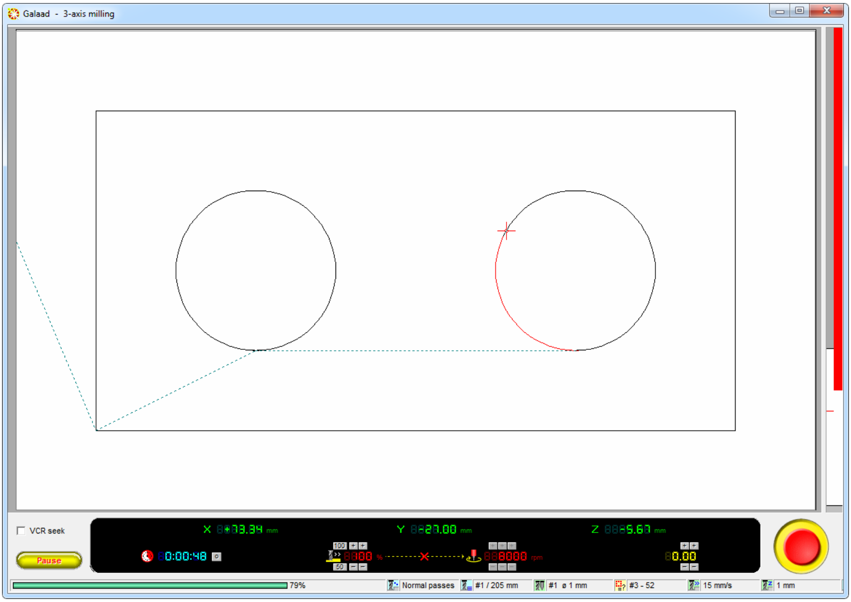
❑ Blast-off

The workpiece origin has been set for all three axes. Galaad now knows all that it needs to know; namely where to find the origin point and where to find the material in relation to that point. Fasten your seat belt, lift-off is imminent. Now you can click on the big yellow button labelled *"Start Machining"*, the one you've been itching to press for ages.



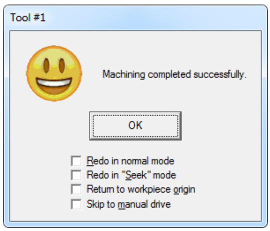
The cutter is retracted if it is low, and one last message warns you that machining process is about to start. You may still cancel the process here, but this is your last chance to chicken out.

Pressing OK starts the automatic machining process straight away. If your spindle does not start automatically, now would be a good time to switch it on and set the required rotation speed. As soon as the message is acknowledged, the tool is moved towards the entry point of the first object to be milled, then plunges down to the contact with the workpiece top surface, drills into the material and starts feeding horizontally, then lifts up for reaching the next object. The sequence followed corresponds to the order of drawing, but we will see that there are several ways of tinkering with that sequence. We will try to remember to say a word about it later on.



Several small buttons at the bottom of the window allow you to override the **Z position** or the **feedrate** in increments.

When the machining cycle is finished, Galaad stops the spindle and sends the tool back to its park position. The workpiece can then be removed from its fixings, unless there are other cutters yet to be used. In this case, the software returns to the workpiece origin window.



Your workpiece is machined, and you now know how to do it. You can return to the other parameters and advanced machining functions later. For now we have covered the essential points of the process.

4

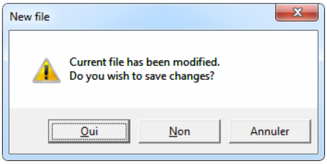
00100

LEARNING TO SAVE FILES

Current drawing

Before moving on to more elaborate creations, temporarily quit Galaad. Click on the window close button or use the menu command *"File / Exit Galaad"*. Galaad quickly shuts down, this software certainly knows who is the boss here.

Now restart Galaad by double clicking on the icon on the desktop or from the Start menu, by using *"Start > Programs > Galaad"*. After using Galaad a few times you will soon realise that when it starts it restores itself to the **same workspace conditions** that existed when it was last closed, with exactly the **same drawing** on the board. This is a peculiarity of Galaad that allows it to be closed without first asking you so save your current work. You can therefore work on the same drawing many times before having to save it.



On the other hand, if you call for a new file or open an existing one, then you will be given the option of saving the current drawing first or indeed risk losing it forever.

If due to principle, habit, shared resources or some other reason you are unhappy with the automatic save feature, then you can disable it using the menu function *"Parameters / Workspace / General settings"*. The choice is yours, but remember that it is activated by default when Galaad has just been installed.

The automatic timed backup during normal work, set in *"Parameters / Auto-save"*, are stored as part of the working environment and not in a named Galaad drawing file.

In addition, double-clicking on a GAL file will automatically start Galaad with that new file. However, if the current drawing has not yet been saved, then Galaad will give you the kind opportunity to save it before proceeding any further. In fact, this double-click on a file is equivalent to starting Galaad and opening that file. Therefore you can fearlessly double-click on any GAL file that appears in Windows Explorer. This is probably not a very useful feature, but it is well worth mentioning.

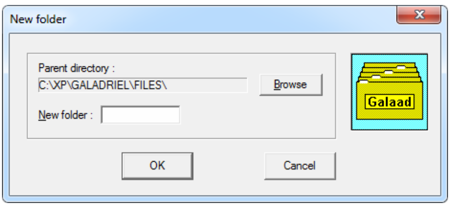
❑ Files and folders

A complete installation of Galaad, complete with all the examples, is spread across several folders. Organising the folders in the directory tree on your hard disk into a clear logical structure is a good way to ensure that your files will be easily found in the future. This is preferable to simply saving your files onto your hard disk, haphazardly, like throwing old clothes into a drawer. Both Windows Explorer and Galaad provide tools to help you organise file storage. When opening a file the dialogue box will also let you delete and rename both files and folders at will.

As far as Galaad is concerned, it suggests that you save your drawings within a sub-folder named **FILES**, in the software installation folder, which itself already contains several sub-folders of examples. Without doubt the best thing is to follow this lead, also placing your drawing folders in this **FILES** sub-folder, that way they will appear at the same tree level as installed samples. This may help copy a complete installation from a workstation to another one, including drawing files. Also, it appears to be more convenient when using the gallery. Thus, Galaad does not impose any disk management model, the location of your folders and files remains your choice. Galaad is polite and will always send the file selection box into the last disk and directory location you used, to avoid spending time climbing up and down the disk tree, be it local or distant. Memorising the last default location remains valid for most file and libraries functions.

The function *"File / New folder"* prevents you from having to leave Galaad to use Windows Explorer for creating new folders.

Here, you are asked simply to provide a name for the new folder. This will then be added into the list of available folders. You can also, if you wish, choose a different path for it.



There remains nothing more for you to do than to save your many future drawings into this folder, whose name will now appear on the list of folders available. You can obviously create **as many folders as you wish,** but be sure to arrange them into some sort of logical order. This is only a tool. To delete a folder, use Windows Explorer or the *"Open"* and *"Save as…"* dialogue boxes that will allow this to be done by selecting the folder or file then pressing the Del key. In the rubbish bin nothing is lost – as long as it is not emptied.

Multiple files

Since the January 2025 version, it has been possible to have several files open at the same time and to skip from one to another, particularly for copying and pasting objects. This is set in the general work environment settings (*"Parameters"* menu). Tabs appear at the bottom of the drawing window.

Saving your data

Even if there are sadder things in life than a big job lost or an unrecoverable configuration due to a PC or hard drive that has given up the ghost, it is still very painful when it happens. Do not expose yourself to this kind of problem and make regular back-ups. The procedure is anything but complicated: simply copy the entire software installation directory to a USB memory stick or network drive if you have one. For example, if Galaad is installed in the *"C:\Galaad"* directory (default location), you open Windows File Explorer, point to this folder, right-click and select *"Copy"* from the context menu. Then point to the USB stick, right-click and select *"Paste"*. That is all there is to it, and it is as simple as that. If you do not make any back-ups, then there is no need to call on your gods or technical support to find your lost data, because unfortunately they cannot do anything for you.

5

0 0 1 0 1

ADVANCED DRAWING TECHNIQUES

This chapter is intended to introduce you to the subtleties of the drawing techniques with Galaad. It is recommended that all users, even experienced ones, read the section covering the basics before proceeding any further, as it contains several useful tips that are often overlooked, so obvious that you do not even notice them.

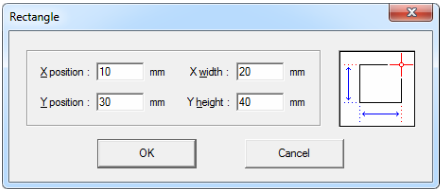
❑ Numerical coordinates

An important feature when designing parts that are to be manufactured is the ability to specify numerical positions and dimensions accurately. You will not have forgotten that pressing the Enter **key** opens a dialogue box that allows the **position of the cursor** to be set without being affected by the current magnetic grid.

Small tip, by the way: you may enter any numerical value using a **mathematical expression**, *e.g.* a chain of dimensions 12+31.2+4*6.35 written like this in its edit zone. Hence you do not need to search for Windows calculator and copy/paste your result in an edit zone. This subtle facility applies to any numerical value you have to type. Please refer to the chapter *"Special functions"* hereafter, for the syntax and the function identifiers that Galaad maths formula analyser can understand.

Now imagine that we wish to construct a rectangle, 65.4 × 32.1 mm, with the southwest corner at (12.3, 45.6) mm. The first point is easy, simply press the Enter key and type in the coordinates. The second point is slightly more difficult as we need to calculate the coordinates of the corner diagonally opposite by adding the dimensions to the coordinates of the point already entered. Galaad understands an arithmetical expression, here an addition, but should do this menial work for you.

This example of a simple rectangle was perhaps not a good choice as it is possible to position the second point using relative dimensions in two ways. Firstly by entering the dimensions, indicating that these are relative to the first point, and secondly by drawing the rectangle in roughly the correct place, then selecting it and numerically modifying the dimension of the selection frame. The effort remains minimal, but all the same there is room for improvement and there is a better way. Try the following: construct a rectangle by entering the first point whichever way you like (with the mouse or numerically) then press the Enter key to position the second point.



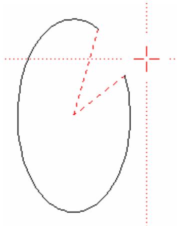
Now, instead of displaying a dialogue box for the coordinates of the second point, Galaad will have anticipated what you intended to do and provides a dialogue box to input the rectangle size directly, along with its starting corner position. Simply enter the width and height of the rectangle, which can be negative if westbound and southbound. Actually Galaad follows your work and attempts to anticipate your requirements, frequently offering context sensitive responses. Consequently, on constructing a rectangle, after entering the position of the first corner, there is a strong probability that you would prefer entering the actual dimensions of the rectangle rather than the absolute coordinates of the second point. Likewise, when constructing other shapes, such as a horizontal line or a circle, Galaad will try to pop up a dialogue box that looks appropriate to the occasion. However, if you still wish to **enter the numerical position of the pointer and nothing else**, then simply use the Enter **combination** which will open the ordinary dialogue box for a classical XY position.

❑ Partial constructions

Constructing a complex geometric figure usually requires several stages. These stages are different in the case of a figure that is constructed progressively, such as an ellipse. First the base is defined and then the form is added to produce the final result.

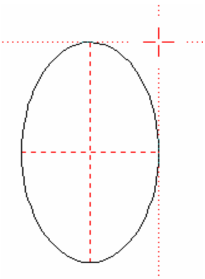
It may be that you need to **interrupt the construction process** before the final stages, quite simply because the shape that you are constructing only requires the basic part of the icon full function. Take, for example, an open elliptical arc starting from a centre point, which has four distinct stages and requires the following data: firstly the centre point; secondly the major and minor Cartesian radii (X and Y); thirdly, the angular start point of the arc and fourthly the angular end point of the arc. The result should look like an elliptical arc.

Now imagine that you wish to construct a centred full ellipse (*i.e.* completely closed). There is not a drawing icon specifically for this simpler construction, so steps 3 and 4 of the above process will have to be skipped. Consequently we need to interrupt the construction without losing what has already been drawn and without aborting the whole process.



Furthermore, some drawing functions are repetitive and have no definite last stage. The simplest example is a polyline, which requires a start point then an indefinite series of next points. It is up to you to decide which point will be the last. When every new point is clicked, the software adds it to the polyline and awaits the next point, until you decide to indicate the final point by clicking with the right mouse button. There are other repetitive figures that follow this pattern, such as Beta-Splines (which have a maximum number of 256 nodes) and Bézier curves which have no such limits on their nodes.

Remember then that entering the **last point** of any endless construction (like a polyline) is simply a case of clicking with the **right mouse** button. This is also true for shapes drawn using a progressive construction technique. If you wish to stop drawing a given construction before its last stage, use the right mouse button as in the following example.



In the case of a closed ellipse it is simply a question of fixing the X and Y radii with the **right** mouse button, which terminates the construction process there. Note that the start and end points will be at the standard trigonometric zero position, *i.e.* at 3 o'clock.

It is also possible to stop the process one step further on which will allow the start and end points to set at any angle required.

The construction of other shapes also works in a similar fashion, for example, a star (before defining the internal circle), radii (before the internal circle and angular spacing) or a spiral (before the internal circle).

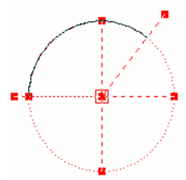
In any case, please remember that you can end the construction of a shape at anytime, without losing what has already been drawn, simply by clicking with the right mouse button to indicate the last stage.

❑ Editing object geometry

Nobody is perfect and sooner or later you will find that you need to modify a drawing without starting from scratch. A typical example is a Bézier curve where the positioning of a node has an effect on the previous one and depends on the distance between them. Almost certainly there will be times when the result is not quite what you had imagined and you will wish to make adjustments to obtain the desired shape.

Four types of object retain the ability to be edited geometrically: **arcs** (of circles or ellipses), **Beta-Splines**, **Quadra-Splines** and **Bézier curves**. The last three still show their nodes after being constructed and can be edited immediately, however, this does not mean that they cannot also be edited at a later date. Arcs do not show these straight after construction, but can nevertheless be edited later on.

Construct an arc or an ellipse, either opened or closed, using the appropriate drawing icon of your choice. Then press either the Esc key or select the object to leave the drawing mode. Now click inside the shape with the **right mouse button** and a special edit frame will appear around the object.



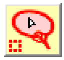
Note that the **mouse double-click** does the same if the object below the mouse is alone. You can then modify the shape by moving the red blocks, this allows the centre, radii, start and end points to be edited. Alternatively, you can press the Enter key to open a dialogue box and access the parameters numerically. When you are finished, select another object or simply press the good old Esc key.

Note that, in the case of Beta-Splines, you can vary the **weight** (power of attraction) of any node by simply pressing the + or - keys. You can also use the Tab key to move between them.

For other objects, that are not editable in this way, clicking with the right mouse button will select a segment or a point. See the specific paragraph on this feature.

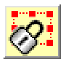
Selections

We saw in the previous chapter how to select an object by clicking on it. You can also select the contents of an entire rectangular area by delimiting it. Either locate the mouse pointer on the left-hand side, **click and drag to the right** while holding down the button: the area covered appears in pale red: you will select all **the objects that lie entirely within this area**. Objects that protrude even a little will be excluded. Either locate the pointer on the right-hand side, **click and drag to the left**: the area covered then appears pale green: you select **the objects that are entirely or partially within this area**. You only need to touch a small part of the object for it to be selected.



You can define a non-Cartesian area to select objects within it. Use the *"**Lasso selection**"* icon to define such an area. In the same way as above, you can move the mouse to the right or left of the starting point to define a red or green zone and grab objects that are either wholly or partly within it.

Locking objects

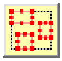


This old but nonetheless useful function allows objects to remain visible without being selected and hence not changed in any way. Simply select an object – one last time – and click on the locking icon, shown here. Once locked, **an object cannot be selected**, nor can any part of its composition. This is particularly useful when you wish to work on a group of objects without affecting others that are within the same area. The locked objects remain visible (displayed in pink) and will be machined normally, but cannot be selected.

To unlock objects, it is necessary to use the menu command *"Edit / Unlock"* or click on the icon again for unlocking all of part of them: if no object is currently selected when you click on that yellow locking icon, then Galaad understands that, instead, you want to unlock objects that you are going to point one by one or in a zone.

❑ Associating objects

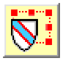
Another classic selection feature is the ability to group some of the objects together then treat them as though they were a single entity. One for all and all for one. This function can be found, quite logically, among the other selection icons. Draw several objects and select them all, then click on this icon. Henceforth, they will act as a single object and **selecting any object of the group will also select all the others**. Note that text blocks use a different system in order to maintain characters together. However, text blocks can be associated with other text blocks and also non-text objects.



To break the association, it is necessary to use the menu command *"Edit / Ungroup"* which provides a range of self-explanatory methods. If no object is currently selected, the association icon switches to the reverse operation and lets you point to the targets. Finally if all selected objects are already associated, then clicking on the icon breaks the whole community.

❑ Protecting objects

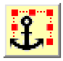
It can sometimes be useful to provide an object with a limited degree of protection, to avoid leaving it alone on the torture table but still keeping it available for selection and position. The protection function, with its little shield, allows only limited operations to be performed. This feature prevents an object from being deleted or subject to any manipulation other than being positioned or dimensioned, neither of which actually change the basic shape. Galaad considers that these operations do not alter the actual geometry of the path and therefore allows them. On the other hand, **the path can neither be cut, incised, reshaped, nor have additional segments welded to it**.



To unprotect an object use the menu command *"Edit / Unprotect"* or click once more on the icon. Protected items can be identified on the board with a small shield located near to their start point, by using the menu command *"Display / Trace / Protection"*.

❑ Anchoring objects

Inevitably there will be times when, in order to position them, it is not very convenient to create a group of associated objects. Conversely, there may be times when you wish to fix an object to an absolute position even though it belongs to a group and the others are to be moved. So, Galaad provides two adaptable methods for this: the **absolute position of an object on the board** can be fixed or, alternatively, the **position of an object relative to other objects** can be fixed, without creating an association.



The absolute anchoring of an object freezes its position and prevents any further attempts to move it. Draw two objects and anchor one in position, then select both objects and move them both. Still selected, the anchored one will not want to move; this is very irritating.

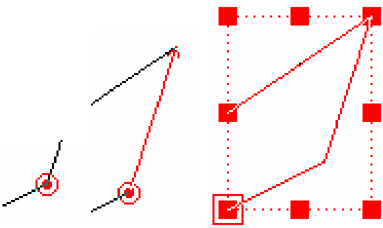
Anchoring objects relative to each other is less restrictive. The position of each object can be changed, but moving one object by any means, mouse, cursor arrows or numerical position, will result in all objects anchored to it being also moved by an equal amount. For example, draw three objects, select them and anchor them together, *i.e.* relatively to each other. Now select only one of them and move it: the others will also move by the same amount, even though they were not selected at that moment, which can look confusing. As well as its practical use for positioning a group of objects, the relative anchoring function has some interesting side effects that will be of interest later on, when we look at the selection and manipulation of points and segments.

To release anchored objects it is once again necessary to use the menu, this time *"Edit / Free anchors"*. As with protected objects, objects anchored in position can also be identified, this time with a small anchor, by using the menu command *"Display / Trace / Position anchors"*. The corresponding icon can also be used for releasing anchored objects, if none are currently selected, to be pointed one by one or in a zone.

❑ Selecting and handling points

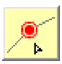
We have seen how to select and handle drawn objects. Whilst this type of selection is fundamental when using Galaad for defining the target of a given operation or change, it is not the only one. Several construction, snapping and manipulation functions require that one or even two points be selected, so that the position can be reused as a reference.

More generally, Galaad has three distinct selection methods for **objects**, **points** and **segments.** Then, for each of these three, there are two possibilities, the selection in **red**, which is the primary method, with the selection in **blue** as secondary method, therefore giving a total of six possibilities, each with a different purpose. What is more, as several of **these selections may be present simultaneously**, Galaad shows the *"focus"* on only one of them at any one time.



This focus takes the form of a contour in the selection frame. Note that the objects selected in red have priority and always have the focus. Shifting the focus to another selection or returning to the drawing mode deselects these objects.

The objects selected in blue never have the focus as they cannot be manipulated in any way: they are used only by some functions that require a normal selection (in red) as target, plus a reference (in blue). On the other hand, points and segments that are selected in blue can have the focus. Returning to point selection, they help to draw, locate accurately and even modify trajectories.



Both the red and blue point selection icons can be found in the group of selection icons. Let us start with the red one. A straightforward example is to draw a simple polyline or a rectangle. Click on the icon shown above then on one of the vertices of your shape. For it to work, the tip of the arrow (cursor) needs to be very close to your chosen vertex. If all goes well, there should now be a red spot on the vertex with a ring around it showing that it has been selected and has the focus.

Big tip: instead of using the selection icon, **simply click directly on the vertex with the right mouse button**. If the Ctrl key is held down during this operation the point will be selected in blue instead of red. As described earlier, clicking on arcs and curves with the right mouse button allows their geometric features to be edited. It is therefore necessary to press the ⇧ key simultaneously in order to select a red point on one of these figures (Ctrl still being necessary to select a blue point). Do you follow?

How to use a selected point. Select a **red point** and click on it with the **left mouse button** and whilst keeping it held down, drag it to a new position then release the mouse button. The whole object will also follow to this new position. Now try the same thing but with the **right mouse button,** this time only the point itself is moved, the rest of the object is unaffected.

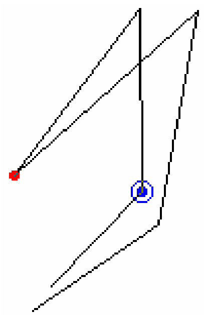
As you would expect by now these operations can also be performed from the keyboard. The now familiar Enter key opens a dialogue box designed specifically for **positioning the selected point**, alone or the with the entire object, this is equivalent to using the right and left mouse buttons. The usual ←↓↑→ keys allow the whole object to be moved along with the point and simultaneously pressing the ⇧ key results in only the point being moved.

If a point is selected in red, then pressing the tab ⇆ key will cause the selection to move on to the next point in the object or the connected objects. If the ⇧ is pressed simultaneously, then the direction of travel will be reversed. Using the Ctrl key over this moves immediately to the end point and ⇧ Ctrl to the start point. These keys are convenient for following step by step the coordinates of a connected path.

Next feature: with a point selected in red, press the Del key. This will simply **delete that point** and thus change the shape of the object. The selection then moves to the preceding point if one exists. This function may help you practice some surgical strikes on the objects.

Slightly more difficult, draw two simple straight lines with an extremity of one at, or very near, an extremity of the other. They are distinct objects that can be selected individually. Now select a red point at the end where they meet and use the command *"Design / Object / Weld"* from the menu. Galaad will join the two objects at the red point to form a new single object, try selecting one of the original ones. The operation can be reversed by using the menu command *"Design / Object / Split"* which will also work on the intermediate vertices of any polyline. You have two objects again. It goes without saying that only end points can be welded together and only intermediate points can be split. Please note that certain objects cannot be joined in this way and that it is only possible to weld objects that have the same properties, *e.g.* an arc and a polyline. However, dissimilar objects can be connected using other methods as will be seen later.

Note that the point selected in red maintains its position and becomes the common point for the new object, this is most apparent in the case where the two original points had different depths and where one point has to change its depth. What is more, should they have different machining speeds the new object also takes its speed from the object that contained the red point. Obviously, when two objects are split, both the position and speed values remain unchanged.



Now let us consider the blue points. Once again, draw a simple polyline or a rectangle, and place both a red and a blue point anywhere on it but of course on different vertices. Now move the blue point with the mouse. You will notice that **the red point remains fixed, the object rotates around it and is also scaled relative to it**.

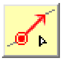
Without the presence of the red point to act as the anchor point the blue point cannot be moved.

Very important: the effects of the blue point are much greater than at first sight. Select a blue point on another object and move it. Although the red point is on a different object, it still acts as the pivot point. What is more, if the blue point is placed on an object within a group of objects that are **anchored together, all those objects will pivot at the same time**. This can help when making fine adjustments.

Apart from being able to snap to, or numerically position, a blue point, there is not much else you can do with one except delete it.

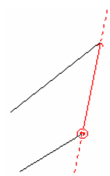
Selecting and handling segments

In the same way as with points, individual segments can also be selected and have their own special functions, notably for the construction of new lines based upon them (parallel, perpendicular, *etc.*). Selected segments are always oriented, that is to say they have a direction, indicated by a start point and an arrow head, which helps when entering details numerically. Warning: **the direction of a selected segment is not correlated to the direction of the toolpath** of which it forms part.

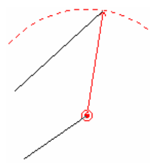


As usual, look in the yellow selection icons to find the segment selection icons. As with points, segments can be selected by clicking with the **right mouse button**, but this time aiming for the segment away from a vertex. If the segment is particularly small, simply zoom in a bit. The other features also apply, *i.e.* using the Ctrl key to select a blue segment and the ⇧ key forces the selection of segments on objects that are sensitive to the right button. In fact, the only difference between selecting points and segments is where you click on the object. If you are aiming at a point then that is what you will get, otherwise it will be a segment. All accessories are the same.

As you might have expected, the ⇆ key moves between segments in a forward direction and the addition of the ⇧ key reverses the direction. Adding the Ctrl key moves directly to the start or end segments.

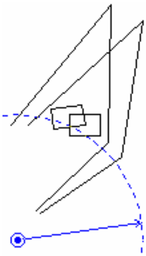


Click on one of the extremities of a red segment with the **left mouse button,** keep it pressed down then move it. The segment can be enlarged or reduced, but only along its axis, without affecting the rest of the object. Both ends of a segment can be manipulated in the same way.



Now do the same thing with the **right mouse button**. This time the segment can be rotated round its other end without changing its length. Combining the use of both mouse buttons effectively allows you to change the polar coordinates of a segment.

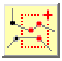
Segments selected in red are used by many of the drawing icons, notably when constructing lines, parallels, perpendiculars, intersections and others, but when it comes to manipulation and adjustments, segments selected in blue are much more powerful. Now select a segment in blue from a polyline in your drawing. As you can see on the screen, and unlike with a red segment **the whole object is either scaled or rotated around a point** when the extremities of a blue segment are dragged with the mouse, left or right button keeping the same functions as for handling the red segment, *i.e.* rotating and stretching. This may also help you to scale and adjust objects according to a basic pair of coordinates.



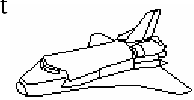
Moreover, if other objects are anchored to the one containing the blue segment being manipulated, then they are all resized and rotated together, as with blue points. This can help rotate objects around a pre-set point. Anchor all of the objects together and to a segment created specifically for the operation. Select this segment in blue and rotate it accordingly. Numerical dimensioning using the Enter key is essential for an accurate result.

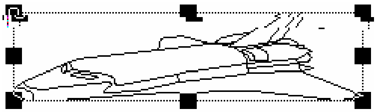
Consequently, it is possible to undertake a difficult adjustment in polar coordinates relative to a point that was not even existing on the target object, for example by snapping the other end of the segment onto the shape to be adjusted, or elsewhere depending on your needs. Once the adjustment is accurately done, the adjusting segment can be deleted or set as *"visual"* (*i.e.* not machined) since it was created only for that purpose.

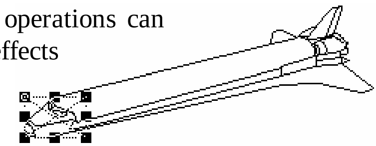
❑ Moving groups of points



This unique little icon from the selection series is well worth a passing mention. This icon allows you to select part of one or several objects so their appearance can be changed without modifying the points one at a time. First, it is necessary to define a selection zone and all points within it, no matter what object they belong to, will be selected and can be moved or scaled, repositioned or stretched independent of the rest of the drawing. Except moving, framing and deleting, no operations can be performed on groups of points. Special effects are not available either. You can also select a group of points without going through this icon, by plotting a classical selection zone with the **mouse right button**.







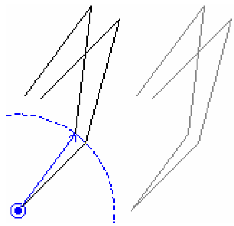
❑ Duplication and cloning

Duplication is a vital feature of drawing software. Obviously, using the copy & paste feature will allow a degree of redrawing. However, a special function is required if multiple copies are to be placed at regular intervals. Galaad also introduces a feature, the **virtual duplication**, intended to save both calculation time and memory space. Something your computer should love, then.

Start as usual by drawing an object. If you copy the object and paste the copy back onto the board, it will then contain two completely independent copies of that object. Modifying the original has absolutely no effect on the copy and vice-versa.

Once again select the object, (or the copy if you prefer) and apply the function *"Edit / Duplicate / Add one virtual copy"* from the menu. Galaad will display a selection frame for the copy, allowing you to position it accurately (of course the  key remains available as usual). When you have placed this copy, try selecting it. As it does not actually exist, it is impossible to select it, only its image is visible, however, **it can still be machined**. Do not worry about it, just because it is virtual it still retains the machining parameters of the original.

Next select the original object and move it, the copy also moves by the same amount. Modify the shape of the original using any function you like, (there are plenty to try), and the copy takes on the same shape. If you select a point on the original and either move or delete it, then the copy is also changed accordingly.

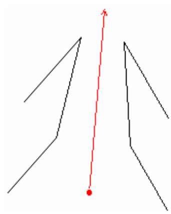


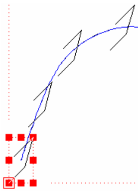
In fact, most duplications in Galaad are virtual by default. Also when you place a series of copies at regular intervals, the copies are all virtual copies and will therefore be modified at the same time as the original. This is most useful when modifications have to be made, as only the original needs changing and all the copies will follow suit.

Having said that, copies do not have to be virtual copies and if you wish, independent copies can be produced. The dialogue box for duplication gives you both options: **virtual or real**. What is more, you can make virtual copies real so that they become independent objects. The *"Edit / Duplicate"* menu contains all the necessary functions to undertake this, but remember that the reverse process is not possible, real copies cannot be made virtual so you must decide beforehand.

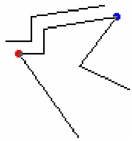
We will not dwell on the simple duplication methods, (in line, matrix, circle or special) which can easily be understood by trying them, but instead move on to the more advanced features.

The **mirror duplication** makes a copy of the selected object about a variety of axes of symmetry, opposite side of a red segment, leaving the original object in place. It is even possible to make the duplication about a red point or a red cross. As the process produces an inverted copy, it is not the same design and therefore cannot be virtual.

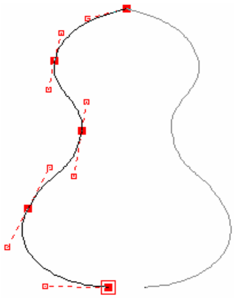




Duplicating along a blue trace places multiple copies of an object selected in red, along the length of a trace selected in blue. Optionally, they can be oriented to lie on the tangent of the trace, but in this case the copies will not be identical to the original because they will have rotated so cannot be virtual. Real objects will be produced then.



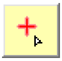
A **duplicate trace between points** takes the portion of an object located between a red and a blue point and makes a single copy of it. Obviously the points must lie on the same object or connected path.

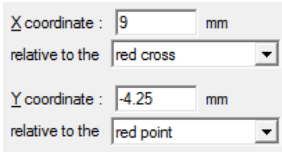


Cloning is a less sophisticated function than duplication because it simply produces a single, virtual copy of the selected object, mirrored in one of the four quadrants. This can help in constructing perfectly symmetrical shapes without having to specifically draw the other half. It goes without saying that the clone is machined in the same way as the original and inherits the same machining parameters.

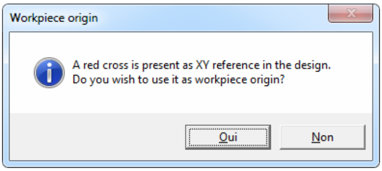
An cloned object is always stuck to one Cartesian border, and cannot be moved any further, which may limit the interest of such a function. You should weigh up the advantages and disadvantages against those of the mirror duplication, which does not suffer from being stuck to a border but, on the other hand, makes a real and consequently independent copy. Mirror duplication or cloning, it is up to you to choose the right function.

❑ Red (or blue) cross





When a point is selected in red or blue, you may enter numerical coordinates – using Enter key, you got it – that take it as a reference. You may also decide to set your origin on that point if it can be useful for something, for example resuming a milling process after the workpiece has been moved. But a selected point is part of a drawn path, even visual. You may get rid of that limitation by positioning somewhere on the board a red or blue cross, using the corresponding icon. **This cross has no relation to the shapes drawn**. It can be located anywhere and of course you can snap the plotter or dimensions for positioning it. In the same manner, you can use it as a reference for adding visual dimensions or for setting the workpiece XY origin, for example if the raw material is not a cuboid and therefore impossible to approach from a corner or two borders.



When setting the workpiece origin, if a red cross exists somewhere on the drawing board, then Galaad asks if you want to use it as reference. This can be disabled in the workspace parameters.

To remove the red cross, just click again on the icon and, instead of pointing a new position, cancel the function by pressing Esc key. A bit faster, you can also click on the icon whilst pressing Ctrl key.

❑ Rapid data palettes (at top)

When creating a complex drawing, it soon becomes difficult to deal with all machining parameters, object by object, using the green icon to access them one at a time. Therefore Galaad provides a shortcut for this. **This function is very important**.

Draw several objects on the board using a variety of depths, feedrates and cutters so that they are all different. Now use the menu command *"Display / Data Palette / Machining depths"*.

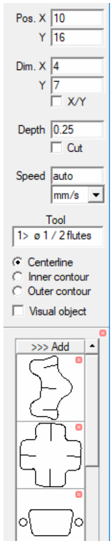
A small horizontal line of data blocks will have been added to the display between the board and the speed buttons. This shows **all the machining depths that you have used** with the current default being highlighted by a green border. When you select an object its depth is highlighted with a red border. Now click on another depth shown in the data palette and the depth of the selected object will be set to this new value without having to go via the dialogue box. With two clicks of the mouse you can change any depth to any other depth currently used in the drawing. This feature also works with layers, tools, feedrates, colours and line thicknesses.

Even better, click on one of the blocks within the data palette with the **right mouse button** and keep it pressed. Only objects which correspond to that depth remain visible. **Double-clicking with the right mouse button** selects all these objects.

To add or remove a palette, in addition to the standard menu, you can click on the left block that displays the palette type (layers, tools, depths, *etc.*) with the right mouse button, which will open a pop-up menu. For vital space reasons, Galaad can display a maximum of ten palette blocks, possibly even less if your vital space is reduced.

If you are displaying layers on the palette, which is rather common, keep in mind that **when objects are selected, clicking on another layer in this palette changes the active layer and transfers the selected objects into that layer**. The selection remains identical so it does not show.

Fast access column (at right)

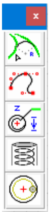


In addition to the dimensioning palettes, the *"View / Fast Access Column"* function sacrifices the right-hand side of the drawing window to house a vertical bar containing the most frequently used functions on selected objects. At the top of this column are the position and dimensions of the selection frame (this also applies to points and segments). You can enter the values directly here rather than using the Enter key, and the same applies to the depth, feed speed and tool selected. You can also add a cut contour or return to the centre-tool path. The last contouring options used will be applied.

At the bottom of this column is an area where you can store selected objects for later use, not necessarily in the same design. When you click on an object in the store, you can immediately position its copy in the drawing. These objects have no name. You can add as many as you like. Right-clicking moves an object up one box. The little red boxes are for deleting.

Personal icon palette

Galaad can display the most recently used design icons at the bottom of the left-hand column, for quick recall without having to deploy a matrix. This



can be set in the general settings of the work environment. It is also possible to predefine your most frequently used icons using *"Display / Personal icon palette"*. Then you may locate the palette wherever you like, with the appropriate orientation. Once activated, the mechanism for enriching the palette is simple: click on a design icon in the left-hand column while pressing the Ctrl key. This icon will be added to the list. If you want to remove an icon from the palette, it is the same thing: you click on it while pressing Ctrl. If you want to change the order of the palette, right-click on the icon to move it up one box. A little help is available in this same sub-menu.

Keyboard shortcuts

We recommend you to print this list out and keep it close to your work station. Not all key combinations may be of use to you, but the main ones most certainly will be.

Default:
- ⇞ / ⇟ (or PgUp / PgDn, or mouse wheel) zooms in around the cursor.
- Enter opens a position (or position + dimensions) dialogue box.
- ⇧ Enter opens the dialogue box for tool / depth / feedrate.
- Esc (or End) interrupts the current operation or deselects.
- ⇧ Esc deselects all: objects, points, segments, blue or red.
- ← (backspace) undoes the last operation, including pointing.
- Ctrl ← redoes the last operation (except pointing).
- ← ↓ ↑ → moves the cursor or selection by one step of the grid.
- Ctrl ← ↓ ↑ → moves the zoomed view.
- + / - sets the magnetic grid sizes 10 times smaller / larger.
- * sets the magnetic grid sizes to automatic (ruler divisions).
- ⇆ (tab) makes the selection jump to the next object, point or segment.
- ⇧ ⇆ makes the selection jump to the previous object, point or segment.
- Ctrl Space refreshes the display.
- Ctrl + quad view or 3D view icon shows a rotating 3D view.

Drawing mode (crosshair cursor visible):
- ⇧ Enter opens a polar position dialogue box.
- Space snaps to the small red point.
- ← (backspace) cancels the last pointing.
- Ctrl when pointing a line, applies the polar magnetic grid.

With selected objects:
- ⇧ + mouse select adds to selection.
- Space select all (in the active layer).
- Del deletes the selected object.
- Ctrl + / Ctrl - increases / decreases the depth (0.01 mm by default).
- Ctrl Home / Ctrl End sends the object to the first / last place in the sequence.
- ⇧ ← ↓ ↑ → inverts the selected object.
- / inverts the selected object about the X/Y bisector.
- . / : selects in red the object start point / segment.

With selected points:
- ⇧ ← ↓ ↑ → moves only the selected point.
- Ctrl + / Ctrl - increases / decreases the depth (0.1 mm by default).
- Ctrl ⇆ selects the last point.
- Ctrl ⇧ ⇆ selects the first point.
- Del deletes the point.
- . selects the whole object.
- : selects the segment starting from this point.

With selected segments:
- Ctrl ⇆ selects the last segment.
- Ctrl ⇧ ⇆ selects the first segment.
- Del deletes the segment.
- . selects the start point of the segment.
- : selects the entire object.

When editing arcs:
- - / + sets the direction as clockwise / counter-clockwise.

When editing a Beta-Spline:
- + / - increases / decreases the weight of a node (attractive power).
- Del deletes the node.
- Ins inserts a new node.

When editing a Bézier curve:
- ⇧ + movement of a control point breaks the tangent at the node.
- Ctrl + movement of a control point locks the angle of the tangent.
- Del deletes a node.
- Ins inserts a new node.
- / cuts the curve in two at the highlighted node.

When setting the rotation or inclination of an object:
- ↑ / ↓ or + / - increases / decreases the angle by one degree.

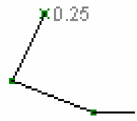
Tip: function keys F1 to F12, alone or combined with Ctrl, plus the keys on the numerical pad, can be easily customised and directly associated to menu commands using *"Parameters / Function keys"*.

Display functions

A screen filled with objects can be very disconcerting, when all of them have different and hidden machining characteristics, and having to select each object in turn to display its tool, depth, feedrate, *etc*. soon becomes very tedious. Galaad provides alternative ways of displaying this data.

As you may have already noticed, the *"Display"* menu is not the smallest around and consists of numerous commands buried in an array of submenus. A selection of the most popular features is detailed below.

The **trace** functions change the appearance of drawn objects. This allows you to make all trajectory points visible, or just limit it to the points on a toolpath where objects are connected. In addition, a small numerical identifier can be displayed at the start point of each object, showing the tool number, depth, feedrate, *etc*.



For example, you may want to display the depth of an object at its starting point or all the variations along the trajectory. The same applies to the tool, the feedrate, and the machining sequence number of each object.

In addition you can use **colours** to highlight differences instead of just having a single colour. Each tool can have a colour assigned to it and Galaad can be set to show objects in the colour corresponding to their cutters. Alternatively, Galaad can highlight all objects using the current default machining parameters (whether this be the tool, the depth or the feedrate). In this case, there will only be two distinct colours on the board. For example, if the display colour is set to *"Default tool"* then all objects on the active layer that use the current default cutter will be displayed in black with all other objects displayed in grey. You can assign a fixed colour to any object once and for all. In any case, keep in mind that this has very little interest for actual millings.

Please also note that selected objects are always displayed in red (or blue), locked objects in pink, visual (not machined) objects in grey, and tool compensated paths in brown. This will remain unchanged unless you mess with *"Parameters / Colours"*.

6

0 0 1 1 0

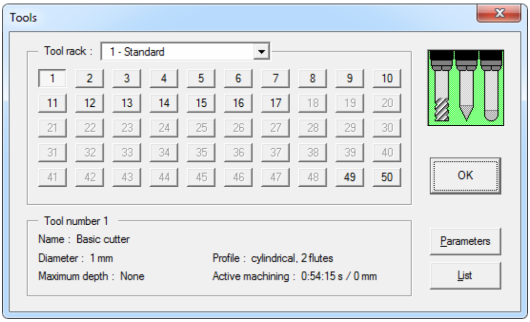
TOOLPATHS

☐ Tool parameters

Galaad uses 10 tool libraries, each containing up to 50 different cutters, for a total of 500 cutters. This should cover most needs. **Filling this library is one of the first jobs that should be done after the software has been installed.** The number and type of cutters will depend on how you intend using your machine. Start by filling it with the cutters that you wish to use right now and simply leave the rest empty until you wish to add more tools. What is important is that the tools you actually mount in the spindle of your machine correspond exactly to those you have set in the tool library.

A reliable old method is to lay out your tools in a numbered rack and ensure that the numbers correspond to those in Galaad. In this way, you can rapidly find the references for each tool used and also quickly lay your hands on the cutter that Galaad asks you to load. Having said that, you can also add the name of your choice to each cutter, this will be displayed beside the cutter number. Note that, in the library, a tool number can be left unused, even between two assigned numbers. **Tool numbers are simply identifiers and have no numerical meaning**, except for a sorted listing.

The management of the tool library relies on two separate commands:

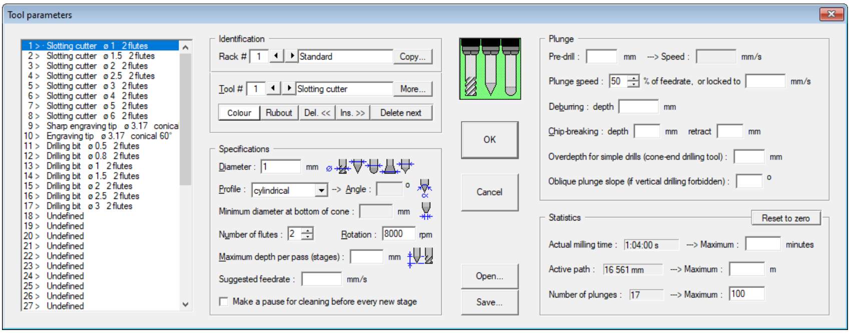


The first one, accessible from *"Machining / Tool library"*, opens a passive window that shows which tools have been defined and which tools are actually used in the current drawing. All displayed values are read-only.

This window provides an **overview of the tool library,** with the tools that will be used for the current workpiece indicated by a tick. Simply click on any tool number to display a summary of the tool parameters. Access to editing the full details is via the *"Parameters"* button. Please note that **Galaad allows the use of a tool that has not been defined**, but in this case it will not be possible to calculate any contouring toolpaths or automatic feedrates, and it will be up to you to decide what to mount below the spindle at the time of machining.

The second command for managing tools gives access to tool definitions. A given tool is defined when at least its two main parameters have been set, namely the **diameter** and the **profile**. This is sufficient for Galaad to calculate the correct compensated toolpath and approach from the workpiece origin, but this should not prevent you from entering a full description of the other physical properties of a cutter.

The alternate and faster way to access the details of individual cutters is via the menu command *"Parameters / Tools"*. This pops up a dialogue box, which allows all 50 tools in each rack to be defined one by one simply by scrolling through the *"Tool number"*, which can be found in the top left corner of the window.



When you are happy with the changes and new tool definitions, then click on OK which validates all changes including masked ones, otherwise click on Cancel to retain the previous settings.

Note: selecting a tool to modify its parameters will neither change the default tool nor the tool used by any selected objects. The tool library is completely independent of any drawing. The cut depth and feed speed largely depend upon the physical characteristics of the tool (as well as the hardness of the material to be cut).

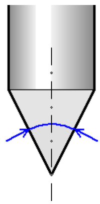
When Galaad is installed, a tool library comprising only one pre-defined cutter exists. It is likely that your personal collection of cutters will have much more. Simply change the parameters of this tool and create the others. To avoid cluttering up the list, you may delete previously created tools that are no longer of any use to you, by clicking on the ***"Rubout"*** button to leave that tool number empty. Alternatively, if you would like to compact the tool library by moving all of the tools down by one position click on ***"Del. <<"***, or on ***"Ins. >>"*** to insert a tool in the middle of the list. The last button ***"Delete next"*** clears all the following tool numbers in the library after the current one (not included).

Be aware that **you can use only one single tool rack** at a time for a given file. Drawn objects only refer to a tool number and not a rack number. If you switch to another rack, these numbers will not be affected but the compensated paths may have to be recalculated.

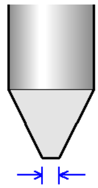
Each cutter can also be given a **name** and a **colour,** which is only displayed if the mode *"Display / Path colour mode / Tool dependant"* is selected. The name of the tool appears in the list and is displayed in the message before setting a workpiece origin. This has no other function than to remember which tool this number corresponds to.

The key parameter of any tool is the **diameter** and leaving it blank tells Galaad that the tool is not yet defined, so there is no point in entering any of the other information as it will not be stored in the tool library until a diameter is entered. It is important to be **as accurate as possible** with this figure, which should be measured over the largest diameter of the active cutting part of the tool. If the tool is not cylindrical, for example with a conical engraving cutter or a 3D hemispherical cutter, this is frequently the same as the shank diameter. For a cutter with a special profile, *e.g.* a surface mill in reversed "T" or a pyramid tool, the right value will be the maximum active diameter.

The **profile** of a cutter, closely linked with its diameter, determines how the width of the cut will vary with the depth. Profile families are: cylindrical (drilling, boring, cutting, flat milling), conical (engraving or chamfered cutting), hemispherical (3D milling ball-end), pyramidal (reversed conical cutter for surfacing), and arcade (conical tool with fillet borders). With a cylindrical tool the width is constant, irrespective of the depth, but with a conical or hemispherical cutter the width will get wider as the cut depth increases, up to the maximum diameter that the tool can cut.



With a conical tool, it is necessary to accurately specify the **point angle**. This angle is the **full included angle, measured from side to side** and not the half angle from the centre line. If the two sides are unequal, measure the largest half angle, then double it as when the tool rotates about its axis it will describe a cone swept out by the largest side.

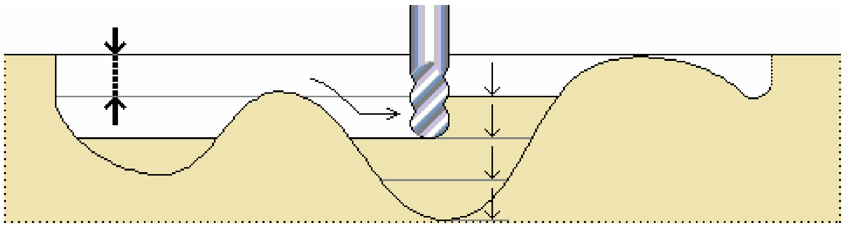


For conical tools with flat or shaved end, you may indicate the **minimum diameter at the bottom of the cone**. The software will integrate this in its calculations, in particular for the compensation trajectory (tool offset). If the end is rounded, then the best is to choose the closest option according to the profile angle, with or without the minimum diameter (sharp or wide angle).

The automatic calculation of feedrates is also based on the **number of teeth** and the **speed of rotation** of the cutter, as parameters of secondary importance. It is true that feed speeds are supposed to be based on these two factors when cutting metallic components. But the fine tools used in engraving or for cutting softer material are very fragile and demand that other factors are of primary importance, namely the tool diameter, profile and cut depth. Galaad takes this approach in its calculations. The spindle speed is displayed immediately before machining, and controlled directly from the software where this is possible, *i.e.* if a remote speed control is available and of course correctly set in Galaad. See chapter *"Machine parameters"*, spindle section, for more details.

If you indicate a **suggested feedrate**, then it will be possible to choose that option in the dialogue box for setting depths & speeds. In that case, the speed can be changed here later on.

The main parameter that prevents expensive cutters being broken is the **maximum depth per pass**, which defines the deepest cut that the tool can make in any one stage. In the case of a cylindrical cutter, the theoretical depth is equal to the length of the cutting (fluted) part, but that supposes that the tool is solid. It is preferable to stick to something a little closer to twice the diameter of the tool to prevent unwelcome surprises and assorted expletives. If the tool is conical (engraving javelin) or hemispherical (3D ball-end), its maximum depth is simply the **height of the cone** or the **radius of the hemisphere**, assuming that the teeth extend that far.



Warning: the maximum cut depth per pass also depends on several other factors, not least the resistance offered by the material. In addition, it goes without saying that a brand new carbide tool will have a much better performance than an otherwise identical, but well used, HSS tool. The unsupported length of the cutter, (*i.e.* that protruding below the collet), must be taken into account. The longer the length, the slower it should cut to prevent excessive deflection and a broken cutter. Then there is vibration, directly proportional to the spindle speed, it can be detrimental to the quality of the finish and fatal to the cutter. Clearly the less cut depth, the less work the tool has to do removing material and the less resistance it will feel. Galaad takes this into account when calculating feedrates.

Note that Galaad manages increasing depth stages and avoids passing twice in deep paths that have been already milled during previous stages. In the case of 3D paths, the tool then only does the active part of the trajectory for the current stage, this also applies for the ending bottom right point. So there no time is wasted with a tool passing several times in the same final path. It is also possible to ask for a **cleaning pause before every new stage**, the time for a quick blow of vacuum cleaner. If enabled, this pause will actually pop up a message for the operator which will have to be acknowledged before the process can be resumed.

On the right hand side of the dialogue box is a section which contains several parameters applicable to vertical motion only *i.e.* when a tool is **plunging** or **drilling** into the material. It also covers the positions for drilling cycles used for clearing material out of deep holes. The first parameter, the **preliminary plotting,** forces the tool to drill the material for a small depth that you can indicate here with the corresponding vertical speed. Obviously, the operation is performed only if the path depth is greater than the plotting depth. This function is useful only with certain tools on particularly hard materials.

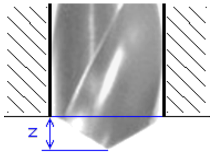
Next, Galaad allows the **plunge speed**, speed at which the tool drills vertically into the material, to be set independently from the horizontal or 3D **feedrate** given to each object. This speed is linked to the characteristics of the cutter itself and can be set at a fixed value once and for all. Whatever its shape, a milling or engraving cutter does not drill as well as a proper drill bit.

In essence, the machining process follows this sequence: the tool moves horizontally at a safe height above the workpiece and at a rapid speed, V1, to a position immediately above the start point of the first object to be machined. It then moves down at a rapid speed, V2, to just above the surface of the work before plotting at a very slow speed, V3, and entering the material at the plunge speed, V4. The part is then machined (2D or 3D trajectory) at the feedrate, V5, and finally the tool is retracted vertically at a rapid speed, V6, to a point above the workpiece. The speeds V1, V2 and V6 are not related to the workpiece, but depend on the performance of the machine, hence these parameters can be found in **machine parameters**. The feedrate, V5, of **each object** is determined when it is constructed. The **plotting** and **plunge** speeds, V3 & V4, are set as discussed in this chapter.

Nevertheless, the plunge speed can also be set in the other traditional way, namely as a **percentage of an object feedrate**. In this case you just have to indicate this ratio in the corresponding edit box, in percentage of feedrates. The two methods are mutually exclusive therefore entering a percentage value will delete an absolute value and vice-versa. At the end of the day, the choice is entirely yours and is likely to be influenced by the type of work that you undertake. Note that it is only possible to have automatic plunge speed if the feedrate itself is set to auto and the plunge speed is set to a percentage of the speed.

The **deburring cycle** is a classic drilling operation, which repetitively retracts the drill completely out of the hole after it has cut a predetermined amount, and returns it back to the depth it has just reached ready to take the next cut. This cycle is repeated until the required final depth is reached. The purpose of this operation is to prevent a build-up of swarf, which can have detrimental effects and is particularly important with deep holes at very accurate diameters. Of course, this also depends on the material.

Another very similar routine is the **chip-breaking cycle**, the difference being that the drill is only retracted by a very small amount and is not lifted completely clear of the hole. The purpose of this is to break up the long spirals of swarf produced by certain materials, into shorter lengths that will be thrown clear.



For drilling works with drill bits having conical ends, you may set an **overdepth for simple drills** that corresponds to the Z height of the lower cone, so the diameter at bottom is the same as the top. Indeed this digs craters in the machine bed under the workpiece.

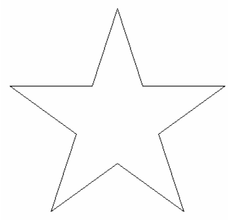
If the tool cannot plunge vertically into the material, which often occurs with a cylindrical cutter, then you may define an **oblique plunge slope**, set in degrees of angle. Galaad will make the tool touch the workpiece top surface without plunging, and then will start the XY feed path following the Z slope, with a return backwards to the start point along the plunge. The XY length of the path is calculated according to this slope angle. If the path is too short, then the process is repeated until the target depth is reached.

For a closed path with one single Z coordinate, which is very common for cutting jobs without support bridges, the option *"Chain Z passes"*, at bottom right of the machining parameters window, enables a **progressive helix lowering** for penetrating the material, *i.e.* a continuous slope around the path, a classic of metal milling. We will see that later on.

Finally, the bottom right area of the dialogue box shows some basic **statistics** about the use of each cutter. These statistics can be active if you want: Galaad will ring the bell when a wearing limit has been reached, defined in machining time, milled path length or number of plunges. This software is well-educated and will not interrupt a machining cycle because a tool trespassed one of these limits. It will just warn you, before launching a new cycle, that the tool about to start has run through a red light, but ultimately you are the one in control.

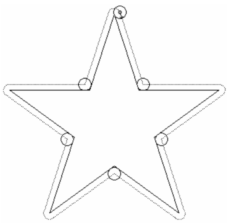
Tool compensation

When you construct an object with Galaad, the drawing is supposed to be an accurate representation in space of the cutter path. This means that it displays the path followed by the point of the tool (or the central axis at its lowest extremity), but not necessarily the actual finished part. Let us look at a suitable example:

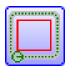


Construct a simple five-point star, remembering that Galaad has an icon specifically designed for this purpose. Imagine that you now wish to **cut out** this star accurately from a thin sheet material, so you set the cut depth to the thickness of the material and select the best cutter for the milling process. Then you are ready to start cutting. Or are you?

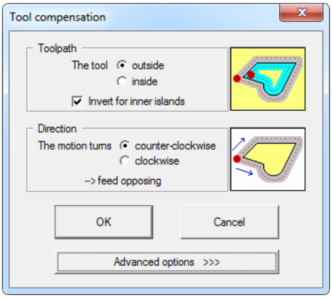
The drawing represents the actual path taken by the centreline of the tool, and the tool does not have an infinitely small diameter, so the finished result will be a bit smaller than it was actually drawn. The difference is half the diameter of the cutter used and as we wanted the star to have precise dimensions the result can be scrapped.



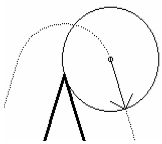
In fact, the toolpath must pass to one side of the path drawn, on the outside if you want the star itself, or on the inside if you want the material with a star shaped cut out. The cutter path must therefore be modified accordingly and moved by half the diameter of the tool. This is called **tool compensation** and thankfully, Galaad does it for you.

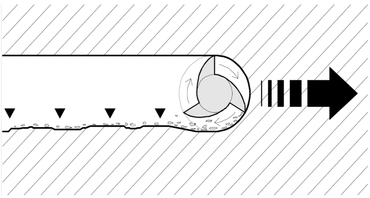


Select your star and find the menu command *"Machining / Tool Compensation / Define toolpath"* which opens the inevitable dialogue box and provides the necessary controls. This function also has a handy shortcut that can be found in the command icon bars at the top of the screen. All that is left to do is define the **path** that the cutter will take around an object. Start by indicating whether the cutter will be offset to the outside (if you wish to keep the object intact) or the inside (if you wish to produce a cut out, the size of the object) of the shape. If several objects are selected, then perhaps a group can be composed of an external path to be cut outside with some islands to be cut inside. Enabling the appropriate option, you can set the whole group without having to redo the reverse operation.



Then specify the **direction** of the toolpath, clockwise or counter-clockwise. This is very important as the machining will only be done in this direction, and a little arrow is shown at the start point of the toolpath to indicate in which direction the cutter will travel. With a cutter rotating clockwise, this equates to a clockwise toolpath when cutting on the inside of a closed shape, and counter-clockwise when cutting on the outside. Moreover, Galaad chooses the direction automatically, but if you want to change the direction, you can do so as you have the last word.

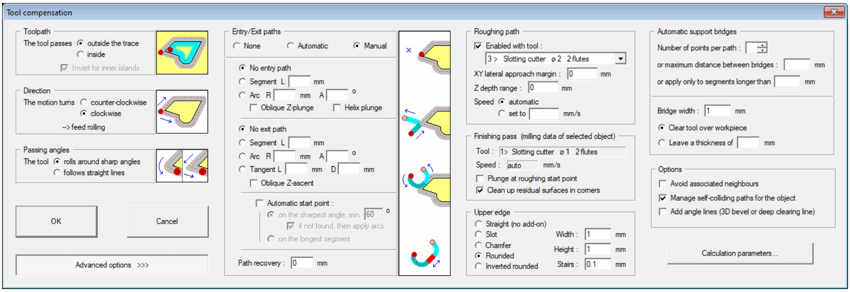




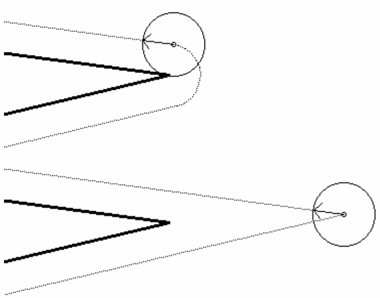
The rotation of the tool combined with its direction of travel is important as it influences the whole cutting process. If the cutter approaches the work as shown to the left it is called *"conventional milling"*, the opposite is *"climb milling"*.

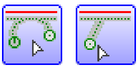
With a light milling machine, conventional milling is likely to produce the best quality finish. But there can be big exceptions to that principle, depending on the material to be machined and the cutter used. For a classical cutter with helix flutes, turning clockwise seen from above for extracting the chips upwards, which is rather common for spindles, it is then better to set a clockwise path when the tool must cut inside the path, and counter-clockwise when it must cut outside.

This small dialogue box for tool compensation contains a big drawer that you can open by clicking on the button for *"Advanced options"*. These options are many, but do not panic, we are going to explain everything in detail.

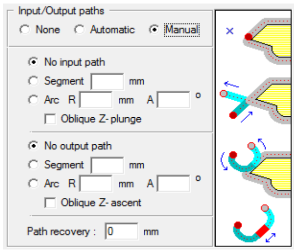


With first option, you can specify how external **angles** are to be machined. Galaad can produce either sharp angles, by extending the toolpath far away to the intersection point of offset segments, so that the tool loses contact with the work, or rolled angles by rolling the cutter around the vertex to keep it in contact with the work. The latter method is often preferred as it helps to reduce burrs.

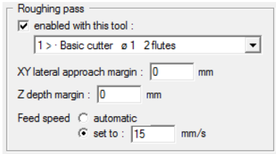




To avoid letting the cutter plunge in the material when near the border of the shape you are creating, which can leave a visible print, you can then define an input path at the feed-in point, a little way from the actual finished shape, and eventually an output path at feed-out point, each of them being possibly a segment or an arc. If you select the *"Automatic"* option, then Galaad will apply what has been predefined in the calculation parameters, shown later on. You may also add a feed-in or feed-out segment that you will locate yourself on the board, by calling *"Machining / Tool compensation / Feed-in point"* or the corresponding icons. But if the calculation of the tool compensation needs to be reset, for example if you change the tool for the selected objects, this manually added segment may be removed. On the other hand, those you preset here are integrated to the basic calculation and will remain valid.

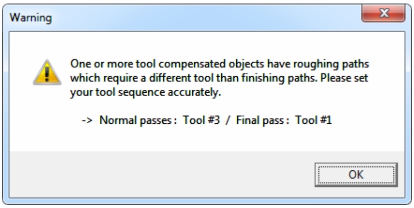


In the same frame, the option for **oblique Z plunge** makes a slope of the feed-in segment or arc, from the workpiece top surface down to the target Z depth (or the staged depth if any). The same reverse function exists for the tool bottom right at output if you do not want the tool to exit vertically. The **path recovery** distance builds a tool compensation path with the output trespassing the input point.

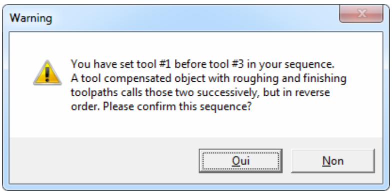


A cutting operation may require a first pass with a **roughing** tool, being fully understood that **the finishing tool is always the one defined for the object** (and also its feedrate), displayed as a read-only reminder. Concerning the roughing, you can call a different tool that feeds at a different speed, and leave a small **XY lateral approach margin**, *i.e.* a thin remaining material that will be taken away when the finishing tool feeds, and also an eventual **depth margin** for reducing the Z coordinate to be reached by the roughing pass. At last, you can ask for a finishing tool **plunge at roughing start point** where the material has already been drilled. In this case, a small linking segment will be added. If the tool is used for both roughing and finishing passes, then it will not even be lifted up.

Keep in mind that the tool sequence has the highest priority, particularly concerning tool compensated cycles with roughing and finishing passes. If

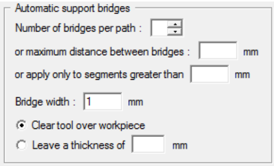


you set a roughing pass with a different tool, then think about inserting it in the machining sequence before the finishing tool to



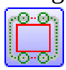
avoid putting the cart before the horse. Nevertheless, if your drawing contains a double tool compensation with roughing and finishing paths, even before setting the tool sequence, Galaad will warn you that a priority order of occurrence must pre-empt this sequence and will offer an automatic set-up. But you always have the last word and, consequently, you may modify the sequence according to your wishes. If by chance you insist on choosing to break the roughing / finishing sequence for one or more tool compensations, a last warning message will ask you to confirm that you know what you are doing. This is your ultimate chance to return to rationality.

When performing the final cut-out of a workpiece that is supposed to have been strongly attached to the machine flatbed, nature's and Murphy's laws being what they are, the cut part often tends to seek a freedom from which it has been unjustly deprived, preferably even before the cutting path has been completed. In order to restrain these emancipating impulses, Galaad can put **support bridges** here and there on the cutting path. The software will leave a small thickness of material between the two parts each side of the groove so they remain connected to one another. Therefore, the cut part remains attached to the main part that is fixed on the machine. A support bridge can either **clear the tool** over the workpiece so the path is completely interrupted, or **leave a small thickness** of material at the bottom of the groove, so at that location, the milling depth will be lesser than the cutting depth. For example, if the workpiece thickness is 3 mm and you ask for leaving a thickness of 0.5 mm, the cutting groove will be at 3 mm depth along the path, except at the support bridge where it will be at 2.5 mm only, leaving at the bottom a thin connecting bridge.

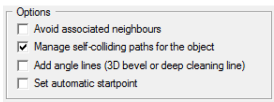


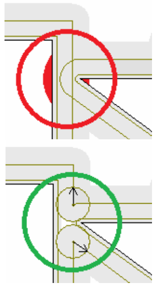
The automatic positioning of support bridges is made either by setting a **number of bridges** that will be located on a regular basis along the path, or conversely a **maximum distance between bridges**, in which case the number of bridges will depend on the path length, or only on **segments** which have a minimum length. The **bridge width** corresponds to the part of the path that will be milled above the cutting depth (the groove width being obviously the cutter diameter).

These support bridges can also be positioned manually on the path using the function *"Machining / Tool compensation / Support bridges"* or the corresponding fly-out icon in the top bar, under the tool compensation icon. But, like manual feed-in and feed-out segments, recalculating the tool compensation may delete them. Those defined here in the dialogue box are integrated to the calculation and therefore maintained whatever happens.



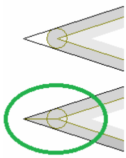
Before closing this big dialogue box for tool compensation, a few little options will let you enable the following functions:





- Avoid associated neighbours prevents the calculated contours of neighbouring objects overlapping one another, giving a possible unexpected result on the machined work-piece, especially when roughing contours are milled with cutters having big diameters. This option enables, for each calculation, the search for collisions with other contours and eventually builds an overall contour path for close shapes. But this can work only if colliding paths are to be made with cutters having the same diameter at current depth.

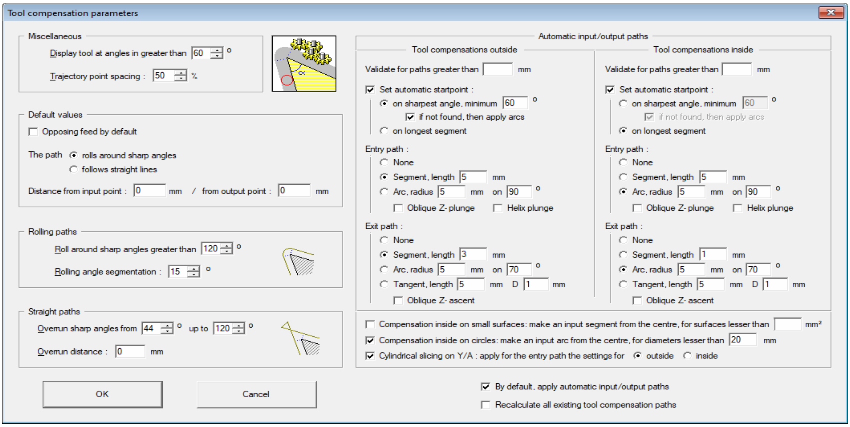
- Manage self-colliding paths for the object is useful only for a 3D path where the trajectory crosses or closely overlaps itself . The typical case is a 3D spiral for which the contour at every turn covers the neighbouring turn. In 2½D, the calculation would eliminate any part of the contour that would be too close to the whole path (current, previous or next turn). In 3D, the overlapping can be accepted since the turns do not have the same depth.



- Add angle lines, for a conical tool, adds a bevel with carving effect in sharp angles. The tool ending cone lifts up obliquely for reaching the deep of an inside corner. If the tool is cylindri-cal, then this option adds a mid-line for clearing the rounded corner.

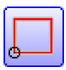
- Set automatic startpoint lets Galaad choose the feed-in point in the contour. This function will be pre-empted by input/output paths if you enable them (see above). The chosen point will be the sharpest inside corner for an inner tool compensation, or the sharpest outside corner for an outer tool compensation, so the point of plunge is as far as possible from the machined contour path.

At bottom right of the extended dialogue box, a button *"Calculation parameters"* gives you access to the internal mechanics of the tool compensa-tion.



On the left hand side, you can set display parameters for the tool compensated trajectories, the threshold angle for rolling around sharp angles and their overrun if the path has been set for not rolling. On the right hand side, you can define how the automatic input/output paths will be calculated. These values remain available in the parent dialogue box. For understanding how they apply, the simplest is to try on some object featuring inside and outside corners, for example a star. The **automatic startpoint**, when enabled, lets Galaad choose where the contouring path will begin (and also end). A classical way is the sharpest angle, which can then be overrun for a little distance, leaving the tool plunge a bit beside the useful part.

Note: it is not possible to modify the geometry of a tool compensation contour that has been calculated by Galaad, except by modifying the shape it is related to. However, a few additional functions allow you to adapt the path to specific needs. To do so, once the trajectories are displayed, just look for the function which interests you in *"Machining / Tool compensation"* submenu. Here you can **create a new offset object** from the calculation of a tool compensated path, and this new object will be fully independent from the original one. If the selected object already has a tool compensation path, then this path will instantly become the new object without asking anything. Otherwise you will be prompted to indicate an offset distance and a few more classical parameters for the calculation.



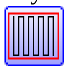
To delete a tool compensation trajectory, just call *"Machining / Tool compensation / Remove toolpath"* or click on the corresponding fly-out icon under the tool compensation icon in the top bar.

Hatching and pocketing cycles

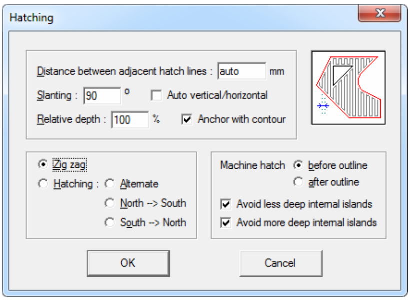
When engraving, there is frequently a need to highlight an area by clearing out its inside. In addition, milling operations sometimes require an area to be hollowed out, to leave what is known as a pocket, for example a surface between two engraved borders such as letters or any closed shape.

Galaad provides two possible ways to accomplish this. Hatching is the most classical and consists of engraving a series of shallow parallel lines across it, very close to each other. Alternatively, pocketing is more mechanical and uses a normal slot drill (or similar) to remove all the material within the pocket down to the required depth. Let us start with hatching.

Find the icon for constructing closed polylines and construct a large one with at least six or seven vertices. Then add a smaller one that is entirely inside the first one. Now select the two polylines, or just the outer one if you prefer, and choose the command *"Design / Hatching"* from the menu.



The dialogue box that opens will let you define the parameters that will be used to hatch out the area enclosed by the outer polyline. There are not too many options and they are not very complicated, or at least they should



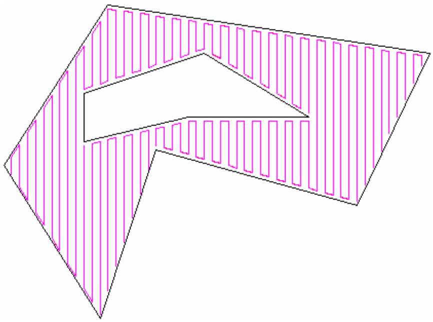
not be after a little practice. The first edit box sets the **interval between two adjacent hatch lines**. In general this distance is presumed to vary with the diameter of the cutter tool used. If you leave the value set to *"Auto"* Galaad sets this distance to half the diameter of the tool. Then hatches will have an overlapping ratio of 50%. Remember that this varies with the cut depth when you are using a conical engraving cutter. However you can change it if you so desire.

Hatch lines can also be **inclined** at any angle, positive or negative, with respect to the vertical, which is the default zero setting. Remember that a positive angle produces a counter-clockwise (trigonometric) rotation and that classic *italic* text, which slopes to the right, therefore defines a negative angle. You can also enable the option *"Auto vertical/horizontal"* so the software will calculate both hatchings and will keep only the one that requires less tool bottom right operations.

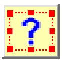
In some types of engraving the hatching out of the interior is not done at the same depth as the outline, therefore Galaad lets you set the **hatch depth as a percentage of the outline depth**. By using a relative depth as opposed to an absolute one, you can work more easily with a collection of objects that have many assorted depths. The option for anchoring links the hatch lines to their contour so they all move together.

The style of hatching produced is affected by the choice of either **zigzag** or **hatching**. Zigzag is simply a hatching of which the adjacent lines are joined at alternate ends. This results in a shorter machining time as the cutter does not have to retract, move to the start of the next line, then feed in again between each line. However, you might want all the hatch lines to be independent, in which case you can also specify the direction they take.

On the right hand side, you can choose if the hatching should ignore (*i.e.* cover) or **avoid internal islands**, for example this polyline inside another one that will be saved. Important: to be considered an island, **the inner polyline must be closed**. The dialogue box lets you filter islands depending on their relative depth.

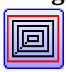


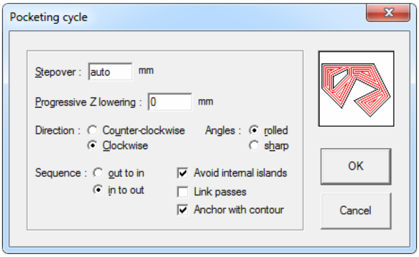
It is also useful to be able to choose whether the hatching is machined **before or after the outline**. Milling the outline last produces a sharper definition of the contour, however as usual, it is entirely up to you.



You can select all the hatching within a given drawing by using the filtered selection icon.

Cancel the hatching operation, or make use of the fact that the hatch lines are currently selected and simply delete it. Now we will look at the **pocketing cycles** so find the menu command *"Design / Pocketing cycle"*. Do not search too much: the command is just below the hatching in the menu or the shortcut icons.

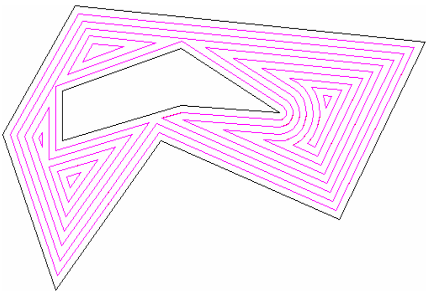




Once again the distance between adjacent passes is required, this time referred to as the **stepover**. The default (or automatic) value is classically set to half the tool diameter for a 50% recovery of the paths, but of course you can override this if you so wish.

Progressive Z lowering is a bit like the duplication function but with the addition that the depth is progressively lowered with each step towards the centre of the pocket, producing a 3D effect. As with the contouring operation, on which this function is based, it is necessary to define the **direction** of the toolpath, which is normally clockwise for internal paths. Likewise you can also choose whether external corners on islands are **rolled or sharp**.

A pocket can be cut out in two basic ways, from the inside to the outside or the other way around; Galaad gives you the choice for that **sequence**. In addition you also have the option to leave any **internal is-lands** intact and anchor the pocket lines with the contour path, like the hatching function.

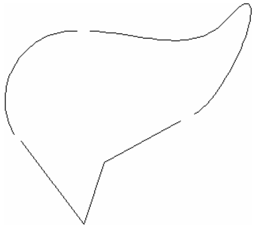


One final parameter, **link passes**, produces a short link between successive contours within the pocket, so that the tool remains at the cut depth. This is similar to zigzag hatching as described above and leads to shorter machining times. This can become quite complicated when there are internal islands involved but Galaad will try to do its best. The hatching function sometimes gives better results than the pocketing cycle in critical cases, *i.e.* outline polylines that are made of very small and erratic vectors.

Connecting objects into one toolpath

You have seen how the software can produce a contour, or clear out an area, defined by a single object. The operation is somewhat more complicated when the path is made up of several different objects. It will work if the objects are drawn (or rearranged) so that they are in the correct machining sequence. Alternatively, you could weld the objects together, but however, there is a problem: when welded together, objects of a different nature lose their geometric properties and this makes it difficult to modify them later.

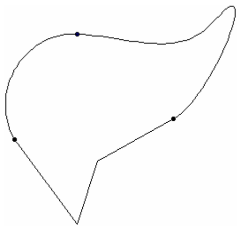
For example, take a drawing comprising an open circular arc followed by a simple polyline then a Bézier curve, which also links back to the arc to form a closed shape. Here we have three independent objects, two of which (the arc and the Bézier curve) have intrinsic geometric properties, but only a single path to contour or clear out.

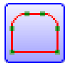


Galaad offers an interesting alternative to simply welding them together. This will not affect the properties of the individual objects. **It integrates the different objects into one single toolpath**, by simply connecting the neighbouring ends of path components.

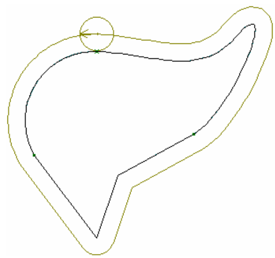
Select an end point of one of these objects and apply the menu command *"Machining / Toolpath / Connect objects"* and if all goes well the object will now be joined. This will be confirmed if you try to select either of them individually, you will now find that they act as a single entity. However, it is much quicker to apply this command globally by first selecting all of the objects.

Nothing seems to have changed except for the appearance of small dots at the connecting points where the objects join, provided that the menu command *"Display / Trace / Paths / Link points"* is ticked. You now have one single **continuous toolpath** made from the group of shapes, which can be handled as a **single object**.





This command also has a shortcut icon in the command toolbar at the top of the screen.



Now that you have this continuous path, it can be **contoured**, **hatched** or **pocketed** just like a single object. However, please note that a toolpath may remain open (as with our previous example) as you can connect and contour along an open shape. The aim is to define a **macro-object** that Galaad will see as a single entity, without having to resort to the less friendly welding technique.

When disconnecting objects, you will find that each object regains its independence, and also its geometric properties that it had prior to being connected. **Your objects have not been harmed in any way.**

There are other associated functions for working with connected toolpaths. For example, you can select a connecting point in red and define it as the start point of the toolpath, or perhaps change the overall direction in which it will be machined, without affecting the end result. If this turns out to be necessary, you can permanently weld the connections, but in this case, the objects will lose their geometric properties and be converted into simple polylines.

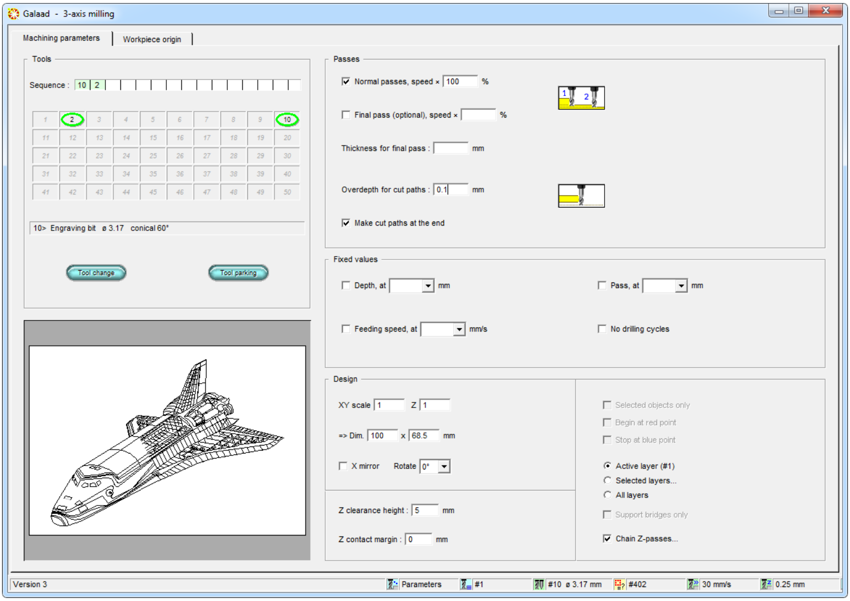
7

0 0 1 1 1

ADVANCED MILLING FUNCTIONS

❑ Tool parking positions

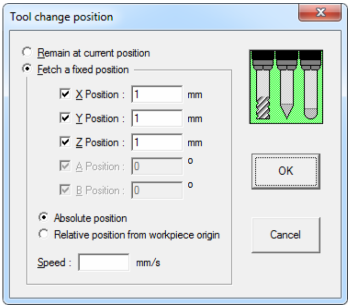
We have briefly seen the basic functionality of the integrated machining module. You have probably already machined something successfully with it and wish to know more about the features available, it is therefore time to delve deeper. Take some drawing of your choice then start the machining module, or the simulation if your CNC is currently unavailable because one happy and selfish colleague does not want to share it.



No doubt you are very familiar with this screen and already know how to control the tool sequence. If not then you have arrived here a little prematurely and skipped the chapter *"Learning to mill"*, then return here when ready for more details. Hurry up, we will wait for you.

Consequently, we are not going to return to the now quite familiar tool sequence occupying the lesser half of the window. But before looking at the numerous parameters on the right, let us quickly review the buttons that control the position of a tool after it has completed a machining cycle.

Click on the *"Tool change"* button. A small dialogue box will appear, to allow the tool change position to be specified. Unless told differently, this position is given in **absolute coordinates**, relative to the machine's fixed zero point. It represents the position where the cutter is sent at the end of a machining cycle when another tool has to be loaded.



If you do not want the tool to move from its position at the end of a cycle, that is, directly above the exit point of the last object machined, simply select the option to remain at the current position. Otherwise you need to enter the coordinates for each axis. Note that if you enter a value beyond the physical travel of an axis, obviously Galaad may find it a little difficult to comply with your instructions and will not trespass the limits.

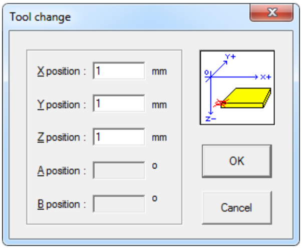
Usually the tool change position is chosen to provide easy access to the spindle for the operator, preferably in a well retracted position towards the front of the machine or at least near the operator.

The default speed is the same as that specified for the XY rapid and the Z bottom right moves, but you can set a different value if you wish, for these cycle end movements.

The neighbouring button *"Tool parking"* defines the position where the very last tool goes to, after having completed its machining cycle. Given that it is assumed that the process is finished – or abandoned, yes this can occur – this position is normally set to park the tool away from the operator so as to allow better access to remove the workpiece. Parameters of this new dialogue box are the same as the previous one, with identical functions.

If you park the tool near to zero on one or more axes, it is recommended that you leave a small distance (say 1 mm), so as not to leave the machine pressing on a limit switch. This can increase the lifetime of the mechanical contacts. Thank you on their behalf.

If your machine has no end-switches, or if you do not wish to work using absolute coordinates with a machine reset, it is obvious that you are not concerned by these previous paragraphs.



Nevertheless, in this case, you can set cycle end positions. These become relative to the one and only existing zero point: the workpiece datum. The coordinates can then be positive or negative. Beware that a **negative** value for the Z axis means that the position is somewhere **below** the workpiece origin. It is always worth mentioning…

❑ Normal pass, final pass, cutting pass

The first parameter frame, at the top right of the window, is entitled **passes**. Unfortunately, this term has several different meanings, the most common being each pass that the cutter makes across the workpiece. In Galaad terminology, the cyclic machining operations are defined as follows, starting from the top:

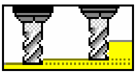
- the **tool sequence**, defined by you,
- the **passes**, based on the order of work (eventual final pass and cutting),
- the **stages**, based on the maximum depth a tool can cut (except final pass),
- the cutting order of the **objects** or **paths**, based on the drawing itself.

A complete machining operation is therefore a repetitive sequence, comprising finding the workpiece origin then automatically milling the part with a series of tools. Each tool can have a 1st pass, an eventual 2nd pass which can be finishing if there has been a roughing, and then a cutting-out pass. When making the 1st pass and the cutting out pass, it is necessary to adhere to the maximum cut depth that the tool is able to make, but not during the final 2nd pass. Finally, either at each such depth the appropriate paths are machined, or for each object all depth levels are machined. Both possibilities exist as we are going to see. Notwithstanding this, a simple machining operation might only use one single tool, only have one single pass and one single depth stage.

We have already seen the tool sequence, so let us return to the passes. When you draw an object in Galaad, you give it a milling depth. If this depth is greater than the material thickness, then it is limited to this value and the object is marked as *"cut-out"*. In this case, changing the thickness of the material will also change the depths of objects to be cut-out. Therefore, on one hand we have **objects to cut out**, and on the other, **objects to engrave or mill** at a lesser depth. 3D paths like wiremeshes are not concerned since their depth varies along the vectors.

For practical reasons, Galaad schedules **all cutting-out operations last in the sequence** for each tool. The reason for this is simple: when a piece is cut-out, there is a chance that it will not remain in place whilst other objects are being machined. When the work comprises a mixture of milling and cutting-out, it is obviously better to leave the cutting-out until last, otherwise you could end up with a piece that has just been cut-out, getting loose, catching a cutter and getting ruined or breaking the cutter. Consequently Galaad overrides the sequence defined in the drawing, and places the objects into two groups: engraving/milling, and cutting-out. Within each group, the drawing sequence is followed. But you can still disable this option, in which case there will be no cutting pass as such: all objects will be machined as milling, whether they are cutting the workpiece or not.

It is often useful to take a cut in more than one pass, rather than just a single pass at the full depth, by starting with a heavy rough cut followed by a light finishing pass. Let us consider an object to be cut at a depth of 10 mm. We could make a single cut at 10 mm or start with a rough cut of 9.9 mm, and follow it with the final cut at the full depth. The normal passes of the cutter will remove most of the material from the grooves machined and the final pass just removes the 0.1 mm **remaining material**, and any glued chips remaining from the normal pass. During the final pass, the cutter has little work to do and consequently the feedrate can be much higher. Obviously, this depends on the material.



Therefore, you can choose to make a final pass with objects that are not cut out (engraving or milling at limited depth).

For objects that are to be cut-out, there is no point in making a final pass since the cut depth is already equal to the thickness of the workpiece. So there is no final thickness to be removed for getting a good cutting aspect. On the other hand, you could make the cut a little deeper to ensure that the piece is cut-out cleanly. Depending on the material, this **overdepth** could well correspond to the thickness of the adhesive fixing the material on to the machine bed. There is no need to exaggerate this overdepth beyond reasonable limits. Your machine bed will appreciate that.

The feed **speeds** for both roughing and finishing passes are adjustable overall. You can specify a multiplication factor for each of the speeds without having to modify the speed of each object individually. These multipliers will be universally applied to the speeds of all objects, whether they were set by you or calculated by Galaad. If you choose to include such a finishing pass, then in general a multiplication factor of 200% or 300% would probably be appropriate, depending on the thickness of the remaining material that has to be removed. Logically, the cutting pass is submitted to the normal pass multiplier.

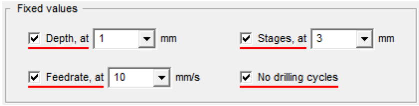
In summary, let us consider a piece to machine, comprising a profile to cut-out and some paths to mill. If you decide to make a final pass, the tool cycle will machine engraving paths at corresponding depths **minus** the thickness for final pass. Then that final pass will be machined faster, at the full depth, and, last of all, the contour cut-out at the workpiece thickness **plus** the overdepth for cut-out paths. If a separate finishing pass is not required, then the normal pass will be directly machined at the full milling depth, the cut-out remaining unchanged.

❑ Fixed values

It may be that you have to machine a workpiece design using feedrates and milling depths other than those specified for the objects. You can, of course, return to the drawing, select all objects and change the machining parameters. However, Galaad offers a shortcut allowing you to override these values with fixed feed speed, depth, stages, and drilling cycles which can be disabled.

For example, you have a drawing which contains a whole range of depths and feedrates to be machined with a tool that makes stages of 2 mm (maximum depth per pass) with deburring cycle. However, you now wish to machine the drawing with this tool in one pass, at a final depth of 1 mm, at a speed of 5 mm/s, with neither depth stages nor deburring cycles. In this case simply specify a fixed depth of 1 mm, a feedrate of 5 mm/sec, stages of 5 mm (or 1000 mm to make sure that the threshold is overrun, do not worry about it, Galaad will target the depth value anyway) and tick the *"No drilling cycles"* option. The machining cycle will now use these new values and ignore those specified in the drawing.

Note: the *"Fixed values"* do not change any of the drawing or tool parameters in Galaad.



This feature can also help you make last minute depth and speed changes according to the tool to be used.

❑ Miscellaneous parameters

The last machining parameter frame contains a variety of features to manipulate the result. The first one allows the **scale** to be changed and applied at the time of machining, without having to modify the drawing itself if it does not match the final dimensions of the physical workpiece you have actually prepared. Just indicate a different scale, and let it go. Note that the depth Z scale is set separately for easier control. The resulting **dimensions** (text box directly below) are immediately changed to show the effects of the new scale, alternatively you can enter the new dimensions directly and the new scaling factor will be displayed. Remember that you may still enter a mathematical formula in these edit zones.

When engraving on the reverse side of a piece of transparent material, it is easier to select the **X mirror** feature when machining, rather than producing your drawing already reversed. With Galaad, the user is free to concentrate on the finished result and leave this sort of detail until the machining stage. Likewise, if you need to **rotate** the complete drawing in 90-degree steps, this can also be left until machining. The small preview will assist in obtaining the correct orientation.

Very important: the **Z clearance height** sets the vertical distance to which the cutter will be lifted before making lateral XY moves to start points of other objects to be machined. This height is the **absolute value** of the distance above the top surface of the workpiece. If the surface is not flat, or if there are any obstacles, such as clamps, then you must specify a value that will allow the cutter to clear them safely, or else there will be trouble. On the other hand, making this distance too large will unnecessarily increase the time taken to machine, unless your machine is able to move at very fast inactive speeds. Changing this value remains memorised for the next processes.

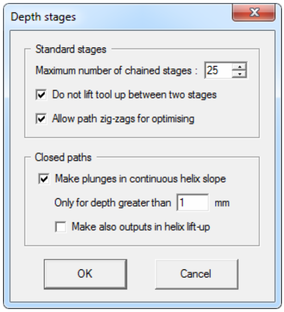
On the same lines, the **Z contact margin** allows you to define the stop point for the rapid move down before entering the material (speed for rapid move down is settable in machine parameters). Zero corresponds to the workpiece top surface. A small margin, still in absolute value, may avoid a hard contact. The slow plunge into the material will begin from that point. If you do not set a value, then the tool moves down at rapid speed until it touches the workpiece top surface, and it will plunge from there at slow speed. This value offsets upwards the start point of the slow plunge. Obviously, the target depth remains unchanged.

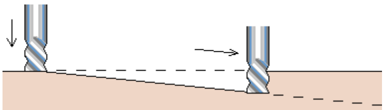
Galaad has the option to restrict machining to **selected objects only,** which is useful if you are returning to a job that has been interrupted, or only wish to machine part of a drawing. This function is like a filter; you simply select the objects required and tick the appropriate checkbox. Galaad will ignore any objects that are not selected. You can also start a machining process at a given point located inside an object if that point is selected in red in the drawing, and in the same way you can stop a process at a point selected in blue, both options being independent.

Similarly you can **filter out layers**, remembering that, by default, objects in the active layer will be machined, and them only. The objects that will be actually milled during the current tool cycle appear in black in the little preview window at bottom left. Other objects, that correspond to other tools, rejected layers or filtered on selection, appear greyed in this preview.

The option for machining **support bridges only** lets you resume a process already finished by milling directly the support bridges that were left intact. This option also exists in the cycle-end message if bridges were found in the process just completed. If you enable that option, then Galaad will not do the normal machining cycle but will only seek the position of every support bridge and will mill them one by one for removing them. This supposes at least that you have something better than your fingers for keeping the parts which often have a good sense of freedom, and also that the workpiece has not moved from its previous position (or that you reset its origin point).

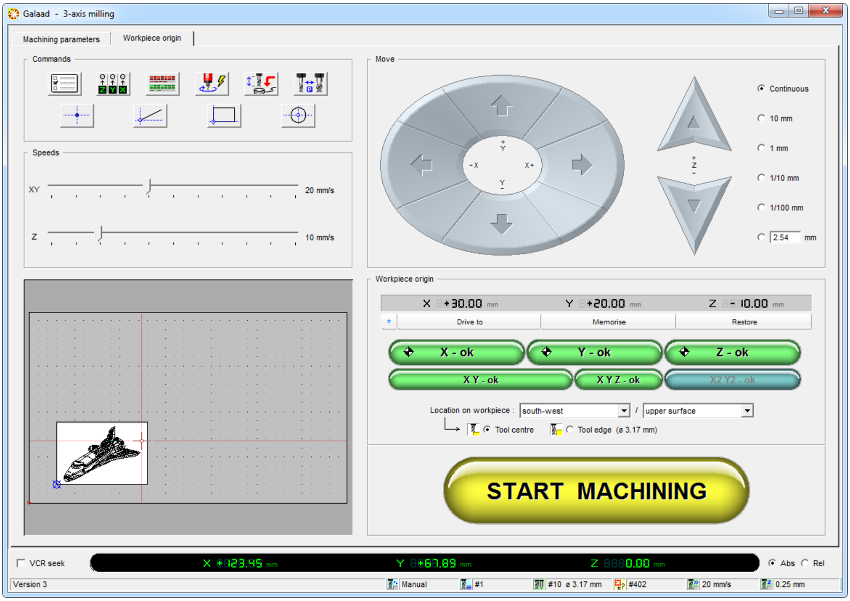
Important feature: just below, **chaining Z stages** breaks the default sequence which normally goes through all objects for every stage of depth. If enabled, then instead, the machining cycle will make all depth stages for every object, before dealing with the next object. In fact, this option pops up a sub-window that offers several related possibilities. In the top frame, you can set the **maximum number of chained stages** or decide that the cutter remains down in the material and will not bottom right for beginning the next stage. For open paths to be milled with stages, you can also **allow path zig-zags** to gain time: once the cutter is at the end point, it plunges down to the next stage for returning backwards, *etc*. In the bottom frame, for closed paths, which is rather common when cutting, you can choose to **make plunges in continuous helix slope**, so that the tool never plunges vertically in the material. In this case, it just goes down touching the workpiece top surface and then starts feeding while moving down gradually, the coordinate of every stage being fetched when it passes again at the start point. Once the final depth coordinate is reached, it makes a last loop at constant Z for finishing the job and then lifts up. That final bottom right can also be done itself along a slope, the feeding direction remaining unchanged here.





❑ Moving the axes

We have nearly finished reviewing the machining parameters and all that is left to do is say a short prayer before experimenting with your fragile cutters on a somewhat more robust workpiece. Before becoming an expert in setting these parameters, it is inevitable that you will suffer a few cold sweats and broken cutters. Do not forget that you can either run a simulation, or perform a dummy run without the workpiece in place so that you can see the path the tool will take and gain an understanding of the complete machining process. Now move on to set the origin by clicking on the *"Workpiece origin"* tab or by pressing the key.



You will have no doubt noticed that the top right section of this window is for manually moving the axes of the machine. The Z and A (rotary axis) buttons may or may not be active, depending on the number of axes fitted to your machine.

After initialising the CNC and performing a reset of the axes to check the machine zero, you can use the jog buttons. Actually there are three possible ways of moving the axes, by using the appropriate buttons on the screen.

There are several motion possibilities: clicking the displayed buttons with the mouse, pressing the corresponding arrows on the keyboard, tilting the joystick, turning a special handwheel, or special inputs for manual control. The movement can be continuous or a fixed distance but in either case it is immediately interrupted if the button, key, joystick or handwheel is released.

When using the keyboard, the X and Y axes can be moved, either along the axes or diagonally, by using the numeric keypad or cursor arrows. The Z axis is controlled with the - and + keys on the numeric keypad. When the Caps key ⇧ is pressed simultaneously, these keys - and + move the A axis. When Ctrl is pressed, the B axis is moved, if it exists.

Without wishing to sound too obsessive, it is difficult not to expound the virtues of a joystick, or a gamepad, as a practical means of controlling the movements of a machine. Its more or less long cable allows you to leave the computer and move closer to the machine. This makes it much easier to approach the workpiece origin points without breaking cutters. Using a joystick is simple, just incline the stick along an axis or diagonal and the machine will move in that direction until you return the stick to the central position. With the default configuration – but you can change it according to your needs – if you press and hold **button #1** (the fire button), pushing the stick forward or pulling it back will move the **Z axis up or down,** the opposite of an aircraft (except if flying on its back). If you press **button #2**, all movement X, Y or Z will be made at a **reduced speed.** Of course both buttons can be pressed simultaneously to combine their functions.

Another interesting interface is the incremental handwheel for making a fine microstep approach, very practical for positioning to the contact. The settings of these interfaces are described in the chapter *"Machine parameters"* a bit later on.

Back to axis motion, now let us look at the radio buttons, which **set the movement distance**. The default setting is **continuous** which results in an axis moving as long as the button remains depressed, though it will obviously stop when it gets to the end of its journey. Unless your machine parameters are incorrect, the halt should be soft and gently decelerated. An alternative and frequently easier way of finding the reference point is to move in predefined steps for which Galaad provides the necessary features. These buttons limit the movement to the distance selected, for example if you specify a distance of 1 mm then this distance will not be exceeded (in each step). **If**

you release the button before the distance is reached then the motion will stop immediately at its current position, not the target position. So you can move the Z axis down, by say 10 mm, without fear of breaking the tool and having to constantly monitor the distance, even if you had less than 10 mm clearance between the tool and the workpiece. Of course the axes are also stopped in this mode before brutalising their end-stoppers.

For diagonal movements, the distance specified refers to the distance that each axis will move and not the length of the resulting path. For example, if a distance of 10 mm is specified and a diagonal move is made in the X and Y axes then the length of the actual path will be about 14.142 mm if Pythagoras was right. He certainly was.

Keeping a button pressed loops the incremental movement after a short pause. This loop is settable in machine parameters, once again see later on.

The **speed** at which manual movements take place is set by the little sliders situated on the left hand side of the window; the range being dependant on the CNC in use. When this window opens, the sliders are set at the default values that have been defined in the machine settings, at the bottom of the *"Speeds"* page. For incremental movements step-by-step, speed is fixed to 1 mm/s unless the cursor speed is lesser, in which case it applies. **A double-click on the speed value, right hand end of the cursor, lets you enter a numerical speed.**

You will remember that, if required, it is possible to **click directly on the axis position LED display** at the bottom, and access a dialogue box to input a position directly from the keyboard. In addition there are hidden commands, namely a **double-click in the preview area** which sends the axes to the corresponding X and Y position, and the ← (backspace) key which goes back to the **last position**. Motion speeds are those for manual sliders.

You now know all the ways of driving the machine manually.

❑ Defining the workpiece origin

You have already seen the ins and outs of setting the workpiece origin in Galaad. Without going into the details of this, remember that the software needs to know where to find the workpiece that it has to machine and consequently asks where to find the reference point and where the workpiece is situated in relation to it. The coordinates of this point are given in 2, 3 or 4 axes depending on your machine and the type of work you are doing. They are displayed at the top of the **workpiece origin** frame.

If you know in advance the exact position of the origin, you can **click on one of these text boxes and obtain a dialogue box to enter the values directly**.

Important: **it is not necessary to actually locate the cutter at the origin point** to confirm it. The physical position of the axes and the position of the workpiece origin are totally independent, although several buttons serve as a bridge between the two. Certainly, you may need to move the axes initially to find where the workpiece origin is situated, but once it has been established and confirmed to Galaad, you can move the axes without the slightest effect on this origin. On the other hand, you may indicate a workpiece origin point without having to locate the tool there, provided that you are absolutely sure about these coordinates or had memorised it.

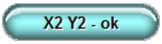
Remember that the classic way of finding this datum is to manually drive the Z axis until it is over the workpiece then bring it down until **the tip of the cutter is skimming the upper surface of the workpiece**, or alternatively, the machine flatbed (or whatever the workpiece is on). This is best performed near to the centre of the work. Once this has been done as accurately as possible, and **do not forget that the mouse wheel can make fine incremental movements**, you can click on the green button marked *"Z - ok"*. The current position will then be transferred to the textbox for the Z workpiece origin. Do not forget to indicate whether this refers to the upper or lower surface of the work, using the combo box provided just below the green buttons. If you have approached and validated the workpiece top surface but left the position combo to *"Lower surface"*, the tool cycle will be made in the air, the workpiece being just scratched by cutting paths if any. Well, it is wrong but without consequence. Conversely, if you have approached the machine bed and validated *"Upper surface"*, then there is a big risk of damage since Galaad will believe that the workpiece top surface is much lower than it actually is. The tool will plunge straight to the flatbed and even deeper, the spindle noise trying in vain to drown out your swear-words.

Now, raise the Z axis, so that it is clear of the work and any obstacles so other approaching movements are not scratching the workpiece or the machine bed. By default, this is automatic: an option for lifting up the axis when pressing *"Z-ok"* is available in machine parameters, *"Advanced"* page. You can set the X and Y values in a similar way with green buttons. XY coordinates of the workpiece origin are updated.

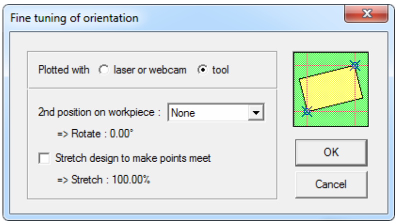
First, we are interested in the combo box that sets the **location on workpiece** of the origin point. **There are no connections between the drawing origin, the machine zero and the workpiece origin,** these three references are absolutely independent. Conventionally, the drawing XY origin is located at the southwest corner of the drawing board, but the machine zero point could very well be at another corner. **The workpiece origin can be anywhere on the workpiece** but Galaad needs to know where to find the work in relation to this point. Consequently the combo box allows you to place the origin in any of the four corners, in the centre of the four sides, in the centre of the workpiece or at a reference point within the drawing, for example a selected point or a fixed cross, red or blue. When you change the reference point, which appears as a small blue marker, the preview window shows the position of the workpiece in relation to it.

You do not have to determine the X and Y position with reference to the centre of the cutter. You can just as well work from the **edge of the tool** and Galaad will correct the position, based on the radius of the cutter. It goes without saying that if you are using a conical or hemispherical tool then the correction will be based on the maximum diameter. This option is not available if you use the centre of the workpiece instead of the edge or a corner as the reference point because Galaad will not know in which direction to make the correction. Note that changing the method (tool edge or centre) is not taken into account after you have clicked on the green buttons: if you confirm a position and then change the method of defining, it is necessary to reconfirm the position.

Possibly, in fact probably, you will have fitted a **reference corner** to your machine, comprising two stopper bars accurately mounted along the X and Y axes of the machine, forming a corner at the northwest or southwest of your bed. The position of the resulting corner will never change whatever the workpiece is fitted into it, consequently there is no point in measuring it each time, though it may be worth checking it occasionally. The Z value may have to be determined each time, as it will vary with the thickness of the workpiece (already taken into account) and the length of the cutter. Beware that these stopper bars must be perfectly parallel to the X and Y axes, the best being moreover to give them two strokes of a flat-end cutter driven in manual mode on their whole length so they go straight.

The blue button 

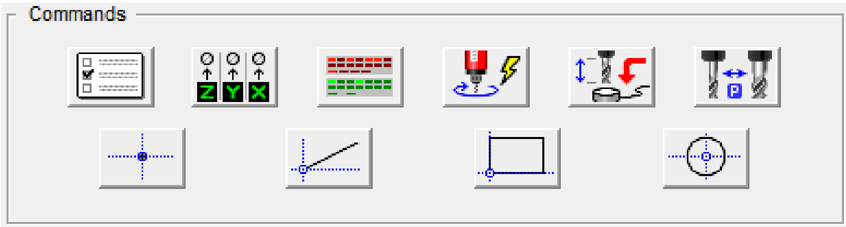
lets you define a **double origin point** if your workpiece is not perfectly Cartesian when fixed on the machine bed, for example if resuming a machining process. A typical case is the printed circuit with a side already engraved and drilled, that must be reversed for engraving the other side. You can legitimately worry about the accuracy of the pad positions around already drilled holes, especially if the card was cut a bit roughly. Here you just have to plot a second reference, for example a drill far from the origin point, and click on that button. The machining paths are going to rotate so they perfectly match the origin point, which is the centre of rotation, and the secondary reference. **You must first point out and validate the primary origin to get this blue button enabled.**



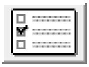
It is useful to actually go to an origin point to check that it is correct. To do this simply click on the *"**Drive to**"* button and confirm the values in the dialogue box that appears. This dialogue box allows you to reduce the Z axis movement rather than actually make contact with the workpiece. The movement and the associated limitation do not have an effect on the origin itself. Just in case, remember that you can stop a motion with the *Space* bar before it turns into a nightmare for the tool.

If you have several possible reference points, you can save the settings until they are needed again. When an origin has been determined and confirmed, simply click on the *"**Memorise**"* button and supply a name to refer to it. To use a previously stored position click on the *"**Recall**"* button and select it using the name you gave it. Galaad also displays the 5 last workpiece origins used with their dates and hours, that you can retrieve directly.

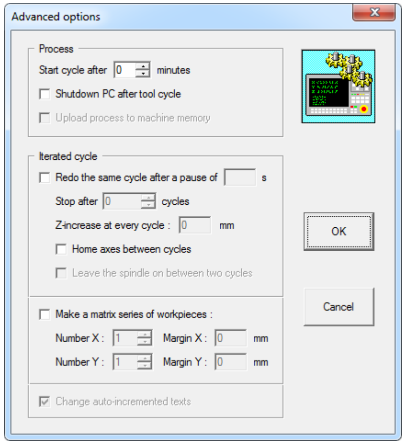
In the top left of the main driving window, above the speed sliders, there are four **command buttons** relating to functions specific to the machine. These buttons can also be accessed from the keyboard using F5 and following keys, unless you have reassigned these keys (see later on, in machine parameters, how to programme an external keyboard). We are now going to explore these commands.



❑ Machining options



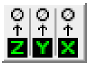
The first of these command buttons gives you access to a few **advanced options** for the machining process, in a dialogue box. Here you can start a delayed cycle, like a washing machine, and even **shut-off the PC** after the cycle has ended. This is good for the planet. If the numerical controller allows it, you can also **upload** the whole machining cycle into its local memory. In this case machining will not be started immediately, but stored in the local memory for later use without the supervision of Galaad. You will have to press the *"Start"* button on the machine anytime you want the process to be completed. Since this process is completely local, the PC can then be switched off.



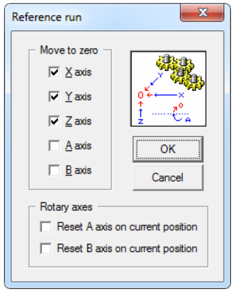
It is possible to perform the machining task **repetitively** at the same location. This is for a production run of the same drawing and allows you to set a delay between every cycle. You can decide if the spindle should be left running and if a homing reset is required before starting a new workpiece so the machine zero is checked. Ensure that you allow enough time to load the new workpieces and remove your hands, unless you want a harsh manicure. If your drawing contains **auto-incremented texts**, then the starting text will have to be set before launching the process, and its content will be incremented at every new cycle. Another type of repetition available here is the **serial machining,** which produces multiple copies of the same workpiece laid out by the machine where the margin defines the distance between the edges of adjacent boards and that the small preview helps you visualise how the results will be laid out. Auto-incremented texts are managed likewise.

Tip: since we are talking about repetition, you can simply set a binary input on the machine linked to a push button to repeat the cycle once the machined workpiece has been replaced by a new blank one. See the complete machine parameters, *"Inputs/Outputs"* tab.

❑ Resetting the axes



The next command button triggers a reset of the axes on the **machine zero point**, if it exists. Under that click, the dialogue box lets you decide to choose the axes that should be reset. **The operation resets the position counters of the numerical controller**. Since limit switches are fixed, the corresponding position does not vary. It is the mechanical reference for having absolute coordinates. Galaad supposes that these contacts are located at the end of linear axes, and preferably at the top for the Z axis. So it is not possible to have coordinates varying from negative to positive or reciprocally by going beyond that zero point. Furthermore, since Galaad knows the length of each axis, it applies strict stoppers at XY opposite ends, which cannot be trespassed (also on Z axis, but the presence of the tool under the spindle reduces a lot the interest of a software stopper). For rotary axes, either there is an angular sensor (inductive or with a cam), or there are no sensors but you can nevertheless reset the machine zero for these axes A and B on the current position. Rotary axes can obviously have negative and positive coordinates, the angular zero point being passable.



In fact, Galaad uses two working modes for driving a machine: **floating coordinates** or **absolute coordinates**. The floating coordinate system is the simplest. You just have to jog the axes to the workpiece origin point which will become the unique reference point. Galaad resets to 0 the axis counters at the position of the validated workpiece origin. But a major drawback of this mode is that Galaad never knows where axes are located in the machine workspace, and therefore cannot check if machining paths are indeed all contained in that space. Consequently, the operator must make sure that the complete tool cycle is possible from that point without bumping any axis stroke limits.

On the other hand, working in absolute coordinates offers several advantages for a classical milling job. First, having an absolute reference allows using and reusing fixed positions, the most important one being the workpiece origin of the last process made. If stopper bars have been mounted on the machine bed for fixing a reference corner, then it is no longer necessary to reset the workpiece origin at every new process, and you waste little time jogging axes manually. It is the same for the Z axis as long as the tool remains unchanged.

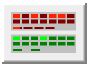
But, for managing absolute coordinates, it is necessary to have a function for resetting the axes on their zero points. The whole coordinate system is based on the reliability and the repeatability of this reset. In the same manner, if the machine has drifted for whatever reasons, a reset becomes mandatory. And obviously, if the numerical controller has been switched off, the current position is forgotten. When opening the dialogue with the machine, Galaad reads the position counters. If the controller says that all axes are at zero, then the software cannot know whether they actually are, or if this is because the machine has just been switched on and therefore all counters are set to zero though the axes can be anywhere for their part. In this case, Galaad automatically launches a mechanical reset. If the machine was already at its zero point, then it does not take much time. To avoid triggering a reset of axes at every new dialogue, the best is to fix an end of cycle parking position slightly before the limit switches, for example X=1 / Y=1 / Z=1 mm so the axes are parked near but not at the zero point. During the next initialising procedure, Galaad will read this position and will conclude that there is no need to reset the axes. But you can also trigger that reset at several moments, by setting the appropriate options in the machine parameters, *"Numerical controller"* page.

The sequence for resetting an axis is as follows:
1 - If the sensor is released, the axis runs towards its direction at rapid speed.
2 - As soon as the sensor is triggered, the axis initiates a braking ramp.
3 - The axis moves backwards at slow speed.
4 - As soon as the sensor is released, the axis stops without braking.
5 - Eventually, a little offset margin is added.
6 - The position counter is reset to 0 in the numerical controller.

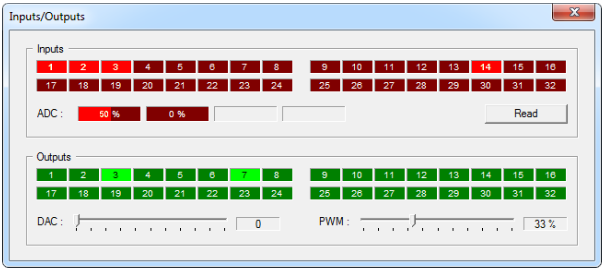
Possible malfunctions: if an axis runs towards the wrong direction at normal speed, then it is or should be reversed (*"Advanced"* page of the machine parameters). But if it does at slow speed, then it is just moving backwards, step 3 of the above sequence, and either the end-switch or its wiring is failing, always triggered, or the polarity of its input is inverted (*"Active NO"* / *"Inactive NC"*) in the special parameters of the controller (*"Controller"* page, small button *"More"* at top right). Corollary of the whole, if the axis runs towards the correct direction at slow speed, then both problems are cumulated. Last, if the axis does not stop in its end-switch, then forget all, it is simply the assigned input number which is incorrect.

If the dialogue with the machine was interrupted due to an unrecoverable error, for example an unplugged cable or an emergency stop, then **clicking on that button reinitialises the communication**. This avoids losing time closing and reopening the machining window. In addition, pressing the Ctrl key when clicking on that button gives access to machine parameters, which will also reinitialise the communication.

❑ Managing the inputs/outputs



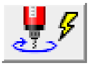
When in basic manual control, the inputs and outputs are accessible in the main window and consequently this button is not displayed, but there is not much room in the workpiece origin window and eventually you may need to check these inputs/outputs there. Remember that the red boxes correspond to inputs, digital or analogue (ADC for *Analogue-Digital Converter*), which are read regularly (twice per second) and therefore are not clickable, and the green boxes correspond to outputs that you may trigger by clicking on them.



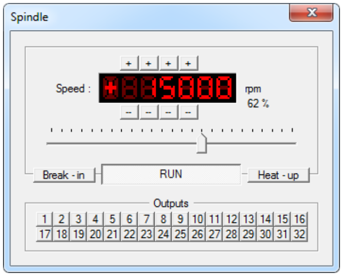
The analogue outputs (DAC for *Digital-Analogue Converter*) or PWM outputs (for *Pulse Width Modulation*) also remain available for a click and you can make them vary by clicking inside (or click and move inside).

Galaad reads the outputs (yes, the outputs) when opening the dialogue with the numerical controller, to know in which state each output is and maintain the coherence between the screen and the machine. When jogging axes, even for a workpiece origin, **it also reads the inputs after every manual movement**, which lets you check if a sensor is working properly, for example a surface probe or an end switch when you are in *"Manual unblock"* mode with no axis limits.

❑ Spindle on/off

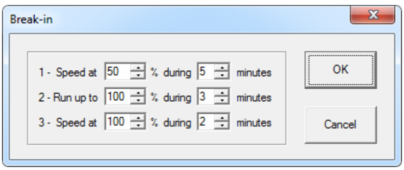


Starting and stopping the spindle is possible in manual mode from this button. If the rotation speed is not under control from the software, then each click will alternatively start and stop the motor. In this case, or even if the spindle start is not automatic, Galaad displays a message before launching the machining process, to warn the user that the spindle must be switched on and set to the given speed. Otherwise Galaad will open a control window for setting the rotation. The spindle controller can be driven through a 0-10 V DAC output (most common case) or a PWM signal, multiplexed outputs or an external control programme or whatever, the window remains the same.



The speed can be driven from a click on the small buttons that increase or decrease by 100, 1000, 10000 and 100000 rpm. The **mouse wheel** does the same with an increment of 500 rpm. [-] **and** [+] **keys of the numeric keypad** change the speed by 1000 rpm. If [⇧] key (Caps) is pressed, the variation is 100 rpm. It is also possible to enter directly a speed value with two digits, pressing one single number being considered a zero followed by the number after 2 seconds, like a TV remote control. At last, if you have an external handwheel, that wheel acts like the mouse wheel.

The small buttons *"Break-in"* and *"Heat-up"* allow you to make the spindle turn at low speed, then progressively increase the speed, then maintain the high speed. The three durations are programmable in minutes. There are two different buttons simply because Galaad will memorise two settings that you can choose to recall by clicking either one.

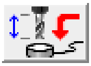


The control for the spindle rotation speed is an open loop, *i.e.* the software sends to the machine a command for setting the speed via an analogue output or its equivalent, but it cannot guarantee that the displayed speed is actually correct. So you must accurately calibrate the match between the value displayed in percentage and the actual rotation speed. Furthermore, all controllers do not compensate the loss of speed due to the hard work of the tool in the machined material.

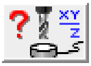
But before trying to set the spindle rotation speed under the control from the computer, **you should start questioning seriously about the real usefulness of such control**. The fact that the software can start and, particularly, can stop the spindle at the end of a cycle, this is normal and even strongly recommended: thus it becomes possible to launch very long machining cycles, knowing that the spindle will be switched off once the process has ended (you can even ask Galaad to switch off the computer itself). The spindle speed is generally set when starting the process and, even if it is under control, Galaad displays the dialogue box for setting the speed. Consequently, the speed setting is manual or semi-manual. Once the automatic process is started, it is easier to turn a potentiometer than click on digits at the bottom of the screen. Since many machines have numerical controllers that cannot manage high priority commands while they are moving the motors along a milling path, the click effect will be delayed in buffer mode with a local memory. Obviously, if the potentiometer of the spindle controller is not accessible when the machine door is closed, then it is a good reason for keeping the control within reach from a mouse click.

In fact, the only case where the software control for the spindle speed is necessary, is when the machine integrates an automatic tool changer (see later on). A machining process calling up several tools will probably require different rotation speeds. Here, controlling the speed is really essential. If you do not have an automatic tool changer, then think about not wasting too much time with a spindle speed control which would just become a nice little gadget without any practical interest.

❑ Automatic tool measurement



Your machine may have been fitted with a device for acquiring the tool length. This device generally looks like a small box topped with a contact surface that triggers some internal sensor, often a basic mechanical switch but it can also be something more elaborate such as an inductive sensor. A controlled lowering of the tool on that sensor then indicates the position of the tool bottom on the Z axis. In that case, it is no longer necessary to perform a Z approach when setting the workpiece origin, even if you have changed the material thickness or the tool length.

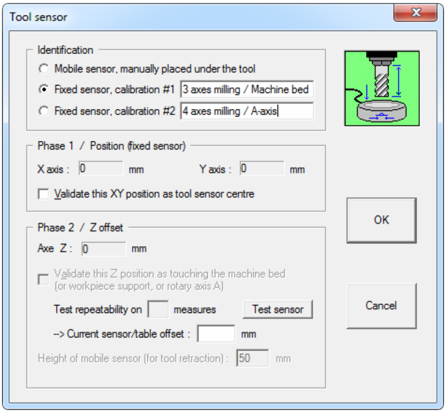


To gain a little space on the screen, only the button for measuring the tool automatically is available from the workpiece origin window. The tool sensor calibration button, as displayed here, appears only in the manual control window, *i.e.* not when preparing a machining cycle. But, if necessary, you may still access the calibration function from the workpiece origin window by clicking on the normal measurement button while also pressing Ctrl key.

Galaad can manage a fixed sensor and a mobile sensor. A fixed sensor is generally located at a corner or a border of the machine, and this location will not change. Furthermore, it can deal with two different calibrations, for example one for a flat machining process on 3 axes, and the other for a rotary machining process on 4 axes. Of course, Galaad can use a fixed sensor only when working in absolute coordinates, meaning that there is a machine zero point which allows memorising the XY position of the sensor. Using only floating coordinates (with no limit switches) allows you to use only the mobile sensor that will be manually laid on the workpiece top surface, under the tool, every time you need to measure it.

1 - Fixed sensor:

Calibrating a fixed sensor begins with **indicating the XY position of the sensor in the machine workspace**. You must bring the tool right above the sensor, as near to the centre as possible. Lowering Z axis so the tool is close to the sensor helps positioning optimally. Once the tool is right above the centre of the sensor, you can click on the calibration button.



The dialogue box for setting the tool sensor lets you choose two possible calibrations for a fixed device, that can be named (this name will be displayed when the automatic measurement function is called).

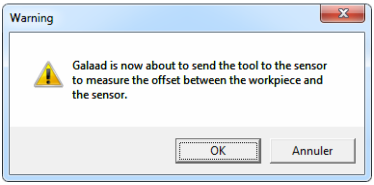
Click then on *"**Validate this XY position as tool sensor centre**"*, phase 1 of the setting, then validate all by clicking on the button *"OK"*.

At the bottom left of the screen, the preview of the machine workspace displays a small round blue icon representing the tool sensor.

Galaad must then **calibrate the Z offset between the trigger point of the sensor and the machine bed** on which the workpieces normally sit. Bring the tool back over the bed and **make a very precise Z approach of the platform with the tool tip**. Here, accuracy is extremely important: any error at this moment will interfere with all future tool length measures, until you redo this calibration process.

Once the tool gently touches the bed, click again on the sensor calibration button and skip to phase 2 of the procedure, now clicking on the option *"**Validate this Z position as touching the machine bed**"*, and click on *"OK"*. Do not change the sensor type at that moment, otherwise Galaad will mix your different calibrations.

As soon as you have validated the phase 2, Galaad wants to measure the vertical difference between the current position where the tool touches the machine bed, and the trigger point of the sensor.



The tool is going to be lifted up to the top of the Z axis, then moved laterally to the XY position of the sensor – let us hope that your positioning at phase 1 was correct – and go down slowly until the sensor is triggered. If you see that it does not plunge at the right location, hurry up to the *Space* bar of the keyboard or the emergency stop of the machine if you do not want to apologise to your priceless tool. Probably the best you can do is a dry test, pressing directly the sensor before the tool reaches it, so you can make sure without causing any damage that the sensor is indeed connected and its input number properly set in the machine parameters. Better still would be to put in the spindle an old tool or a suitable piece of dowel that does not risk much and would cause no regrets in case of accident. The accuracy will not be affected anyway.

We are going to suppose that everything goes well and that the tool stops before slaughtering the sensor. By the way, please do not switch the spindle on before measuring the tool, just for seeing what it makes (it makes a crater). Once the sensor is triggered, the tool brakes for smoothly stopping its plunge, then lifts up slowly and stops accurately at the trigger point. Then Galaad requests the position counter of the machine, so it can calculate the offset between the touch point on the flatbed, that you have set just before, and the precise trigger point of the sensor. This Z offset will remain memorised until you perform a new calibration.

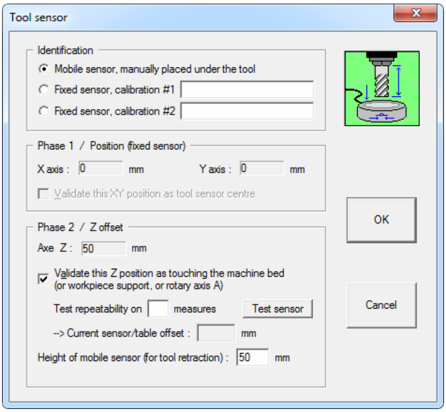
If you planned to build the tool sensor yourself, using some push-switch, it is better to have 2 or 3 mm extra stroke beyond the trigger point so the tool can perform its braking ramp. Otherwise you must set a slow lowering speed (see the machine parameters, *"Speeds"* page) so the tool does not get stuck in a stopper which would cause an inaccurate measurement and coordinate offsets. Also, think about using a switch that provides an excellent repeatability, just to avoid making your sensor a new source of problems that will probably lead you to blame very unfairly a splendid piece of software. Vandalproof push-switches with an accuracy of 0.01 mm are easy to find.

Important: if you surface the machine bed because it begins to be too damaged, the Z position of its top surface is obviously lowered by the thickness that has been removed when surfacing. **Here it is necessary to redo the Z calibration** of the tool sensor which is no more valid. Galaad does not cancel the calibration when you are performing an automatic surfacing of the machine bed. It probably should, but it does not and it is up to you to decide. On the other hand, this changes absolutely nothing for a mobile sensor, since it has no relations with the machine bed. By the way, try to avoid surfacing the sensor with the bed. It would probably not appreciate that.

Equally important: if you have tools with offsets (*"Extensions"* button of tool parameters), calibrating the XY position of the fixed sensor must be done with the tool #1. It is recommended to always use that tool #1 as a reference and therefore not assign any offsets to it.

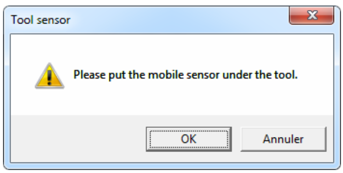
2 - Mobile sensor:

Calibrating a mobile sensor is easier. Since **the operator puts the sensor on the workpiece top surface just under the tool**, Galaad has only to know the Z offset between the sensor click and the surface it lays on. So just move the tool down so it touches any surface, anywhere, the location is not important. Once the tool is into contact, open the dialogue box for the tool sensor calibration.

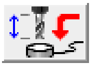


Select *"Mobile sensor"* at the top. Then, in the frame for phase 2, click on option ***"Validate this Z position as touching the machine bed"***, no matter if the touched surface is in fact not the bed: only the offset between the surface and the click is of interest. You must also indicate the **sensor height** using an exaggerated value. This helps Galaad lift up the tool high enough so you can put the sensor box under.

As for a fixed sensor, the tool is then going to be moved down until it triggers the sensor. The distance between the surface at bottom and the click point now becomes the Z calibration offset, used for future automatic measurements.



3 - Automatic measurement:



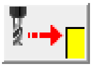
Once the sensor has been positioned and calibrated, **you can use the button for measuring the tool**. Galaad will just have to lower the tool down to the sensor until the trigger point is reached, and add the calibrated offset for having a reference point Z. In the case of a fixed sensor, this position will correspond to the tool touching the machine bed (or any workpiece support, or the 4th axis) with the same tool, making it possible to calculate the position of the top surface since the workpiece thickness is known. With a mobile sensor, this position corresponds directly to the surface on which the sensor box has been placed. We can suppose that it is the workpiece top surface.

Measuring the tool on the sensor automatically changes the Z value in the workpiece origin position. You have nothing to validate or confirm then: do not click on green button ***"Z - ok"*** which would cancel the result of the measuring operation by validating instead the current position of the Z axis.

If, in addition to the tool sensor, you have fixed stoppers or rulers for wedging your workpiece always at the same XY corner, then you will not have to jog axes anymore. When arriving in the workpiece origin set-up window, just initiate a tool measurement and launch the cycle. Of course this is valid only for a fixed sensor. By the way, this measurement can be done automatically, by setting option *"Always measure the tool before workpiece origin"* in Galaad advanced workspace functions.

Using the tool sensor requires a few adjustments with Kay module which cannot know the workpiece thickness since this is not formally indicated in machining files. But let us not anticipate, we will get back to it in the chapter dedicated to 3D millings with Kay.

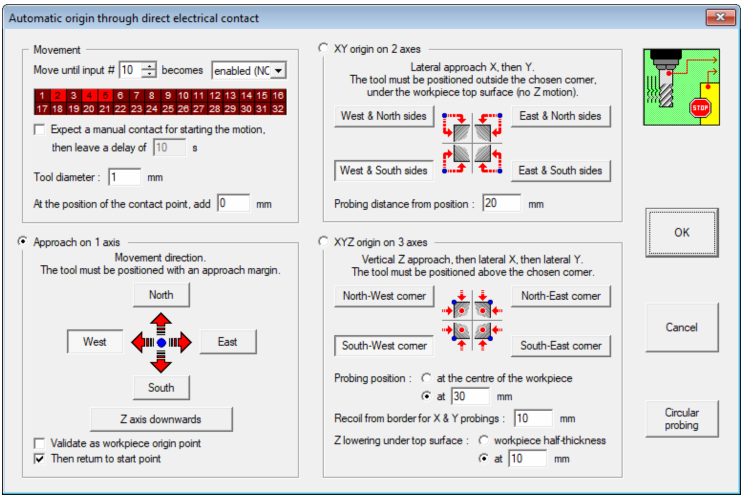
❑ Workpiece origin from direct electrical contact



It is possible to perform an approach of the workpiece with an automatic stop when the cutter touches its surface or one of its lateral sides. Of course, an electrical contact must close between the cutter and the workpiece, wired to a binary input of the machine. A couple of crocodile clips will do: one will clamp the spindle chuck or even the spindle itself it it has a metal body; the other one will be connected to the workpiece. If that workpiece is not made of a conducting material, then a trick consists of sticking at the right location a metallised adhesive tape with a folded edge for clipping. Warning:

if the body of the machine is connected to the 0 V ground and the inputs use a positive voltage (+5 V, +12 V or +24 V), then it will be necessary to **galvanically isolate the spindle** from the rest of the machine, to avoid seeing the numerical controller emit smoke.

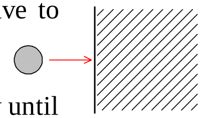
Locate the tool above or beside the workpiece, then click on the button shown at the top of this page (located at the bottom-right of the screen). A big dialogue box will pop up:



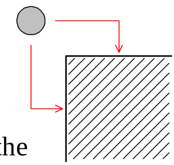
Here you must set up the **input** of the numerical controller that corresponds to the electrical contact, and also the polarity which should be *"Active (NO)"* since the circuit is presumed normally opened. If you are in doubt about this input, a short trip to the function *"Parameters / Machine / IO test"* will do you the greatest good. The input indicated here remains memorised. If you are using a thick metallised adhesive tape or even a conducting blade that you press against the surface to be touched, then also indicate its thickness to be **added** to the position measured.

Four operating modes are available:

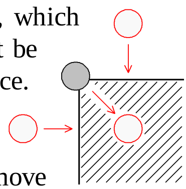
1 - Approach on 1 axis: this is the simplest, you just have to locate the tool somewhere above or beside the workpiece, then choose a direction of approach on X or Y or Z. As soon as the dialogue box is validated, the tool moves slowly until it touches the workpiece, and there stops immediately. The option *"Validate as workpiece origin point"* avoids clicking on the green buttons *"X/Y/Z - ok"*. If the approach is driven for X or Y laterally, then Galaad will automatically change the location on the workpiece for facing the motion direction and set the mode to *"Tool edge"* instead of *"Tool centre"*. If the approach is done for the Z axis, then the location on workpiece remains unchanged since it is possible to touch the machine bed as well as the workpiece top surface.



2 - XY origin XY on 2 axes: this lateral approach is done by positioning the tool under the workpiece top surface, outside one of its corners, far enough away so the tool can move along the lateral sides without touching them. Galaad moves the Y axis quickly northbound or southbound depending on the chosen corner, so the tool faces a side of the workpiece. Then it moves slowly along X axis eastbound or westbound until it touches the edge, closing the electrical circuit that stops the motion. The measured position validates the X origin. Then the tool moves backwards, returns to its starting position and does the same operation for the other side, touching the workpiece edge and validating the Y origin. So, if you stick a metallised adhesive tape on the edge of the workpiece, of course it must cover both contact zones. The distance from the corner is extended by twice the tool diameter because the start point is indeed beyond that corner.

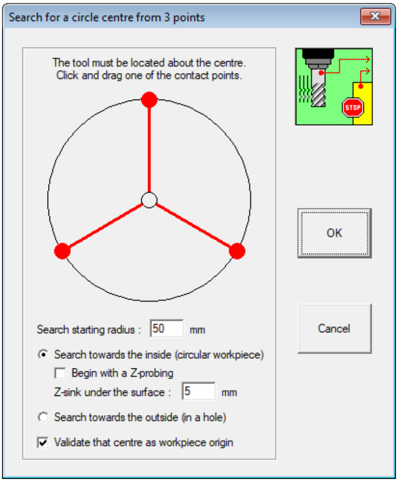


3 - XYZ origin on 3 axes: this is the most complete mode, which validates the workpiece origin on the all axes. The tool must be positioned approximately above a corner of the workpiece. Galaad is going to move it a bit towards the centre where it will touch the top surface and validate Z, then move up and go outside the workpiece, get under the top surface and move

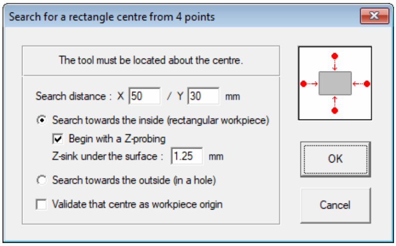


slowly until it touches the edge for validating X and Y successively. If you have to stick a metallised adhesive tape, that will be on the workpiece top and lateral sides where the tool will perform its approaches. Do not forget to link them all to the circuit. The **distance from the corner** sets the lateral touching positions.

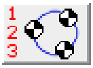
4 - XYZ origin by 3 points in a circle: at the bottom right of the dialogue box, a ***"Circular probing"*** button allows you to make an automatic measurement by contact on a cylindrical part or a thick disc. As with the three previous modes, you need to setup an electrical loop between the tool and the workpiece. The method is not very complicated: you **locate the tool over the approximate centre of the part and specify the starting radius for the search**, which must be greater than the radius of the part. The process can start with a vertical probe at this centre to validate the Zo origin. If this is not the case, you need to lower the tool so that it is close to the surface with a little margin to avoid scratching it. The tool will move to a first edge and descend below the surface, to the depth indicated, then it will return gently towards the centre until it stops on contact. Point XY is taken, the tool moves outwards, rises again and approaches the next two points in the same way, at 120° and 240° from the current point. You can click and drag the points on the three-pointed star displayed. Once the three points have been taken, Galaad knows the exact position of the centre of the circle and can therefore validate the origin of this circular piece. It is possible to do the same thing from a large hole in the surface of a part with the **search outwards** option, but of course the hole must be circular.



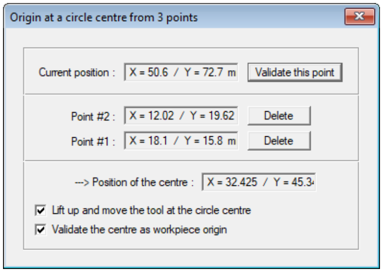
5 - XYZ origin by 4 points in a rectangle: still at the bottom right of the dialogue box, another "**Rectangular approach**" button helps you find in the same way the centre of symmetry of a classic block workpiece or in a rectangular cavity. You bring the tool over the approximate centre of the workpiece or cavity and indicate the horizontal and vertical search distances, slightly greater than half the dimensions of the workpiece for an inward search, or slightly less than half the dimensions of the cavity for an outward search. It is better for the tool to go further than the edge before plunging in and coming back (or less than the end of the hole). The process will seek the 4 points of contact at west, east, south and north of the current position.



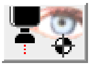
❑ Workpiece origin from three points



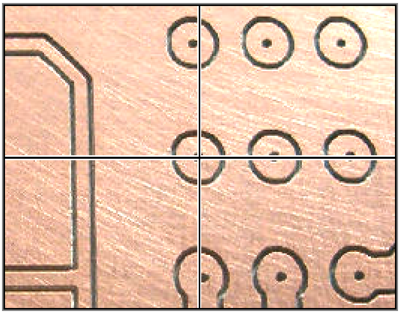
In the same manner as above, it is possible to **find the exact centre of a circular part**, but this time in manual mode. The idea is to take a manual reading of three points on the circumference. These points may correspond to the centre of the tool or to a flush approach to the edge, it does not matter, but you need to keep the same mode for all the readings. Be precise for each positioning, as the final result depends on it. Start by approaching a first point and clicking on the button above. A dialogue box will appear to validate this first measurement. Then do the same with the next two positions: precise approach and click on the button. On the third measurement, Galaad indicates the position of the centre of the circle defined by these three points and suggests that you move the tool to this centre or even validate it as the workpiece origin.



❑ Pointing the origin with a video camera



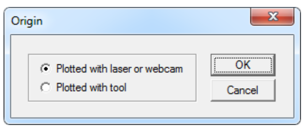
Still in the series of commands, this buttons enables or disables the **video** camera whose display is embedded in the manual control window. The option for pointing with a video camera and the few related functions are settable in the advanced machine parameters that we will see later on. If you have no webcams in your parameters, then this button does not appear.



The purpose of the video assistance is to ease pointing on workpieces where the XY origin is not in a corner, rendering the stoppers and any clamping device useless, for example a printed circuit board that has been reversed for engraving the other side, targeting a drill hole. It can also help for workpieces on which pointing the XY origin is difficult for any reasons.

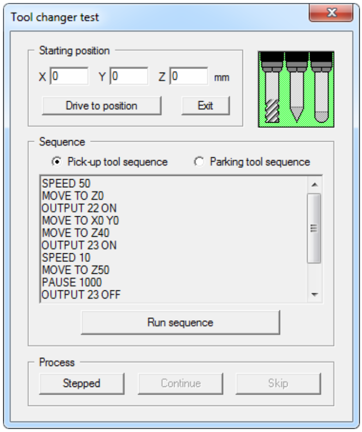
Galaad provides a small video display that overlaps the application window, including crosshairs. But it may not work with all webcams so you can also use any external programme that will be indicated in the machine parameters. **The camera must be accurately vertical, parallel to the Z axis** if it moves with the machine. If the alignment is not perfect, then moving the axis up and down will result in an XY offset of the image centre to the sides and your pointing will be inaccurate. So, if your webcam has been mounted on the Z axis, then make sure that its support is stable despite the vibrations, and that fine tuning its apex is easy enough.

Obviously, the XY offset between the tool centre and the position of the camera must be calibrated with care. A special function exists in the machine parameters, *"Advanced"* page. If a video camera has been set, even if it is not activated, when you validate the pointing of the workpiece origin, Galaad will ask you whether you have pointed with the tool or with the camera, to avoid a harmful confusion. But you can disable this in these webcam parameters.



Tool change

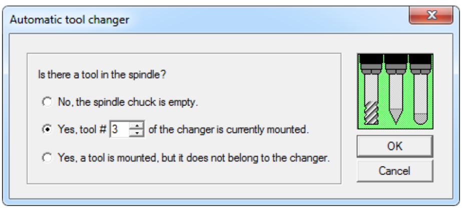
Since a milling process can integrate several tools according to the sequence that is defined in the machining parameters, Galaad can obviously chain the cycles. If your machine does not have an automatic tool changer, then the button here is just for **sending the axes to the tool change position** and nothing else. Let us remember that this position is settable in the machining parameters page, not far from here, the small blue button *"Tool change"* under the tool sequence table.



Now, if your machine is equipped with an **automatic tool changer**, the sequence of which is programmable (see this in tool parameters or in machine parameters), clicking on this button will pop up the dialogue box for managing the tool changer.

When a machining cycle has been abruptly interrupted with the tool left on the spindle chuck, then this function helps you bring it back semi-manually to its position in the rack.

Be careful never to park a tool in a position already occupied, since there are no tests performed for checking if the position is available. Conversely, trying to pick a tool in an empty position will have no consequences except leaving the spindle disappointed.



When opening the workpiece origin window, Galaad asks you if a tool may have been left in the spindle chuck, to avoid a possible conflict. If the chuck is empty, the software will grab the first tool in the sequence before you can set the workpiece origin (with an eventual measurement of the tool on the sensor). If a tool was mounted in the chuck and it is the one that is about to start working, then Galaad goes to the workpiece origin validation. If there is a tool but not the one needed, Galaad will park it and pick the good one from the range. Also, if an external tool, *i.e.* not belonging to the rack, is present in the chuck, then Galaad will invite you to remove it before grabbing the good one, provided that it is a member of the restricted brotherhood of automated tools.

In a machining process that is organised in a smart way, and therefore driven by Galaad, **the automatic tool change requires no operator's action**. Either the tools are all calibrated and their relative Z offsets will be considered, or they are not and the measurement on the tool sensor is necessary. Obviously, only a fixed sensor is usable since the mobile sensor remains semi-manual. So, an automatic tool changer requires a tool sensor for measuring the lengths and validating the Z origin, as well as a remote control for the spindle rotation speed. If you are lacking one of these two, then the tool changer will avoid the operations for removing/mounting the tool from/into the chuck, but resuming the workpiece origin and tuning the spindle speed before each cycle will remain manual.

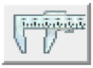
At the end of the last cycle of a process that called several tools, Galaad stores the last one used back to its position in the automatic changer rack, unless you have defined this end another way, then sends the axes to their parking positions. However, and whatever the set-up is, **if the process required only one tool, Galaad will not store it back at the end of the cycle**. This is for easing multiple processes avoiding useless movements to and from the rack. And it renders more important the dialogue box for checking if something is present in the spindle chuck before each process.

The set-up and the driving scripts for the tool changes are described in the chapter *"Machine parameters"*, section *"Advanced"*.

❑ Parking

The button hereby is used to send the machine to the parking position defined in the *"Machining parameters"* tab (see at the beginning of this chapter). By simultaneously pressing Ctrl button, you can redefine this position on the fly.

Calibrating screw pitches

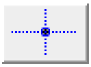


In basic manual control, it is possible to check a screw pitch or other transmission factor by requesting a movement on an axis followed by a precise measurement made by you with a calliper or a ruler. Galaad will transfer the calculated value to the mechanical parameters.

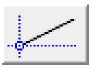
Semi-manual milling commands

Still in the *"Commands"* frame at top left of the manual drive or workpiece origin windows, in a second line, you have access to four buttons that help you make brief millings in semi-automatic mode, **at the current position of the axes**, which consequently supposes that you have first moved the tool manually to the correct location on the workpiece, including Z axis to the contact with the surface to be machined. Be aware that, if you are in the workpiece origin window, **these commands do not consider the validated origin** but only the current position of the tool in the machine workspace. Obviously, the spindle is switched on before milling and switched off after: *"semi-automatic"* does not mean *"completely manual"*.

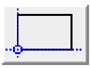
Let us review these commands briefly:



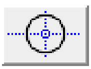
The button *"Point"* makes a basic drill, at given depth and speed. No deburring or chip-breaking cycles apply here.



The button *"Line"* mills a segment with the cutter tool, from the current position to a target XY position, at given depth and speed. Plunge speed is also available. This function can be useful for milling X and Y stopper bars beside the machine bed, to which the workpieces can be clamped, and make them perfectly aligned on the axes.



The button *"Rectangle"* uses the same parameters as the button *"Line"* but will mill a rectangle rather than its diagonal.



The button *"Circle"* mills a circle with given diameter, which centre is defined by the current position.

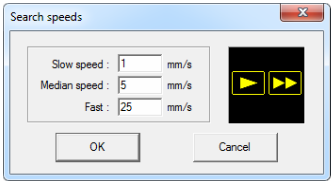
❑ VCR-Seek

Hidden at the bottom left of the workpiece origin window, this small and discreet option is important and even practical for helping you **find the resuming point of an interrupted machining** or run a partial machining. Click on it and launch a normal process (it is also available when simulating). At the bottom of the machining window, new buttons appear:

We are seeking the path from which the machining process must restart. If there has been an interruption of the previous cycle and the drawing remains unchanged, then Galaad suggests a resuming at this memorised path that is highlighted. Then you just have to validate if it looks correct. The resuming path is displayed in red, the previous paths in the sequence that will not be machined are in black, and those that will follow hereafter are in blue. The cyan buttons allow you to jump to the previous or next path, or 10 or 100 paths backward or ahead for speeding up. Every time, the software moves the machine to the start point of the highlighted path. This positioning corresponds to the laser spot or the video camera if any has been configured, otherwise for the tool itself. The Z axis remains wisely at clearance height. You can also click directly on a path appearing on the screen to choose it as the resuming path. It is possible to zoom on the path or return to the main view using the white button at left hand side.

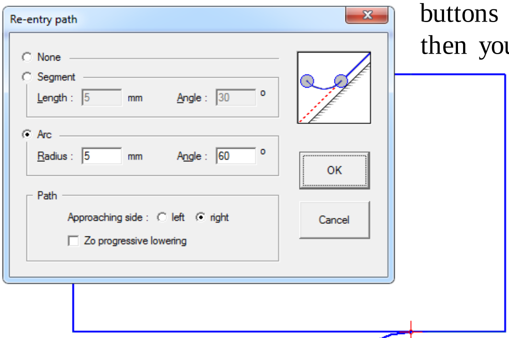
When the path has been found, then you can validate it at its start point by clicking on the green button (or stop all by clicking on red button or on the emergency stop). In this case, the resuming will start at that entry point with no more questions. But you can also explore inside the path for finding a more accurate resuming point further than the start point, by clicking on the yellow button at centre. This will avoid wasting time redoing long paths, and it is an absolute necessity when cutting with a plasma torch since it is then not possible to cut again over paths already made. The bottom of the screen is going to change consequently, with new yellow buttons:

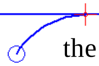
Here the expression *"VCR-Seek"* makes sense, at least for those who can remember which paleolithic device it refers to. The buttons allow you to move the tool (or the laser spot or the video camera) along the path, forward or backward, at slow, medium or fast speed. These three motion speeds are available from the white button  at left hand side, and their values remain memorised.



The yellow button  at centre lets you stop the movement along the path when you think you have found the proper resuming point. Anyway it is possible to reverse the motion direction to ease the finding. When the machine has been positioned to the resuming point, you just have to click on the green button  (or press  on the keyboard).

If the path has been validated at its start point, then the resuming is performed directly, with no entry trajectory. But if you have used the VCR buttons for seeking inside the path, then you can add an entry segment or arc, to avoid marking the workpiece at the resuming point. This is particularly valid for cutting works using a plasma torch where the ignition digs a much greater hole than the expected kerf. Here it becomes much better to fix the resuming point at a small distance away from the path. The screen displays the entry segment or arc as it will be machined. Now, in classical milling, a last dialogue box lets you indicate at which depth stage the resuming must be done, to avoid long useless paths before reaching the material.

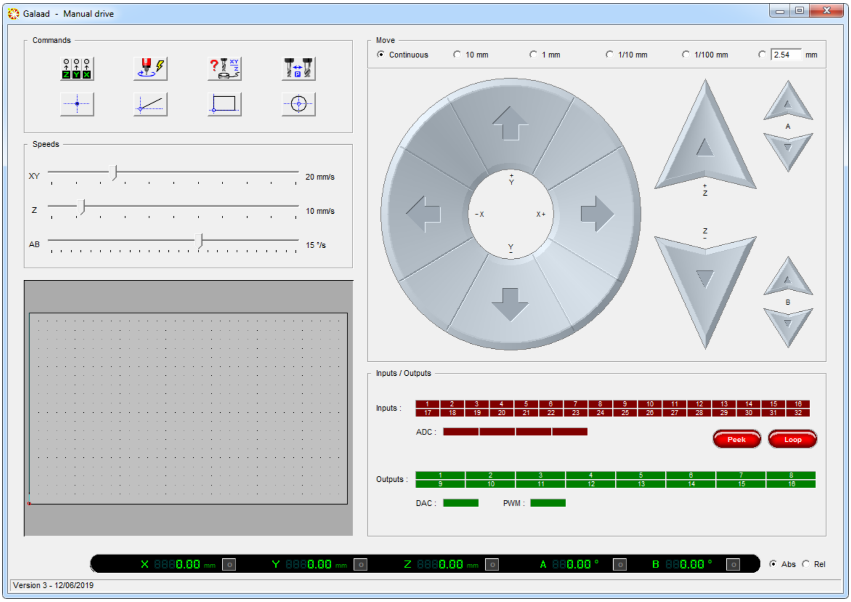




Once you have validated the resuming point, the VCR-Seek remains enabled. If you were just seeking one single resuming point, then you can disable it while it machines the path. The rest of the process will be managed normally.

❑ Manual jogging

Besides the workpiece origin for a machining process with a tool, Galaad also offers the possibility to drive the numerical controller manually, just for the fun or for checking that it indeed works, we never know. Just call function *"Machining / Manual control"* for finding the main control items for the machine, all gathered in a new window.



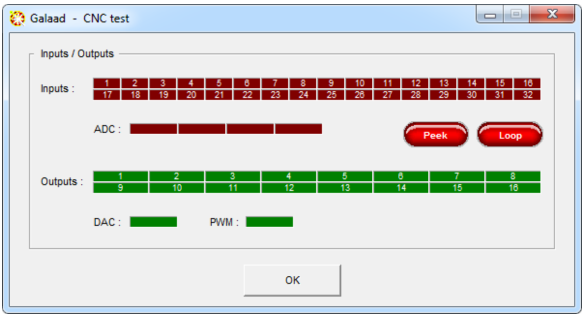
We are not going to spend much time on control objects that you already know. Let us just point out that you can check the statuses of the inputs by clicking on the red button *"**Peek**"*, or on the button *"**Loop**"* which starts a cyclic read of the inputs, allowing you to monitor any status changes. You can trigger the outputs by clicking on green boxes. Caution! One of them may start the spindle or any dangerous peripheral device. You can also set the value of the DAC (analogue) and PWM outputs by clicking in their green boxes at a given location, or clicking and moving the mouse with the button pressed.

Note: the inputs are automatically read and refreshed after each manual movement, as soon as the axes have stopped.

The function *"Machining / Manual unblock"* gives access to an even freer manual control, because **the machine does not reset its axes** on the zero point and therefore does not manage any limits on any side of the axes. This function has several possible uses: after a machining process that ended abnormally, you may need to move the machine avoiding an automatic axis reset; you may also check the limit switches or simply need to know which one is connected to which input with which polarity. Just move towards the limit switch, going slowly in the final phase (right mouse button, you remember), sink the switch and see on the screen which input red box has changed, take its number and note if it is on (*"Enabled - NO"* polarity) or off (*"Disabled - NC"* polarity) when the contact is triggered. Now you have only to transfer these values into the machine parameters, *"Numerical controller"* page and small button *"More…"* at top right).

❑ Test of the inputs/outputs

A bit in the same way as the manual unblocking, if you need to check the connection of switches, relays and other electric devices connected to the machine, the command *"Parameters / Machine / I-O test"* lets you open a dialogue with the machine in a restricted mode, with no motion commands nor position feedback. Functions are identical to the control frame of the manual drive, with a cyclic read of inputs (twice per second) which loops immediately after initialisation.



For testing the connection and the polarity of a contact that is opened or closed, for example a limit switch, a tool sensor or a safety device, you just have to trigger it and read on the screen which red box has changed. The number corresponds to the input seen from Galaad. The polarity is given as *"enabled"* if the box turns to light red when the switch is triggered and returns to dark red when the switch is released; it is *"disabled"* if the box is by default light red and turns to dark red when the switch is triggered. The ADC box displays the status of the analogue input, if any.

As for the manual control, you can click on green boxes that correspond to outputs, to checkout how your peripheral devices work and if they are indeed under control from Galaad.

❑ Managing collisions with obstacles

You may define, either on the machine flatbed (absolute position) or on the workpiece itself (relative position), obstacles that the tool will avoid when it is moved. These obstacles have a very basic shape, rectangle or circle, but they can be combined for representing more complex zones. **For each obstacle, you must indicate a Z-height**. When moving above the workpiece, the spindle will lift up high enough to avoid a collision. You may draw the machine bed directly within Galaad (the dimensions of the drawing must correspond to the strokes of the axes X & Y) with rectangles and circles, then select them and call *"Machining / Obstacles / On the machine"*. You can also, when jogging the axes manually, bring the tool above the centre of an obstacle and click on the button here for memorising it.

If some fixed obstacles have already been defined, displayed in orange on the view of the machine bed, then you may click on them with the **right mouse button** for relocating them, or use the pop-up menu under the right-click for editing or deleting them. On the other hand, the obstacles that are related to the workpiece can be edited only in the drawing (accessible from a right-click).

In all cases, the option *"Physical obstacles to avoid"* must be enabled in the machine parameters, *"Advanced"* page.

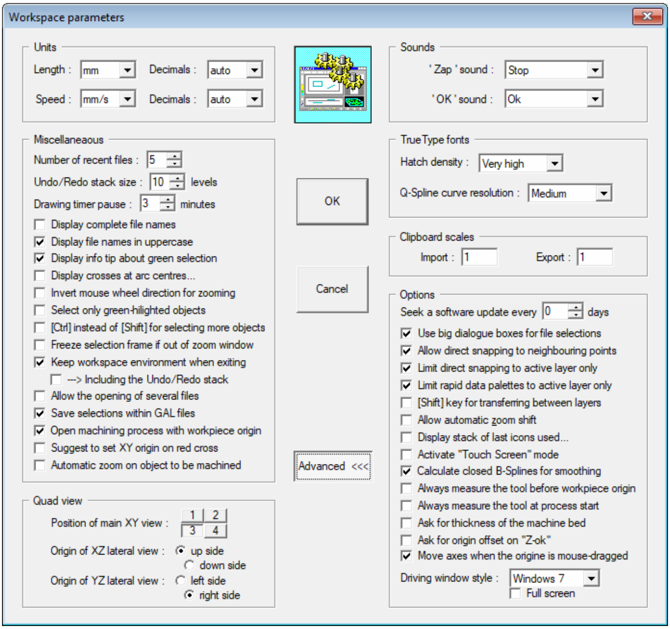
8

0 1 0 0 0

APPLICATION WORKSPACE

❑ General settings

This chapter is concerned with the parameters that control the working environment of Galaad, namely how the information is presented on your screen, but excluding the display functions discussed earlier. Use the menu command *"Parameters / Workspace"* to access these features, the first of which covers the basic application settings.



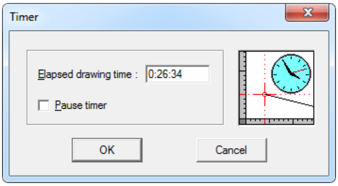
The underlying dialogue box can be extended via the button *"Advanced"* for accessing all options. And they are many.

At top left, the frame *"Units"* lets you indicate your **length and speed units**. The units for the length are passive and do not actually affect anything, only serving to provide the symbol to be displayed after the numerical value. On the other hand, the speed changes all values that have been used until now. The software stores the speeds as integers that represent mm/minute. If you wish to use another unit, then it is converted before displaying or after entering a given speed. If you change the speed unit, the internal encoding remains in mm/min and does not vary. Only the display does.

In the frame *"Miscellaneous"*, just below, you can define the **maximum number of recent files** that will be shown at bottom of the *"File"* menu (10

maximum), and the **size of the undo/redo stack**, *i.e.* the maximum number of successive drawing operations that can be cancelled. The limit is 50, but a smaller number will speed up operations and save memory use. In fact, it is very unusual to have to backtrack so far, but you may seek a better balance point between processing speed and undo comfort. If your workstation is recent and fast, it is probably better to use the maximum value. The default is 10, which looks reasonable.

The **drawing timer pause** controls the duty cycle of this counter which is displayed at the bottom left of the main drawing window. When you stop

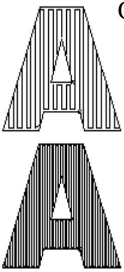


drawing, the timer automatically stops after this preset delay to avoid counting your lunch break. The time is set in minutes, but you choose how long. The timer itself remains under the control of *"File / Elapsed time"*.

Displaying the complete file names adds to each file name the disk path to it for the title bar and the recent files. **Displaying the file names in uppercase** means exactly what it means. **Displaying the info tip about the green selection** corresponds to the small help tip when the mouse flies over an object that it highlights in green, to check its geometry, tool number, depth and feedrate without requiring further selection. **Displaying crosses at the arc centres** adds a grey cross indicating the centre of a circle or an arc, with sub-parameters for that option. **Inverting the mouse wheel direction for zooming** lets you change the zoom in/out function using the wheel, to match your habits with other software applications almost as good as Galaad. The option for **selecting only the pre-highlighted objects** forbids the selection in the simple Cartesian zone of an object: the mouse must be located on the actual path, and therefore highlighted in green (remembering that a small menu pops up if there are several possibilities, even if one single object is highlighted). **[Ctrl] instead of [Shift] for selecting more objects** checks one of these keys for adding to selection. **Freezing the selection frame if it is out of the zoom window** blocks any movement of the selected objects if the selection frame is completely out of the view, to avoid unwanted repositionings. If you wish to **keep the workspace environment when exiting**, which is the default after installing, then you do not need to save your current drawing when you quit Galaad: when restarting the software, you find again the same drawing in the same state. It is also possible to **save the Undo/Redo stack** so that you can go back to the previous session (the environment file will be much larger). If you authorise the opening of several files, then Galaad switches to *"Multiple files"* mode with tabs for each one at the bottom of the screen. Opening a file no longer replaces the current file. The option for **saving the selections within the GAL files**, when enabled, keeps the selection frame and the selected segments or points in the saved file. **Opening the machining process with the workpiece origin** automatically skips the page of machining parameters (of course you may return to it), except if there are several tools to be sequenced in the calling table or if an unusual option is enabled, for example a fixed value or an adjusted scale. **Suggesting to set the XY origin on the red cross** pops up a message asking you if you want to have that red cross used as the datum point for the workpiece origin, supposing that there is a red cross in the drawing. At last, the **automatic zoom on the object to be machined** magnifies each path when the tool is about to mill it. It is a simple gadget, we can admit that, since it is too late to stop the process when the display shows something wrong. Unfortunately, there are no *"Undo"* functions for a machined workpiece. There is room for invention here. On the other hand, the function *"Redo"* is well-known to machining beginners.

You can choose to place the main plan view in any of the four positions of the **quadruple view**. When you change its position, the view on the main screen, remaining visible behind the dialogue box, is immediately changed so that you can see the result. At the same time you can also change the orientation of the lateral views, namely the position of the upper surfaces that correspond to Zo.

When this dialogue box is extended to the right hand side with the button *"Advanced >>>"*, then the top frame sets the two system sound functions that are called to produce a warning *"zap"* when you make an error – which of course occurs rarely – or a confirmatory *"OK"* when a long operation finishes without a problem, which does happen sometimes. The choice corresponds to the event sounds set in the **Windows Control Panel.**



Closely related to your Windows environment, drawing texts with **TrueType** fonts uses some graphical parameters. First, the general density of the automatic hatching for filling up character surfaces, which calls Galaad hatching function, and second, the basic resolution of Quadra-Spline curves that are part of the TrueType format. If a text is selected when these parameters are modified, it will be rebuilt according to the new settings. Note that these values have no relations with the few Galaad fonts that are pre-hatched and do not use Quadra-Splines.

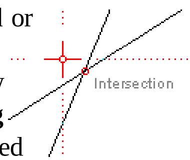
Mainly designed for the screen or the printer, TrueType or OpenType fonts that are used by Windows cannot encode simple lines without thickness. For example, a basic capital "I" sans-serif will be represented as a rectangle, or two zigzag lines if the thickness is null, but never with one single vector. The starting and ending points of a TrueType path always match, defining a closed contour. If the font was built without thickness, *i.e.* null weight, then Galaad searches and **deletes overlapping lines** for each drawn character, to build up an optimal path with single vectors and no useless zigzags.

Clipboard scales apply to drawing entities that transit from and to Windows clipboard for exchange with other software applications. Objects

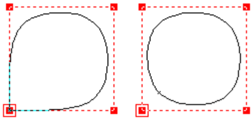
that are copied and pasted from Galaad to Galaad without closing the application are not concerned. They even carry more properties such as the tool, depth, feedrate, *etc.* that cannot get through the standard clipboard.

The lower frame *"Options"* provides a long list of settings. The **seek a software update** defines after how long Galaad automatically searches on its own website for a more recent update and, if there is one, will offer to download it. The **big dialogue boxes for the file selections** give access to all folders on the left hand margin, not just a drop-down list.

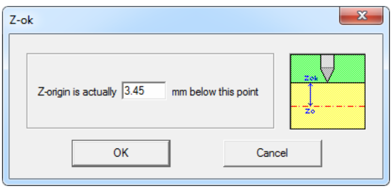
Direct snapping to the neighbouring points are these small red points that appear near the cursor when you are drawing and that can catch the coordinate when pressing the Space bar on the keyboard or clicking the mouse central button, as you already know (if not, then please see the section *"Learning to draw"* a few pages back). You can disable them or **limit direct snapping to the active layer only** if your drawings are heavily loaded



and the objects in the background layers are not of interest for drawing. **Limiting rapid data palettes to the active layer only** decides if the objects that are present in the background layers must or must not appear in the quick data palettes at the top of the screen. If enabled, your palette may become saturated by all quotes. On the other hand, this helps you quickly assign to an object the same value as another object in an alternate layer. The **Shift key for transferring between layers** lets you duplicate from one layer to another one using the rapid palette. **Allowing the automatic zoom shift** slides the visual window when you are moving a selection and the mouse comes close to the border of the view. If you **activate the touch-screen mode**, in which case the mouse cannot fly over a parent icon, then underlying icons appear when you click – or touch – the parent icon, its own function being called up only if you re-click on it. **Displaying the stack of the last icons used** displays at the left hand side of the screen the last drawing icons that have been clicked, below the normal icons. **Calculating the closed B-Splines for smoothing** changes the calculation mode for the Beta-Splines when their path is closed. Normally, the ends are the only points that the path actually match, other points being attractors. But if the path is closed, then we can considerate that the starting/ending point is also a simple attractor and, in this case, there will be no sharp angle.



Concerning functions that are related to the machining process, you can decide to **always measure the tool** when arriving at the workpiece origin window, which avoids clicking on the button. This automatic measurement can also be made afterwards, **at the process start**, namely when clicking on the big yellow button. If you have a variable support for your workpieces, then the tool measurement on a fixed sensor must be corrected after **asking for the thickness of the machine bed** which can be variable. Then, when arriving at the workpiece origin window, Galaad will ask which bed is under the workpiece, or rather its thickness, for correcting the measured Zo origin. Finally, **asking for the origin offset on *"Z-ok"*** will pop up a small dialogue box when you click on the green button for validating the Z position of the workpiece origin, in which you may clarify that the origin point is actually not where the tool is, but lower in the work-piece material, at a coordinate that cannot be reached.

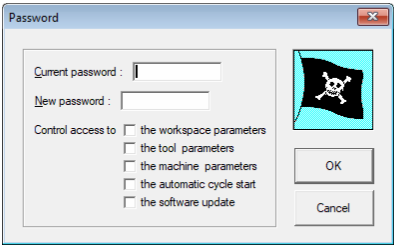


Moving the axes when the origin is dragged with the mouse positions the tool at the new origin the workpiece is dragged with the mouse in the viewing window at the bottom left of the part origin screen. Finally, the **driving window style** lets you choose the theme of the machining module graphical interface, between the old raised rectangular buttons of Windows 95, the semi-transparent *"candy"* buttons of Windows 7, the thin click zones of Windows 10, or a black interface with haloed buttons. You can also choose to display the machining window in ***"Full Screen"*** mode, so that it takes up all the available desktop space, without titles or borders. In this case, remember to ask Windows to automatically hide the taskbar in Desktop mode.

Important: in a dialogue box or even a complete parameter window, when the text of a checkbox ends with **suspension dots**, this indicates that a sub-dialogue box will pop up like a furuncle if you enable that option, and only in that case. This means that when you disable the checkbox, nothing happens. To access this sub-dialogue box when the option is already enabled, either you click twice on it, once for disabling and once again for re-enabling, which will open the underlying dialogue box, or you press Ctrl key when clicking. In this case, the option is not disabled and the dialogue box appears.

Password protection for parameters

If by any chance your workshop or classroom is visited by curious, playful little mice who find nothing more amusing than tinkering with your working environment, the simplest and most radical solution is to surround it with a barbed wire fence and armed guards. In other words, to lock down the various access points with a password: environment settings (described in this chapter), tool settings, machine settings, automatic cycle launch, and finally software updates.



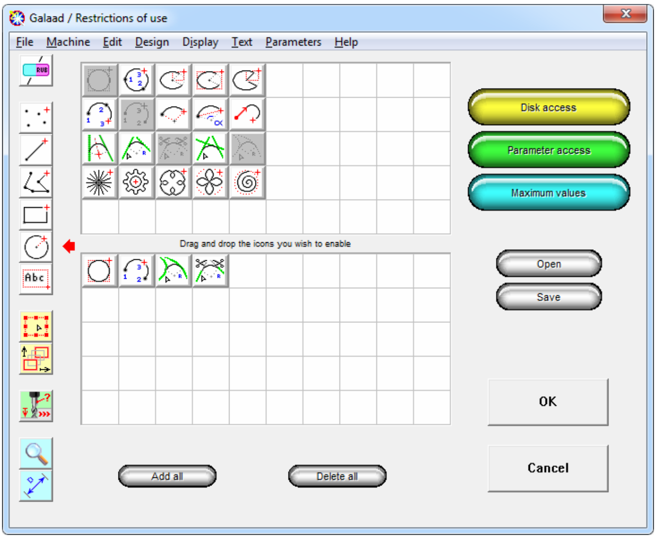
On installation, the default password is *"**galaad**"*, which admittedly is not very original. Entering the password is not case-sensitive. Nor are you asked to use a combination of letters, numbers and special characters. In fact, you can enter whatever you like, even a single letter, it is up to you.

Do not shout this from the rooftops: **if you lose your password**, you will find it in the **PASSWORD.TXT** file in the Galaad installation directory. Simply double-click on this file and Windows Notepad will display its contents.

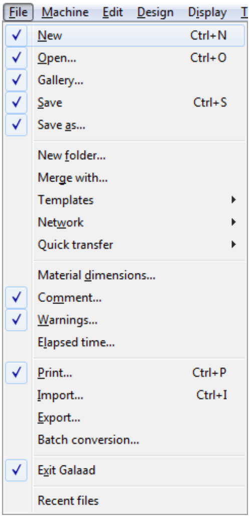
Restrictions for training purposes

Galaad gives the possibility to modify its own appearance for the user's eyes. The software is richly supplied with icons and functions, but this wealth can become layer of complexity so you have the means for depleting it. By default, Galaad offers its full palette of flavours without restrictions, but suggests that you trim it according to the needs of your educational project. Who can do the most can do the least. The purpose is nothing less than **suppress the functions that you consider useless**, and impose limits to the user. This aims to slim your software, at least in its appearance.

The trimming is done through the command *"Parameters / Workspace / Restrictions / Change"*. As its name suggests, the game will consist of eliminating menu functions and drawing icons. **Three restriction levels are pre-defined**, that you may pick directly without having to set them up. These levels correspond to files *"**Level-N.cus**"* that you can modify as you wish (see later on). You can also open a restriction set that has been previously defined. This gives you access to a set-up window for this workspace:



When this command is called, Galaad disappears and leaves the unoccupied room for a new window where you can select all features. This window uses the same menus and the same drawing icons as the central application, plus a few buttons that give access to complementary settings. You can save a set of restrictions from the definition window, using *"Save"* button, and *"Open"* for loading it back.



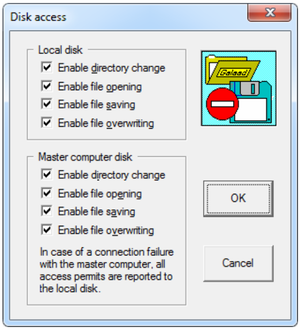
For suppressing the access to a command in a menu, you just have to **open the menu and tick-off the line by clicking on it**. For example, if you want to delete the line *"New folder"* of the *"File"* menu, click on it to remove its checkmark. When you will return to the normal drawing window, that line will have disappeared in the menu.

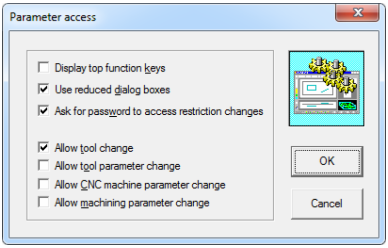
If all commands of a cascaded submenu have been suppressed, Galaad itself will then suppress the corresponding parent line in the previous menu, to avoid displaying an empty submenu. **Will remain visible only the menu lines headed by a check-mark** or which open a non-empty cascaded sub-menu, one single line being then sufficient for keeping the parent line.

Neither is it complicated for **drawing icons**. The window displays two bunk matrixes. When you click on a parent icon at the left hand margin, then all available icons of the series are displayed in the top matrix, at their default location. Just **drag and drop the icons of interest down to the bottom matrix**, choosing for each one the location that is right for you. Conversely, drag up an icon from the bottom matrix to the top one for removing it from your list.

Galaad abhors emptiness. Consequently, it demands that you do not leave any boxes blank in your personal matrix. **The number of lines and columns must be homogeneous, with no blank icons**. Anyway the software quickly finds them and marks them with an infamous red box. Clicking another parent icon at the left hand side is sufficient for validating a matrix. But, as long as you have left a red empty box, you cannot validate. You must arrange your icons for obtaining a compact set. It is possible to move the icons in the bottom matrix for changing the arrangement.

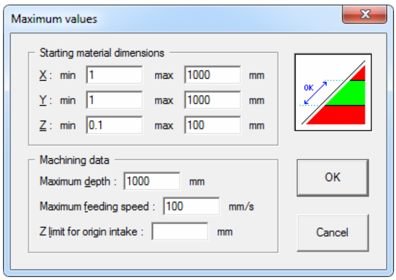
The yellow button for **disk access** opens a dialogue box that helps you manage the authorisations for accessing all disks available from the user's workstation. In addition to the local disk drives, a networked computer can eventually read and write from/to the distant disk of the main workstation. If you disable these checkboxes, then you forbid the access to other directories and restrict access for read, write and overwrite operations.





The green button for **parameter access** calls up the dialogue box that decides whether the user is allowed to modify some sensitive technical settings. The few restricted dialogue boxes that have been defined substitute, for the most used, light versions. So they limit the access to corresponding data and help towards a progressive understanding. Only some complex dialogue boxes have their equivalents restricted. The **password for accessing the restrictions** allows you to forbid any change in the given restrictions, except for the user who has the password. This password is ***"galaad"*** when installing (case not sensitive). Do not shout it from the rooftops, but **if you lose your password**, then you can recover it from the file **PASSWORD.TXT** in Galaad installation directory.

The blue button for **maximum values** opens the dialogue box which will let you fix some limits to the numerical data that the user is allowed to enter. These values concern the board size and the machining data (depth and feedrate) for the objects drawn, plus a maximum stopper for the workpiece Z origin.



Do not forget that you may forbid the use of another tool than the current one, with the dialogue box which is related to the green button (see above).

Now let us summarise the restrictions that you can impose on Galaad from this parameter window:

- **Suppression of lines deemed unnecessary in** menus. The surviving lines that will remain displayed are headed with a checkmark; the lines that open cascaded submenus disappear if the submenu has been emptied.

- **Set-up of drawing icons**. For each series of icons, the bottom matrix shows those that will be displayed and their layout.

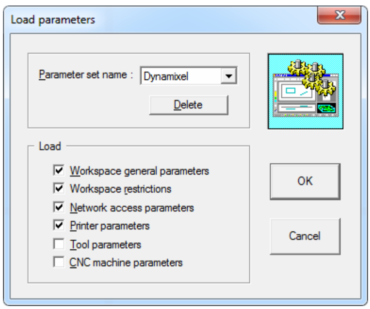
- **Disk access authorisations**. The operations of files are regulated for the local disk drive and for eventual networked disks.

- **Permission for modifying parameters**. The access to machining technical data is subject to a set of authorisations.

- **Guidance for numerical dimensions**. Limits are imposed to the data that the user may enter.

With these restrictions, you will see that it is possible to build up a very undressed Galaad, and return to a normal display without losing your customisations by calling the command *"Parameter / Workspace / **Ignore restrictions**"* (this command is password-protected, if any). Ignoring the restrictions rebuilds a full-power Galaad.

You can also save your restrictions using commands *"Parameters / Save parameters"* and *"Load parameters"*; or transmit them from one workstation to another one using *"Parameters / Quick transfer / Send parameters"* and *"Receive parameters"*. Then a small dialogue box helps you filter the set of parameters you have received.



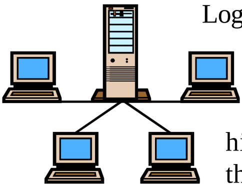
9

0 1 0 0 1

USING A NETWORK

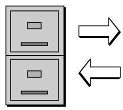
Sharing disks and folders

One of the important functions in Galaad concerns the interconnection of several workstations and the means of exchanging drawings and parameters. The professional user will find it advantageous to reserve one computer for design work and dedicate another specifically to the machining work. In an educational environment, it is useful to be able to prepare the working environment for all students, including the drawing and organisation of their files, from one single workstation.



Logistically, it is necessary for all the computers to be connected to a local area network and able to run Windows. This network does not need a client / server architecture or any particular hierarchy. as Galaad defines its own hierarchy between the master workstation and all its slave units.

Let us consider the following network, with a master machine M, and several slave units S1, S2, S3, *etc*. Galaad is installed on M, in the folder C:\GALAAD, and also on all the S slave units in folder C:\GALAAD. As well as accessing its local hard disk C:, each slave unit S can also access the hard disk C: on the master computer M, which will be remapped to another drive on the slave units, for example X:. In this way, Galaad running in the folder C:\GALAAD on workstation S1 sees the copy of Galaad installed on the master machine as being in folder X:\GALAAD. Thereafter, the slave unit can load its parameters, and current drawing, from the master machine M instead of taking them from its local hard disk. This gives the user of the master machine M the ability to control the configuration of each slave machine S. What is more, the slave units S can store their work centrally on the master machine instead of on their local hard drives.



To enable data to be exchanged between workstations, each slave unit must be able to access the hard disk on the master computer, or at least the folder where Galaad is installed. It is assumed that your network is already configured and that disk sharing has been enabled. If not, see the **file sharing** parameters in Windows Control Panel.

However, the point is not to have a close look at your network, but simply to show the principle that part of the hard disk on the master computer has to be accessible to each slave unit. If this is not the case then you should consult your network administrator. Note that it is not necessary for the master workstation to be able to access the hard disks on the slave machines, as Galaad does not need this.

Main workstation

To use Galaad on a network, it is necessary to define one workstation as the master and all others as slaves to it. The slave units will copy the working environment on the master computer by taking their parameters from it, and possibly save their drawings back to it. It is normal for the CNC to be connected to the master machine but this is not mandatory. You have probably only got one dongle for the whole work group. **The dongle must be plugged into the master computer**. By loading their parameters from across the network the slave units will see the licence and will validate it locally.

If your network only comprises **two computers, one for design work and one for the machining process**, plug the dongle into the machining computer and consider this to be the master, the drawing computer does not need a licence except for exporting files. It is better for this master computer to be running when you start Galaad on the secondary machine, so that it will be able to access the dongle and allow exports.

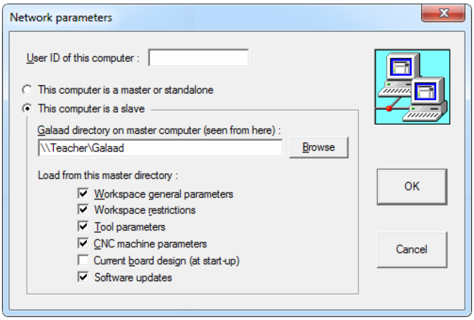
The master workstation acts as the model that the slave units will copy when Galaad is started on them, and consequently is the same as a stand-alone installation. It operates the same whether or not a network is present. The slave units collect their parameters from the master machine; it does not send them to the slave units one by one. **There are no messages exchanged between computers through the network, but only an access to the main computer from the secondary computers.**

Configuring a master installation is done in the simplest way possible. In the network parameters, simply indicate that this computer is a master or standalone, that is it.

The user of a master installation of Galaad – generally the teacher – opens the software and configures the working environment to suit themselves, or perhaps loads a profile, saved somewhere. He or she can also open a default drawing, that secondary workstations will get when starting their own Galaad. He/She then uses the *"Parameters / Network / Upgrade workspace"* command to save the changes to disk. The users of the slave units – usually the students – then start Galaad, which having copied the parameters from the master computer, display the default drawing.

❑ Secondary workstations

You have seen that the students need to load their environment parameters from the copy of Galaad installed on the teacher's workstation. This requires them to have access to the hard disk on the teacher's machine via the local area network. For a simpler installation, with functions limited to down-loading the master parameters, you can restrict access to the Galaad folder on the master workstation, but the other functions would require write access.



Defining a workstation on a network is done from the *"Parameters / Network / User"* command. The dialogue box that appears then allows you to choose the type of workstation, main or secondary, and control the functionality for a slave installation. All computers can also have an ID.

Off-topic: if you do not have a network but **several Galaad installations coexist on one single computer** with different settings, for example to control several machines, then **the ID of the computer** (here in fact of the installation) will be displayed in the window title bar, to help you remember which Galaad you are currently running. Nothing forbids storing the files related to these different installations in a common folder. But let us come back to the use of a network.

If your workgroup comprises several slave workstations, it can be useful to specify here also the identity of the workstation. We will see the point of this a little later. The most important parameter is obviously the definition of the **Galaad directory on the master workstation,** as seen from each slave unit. For example, if Galaad has been installed in the folder C:\GALAAD on the master computer, this folder can be mapped as X:\GALAAD or \\PC-1\GALAAD or other on the secondary workstations. You can search for this distant drive by clicking on the *"Browse"* button. It is obvious that access to the network must be available in order for you to browse it.

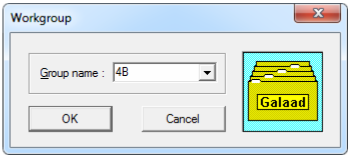
All that is left is to define which parameters will be copied from the master installation. Besides the main environmental and machining parameters, it is also possible to retrieve the **current drawing** from it. In this case, the copy of Galaad on the slave unit will open and display the drawing that has been previously prepared by the teacher.

If the network is down or there are difficulties accessing the master workstation, the slave units will temporarily revert to stand alone mode and forget about the master installation until they are restarted. This allows the group to carry on working in the event of a network problem. It is pointless dwelling on the effects of the network restarting, as the slave units will return to their normal mode of operation.

When the parameters have been changed on the master machine, it is not absolutely necessary to restart the copies of Galaad currently running on the slave units in order to update the workspace. Simply use the *"Parameters / Network / Upgrade workspace"* command on the master machine then on each slave unit to first save the current environment then load the new parameters.

❑ Exchanging files

Using Galaad on a network is not limited to downloading the environment parameters. To help workgroups, the software allows files to be stored centrally on the master computer. The method is very simple, but of course best explained with an example: Galaad is installed in the folder *"C:\GALAAD"* on the master workstation, which will be mapped as, say *"X:\GALAAD"* (or *"\\TEACHER\GALAAD"*) when seen from the slave unit *"TOTO"*. In addition, the teacher has designated the workgroup as *"4B"*.



This **workgroup** name, not previously mentioned, is set by the *"Parameters / Network / Workgroup"* command. The dialogue box asks for a name and offers a list of those previously used.

One or more students can work on *"TOTO"* and run Galaad, with their workspace parameters loaded from *"X:\GALAAD"*. They will open and modify the drawing, which can then be saved under the name *"DRAWING"* (for example). Given that their restrictions only allow them access to the hard disk on the master computer, their work will be saved to the master folder *"X:\GALAAD\PUBLIC\4B\TOTO\DRAWING.GAL"*. However, the teacher will collect all the drawing files of that workgroup into his/her master folder *"C:\GALAAD\PUBLIC\4B"*, with each sub-folder being allocated to a workgroup. If the workgroup changes, for example from 4B to 3A, the drawing from workstation *"TOTO"* will be saved into folder *"X:\GALAAD\PUBLIC\3A\TOTO"*. Therefore there will be no confusion between the different groups using the same workstations.

To check the work produced by the students working on the workstation *"TOTO"*, the teacher opens their drawings in the corresponding subdirectory. They can be corrected or marked ready for the next session. Most importantly, he/she can open them to start the machining operation. At the end of the year, the teacher tidies up by deleting folders 4B and 3A, using either Windows Explorer or Galaad, and can then go away on holiday with a conscience as clean as the hard disk.

The teacher also has the option to save some **public files** in his/her own local folder *"C:\GALAAD\PUBLIC\4B"*. These files can then be opened by the students using the *"File / Network / Open a public file"* command. They will then be saved into a public file available to the entire group.

10

0 1 0 1 0

DRAWING ICONS

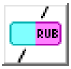
This chapter does not aim to go into a lot of details about special drawing techniques for every icon, but to provide a brief overview of their use and add a few clarifications in some cases. They are particularly numerous and we are not going to cut trees down for more pages. The best way to become familiar with their purpose is to actually use them rather than simply read about them.

As a reminder, let us start with a few constants already forgotten since the far off chapters 2 and 5:

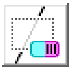
- When you draw an unfinished shape (isolated points, polyline, curves, *etc.*), you must click the ultimate point with the right mouse button for stopping the construction.
- When you draw a shape with several different steps (ellipse, star, radii, *etc.*), you can freeze the construction as is by clicking here again with the right mouse button (or press Esc or End buttons after the last point has been clicked).
- The ← key (backspace) undoes the last click without cancelling the construction in progress.
- Once the shape has been completed, whatever it is, the drawing functions reiterate the construction of a new shape of the same type. To stop there, press Esc key or the yellow icon for selecting (parent icon at left).
- The keyboard arrows ←↓↑→ move the plotting cursor by one unit on the magnetic grid towards the corresponding direction. The + / - keys set the magnetic grid to a step 10 times smaller / bigger. The * (star) key resets the grid to automatic mode, which always corresponds to the smallest scales of the rulers, whatever the zoom level is.
- The mouse central button and the *Space* bar on the keyboard snap the plotter position to the suggested point that appears in orange near the cursor. If there is no suggested point, then the current cursor position is validated as is.
- The shapes that need a line drawn from the previous point (dotted line, line, polyline, circle radius, *etc.*) are submitted to the polar magnetic grid when you keep Ctrl key pressed while moving the mouse.
- In the same way, the combination of ⇧ Enter keys pops up a dialogue box for a polar dimension (distance and angle).
- You can zoom in or out without leaving the current drawing function, using the mouse wheel or ⇟ / ⇞ (Page Down / Page Up, or PgDn / PgUp) keys. This helps a lot for snapping to a suggested point among a big jungle of objects.
- Finally, you can shift the zoom window, without interrupting the drawing function in progress, by clicking-moving with the mouse central button.

❑ Rubbing out

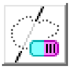
The series of rubbing icons gets Galaad a bit closer to classical applications for drawing bitmap images. Be careful not to leave little bits of paths almost invisible on the screen or a printer, but that will no less be machined. The function *"Edit / Select the smallest object"* may eventually help you clean up the drawing board.



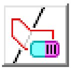
Rub - lets you erase directly everything that is located under the mouse cursor, which then takes the form of a rectangular eraser. You can click and release for a one-shot erasure under the rectangle, or click-and-move for erasing more.



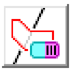
Rub zone - erases everything that is under a rectangle to be defined by a classical pointing.



Lasso rub - erases everything that is inside a zone to be contoured manually by the mouse cursor while the left button is held down.



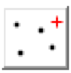
Mask inside - erases everything that is located inside the boundaries of the selected object.



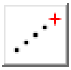
Mask outside - erases everything that is located outside the boundaries of the selected object. Really everything, be careful!

❑ Points

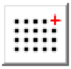
This series of tools allows you to create points, to indicate where holes have to be drilled. When drilling, a feedrate (that is the horizontal motion speed) is meaningless, unless the plunge speed is expressed as a proportion of the feedrate. See the command *"Parameters / Tools"*.



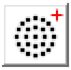
Single points - allows points to be placed anywhere on the board. Each point is a completely independent object, but if you are using the yellow icon *"Group selection"* on a set of single points, then they will fuse into one single object.



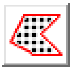
Line of points - places points at regular intervals along a line. The group of points is a single object.



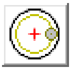
Rectangular array of points - creates a rectangular matrix of points at regular intervals.



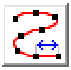
Circular array of points - creates concentric circles of points, each point remaining at a minimum distance from its nearest neighbours, which defines the number of points per radial layer (single object).



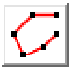
Fill with points - places points at regular intervals (to be defined) inside the boundaries of the selected objects.



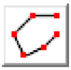
Bored point - drills a hole of a given diameter by integrating the tool compensation immediately. Either the drill-down and bottom right can both be performed at the centre, or the drilling can be helical. A zoomed 3D preview will show the detailed result.



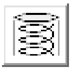
Points on a trace - drills points at regular intervals along a trace selected in red. It is possible to add a small random variation of the distance, to create irregular intervals.



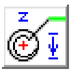
Points on vertices - drills points at the vertices of a polyline selected in red.



Connect points - makes the reverse operation by creating a polyline that links the group of selected points.



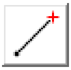
Helix boring - makes an helical plunge for boring or threading using a thread-milling cutter, integrating the tool compensation. A roughing pass is possible, and also a prior drilling with a drill bit. The boring or threading can be outside a cylinder.



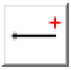
Helix plunge - adds a progressive helical plunge near the entry point of a path to be pointed. Yet again, a zoomed 3D view will show you the detailed result.

Lines

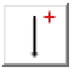
This collection of drawing tools is particularly rich and with good reason, as there are many ways in which choosing a couple of coordinates to draw a line across a battlefield can be done. However, it is probably not useful to state here the various possibilities for plotting, snapping or entering positions of ends, whatever the line may be.



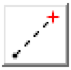
Simple line **-** constructs a basic line between two points by clicking with the mouse or snapping to existing coordinates. Pressing the Ctrl key applies the polar magnetic grid.



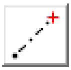
Horizontal line **-** constructs a horizontal line whatever the vertical position of the cursor when adding the second point.



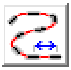
Vertical line **-** as above, except that it is now for a vertical line, wherever the mouse is on X axis.



Dashed line **-** constructs a line comprising regularly spaced dashes, at intervals defined by yourself.

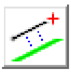


Dot-dash line **-** as above, but comprising alternate dots and dashes, at intervals defined by yourself.

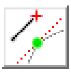


Segments **-** constructs a series of segments as defined by you along the path of an object selected in red.

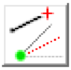
Duplication **-** constructs a copy of the segment to be pointed. The copy must be positioned on the board from the segment starting point. If a segment is already selected in red, it is the one that will be copied.



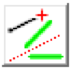
Parallel **-** constructs a line parallel to the segment to be pointed first, and passing through a point to be positioned then. If a segment is already selected in red, it will be the actual reference.



Tangent at point **-** constructs a line on an axis that is tangent to a vertex, which must be pointed first, and passing through another point to be positioned then. If a point is already selected in red, it will be the actual reference.

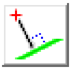


Bisector - constructs a line parallel to the angular bisector of the vertex to be pointed. If a point is already selected in red, it will be the actual reference.



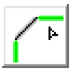
Bisector of segments - constructs a line whose axis is the bisector of the vertex between the segments to be pointed first, and defined by two points to be positioned then. If two segments are already selected in red and blue, they will be the actual references.

Angular - constructs a line at an angle that you must indicate, from the segment to be pointed, then defined by two points to be positioned. If a segment is already selected in red, it will be the actual reference.

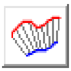


Perpendicular - as above, except that the angle is fixed to 90° relative to the pointed segment.

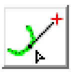
Intersection - constructs a line from a point defined by you to a segment to be pointed first. If a segment is already selected in red, it will be the actual reference.



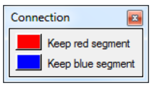
Chamfer - breaks the angle between two joined segments to be pointed, on a length to be defined.



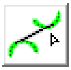
Sheaves - creates a sheaf of lines at regular intervals between an object selected in red and one in blue. If the lengths of both objects are different, then the interval entered corresponds to the red object. If there is no object selected in blue, then the sheaves are constructed between the object selected in red and a free point to be positioned. See also 3D meshes or the function *"Design / Transmutation"*.



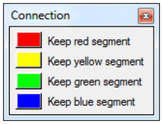
Arc tangent - constructs a line whose starting point slides along the arc of circle to be pointed first. Both tangents to the full circle that support the arc are displayed, one of them being chosen after the final pointing (you can then cancel for keeping both). If a segment is selected in red, then a dialogue box offers the possibility to refer to it for creating the tangent that would also be parallel to that red segment.

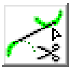


Trimmed arc tangent - as above and following the same sequence, but also trims the arc to the point of intersection once validated. The part of the arc to be deleted has to be selected.



Double arc tangent - constructs a tangent to two arcs to be first selected in green. There are four possible tangents between two circles, including the crossing ones. The one you want to keep has to be selected.



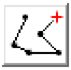


Double arc tangent with single trim - as above, but also trims the first arc selected to the point of tangent, with the choice of which part must be kept.

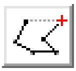
Double arc tangent with double trim - as above but also trims both arcs to the point of tangent.

❑ Polylines and spline curves

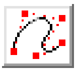
The small family of polyline shapes includes basic polylines, and several non-circular mathematical curves, as well as the shortcuts possible.



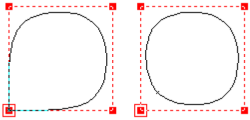
Polyline - constructs a simple open polyline by successively plotting its vertices. All position snaps are valid, and pressing the Ctrl key while plotting applies the polar magnetic grid.

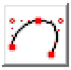


Closed polyline - as above, but the polyline drawn remains closed automatically.

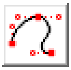


Beta-Spline - constructs a *Non-Uniform Rational Beta-Spline* by plotting the vertices. The NURBS curves have two end points the path goes to, and a various number of attractor points. When the curve is edited (double-click or right mouse click or *"Design / Object / Geometry handles"*), all points can be weighted to increase their attractiveness, by pressing + and - keys. It is possible to jump from point to point forward or backward with ⇥ or ⇧⇥ keys. The NURBS curves are easy to handle. Whenever possible, it is better to limit the number of attractor points and rather change their weights: this is much easier and often more efficient. Note that, for a closed curve, only the ends are part of the actual path, but if you have enabled the option *"Calculate the closed Beta-Splines for smoothing"* in the workspace parameters, then even these end points will become simple attractors, and the curve will have no sharp angle.



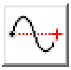


Quadra-Spline - constructs a quadratic curve with joined tangents by plotting the nodes. Warning: editing is sometimes difficult. Note that, when handling a tangent point, pressing ⇧ key allows you to break the curve by dropping the alignment of the facing neighbour, for example to make an inside corner after plotting.

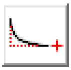


Bezier curve - constructs a curve by plotting the control nodes, the semi-tangents being calculated automatically. As for the B-Spline or the Q-Spline, it is possible to jump from point to point forward or backward with ⇆ or ⇧ ⇆ keys, for example for splitting the curve or conversely for inserting a new sector. By the way, pressing Ctrl key when handling a semi-tangent lets you freeze its angular direction, and the ⇧ key lets you break the curve by dropping the alignment of semi-tangents.

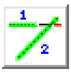
Manual trace - constructs a basic polyline by following the mouse cursor while the left button is held down. Not very useful unless you have children.



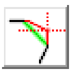
Sine curve - constructs a sine curve along an axis between two points with user-defined amplitude, period and number of cycles.



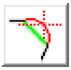
Hyperbole - constructs a hyperbolic curve by plotting directly its amplitude, and then its length and direction.



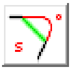
Link/Trim - extends or cuts a segment to be pointed, for adjusting its end onto a second segment, also to be pointed. In the case of a trim, Galaad asks you to indicate the segment side to be deleted.



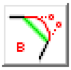
Insert vertex - adds a vertex in a segment to be pointed. That vertex can be repositioned immediately. The double-click with the right mouse button does something almost equivalent.



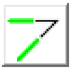
Insert arc **-** replaces a segment to be pointed by an arc whose passing point can be repositioned immediately.



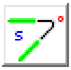
Insert B-Spline **-** replaces a segment to be pointed by a NURBS curve with one single attractor point which is pre-positioned at the intersection of the axes of the neighbouring segments.



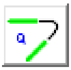
Insert Bezier curve **-** replaces a segment to be pointed by a Bezier curve whose semi-tangents are pre-positioned in the axes of the neighbouring segments.



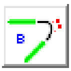
Simple link **-** constructs a polyline link between two segments to be pointed, through the projected point of intersection. If two segments are already selected in red and blue, they will be the actual references.



Beta-Spline link **-** constructs a NURBS curve with three points, linking two segments to be pointed, and using the projected point of intersection as attractor point.



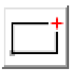
Quadra-Spline link **-** constructs a curve with two nodes, using the segments as semi-tangents.



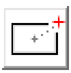
Bezier curve link **-** constructs a Bezier curve with two nodes, using the segments as semi-tangents.

Rectangles and polyhedrons

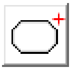
This abundant collection of tools includes several non-Cartesian shapes and also ones that have not got much at all to do with rectangles, other than being regular closed polylines.

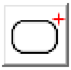


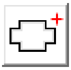
Rectangle **-** constructs a Cartesian rectangle in the classic way by defining one of its diagonals.

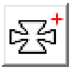


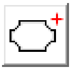
Centred rectangle **-** constructs a Cartesian rectangle by defining the centre and one corner.

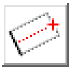

***Rectangle with chamfered corners* -** constructs a Cartesian rectangle with an extra pointing of a chamfer applied to corners.

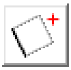

***Rectangle with fillet corners* -** as above, except that the corners are rounded with 90° arcs.

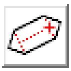

***Rectangle with inverted corners* -** constructs a Cartesian rectangle with the corners turned inwards.

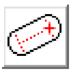

***Rectangle with clipped corners* -** as above, except that the corner can be placed anywhere within the rectangle up to the centre point.

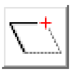

***Rectangle with inverted fillets* -** constructs a Cartesian rectangle, with fillet radii that are inverted about their points of tangency.

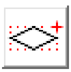

***Oblique rectangle* -** constructs a non-Cartesian rectangle by drawing two adjacent sides. Remember that pressing Ctrl while pointing applies the polar magnetic grid.

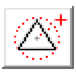

***Oblique square* -** constructs a non-Cartesian square by drawing one side then indicating the direction to complete the square.

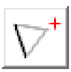

***Cartouche with two sharp sides* -** constructs a non-Cartesian polyhedron, with 90° end angles, by drawing two adjacent sides.

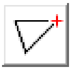

***Cartouche with two fillet sides* -** as above, except that ends are semi-circular instead of angular.


***Parallelogram* -** constructs a non-Cartesian parallelogram by drawing two adjacent sides.

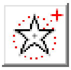

***Diamond* -** constructs a diamond by defining one of the diagonals of a Cartesian rectangle that surrounds it.


***Centred equilateral triangle* -** constructs an equilateral triangle by marking out a circle that inscribes it.

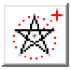

***Equilateral triangle* -** constructs an equilateral triangle by marking out one side and a general point in connection to this side.



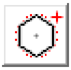
Isosceles **-** constructs an isosceles triangle by marking out its base followed by its height.



Star **-** constructs a star with N points, to be defined by the external circle in which it inscribes, then the interior circle which inscribes in it. *Important:* click the first circle using the right mouse button to freeze the construction as it is.



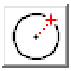
Crossed star **-** as above, except that the transverse lines of the vertices are self-crossing.



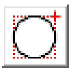
Centred polyhedron **-** constructs a regular polyhedron, defined by drawing the circle in which it is inscribed.

❑ Arcs and cyclic shapes

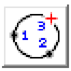
The large and powerful tribe of arcs gives access to tools for constructing all kinds of circular curves by a variety of different methods. Some trigonometric curves are also included in this series. It should be remembered that when drawing a toolpath for machining purposes, circles are not hermetically closed shapes but open 360° arcs that have a start point and an end point, defining how the cutter will travel.



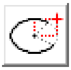
Circle **-** constructs a closed circle by plotting its centre and a point on its circumference that will become the start/end point. Here again, pressing the Ctrl key when setting the start point position, applies the polar magnetic grid.



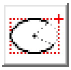
Inscribed circle **-** constructs a circle by defining one of its diagonals within the Cartesian rectangle that inscribes it. The start point is set at 0° according to the trigonometric reference (3 o'clock in the aviation lingo).



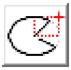
Three points circle **-** constructs a closed circle by plotting three points on its circumference. The starting point is at 0° and the path runs counter-clockwise.



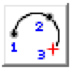
Ellipse **-** constructs an open Cartesian ellipse by defining its centre, X and Y radii, and start and end points. To draw a closed ellipse, right mouse click or Esc key in due course.



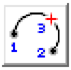
Inscribed ellipse - constructs an open Cartesian ellipse by defining one of the diagonals of its inscribing rectangle.



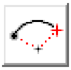
Pie sector - constructs an open Cartesian ellipse (as above), then closes it by drawing the radii from the start and end points. The whole path consists of two lines and one elliptical arc.



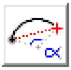
Arc from three points, end point last - constructs an open arc by plotting the start point, then an intermediate point, and finally the end point, clicked last.



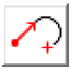
Arc from three points, intermediate point last **-** constructs a circular arc by plotting the start point, then the end point, and finally the intermediate point. Probably more efficient than the previous icon.



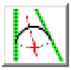
Arc from centre and aperture angle - constructs an open arc by plotting its centre, then its start point (giving the radius), and last its end point direction, the radius then varying no more.



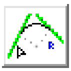
Arc from aperture angle and two points **-** constructs an open arc by plotting its ends with a pre-defined aperture (the centre is floating). Be careful with the plotting sequence: the arc always turns counter-clockwise.



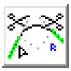
Arc from tangent - constructs a circular arc, tangent to the endpoint of a segment selected in red, by plotting its end point.



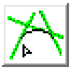
Arc sliding on two tangents **-** constructs an arc with variable radius, tangent to two pointed segments, then positioning the centre.



Arc with fixed radius on two tangents- constructs an arc with a given radius, tangent to two pointed segments.

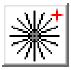


Arc with fixed radius on two tangents, with trims - as above, except that the tangent segments are trimmed.

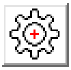


Arc from three tangents - constructs an arc defined by three tangent segments to be pointed.

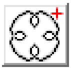
Arc from two tangents on curves - constructs a linking arc by pointing two tangents paths that can be either lines or curves. When several solutions are possible, then the closest to the click position is validated. Then, for each tangent curve, the software asks which part must be kept, the other one being deleted. To keep both parts, just press Esc key for closing the small window.



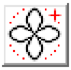
Radii - constructs a group of radial lines by defining the inner & outer enclosing circles and the start & end angles with the cursor. Right mouse click for freezing the construction as is.



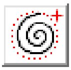
Gear wheel / Rack - constructs a simple rack or pinion by defining the key parameters and positioning the result with the cursor. In the case of a gear wheel, a coaxial circle is added to the construction.



Epicycloid - constructs an epicyclical curve within a circle defined with the cursor. The shape can be built regardless of its size, then extended or reduced.



Rosace - constructs a cyclic shape with petals inside a circle defined with the cursor.

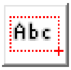


Spiral - constructs a spiral from a number of turns, to be defined, and by plotting the inner & outer enclosing circles with the cursor. Right mouse click freezes the spiral at the first circle (*i.e.* external).

❑ Text

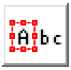
The tools for working with text are few. Except for the initial creation and manipulation of individual letters, most work is performed with the text editor, as it remains in text form for later editing, unless converted by you to simple polylines. Please refer to the *"Text"* menu and its advanced functions, plus the ***"AutoText"*** argument from the command line for automatic updates on the board (see the chapter that explains the technical matters, at the end of this manual).

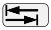
Important: a straight text or a text written along a path keeps its own properties unless it suffered a geometrical torture. So you can modify its style or its content afterwards.

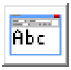


Text block - creates a paragraph of text with the current font settings, contained within a rectangle defined by the cursor. It is possible to write vertically or reverse the text by setting the corresponding parameter in the text entry dialogue box.

Text on trace - places text along the path of a shape selected in red. The text is written from the start point to the end point of the shape, that represents the water-line. The shape remains memorised even after it has been modified or deleted.



Arrange letters - selects individual letters in a piece of the already selected text, so that they can be dimensioned or repositioned one by one by pressing  keys or simply using the mouse. You can jump quickly between letters with the tab  key (next) or  (previous). Warning: these manual changes are lost if the text or its style are modified thereafter. Note that a click on a text with the **right mouse button** lets you handle directly the pointed letter.

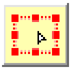


Edit - opens an edit and format box for the red-selected text so that it can be modified, including its layout in a paragraph. A double-click on a text block does the same, and also the key combinations that are indicated in the lines of the *"Text"* menu.

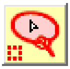
Auto-incremented text - allows you to write a block of text that will be incremented automatically during the milling process, single or serial. When starting the cycle, all auto-incremented texts are reviewed and you are prompted to give their new starting content. A normal text that has already been written and selected in red can also become auto-incremented afterwards, using this icon.

Selections

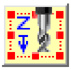
The selection tools cover an area wider than simply highlighting objects, points and segments, offering specialised functions for editing routines.



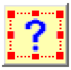
Select objects **-** switches from drawing plotter mode to selection mode (selection mode is the default). The cursor is shown as an arrow. The Esc key does the same to return from active drawing to selection mode. If an object has just been drawn, then clicking on this icon selects it immediately. In the same way, if only one object of the drawing can be selected when clicking this icon, then it will be. By the way, take a look back to the *"Selections"* paragraph in "Advanced drawing techniques" chapter.



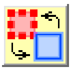
Lasso selection **-** manually traces the contour of a zone in which will be selected all objects of the active layer that are completely inside the boundaries. The contour follows the mouse in clockwise direction (red) while the left button is held down, just like a manual trace. In counter-clockwise direction (green), the objects partly inside are also eligible.



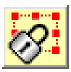
Select equal depths **-** selects all objects with the current machining depth, or with the same depth as the current selection. See also the rapid data palettes with right mouse double-click.



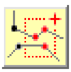
Filtered selection **-** selects objects by their machining parameters or by selected graphical characteristics.



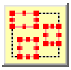
Swap red/blue objects **-** makes objects selected in red become blue selected objects, and vice versa. Pressing the Ctrl key while pointing an object (or a segment or a point) selects it directly in blue. Blue selections are used as secondary selection, for example for duplicating a selected object along a blue-selected path.



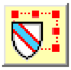
Lock selection **-** locks all objects selected in red, which avoids them being selected in the future. See also *"Edit / Unlock"* menu.



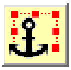
Select range of points **-** allows you to handle together a set of points that belong to various objects. Please refer to the chapter *"Advanced drawing techniques"* for moving and stretching groups of points. Also note that **selecting a zone with the right mouse button** makes a selection of the group of points.



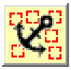
Group selection - associates together all objects currently selected in red so that they can be handled as a single item in the future. If all selected objects are already associated, then Galaad asks if you actually want to ungroup them without going via the *"Edit"* menu. Please refer to the chapter *"Advanced drawing techniques"* for a more detailed description of this icon and the following ones.



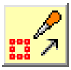
Protect selection - protects all objects selected in red, which prevents deletion and changes made to their shape. If all selected objects are already protected, then Galaad asks if you actually want to unprotect them without going via the *"Edit"* menu.



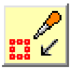
Anchor to position - anchors all objects selected in red to their current position on the board. They can, however, be re-scaled. If all selected objects are already anchored to position, then Galaad asks if you actually want to free them without going via the *"Edit"* menu.



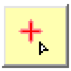
Anchor together - ties together two or more objects selected in red so that they maintain their relative position and act as a group. This means that when you are moving a selected object and other objects are anchored with it but are not selected, they are also moving. If all selected objects are already anchored together, then Galaad asks if you actually want to free them without going via the *"Edit"* menu.



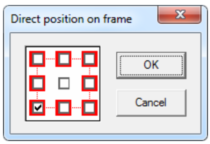
Copy position & dimensions - loads the position and dimensions of the selection frame, so they can be injected after into another selection (see related icon just below).

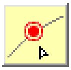


Paste position & dimensions - applies the previously copied position and dimensions into the current selection frame.

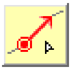


Plot red cross - places a fixed red cross on the board as a reference marker when drawing or as the workpiece origin. This cross is plotted directly on the board and has no link with any objects. For suppressing a red cross, just click again on that icon and abort the plotting operation with Esc key (or press Ctrl). If a red cross already exists, Galaad suggest repositioning it. And when objects are selected, instead of pointing the position, Galaad can snap it onto a selection frame handle.

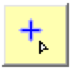




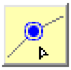
Select red point **-** selects a point in red, even on objects that have geometrical properties (arcs and curves). The right mouse click on a polyline vertex or a single isolated point directly selects that vertex or point with no need to call up this icon, except if the whole object is itself already reactive to the right click.



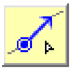
Select red segment **-** selects a segment in red, even on objects that have geometrical properties. The right mouse click on a segment directly selects that segment with no need to call up this icon, except if the whole object is itself already reactive to the right click.



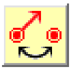
Plot blue cross **-** like the red cross, mentioned above, but in blue so that a second reference marker is available. That blue cross can be helpful for the drawing or the workpiece origin, for example for setting a double origin allowing adjustments.



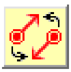
Selection blue point **-** does the same as above for a blue point. The right mouse click on a polyline vertex or a single isolated point, while keeping Ctrl key pressed, directly selects it with no need to call up this icon, except if the whole object is itself already reactive to the right click.



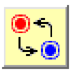
Select blue segment **-** does the same as above for a blue segment. The right mouse click on a segment, while keeping Ctrl key pressed, directly selects that segment with no need to call up this icon, except if the whole object is itself already reactive to the right click.



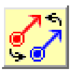
Swap red point/segment **-** jumps from red point selection to red segment selection, and vice-versa.



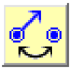
Flip red segment **-** reverses the direction of the segment selected in red, without changing the overall toolpath of the object that it is on.



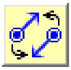
Swap red/blue points **-** turns a point selected in red into one selected in blue and vice-versa.



Swap red & blue segments **-** turns a segment selected in red into one selected in blue and vice-versa.



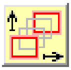
Swap blue point/segment **-** jumps from blue point selection to blue segment selection, and vice-versa.



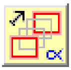
Flip blue segment **-** reverses the direction of the segment selected in blue, without changing the overall path of the object that it is on.

❑ Special effects

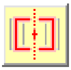
The special effects provide a comprehensive set of tools that allow the objects selected in red (and these only) to be tortured according to your darkest fantasies. Manipulation can be from simple 2D positioning to 3D bending, including miscellaneous rotations and projections. Please note that objects lose their geometrical properties (arcs, curves and texts) if the torture applied changes their shape.



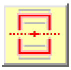
Cartesian move **-** allows the object selected to be moved in all directions by values entered for X, Y & Z.



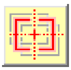
Polar move **-** allows the object selected to be moved in the X & Y directions by values entered for the angle and radius.



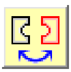
Centre horizontally **-** centres the selected objects about the vertical axis of the cursor. See also *"Design / Align-centre"* that allows you to use many references on the board. Clicking on this icon while keeping Ctrl key pressed makes an **automatic centring on the board X centre**.



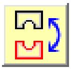
Centre vertically **-** centres the selected objects about the horizontal axis of the cursor. See also *"Design / Align-centre"* that allows you to use many references on the board. Clicking on this icon while keeping Ctrl key pressed makes an **automatic centring on the board Y centre**.



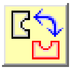
Centre both **-** simultaneously centres the selected objects about both the horizontal and vertical axes of the cursor.



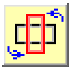
Flip horizontally **-** inverts the selected object about the vertical axis within the selection frame.



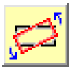
Flip vertically **-** inverts the selected object about the horizontal axis within the selection frame.



Flip diagonally **-** exchanges the X and Y coordinates of the selected objects, without changing the frame origin (southwest corner).



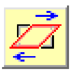
Rotation 90° **-** rotates the selected objects by 90° counter-clockwise, maintaining the original position of the selection frame.



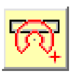
Rotation **-** rotates the selected objects around the centre point of the selection frame. *Very important:* the [Enter] key allows you to enter directly of the angle value or change the pivot without ending the operation.

Tip: you can rotate the selected objects around the centre of the selection frame directly from the **mouse wheel along with the** [Ctrl] **key** for a rotation of 1° per increment of the wheel, or 5° per increment if you also press [⇧] key. This can help you optimise the wastage by nesting objects manually: positioning and rotating are then available from the mouse.

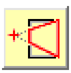
Small tip, delightfully useless: along with [Ctrl] + [⇧] keys, the rotation icon animates the drawing by rotating continuously the red selected objects (counter-clockwise) and the blue selected objects (clockwise). The rotation speed of each object is in proportion to its depth, 1 mm corresponding to 45 °/s. Each object turns around its own centre of symmetry, unless it is associated with other objects, in which case it turns around the centre of symmetry of the group. The [Esc] key ends the animation. This function can simulate the rotation of a set of pulleys or gears, but even so it does not verify the validity of the speeds. Among the files that are supplied with Galaad, **SAMPLES \ ROTARYGEARS.GAL** gives a good example.



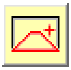
Slant **-** tilts the selected objects about the vertical axis at the centre of the selection frame. The [Enter] key allows the direct entry of the angle.



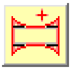
Wrap **-** rolls up the selected objects along a double circle. The operation follows several steps: plotting the wrapping centre, plotting inner and outer radii, plotting the aperture angle (right mouse button for ending), and last of all plotting the orientation.



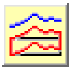
Vanishing point **-** modifies the selected object so that segments in the chosen direction project back to the vanishing point.



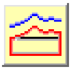
Perspective - modifies the selected object so that it appears to have depth and disappear into the page.



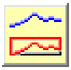
Panoramic - modifies the selected object so that it appears to be a wrap-around, panoramic view.



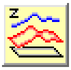
Bend **-** modifies the object selected in red so that its Y profile follows the variations of the object selected in blue. The blue object must occupy on the board a horizontal space X at least as large as the red object to be modified. On the other hand, its Y position has no importance. This remains valid for the next three icons.



Stretch **-** as above, but only applied to the top of the object selected in red, for expanding it vertically.

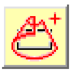


Compress - as above, but only applied to the base of the object selected in red, reducing it vertically.

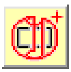


Modify Z depth **-** modifies the object selected in red so that its depth follows the Y path of the object selected in blue. The XY path of the blue object becomes the XZ reference for distorting. So the appearance seen from the top remains unchanged.

Project onto cylinder - modifies the object selected in red to produce a 2D visual representation of its projection onto a transparent cylinder. Be aware that this icon and the two next do not make 3D distorsions but simple projections for artistic effects.



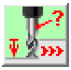
Project onto cone - modifies the object selected in red to produce a 2D representation of its projection onto a transparent cone.



Project onto sphere - modifies the object selected in red to produce a 2D representation of its projection onto a transparent sphere.

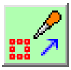
Milling data

Within this small group of features the first one will probably be used frequently. The pop-up items are little gadgets that may actually turn out to be practical in some cases. See also the rapid data palettes to copy/paste depth, feedrates and tool numbers.

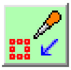


Machining data **-** gives access to the milling tool, the depth and the feedrate. If some objects are selected, then the new values apply to these objects, otherwise to the next objects to be drawn.

Colour / thickness **-** defines trace colour and thickness as default settings or for selected objects. Of course this is useful only for the screen display and the printout.



Copy machining data **-** copies the tool, depth and feedrate of the selected object for later use. See below.



Paste machining data **-** applies the previously copied tool, depth and feedrate to the selected object. See above.

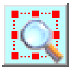
Zoom

The zoom icons allow you to change the magnification of the view so that the drawing can be seen more clearly without affecting the objects. Several successive views are memorised, therefore allowing you to backtrack, which is entirely independent of the Undo/Redo functions in drawing mode. Note that **fast zoom in/out** features around the cursor are also available using the ⇟/⇞ (or PgDn/PgUp) keys without losing the current drawing function. If your mouse has a **central wheel**, which is highly recommended for the 21th century, then the wheel has the same quick zoom function.

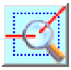
Zoom in **-** magnifies the plotted area. Defining this Cartesian zone must be done by keeping the mouse button pressed (click-drag-release). If you click and release the mouse button at the same location, then the zoom is made by a factor of 2 around the clicked point.

Global view **-** cancels the current zoom and returns to the view of the whole board.

Temporary zoom **-** magnifies the plotted area as *"Zoom in"* above, but it is cancelled by the next click of the left mouse button (not right button). This helps make visual inspections here and there.



Magnify selection **-** magnifies the view so that the selected object fills the screen with a small margin.



Magnify ends **-** magnifies the ends of the selected object, so you can check whether it is opened or closed.

Undo previous zoom **-** cancels the last zoom operation, whatever the direction was. There are four levels of cancellation, and these have no link with the *"Edit / Undo"* function.

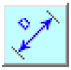
Redo previous zoom **-** redoes the last zoom operation cancelled by the previous command.

Zoom out ×2 **-** reduces the magnification by a factor of 2 (visible objects appear smaller).

Move view **-** slides the window when the mouse is moved with the left button held down. This function has no interest if you have a mouse with three buttons (or two buttons plus wheel): **click & drag with the central mouse button (or the wheel) moves the view** without ending the current drawing function.

❑ Visual dimensions

These icons have no link with machining functions, but represent a little bonus for technical drawing. You can use them to add numerical indications on the board, and they can be printed. **Visual dimensions are dynamic**, *i.e.* they are updated automatically while changes are made on drawn objects. Their style can be changed using the *"Parameters / Visual dimensions"* dialogue box. Other commands are also available through the *"Display / Visual dimensions"* menu.



Multi-purpose **-** is a single, multiple use icon for setting most visual dimensions. The scenarios are various:

- As a preamble, you can consider a clickable point the centre of a rectangle or a circle, even if the path does not go through it.

- If you plot a **point** with the **right mouse button**, then you get a single coordinate dimension. If you have clicked very close to the point, it will be an XY arrow dimension according to the table modulo 45°. If you have clicked a bit on the left hand or right hand side of the point, then it will be a Y dimension alone. If you have clicked a bit above or below the point, then it will be an X dimension alone. Both can be positioned later.

- If you plot **two points** (or vertices), then you get a distance dimension. Depending on the location where you will then plot the dimension, that will be either a horizontal, or vertical, or oblique distance.

- If you plot **a point and a segment** (whatever the order is), then you get an oblique distance dimension, perpendicular to the segment.

- If you plot **two parallel segments**, then you get a distance dimension between these segments.

- If you plot **two non-parallel segments**, then you get an angle dimension between these segments.

- If you plot **a segment** with the **right mouse button**, then you get a slope angle dimension for that segment.

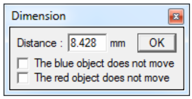
- If you plot a **closed circle**, then you get a horizontal diameter dimension, between vertical reminder lines.

- If you plot a **closed circle** with the **right mouse button**, then you get directly an arrow dimension, for a diameter if you have clicked outside the circle, or for a radius if you have clicked inside.

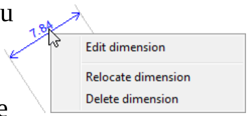
- If you plot an **open arc of circle**, then you get an arrow dimension, positioned at the nearest angle modulo 45°. If, when clicking, the mouse

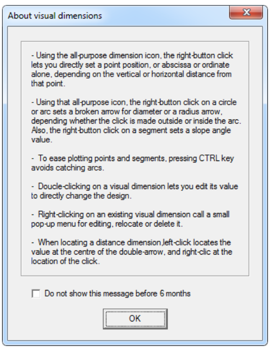
cursor is outside the arc, then it will be an external diameter dimension, otherwise it will be an internal radius dimension.
- If you plot an **ellipse**, then you get an arrow dimension for radii, to be located according to a table modulo 45°.
- When you try to plot an item, **pressing** Ctrl **key simultaneously** avoids catching a circle or an arc.
- When fixing the position of a distance dimension, the **left mouse button** sets the numerical dimension at the centre of the double-arrow, and the **right mouse button** sets it at the clicked position, without going outside the arrow head (if you click outside the reminder lines, the position of the numerical dimension will be on the border but inside).

Very important: **visual dimensions are reversible**, *i.e.* you may double-click on a dimension for editing its value, which will act on the drawing. For example, for a distance dimension, you can change the value for adjusting the items that have been dimensioned, eventually indicating if one of them must remain fixed at its current position.



Equally important: the **right mouse click** on an existing visual dimension pops up a small context menu which lets you edit the dimension (see above) or relocate it, or even simply delete it. Of course, you must catch the dimension successfully, without letting Galaad consider that you want to select a point or a segment or edit the geometrical properties of the neighbouring object.




About visual dimensions

- Using the all-purpose dimension icon, the right-button click lets you directly set a point position, or abscissa or ordinate alone, depending on the vertical or horizontal distance from that point.

- Using that all-purpose icon, the right-button click on a circle or arc sets a broken arrow for diameter or a radius arrow, depending whether the click is made outside or inside the arc. Also, the right-button click on a segment sets a slope angle value.

- To ease plotting points and segments, pressing CTRL key avoids catching arcs.

- Doucle-clicking on a visual dimension lets you edit its value to directly change the design.

- Right-clicking on an existing visual dimension call a small pop-up menu for editing, relocate or delete it.

- When locating a distance dimension,left-click locates the value at the centre of the double-arrow, and right-clic at the location of the click.

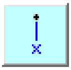
Do not show this message before 6 months

OK

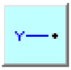
Note: at the very first click of the day on any visual dimension icon, a **reminder message** pops up for refreshing your knowledge of all the above mentioned. You will have to wait until the day after to see again that message, unless you press simultaneously Ctrl + ⇧ keys while clicking on the icon, in which case it kindly appears.

Now we can see the other icons of that series.

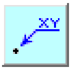
Concerning the next three icons, it is possible to get the same result from the multi-purpose icon, already seen, and the right mouse click when plotting (see all details above).



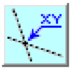
Abscissa - adds a vertical axis with a numerical indication of the corresponding coordinate.



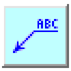
Ordinate - adds a horizontal axis with a numerical indication of the corresponding coordinate.



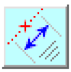
XY position - displays a couple of numerical values over a position arrow, associated to a vertex or point you have to select.



XY intersection - adds an XY position couple at the intersection point of two segments you have to select, which is not possible with the multi-purpose icon. The operation can be done from one single plotting on the intersection, which highlights the two segments concerned. If, during later manipulations, it appears that the segments do not cross each other anymore, then the dimension is suppressed without mercy.

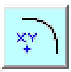


Custom indication - adds a mark with a free comment somewhere on the board, with no links to existing objects. Even if all objects are deleted, this indication remains at its location. The text cannot exceed 80 characters. The direction and the length of the arrow are settable, and can be modified later on.

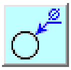


Change location - allows you to move a visual dimension that has been previously positioned on the board, without changing what it refers to. If the visual dimension is a broken arrow, then the corresponding dialogue box is recalled. Otherwise the positioning function is restarted. You can get the same thing with a right mouse click on the visual dimension, through the small context menu.

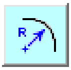
Concerning the next five icons, it is again possible to get the same result from the multi-purpose icon with or without the right mouse click.



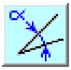
Arc centre - displays the position of the centre of an arc of circle or ellipse, to be selected.



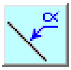
Diameter **-** displays an external arrow dimension that points to the circumference of an arc.



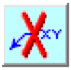
Circle radius **-** displays a dimension arrow between the centre and the circumference of an arc.



Angle between segments **-** displays an angular dimension between two segments to be selected, the arrows being outside, which makes it different from the multi-purpose icon.



Segment angle **-** adds an arrow dimension to a segment, indicating its slope angle with a horizontal axis. The value keeps the indicated orientation even if the angle has changed.



Delete **-** erases visual dimensions to be pointed, more easily accessible from the right mouse click and the small context menu. See also the *"Display / Visual dimensions / Delete all"* command.

Like everything else, trial and error is the best way way to learn how Galaad’s visual dimensions work, and particularly the multi-purpose icon, with left and right mouse clicks.

11

0 1 0 1 1

MENU FUNCTIONS

Here follows a complete list of all the commands found in Galaad menus, together with a brief summary of their function. They are too numerous to describe in full here. Anyhow the best way to understand their actions is to actually use them, crossing your fingers.

In the following descriptions, when a function is also accessible from a shortcut icon in the top bar just below the menus, that icon is show, or even two icons if the one of interest is dropped down from a parent icon. All icons in this top bar are also described together at the end of the chapter.

"File" menu

File / New - initialises a new board and lets you override the default values for the material and machining parameters (material type and size, choice of tool, cut depth, feedrate). You may set the default values from *"Parameters / New file defaults"*. Simultaneously pressing the Ctrl and ⇧ keys when the command is called on (or its corresponding icon) opens a new board with the same dimensions and milling values as the current one, in this case the dialogue boxes do not appear.

File / Open - loads from the disk an existing drawing under Galaad format (**.GAL**). The default directory in the file selection dialogue box is the last one accessed for opening or saving.

 / 

File / Gallery - displays all files in a given folder and allows you to select one by double-clicking.

File / Save - saves the current drawing to the disk. If it has not yet been named, this function is the same as *"Save as"*, detailed below.

 / 

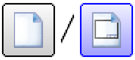
File / Save as - saves the current drawing to the disk, prompting you to give it a name.

File / Close closes the open file and returns to a blank drawing board once you have confirmed if you want to save.

File / Close all closes all open files, through saving confirmation for each. This function only appears if you have enabled the possibility of having several files open at the same time in the work environment settings (see the corresponding chapter on previous pages).

File / New folder - creates a new folder in which to save drawings. It will be located within the ***"File"*** folder of the installation directory.

File / Merge with - loads an existing file and adds what it contains to the current work, with possible offset of its XY coordinates.



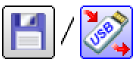
File / Templates / Open template - loads a template drawing to be completed, from the ***"Templates"*** folder.

File / Templates / Template gallery - displays all files in a given template folder.

File / Templates / Save as template - saves the current drawing as a new template in a folder.

File / Network / Open a public file - loads a drawing from the appropriate workgroup (if one has been defined) located in the ***"Public"*** folder on the master computer. Please refer to the chapter dedicated to network functions for more details.

File / Network / Save as public file - saves the current drawing in the appropriate workgroup (if one has been defined) located in the ***"Public"*** folder on the master computer.



File / Quick transfer / Send drawing - performs an immediate save of the current drawing to a removable disk or a mapped network drive, using a default filename (not visible), so that it may be quickly transferred to another workstation, using the corresponding *"Receive drawing"* function of the same submenu. One single file can be transmitted at once since the next sending overwrites the previous file on the transfer disk. The drive and directory are to be set every time (defaults remain the last used).

File / Quick transfer / Send tool library - performs an immediate save of the tool library to a removable disk or a mapped network drive, like the above function for the drawing file.

 /  **File / Quick transfer / Receive drawing -** replaces the current drawing with one previously sent from another workstation to a removable or network disk. If the current drawing has been changed, you are prompted to save it. File name is not transmitted since it may refer to a directory tree that does not correspond on the local recipient.

File / Quick transfer / Receive tool library - replaces the current tool library with one previously sent from another workstation to a removable or network disk, like the above function for the drawing file.

 /  **File / Material dimensions -** displays a dialogue box allowing the dimensions of the raw material to be entered. If you have already drawn objects, you cannot specify dimensions that leave any of them outside the board. You may also rescale the whole drawing, including from an adjustment between two points whose distance is known.

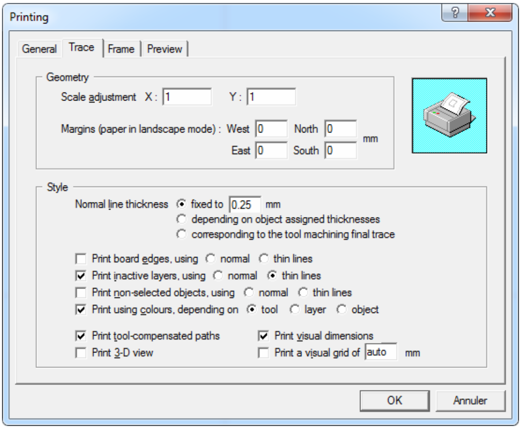
File / Comment - allows you to add a comment to the drawing in the top left of the screen. This remains visible irrespective of the magnification unless it is turned off with *"Display / Comment"*. It is not machined.

File / Warnings - prompts you to indicate free comments that will be displayed when the operator, who might be you, opens the file, or sets the workpiece origin, or launches the machining cycle.

File / Elapsed time - shows the time spent working on the drawing (the elapsed time is displayed at the bottom left corner of the screen) and lets you pause or reset the timer. Every new active drawing operation restarts the timer for three minutes (this duration is settable in *"Parameters / Workspace / General settings"*).

 **File / Print -** defines the print parameters for the drawing, and launches the printer selection process. If the printout scale is left undefined (*"Scale"* edit zone not ticked, or scaling factor set to *"auto"* or empty), then Galaad will try to fill up the whole available space on the paper. All other printing parameters remain memorised, including general identification data.

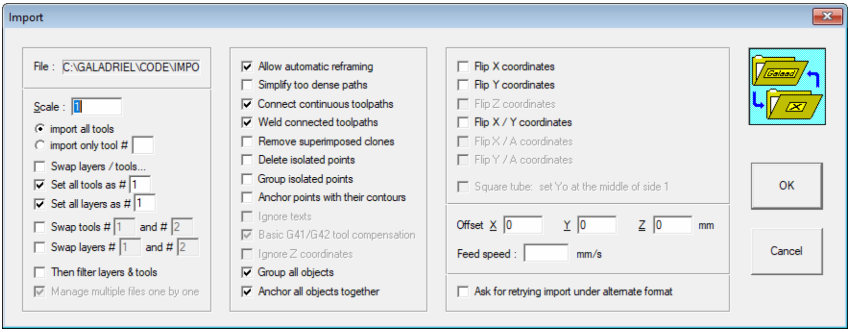
The accurate set-up of the printing is done from *"Trace"* page. Here you can set the X and Y scales that correspond to your printer, to get accurate dimensions on the paper. Many other parameters let you define the pen style for each object, or print the 3D view. A custom page frame can be defined from the *"Frame"* tab of the printout dialogue box, this frame will be located at the bottom right corner of the page and will display the identification texts that will be written at corresponding positions (title, date, reference, *etc.*) using the style selected for each. The custom frame must be drawn separately and saved under Windows *Enhanced Meta-File* (EMF) vector format or bitmap image (BMP). Therefore you can add your logo or any invariable information to this frame drawing. Galaad will stretch or compress the custom frame so that it matches the predefined frame dimensions. XY positions of the identification texts are relative to the top left corner of the custom frame. Even though there is a preview available, you will probably waste a bit of paper before it looks perfect. Using a PDF virtual printer is probably the simplest.



File / Import - loads a **vector drawing** saved by another CAD software package as a standard 2D, 2½D or 3D exchange format. This drawing can include points, lines, arcs, curves, *etc.* but **neither images nor 3D surfaces** are loadable.

After selecting the file, a dialogue box lets you set a scaling factor for the import, which will remain memorised for every format used. You may import the entire file or filter one particular tool, the concept of tool being substituted for the pen colours for classical drawing formats. You may also ignore the colours/tools and assign all objects a given tool (the current tool if none is specified), and also for the drawing layers (the current layer if you enable the option but specify no layer).

The coordinate system of an imported file can be different from the one Galaad uses, which does not allows negative coordinates, so you can activate an automatic reframing which will relocate the drawing in the board, but then absolute positions will be lost and adjustments of successive imports must be handled manually. You may also offset the input coordinates directly. The isolated segments or shapes can be connected for obtaining continuous paths. You may eliminate overlapping clones that are not visible on the screen but cause useless redundancies when milling, or group together isolated points, or anchor them together with paths that are contouring them. You can ignore texts, for example in a DXF file which contains many visual dimensions. Finally, for ISO G-code files, you may enable a basic tool compensation (simple tool offset) without the advanced collision management. Some highly specialised formats use other sets of parameters.



Importing a file adds to the existing drawing without replacing what already exists on the board, and all objects of the newly imported drawing are automatically selected, even if they spread over several layers. Galaad offers a quite large number of import/export formats, which should ease transfer operations in both directions. If you do not succeed in loading a file under a given format, then the best solution is to try another. Available formats are as follows:

- **HPGL** is a format dedicated to **HP plotters** and has become a little standard for 2D vector basics. Galaad is compatible with 2½D extensions that encode Z depths and drilling speeds, mainly used by GravoGraph and Roland tables.
- **PS HPGL** is derivated from HPGL, integrating Z coordinates for 3D paths. It is used mainly by Roland engraving machines.
- **Text-DXF** is the **AutoCad** data exchange format, almost a standard for CAD software. This format can contain surface information that makes no sense for Galaad, which will therefore load only the vector paths (lines, arcs, curves, *etc.*). Please note that successive versions of DXF may have quite important differences. In certain cases, Galaad may read more easily one version of DXF than another. The best bet is to try and hope.
- **Converted-DXF** is a variant of the above, to be used when Text-DXF fails. Its analyser performs two passes so it is slower for big files.

- DWG is the **AutoCad** software native format and can be imported directly by Galaad. Well, in most cases…

- IGES is a somewhat older format which manages drawings projected onto different planes. If the file is a cylindrical projection, then Galaad tries to pick the flat development, machinable as is with 4 axes.

- EPS is the **Adobe Encapsulated PostScript** format for 2D vectors, based on straight lines and Bezier curves. It is dedicated to printers and other graphical systems.

- AI is the **Adobe Illustrator** format, extension of the EPS format above mentioned. An AI file may also contain bitmap images and font definitions, which Galaad will not read.

- WMF is the old 16-bit *Windows Meta-File* format and should not be used much now. Its encoding possibilities are limited.

- EMF is the Windows *Enhanced Meta-File* format for 2D vectors, including colours and line thicknesses, with a relative accuracy. It is more focused on graphics than real technique. Galaad does not import images that are encapsulated in EMF files.

- NCP is an **Isel-Automation** proprietary format for 3D vectors, dedicated to *Isel-Remote* drivers and software applications. This format is specific for Isel CNC machines.

- ISO G-Code is the standard format for files that are dedicated to CNC machine tools. In the drawing module, Galaad will load only 2½D or 3D files with **XYZ** or **XAZ** coordinates.

- G-Code / NUM is a variant of the above that includes arcs with absolute IJ coordinates (centre of arcs). If your ISO import displays fanciful arcs, then try the other available version of the format among those two.

- G-Code for turning is another variant for 2 axis lathes that uses XZ coordinates instead of YZ, but for importing in an XY plane.

- NCI is the 2½D and 3D data exchange format for **MasterCam** software, dedicated to CNC drivers.

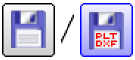
- ACL is the vector 3D format for **MecSoft VisualMill**, dedicated to post-processors.

- CF2 is a 2D vector format for cutting packaging cardboard.

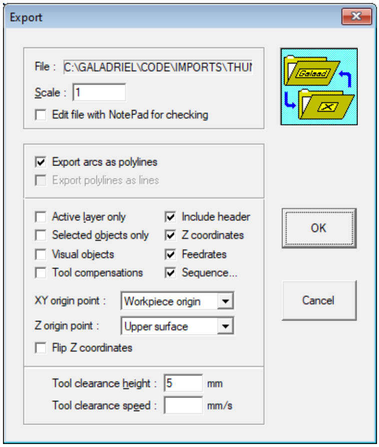
- ESSI, **B3**, **MDA** and **DSTV-NC1** are 2D formats for cutting metal sheets with plasma torches or high-pressure water jets.

- EXL and **DRL** are the formats of **Excellon-Automation** machines that are dedicated to drilling operations on printed circuits. This format contains only XY coordinates for drill points, with tool numbers. Two variants exist, depending on the data format, with or without trailing zeros to the right. If your import displays a fanciful result, then try the other available version.

- **GBR** is the format of **Gerber Scientific Instruments** machines that are dedicated to flashing operations of printed circuits. In the drawing module, Galaad reads only XY coordinates regardless of the shapes and sizes of the plotter aperture. See the Percival module for printed circuit works. Percival loads all Gerber file data and allows you to engrave contours.
- **UIUC** is the format of **Airfoil Coordinate Database** that is dedicated to wing profiles for aeroplanes (DAT files), as defined by the Aerospace Engineering Dept. of the University of Illinois at Urbana-Champaign.
- **OMA** is the 2D polar format of **Essilor** machines that are dedicated to cutting operations on lenses. Two variants exist, with 400 or 800 points, or in complete polar coordinates OMA-3.
- **CLC** is the 2D format for cutting solar masks for glasses, developed for **Opti-3** / ColorClip.
- **LGD** is the 2D format for drilling frameless spectacle lens, developed for **Opti-3** / LogicDrill.
- **SSL** is a format for cutting 2D slices constructing a 3D workpiece assembled layer by layer.
- **DIS** is a 3D Cartesian wire mesh format that is used by the French **Institut Géographique National** for topographic coordinates between two referenced altitudes.
- **DEM** is the 3D Cartesian wire mesh format for geographic or topographic coordinates that are constructed from satellite pictures through the ISTAR stereoscopic system.
- **MNT** is a 3D format for topographic applications containing clouds of points with XYZ coordinates in Lambert projection. Galaad is able to build up a non-rectangular wiremesh from these points.
- **TXT** is a filter for importing coordinate files in decimal text format, each line corresponding to an isolated point or a polyline vertex. The coordinate separators and the line separators are fully settable.
- **BUG** is an appendix for re-importing into Galaad a debugging file that has just been extracted after a problem when machining, for checking its coordinates graphically, including tool-up movements. The workpiece dimensions correspond to the axis range, and the drawing may be inverted depending on the position of the machine zero point. Before prejudging that Galaad has made an ugly offset error while driving your machine during a milling cycle, preferably on an expensive workpiece, you can check this way that coordinates are coherent and perhaps it is rather the machine that drifted by itself.



File / Export - saves the current drawing under a standard vector format, for transmission to another CAD software. Please note that only the drawn objects are exported: the visual dimensions and the background image are not exportable.



After choosing the target folder and the file name, the export function gives the possibility to set up a few parameters. If changed, the scaling factor will remain memorised for the selected format, separately from the import. The arcs of circles in the drawing can be generated as arcs or as simple polylines (vectors). The sequence of the tools that are used for the output can be set from a sub-window. Other options are quite self-explanatory and do not require a long speech. For milling-oriented formats, it is possible to set the tool clearance height and the cleared tool speed.

Most import formats are also available for export, except a few specialised geographical formats. It would be useless to list again all these formats. Please refer to what has already been exposed above for the import function. In addition to these, Galaad offers the following export formats:
- **KYN** is the text format of programming files in the **Kynon** module, which allows you to create a programmed path from the Galaad drawing module. Only motion commands will be exported.
- **CM3** is the vector format for **Roland DG** CAMM-3 or MDX engraving machines, which accepts 3D interpolations.
- **EGX** is the vector format for **Roland DG** Modela EGX, which accepts 3D interpolations.
- **SML** is the data format for **Suregrave** machines.
- **CBR** is the data format for **Colinbus** machines.
- **XYA** (G-Code for cylindrical revolution) allows you to produce a revolving wiremesh with a transmutation of a red-selected profile to a blue-selected profile. The axis of revolution is the bottom of the drawing board (coordinate Y=0). It is possible to make several revolutions, with a progressive X offset, for example to produce a screw.
- **STL** can export surface facettes of a Cartesian 3D wiremesh that has been drawn using the wiremesh functions of Galaad.

- **C** is a text format that uses the MoveTo(…) / LineTo(…) instructions of a programming language based on the same syntax as the **C language**, which is rather easy to transcode into another programming language. Note that there is nothing better than C except C++ (sorry, private joke for developers).
- **Post-Processor** is the custom format that you can define from the big dialogue box available from *"Parameters / Post-processor"*. Almost all syntaxes and coordinates systems are allowed, provided that they are based on decimal or hexadecimal text.
- **BMP** is the Windows standard **bitmap image** format, for a non-vector output of the drawing.

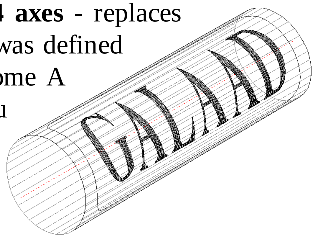
File / Batch conversion - helps you convert a set of files located in a directory or a whole tree, including Galaad files, into another format. This avoids a big conversion work of a whole library of old files.

File / Exit Galaad - saves the current working environment, including the current drawing, and exits the software. Please refer to the chapter on the working environment in the previous pages, if you prefer a classic mode of closing the current file when you exit Galaad, with a blank design board at the next launch.

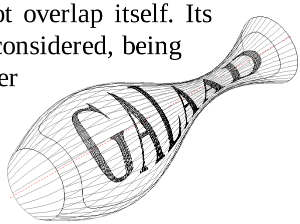
❑ *"Machining"* menu

Machining / Standard machining on 3 axes - gives access to the automatic milling process for the current drawing (in fact on 2 axes XY or 3 axes XYZ depending on the machine type), using either the internal machining module, namely Lancelot program, or the predefined external driver. No need to revisit that function since it has been widely described in previous chapters. In case of doubt, then please go back to the chapters *"Learning to mill"* and *"Advanced milling functions"*.

Machining / Cylindrical machining on 4 axes - replaces the above command if the raw workpiece was defined as cylindrical. Y coordinates in the drawing become A coordinates, the board becoming a cylinder. If you wrap this cylinder on 360°, then the board is completely wrapped and the coordinates at the top match the coordinates at the bottom. The



diameter that is suggested makes the board Y height coincide with the resulting circumference. It is possible to use a blue selected object for defining a non-cylindrical overall profile (Z depth then vary depending on this object). That profile runs from left to right and must not overlap itself. Its position has no importance, only its variation is considered, being fully understood that the indicated diameter corresponds to the highest point of the blue object. The function *"Display / 3D view"* helps you visualise the projection of the drawing onto the cylinder or the profile of revolution.



 /  **Machining / Multi-sided milling on 4 axes -** gives access to the milling process on 4 axes of the current drawing, splitted in several normal cycles on 3 axes XYZ, each cycle being driven from different positions of the A axis. The rotary axis turns on an angular value to be defined before the process is resumed. The cycles can be identical (repeated machining on all sides), or correspond to different drawing layers, each layer corresponding to a single side. Since the workpiece can have a rectangular section, it is possible to define offsets for Y and Z axes for each side, so the origin point can be located at a corner and obviously Z coordinates will compensate the difference when the top surface of a side is higher than the one on which the workpiece origin has been made.

 /  **Machining / Simulate -** simulates the normal milling process for the current drawing, using the internal machining module. Process steps are identical to the real milling cycle, except that there are no dialogues with the machine nor a chaining to an eventual external driver.

Machining / Upload - starts the normal milling process for the current drawing, but stores all commands in the local memory of the machine. This is the same as launching the cycle after having enabled *"Upload machining to the controller memory"* in the options of the milling module. The process is the same, but the start is automatic (tool retraction height and current workpiece origin are preset here). Obviously the functions for storing the cycle in the machine memory must be available, and this memory must be big enough. If it is not the case, then this function does not appear in the menu.

Machining / Milling zone - allows you to define a rectangular area for milling in your drawing board. The workpiece will be confined to every object inside this zone. This function can help make successive shifted machinings on a workpiece which does not fit in the machine space. A zone is machined, then the workpiece is shifted by the width of that zone and the machining is repeated, *etc*. Obviously, the physical offset of the workpiece must be supervised with the best accuracy and, whenever possible, it is better not to split drawing paths that will be marked at the joint if they are dispatched in two different cycles.

Machining / Manual control - displays a control panel allowing the machine to be controlled manually, thus with no relation to the drawing. In addition to the ability to jog axes, the manual control can be used as a test to check the communication with the machine. See also the *"Parameters / Machine / I-O test"* function in this case.

Machining / Manual unblock - does the same as above, except that the control does not reset the axes on the machine zero point and consequently uses floating coordinates, so with the risk of bumping the stoppers at the axis ends. The purpose is to make the machine move freely, for example if a tool remained stuck in a workpiece, or for checking manually the axis end switches (here they can be approached one by one until they are triggered and the screen displays which input has changed).

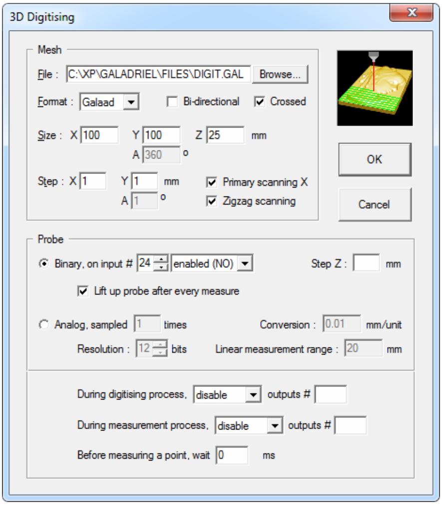
Machining / Surface / Table - allows the top surface of the machine's removable bed to be cleaned up to provide a good flat working face. The default surface range corresponds to the active length of XY axes but you may change it. The angle cross avoids leaving the tool fillet in the corner that will be used as the XY workpiece origin, the lateral borders becoming stopper bars for the workpieces. When setting the origin before surfacing, **it is necessary to do the flatbed approach manually** and validate with *"Z - ok"* button.

Important: After surfacing the flatbed, the Z calibration of the fixed tool sensors is invalidated since the reference has changed. If indeed you are using a fixed tool sensor, then you must calibrate again its phase 2 (Z offset).

Machining / Surface / Workpiece - allows the top surface of the current workpiece to be cleaned up to provide a good flat face. The Z origin must be done on the raw surface.

Machining / Surface / Thickness - rectifies the workpiece so it matches the parametered thickness. The material layer to be removed must be given by its excess. It will help manage the stages and the tool clearance height.

Machining / Digitise / Surface - calls the 3D digitising process on 3 axes XYZ for a Cartesian volume. This allows a reverse engineering of an existing workpiece to be copied. The function in Galaad creates a rectangular wiremesh from automatic probing measurements of points on the piece. The machining module is independent from the drawing module. It will create a new file for that wiremesh, under a format, a density and parameters to be defined. The characteristics of the probe that are defined here remain memorised. Two types of probes can be used:



- A binary sensor, generally a switch connected to an input. For every point to be measured, Galaad lowers the Z axis down to the contact with the piece, then quickly lifts up and moves the X or Y axis towards the next point.

- An analogue sensor, generally a measuring laser connected to the ADC input of the controller. It is necessary to enter the conversion factor between the analogue value and the distance unit. Galaad will just position the sensor above every point to be measured, moving the Z axis only for keeping the sensor inside its linear range. It is not possible to grab fly-out samples along the continuous X or Y movement.

Once the parameters have been validated, the process is the same as a workpiece origin for a standard machining, except that the piece is already machined. You must set the origin point for the volume to be digitised, using the green buttons, and the sensor stem must touch the flatbed instead of the tool (with an analogue sensor, just lower it to its linear range). When the origin is set, then the process starts like a machining cycle.

Machining / Digitise / Cylinder - calls up the 3D digitisation process on 4 axes for a cylindrical volume. This process is identical to a surface digitising process, except that the A axis replaces the Y axis, so with a workpiece origin that is the same as a 4-axis machining. But the wiremesh created this way will be unwrapped into a flat XYZ.

Machining / Digitise / Manual - calls the digitisation process requiring you to manually control the machine and confirm every point. The major interest of this function is to physically pick up one by one the coordinates of a shape that has been placed on the machine. The three available entities are the isolated point, the polyline by its vertices, and the 3-point arc. The coordinates are absolute, the origin being the machine zero, so you must reframe the result once the digitising is completed. Newly digitised entities are appended to the file, so that it is possible to resume a digitising process.

Machining / Obstacles / On the workpiece - is used to define obstacles to be avoided during machining. The idea is simple: you draw rectangles or closed circles, select them and call up this function to indicate the clearance height. These rectangles and circles become obstacles. They will not be machined, of course, and will be saved with the file. You can create as many as you like and overlap them to create a complex shape.

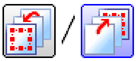
Machining / Obstacles / On the machine - is used to define obstacles to be avoided in the machine workspace. Unlike obstacles on the workpiece, these are linked to the machine and are not saved with the current design. You need to create a drawing whose dimensions correspond to the axis range, and draw the obstacles as above. The depth of each will actually give the avoidance height. These obstacles can be globally enabled or disabled in the machine parameters, *"Advanced"* tab.

Machining / Obstacles / Open and **Save -** helps you create obstacle files on the machine, which can be recalled individually.

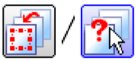
Machining / Tool library - opens a window where you can choose the tool rack if several have been defined, shows the key parameters of the tools currently defined within that tool rack, and indicates which are used in the current drawing. Access is also available to set the full parameters. The *"List"* button generates a text output of the list into a file named *"GALAAD.TXT"* that can be printed from Windows Notepad, which is opened automatically.

Machining / Recent tools - browses a list of the tools used in the current drawing plus the last five.

Machining / Sequence / Set as first - moves the selected object to the first position in the machining sequence (tool cycle and main passes). If several objects are selected, of course the internal sequence of the group is unchanged. This remains valid for all other sequence commands that are listed below. Always keep in mind that the drawing sequence set here is not absolute: by default, the machining process will follow first the tool sequence and, for each tool, the cut-out paths will be made last (when object depths are greater or equal to the workpiece thickness). It is only inside these main passes that the sequence may be changed. You can display the position of each object in the sequence with *"Display / Trace / Identifiers / Sequence"*.

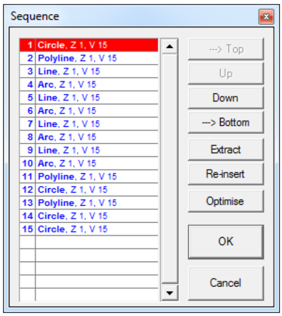


Machining / Sequence / Set as last - moves the selected object to the last position in the machining sequence.



Machining / Sequence / Mouse select successively - allows you to set the machining sequence by pointing objects one by one from the first to the last. Sequence numbers are displayed during the operation, near the start point of each object.

Machining / Sequence / Table - pops up a table of all objects present in the drawing, by order of appearance. These objects are displayed in the same colour as the lines, *i.e.* red for the objects selected in the table (not related to the eventual selections in the drawing), in black for those that precede, and in blue for those that follow. So you can select objects in the table and drag them in the sequence or use the corresponding buttons on the right hand side.



Machining / Sequence / Set order of appearance - allows the selected object to be placed in any position in the machining sequence.

Machining / Sequence / Reverse - reverses the order of the selected objects, within the selection only. Objects not selected are not affected.

Machining / Sequence / Optimise - rearranges the machining order of all objects to minimise the inactive movements between them. The drawing is divided into several zones depending on a matrix that you define, and all objects in each zone will be completed before passing on to the next zone, for limiting the cumulated distance by always seeking the nearest. The first object in the sequence is unaffected. Open objects may have their toolpaths reversed to start from the nearest end.

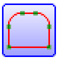
Machining / Sequence / Pessimise - changes the machining order so as to increase the number of inactive movements, which may seem counterproductive but gives the cutting tool time to cool down.

Machining / Sequence / Sort by increasing depths - makes an automatic sequence related to the depths of each object (you can then reverse the order if necessary by using *"Machining / Sequence / Reverse"* above mentioned).

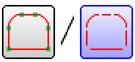
Machining / Sequence / Inner objects before their contours - sets in the order of appearance all objects included inside a path before that path, even if they have not been given a tool compensation.

Machining / Sequence / Order by layers - sets the order of appearance by placing all objects of the layer #1 as first, then all objects of the layer #2, *etc*. The internal sequence of each layer remains unchanged.

Machining / Sequence / Automatic before machining - indicates how the design should be globally sequenced when machining is called up.

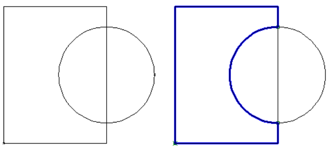


Machining / Toolpath / Connect objects - creates a single toolpath to join independent objects that are adjacent to each other. This single trajectory can be used for contouring or hatching, but unlike objects that are welded together, they can be disconnected and regain their independence and geometric properties (arcs and curves). Please refer to the previous chapter *"Toolpaths"* for more details about connected paths and related features.

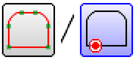


Machining / Toolpath / Disconnect objects - disconnects objects that have been connected using the above command.

Machining / Toolpath / Set path object after object - lets you point one by one the objects that cross one another. These objects will be incised at their intersections for constructing the path. The split parts left beside the path are not deleted. See also "Design / Polyline / Delete parts where clicking" function and its shortcut icon in the top bar.



Machining / Toolpath / Weld connections - welds together all connected objects into a single trajectory. The objects having geometric properties (arcs or curves) lose these properties and become polylines. Only neighbouring arcs that share the same centre and radius, or neighbouring Bezier curves, can be welded together without loss of properties. But Beta-Spline and Quadra-Spline curves cannot be welded unless they become simple polylines.



Machining / Toolpath / Define as start point / The red point, the point to be clicked - allows the end point of an object or connected toolpath to be redefined as the start point. In the case of an open object, of course that point can be only one of its ends.

Machining / Toolpath / Define as start point / The highest point, the lowest point, the point which is more at centre, at north, *etc.* - sets as start point of the selected object or path the one that is the closest to the chosen reference. If the object is open, then the closest end is used. Objects having geometrical properties are not cut.

Machining / Toolpath / Define as start point / The sharpest external vertex - sets as start point of the selected object or path the one that corresponds to the sharpest angle outwards.

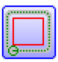
Machining / Toolpath / Define as start point / The sharpest internal vertex - sets as start point of the selected object or path the one that corresponds to the sharpest angle inwards.

Machining / Toolpath / Close path - closes the trajectory by adding a segment or adjusting ends.

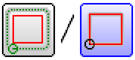
Machining / Toolpath / Set clockwise - sets the machining direction of selected objects to clockwise.

Machining / Toolpath / Set counter-clockwise - sets the machining direction of selected objects to counter-clockwise.

Machining / Toolpath / Reverse direction - reverses the machining direction (clockwise or counter-clockwise) of selected objects.

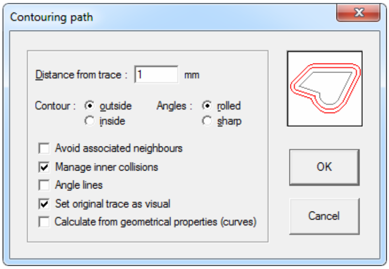


Machining / Tool compensation / Define toolpath - sets an inner or outer contouring trajectory of the selected object or path, depending on the tool diameter. Please refer to the previous chapter *"Toolpaths"*.



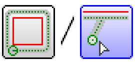
Machining / Tool compensation / Remove toolpath - removes the compensated toolpaths of the selected objects.

Machining / Tool compensation / Create a new offset object - creates a new independent object from the internal or external tool compensation applied to the selected object. If this selected object already has a tool compensation, then its offset contour trajectory becomes the new object without asking more. If not, then you are prompted to define the offset with simplified calculation parameters.

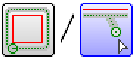


Machining / Tool compensation / Recalculate toolpath - rebuilds the contour trajectory of the selected objects, removing any entry or exit paths that have been added manually.

Machining / Tool compensation / Define as start point - lets you click directly the positions of the start points of closed compensated paths. If the tool compensation has a roughing and a finishing path, both must be clicked separately. The original object trajectory is not modified, only its offset path changes. If the tool compensation path is recalculated later on, then its start point returns to the one of the object. Consequently, this operation must be done last, once the object shape and its tool compensation path have been definitely validated.

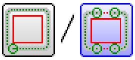


Machining / Tool compensation / Add feed-in segment / … - allows you to control how the cutter approaches a tool compensated path by defining a horizontal new entry segment, that keeps the same depth as the previous entry point, thus with a standard drill-in cycle, or uses an oblique Z slope starting from the top surface. It can also be chained with the exit point, in which case you click twice on the same object the feed-in and feed-out segments. Warning: the validity of the new segment is not checked by the software, especially concerning the fact that it may collide with the compensated path trajectory or other neighbouring objects. It is possible to add successively as many feed-in (or feed-out) segments or arcs as you like. But if the tool compensation is recalculated (object shape modified, tool changed, depth changed if the tool is non-cylindrical), then these add-ons at ends cannot be recalculated and consequently are lost.

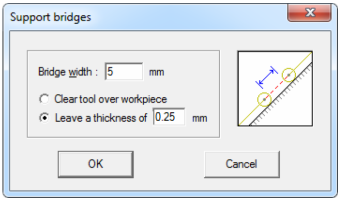


Machining / Tool compensation / Add feed-out segment / … - as above for an exit segment that brings the path away from the compensation trajectory.

Machining / Tool compensation / Add angle lines - lets you choose which corner of the contour path should receive a 3D oblique dash or a flat clearance, and to which vertex of the layout it should be linked.



Machining / Tool compensation / Support bridges / Add - allows you to put, on the tool compensation paths, bridges of remaining material which will prevent the cut part from vibrating when the tool separates it from the rest of the workpiece, or from flying away if the path is closed, which is rather usual. The software asks you to indicate the bridge width, and if the tool must be completely lifted up at the clearance height or just leave a small thickness of material, *i.e.* the cutting depth becomes a bit less along the bridge, of a given value. Once the machining is completed, material bridges can be removed manually for freeing the cut shapes. Please note that these support bridges cannot always be kept when a change in the object shape requires a recalculation of its tool compensation path. So this operation must be done last, once the object shape and its tool compensation path have been definitely validated. The best solution is to set automatic support bridges in the extended window of tool compensation parameters. Then the calculation integrates them.



Machining / Tool compensation / Support bridges / Move - lets you change the position and the parameters of the existing support bridges, to be pointed in the drawing. In fact, the operation consists of deleting the support bridge and immediately creating a new one.

Machining / Tool compensation / Support bridges / Delete - allows you to suppress existing support bridges, to be pointed in the drawing.

Machining / Tool compensation / Support bridges / Change depth - lets you modify the remaining thickness of the support bridges.

Machining / Tool compensation / Support bridges / Delete all - suppresses all support bridges of the selected objects.

Machining / Tool compensation / Support bridges / Check all - reviews all support bridges of all tool compensation paths for ensuring that none has a thickness greater than the machining depth.

Machining / Tool compensation / Internal Sequence / Point the contour to be machined first - lets you change the machining sequence of the multiple offset paths for an object. The tool compensation calculation can sometimes create several pockets or paths if the object shape has narrowings which do not allow the tool to pass through. This function helps you define the internal sequence for these multiple paths, without changing the object sequence in the drawing. The pointing is done directly under the mouse, by clicking the highlighted path. Warning: if the tool compensation has to be recalculated, then this change of the internal sequence is reset.

Machining / Tool compensation / Internal Sequence / Point the contour to be machined last - as above, except that now the tool compensation path that is clicked becomes last.

Machining / Tool compensation / Flip inside/outside - resets the outer tool compensation paths to inner tool compensation paths for the selected objects, and reciprocally.

Machining / Tool compensation / Reverse feed direction - inverts the machining direction of the tool compensation paths of the selected objects.

Machining / Tool compensation / Select all inner compensations - selects or adds to the current selection (⇧ key) all objects having a tool compensated path inside.

Machining / Tool compensation / Select all outer compensations - as above for all objects having a tool compensation path outside.

Machining / Tool compensation / Filter toolpaths - selects objects depending on the settings of their tool compensation paths.

Machining / Tool compensation / Move contents with outer compensations - activates the mode in which all objects contained in the selected object are moved with it, if itself has an external contour.

Machining / Tool compensation / Enable anti-collision repulsion -

activates the mode in which an object with external tool compensation is kept at a distance from the others or from the edge of the workpiece when it is moved. The others must also have an external contour.

Machining / Tool compensation / Check all collisions - reviews all the objects with an external tool compensation to check that their trajectories do not collide with each other.

Machining / Tool compensation / Advanced parameters - allows you to control how compensated toolpaths are produced. The angle display threshold indicates the minimum angle of the trajectory that will show a circle to represent the tool. The rolled angle threshold defines the minimum angle that will trigger an arc around the path point, and the vector stepping. The spacing percentage smoothes the path before calculations, erasing the points that appear to be insignificant

Machining / Depths / Summarise - if there are no objects selected, this displays a summary of all objects and their cut depths, otherwise the list is limited to the selected objects only. See also the rapid data palettes.

Machining / Depths / Change all - allows the cut depth to be set to the same value for a group of selected objects.

Machining / Feedrates / Summarise - if there are no objects selected, this displays a summary of all objects and their feed speeds, otherwise the list is limited to the selected objects only. See also the rapid data palettes.

Machining / Feedrates / Change all - allows the speed to be set to the same value for a group of selected objects.

Machining / Duration - calculates the theoretical time required to machine existing paths, the cleared moves being ignored since they are related to machining parameters (tool plunge, depth stages, clearance height and finishing pass) and not considering the kinematical parameters (accelerations and decelerations) that are necessary for executing the movements.

Machining / Timers - displays the status of the machining timers that can be used to monitor the cumulative time spent on different projects. Each timer can be set and activated, and several counters can run simultaneously.

"*Edit*" menu

Edit / Undo - successively undoes changes made to the current drawing from the *"Machine"*, *"Edit"* and *"Design"* menus. This only applies to the drawing process. File, display and parameter functions are not concerned. The size of the undo stack can be set in the workspace parameters (up to 50 levels).

/ **Edit / Redo -** successively redoes the last undone operations on the current drawing.

/ **Edit / Repeat -** repeats the last operation performed on the current drawing from the *"Machine"*, *"Edit"* and *"Design"* menus, using the same parameters. This only applies to the drawing process.

/ **Edit / Restart -** undoes the last operation performed on the current drawing and restarts the function that has been undone, with new parameters, *i.e.* again using the dialogue box.

Edit / Delete - deletes the object, point or segment currently selected in red, *i.e.* having the focus. Same function as the Del key.

/ **Edit / Cut -** places a copy of the current selection onto the clipboard then removes the selection from the drawing. See also *"Edit / Copy"* and *"Edit / Paste"* below.

Edit / Copy - places a copy of the current selection onto the clipboard. This object can then be pasted onto the board or in another drawing application that accepts vector graphics under EMF format. See also *"Edit / Cut"* above and *"Edit / Paste"* below.

Edit / Paste - places a copy of the objects currently on the clipboard into the current drawing. This is only supported if it was either copied from Galaad or from another software package as a vector graphic. See also *"Edit / Copy"*. You can paste a background bitmap image from the clipboard using *"Display / Background image / Paste"*.

/ Please note that a sub-icon *"Paste the last but one copy"* allows you to paste the last-but-one object that was copied onto the clipboard. This icon has no corresponding line in a menu.

Edit / Paste from centre - places a copy of the object currently on the clipboard into the drawing, positioning its centre instead of its southwest corner. See comments above.

Edit / Reframe and paste - places a copy of the objects currently on the clipboard into the current drawing and allows it to be resized at the same time. See comments above.

Edit / Selection / Select all - selects all objects of the active layer. *Tip:* the Ctrl key selects all across all layers.

Edit / Selection / Deselect all - deselects all the objects. The ⇧ Esc combination does the same.

Edit / Selection / Invert selection - deselects the selected objects and reciprocally selects all objects that were not selected.

Edit / Selection / Select smallest object - is a self-explicit function. Its major interest is to search for and delete residual micro-objects that cannot be distinguished because they are too small or they overlap other objects. As usual, the ⇧ key lets you add to the current selection, since the objects already selected are not eligible for that search.

Edit / Selection / Select cut objects - selects objects whose depth corresponds to the board thickness.

Edit / Selection / Select 3D objects - selects objects whose depth varies along their path.

Edit / Selection / Select superimposed clones - searches and selects the identical paths that overlap each other. It can happen that imported files include objects that have been duplicated over one another. This is not visible on the screen or the printer, but the machining makes these objects twice (or more). The import parameters dialogue box offers an identical option.

Edit / Selection / Delete smallest object - helps you remove unwanted items from the design, particularly after importing a file that may contain objects that are as useless as they are invisible.

Edit / Duplicate / Add one real copy - copies the selected objects and pastes them at an XY position to be pointed. Machining data (tool, depth, feedrate) remain unchanged. The original objects and their copies are independent, unlike the virtual copy (see below).

Edit / Duplicate / Add one virtual copy - copies the selected objects and places a virtual copy of them at a position to be plotted. The copy remains linked to the original. In fact it does not exist in the memory or in the file, which saves space, but it is displayed and machined normally. A virtual copy follows all modifications of the original.

Note: if you make one or more virtual copies of a set of objects, each series of copies will be machined immediately after the original. But if your original objects are associated (see yellow icon of association), then each group of copies will be fully machined before the next copy. Hence this association helps to save time when milling. Of course, you may associate objects after they have been duplicated.

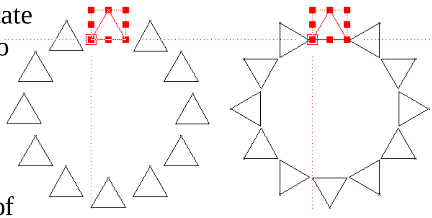
 / 

Edit / Duplicate / In line - duplicates the currently selected objects in a straight line at regular intervals from a position to be plotted. The parameters for a line duplication are the number of copies, a possible progressive Z sinking (depths increase for every new copy), and the ability to make a virtual duplication (ghost copies) or a real duplication (copies are independent from the original). You must plot the position of the nearest or the farthest copy.

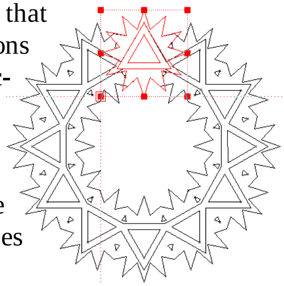
 / 

Edit / Duplicate / In matrix - duplicates the currently selected objects in a rectangular matrix at regular intervals to be defined (number of columns and rows, intervals between columns and lines).

Edit / Duplicate / In circle - duplicates the selected object at regular intervals on an arc to be pointed (centre, radius, starting angle). In addition to the number of copies, you must indicate the angular interval between two neighbour copies and the direction of rotation (clockwise or counter-clockwise), plus an eventual Z sinking. You can make copies rotate themselves around the centre of duplication. But then, the copies being different from the original, they cannot be virtual copies.



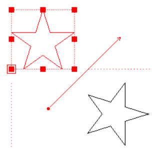
Edit / Duplicate / In kaleidoscope - duplicates the selected object as for a circular duplication with a rotation about the centre, but welding the intersections from one copy to the next and deleting parts that overlap. It is possible to keep either the unions (intersecting polylines are deleted) or the intersections (polylines located off intersections are deleted). This function is for artistic purposes only, for helping you make repetitive shapes such as circular arabesques by deleting automatically the surfaces that overlap each other and welding polylines between them.



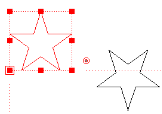
Edit / Duplicate / Mirror / To left, *etc.* - makes one real copy of the selected object, mirrored about a vertical (or horizontal) line passing through the left-most (or right, or above, or below) extremity of the object. Copies are real and therefore independent from the original. See also *"Edit / Clone"* hereafter.

Edit / Duplicate / Mirror / Horizontally (or Vertically), about centre - duplicates the selected object onto the opposite side of the board about a central axis of symmetry that can be either vertical or horizontal. Obviously, the copy is inverted and therefore cannot be virtual.

Edit / Duplicate / Mirror / About red segment - duplicates the selected object so the copy becomes symmetrical relative to the segment selected in red. This mirror copy is obviously inverted compared to the original and therefore cannot be virtual.

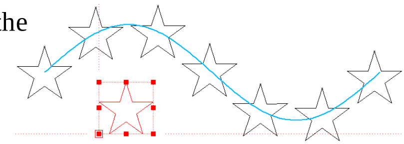


Edit / Duplicate / Mirror / About red point - duplicates the selected object so the copy becomes symmetrical relative to the point selected in red (consequently with a double inversion).



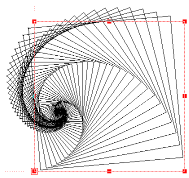
Edit / Duplicate / Mirror / About red cross - as above about red point, except that here the red cross is used as centre of symmetry.

Edit / Duplicate / Along blue trace - places copies of the selected object at regular intervals along the course of a blue trace, and if necessary moving the position of the original onto the trace start point. The Z depth variations of the blue trace can be applied to the copies. It is possible to rotate every new copy according to the tangent of the blue trace at the point of duplication.



Edit / Duplicate / On all blue points - places a virtual copy of the selected object at every point of a group of points selected in blue. Then, for recentring or adjusting the copies on the points, you just have to move the original and obviously the virtual copies will follow. The procedure to be followed for selecting a group of points in blue simply consists of selecting them one by one or in a Cartesian zone, then click on the yellow icon for inverting red and blue selections.

Edit / Duplicate / Special - duplicates the selected object with progressive scaling, rotation, offset and eventual Z sinking features at every new duplication. Since copies are different from the original, they are real and cannot be virtual.



Edit / Duplicate / Trace between points - copies the portion of the trajectory that is located between a red point and a blue point. Both red and blue points can be on different objects if these are connected in a single path (see *"Machining / Toolpath / Connect objects"*) so Galaad can find its way. The copy is located on the trajectory itself but can then be repositioned with the mouse or keyboard.

Edit / Duplicate / At 4 corners of the board / Virtual - makes three copies of the selected object at the other corners of the board, without changing its orientation (virtual copies).

Edit / Duplicate / At 4 corners of the board / Symmetrical - makes three copies of the selected object at the other corners of the board, each of them being reversed horizontally and vertically.

Edit / Duplicate / Suppress copies - deletes all virtual copies of the selected object.

Edit / Duplicate / Suppress one copy - deletes one virtual copy of the selected object by specifying its number (order in which they were created).

Edit / Duplicate / Make copies real - converts all virtual copies of the selected object into independent real objects. There is no reverse function; only the immediate undo is possible.

Edit / Duplicate / Make homogeneous copies - this function is used when two or more objects have virtual copies attached to them and only affects the virtual copies. The selected object with the largest number of virtual copies is taken as the datum. All other selected objects with virtual copies are then modified so that they have the same number of virtual copies with the same relative offsets in X, Y and Z.

Edit / Clone / Mirror to left, *etc.* - makes one virtual copy of the selected object, mirrored about a vertical (or horizontal) line passing through the left-most (or right, or above, or below) extremity of the object. Only one clone of an object can exist at a time. The clone is virtual, and is also modified when the original shape is. But it will be machined like a normal path.

Edit / Clone / Suppress clones - deletes the clones of all selected objects.

Edit / Clone / Make clones real - converts the virtual copy of the selected object into an independent real object.

Edit / Libraries / Open - loads an existing library file from the disk and allows it to be positioned on the drawing, *i.e.* not whole drawing boards but only objects. A library object keeps all machining data (tool, depth, feedrate). The default directory is the last one accessed.

Edit / Libraries / Gallery - displays all objects in a given library and allows you to select one using a double-click.

Edit / Libraries / Open & rotate - chains the opening of an object in a library with its positioning in the drawing (about its symmetry centre) and then automatically calls the rotation function.

Edit / Libraries / Save as - saves the selected objects to the disk, as a library object, allowing you to give it the name of your choice.

Edit / Libraries / New - creates a new folder within the ***"Library"*** folder.

Edit / Libraries / Keep layers - indicates to Galaad that objects retrieved from a library must retain the layer references with which they were registered. By default, they are placed entirely in the current layer.

Edit / Library / TrueType symbols - loads a symbol or a single character from a Windows TrueType font. A big dialogue box allows you to choose the font and browse all the symbols it contains. The size (Y height) is settable, and so is the automatic hatching for filling in the surface of the shape. A double-click on the symbol selects it.

Edit / Unlock / All - unlocks all locked items in the active layer of the drawing. Remember that locked items cannot be selected (as objects, but neither internal segments nor points can be selected).

Edit / Unlock / Mouse select - allows you to unlock the locked object by clicking on them with the mouse.

Edit / Lock / Lock selected objects - does the same as the yellow icon for locking, but having it in a menu allows you to create a keyboard shortcut.

Edit / Ungroup / All - ungroups all items in all groups of the active layer, whether they are selected or not. Bear in mind that the association of objects makes an automatic selection of the whole group when one of its members is individually selected.

Edit / Ungroup / Selected objects - ungroups all items in any groups selected in red. Note that clicking on the yellow association icon, when selected objects are already associated, ungroups them if confirmed, without passing through the menu.

Edit / Ungroup / Mouse select - allows you to extract a single object from a group by clicking on it with the mouse.

Edit / Ungroup / Mouse selection and associates - allows you to break a group by clicking on one object with the mouse. All objects in that group are then ungrouped, even if one only was clicked.

Edit / Ungroup / Select all grouped objects - filters and selects all objects of the active layer that belong to a group. These objects are not ungrouped but just selected.

Edit / Ungroup / Group selected objects together - does the same as the yellow icon for grouping, but having it in a menu allows you to create a keyboard shortcut.

Edit / Unprotect / All - lift the protection from all objects in the active layer, whether they are selected or not. Remember that protected objects can just be moved and resized, but neither deleted nor reshaped in any way.

Edit / Unprotect / Selected objects - lifts the protection from the objects that are selected in red. Note that clicking on the yellow protection icon, when selected objects are already protected, removes their protection if confirmed, without passing through the menu.

Edit / Unprotect / Mouse select - allows you to remove the protection of the objects by clicking on them with the mouse, or select a group of objects by dragging over them with the mouse.

Edit / Unprotect / Select all protected objects - selects all objects in the active layer that are protected.

Edit / Unprotect / Protect selected objects - does the same as the yellow icon for protecting, but having it in a menu allows you to create a keyboard shortcut.

Edit / Free anchors / All - releases all anchored objects in the active layer. This applies to both objects anchored in position and objects anchored together. Bear in mind that absolute anchorage fixes the position of the objects on the board, and relative anchorage moves simultaneously anchored objects when one of them is moved in its own.

Edit / Free anchors / Selected objects only - releases all anchored objects that are selected in red. If anchoring is relative, only the selected objects are released from the group. Note that clicking on the corresponding yellow anchorage icon (relative or absolute), when all selected objects are already anchored in the same mode, releases them if confirmed.

Edit / Free anchors / Selected objects and acolytes - releases all anchors on objects selected in red and those existing between other objects in an anchored group of which they are members.

Edit / Free anchors / Mouse select only - allows you to release a single anchored object by clicking on it with the mouse, or select a group of objects by dragging over them with the mouse. If anchorage is relative, then only clicked objects are released from the group.

Edit / Free anchors / Mouse select and acolytes - releases the anchors on objects selected with the mouse and those existing between other objects in an anchored group of which they are members.

Edit / Free anchors / Objects located in different layers - releases the anchors between objects belonging to different layers. Of course, these objects do not have to be selected.

Edit / Free anchors / Select position-anchored objects - filters and selects all objects anchored in position. These objects remain anchored. The function does not free them.

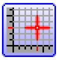
Edit / Free anchors / Select relation-anchored objects - filters and selects all objects anchored together. These objects remain anchored. The function does not break the anchoring links.

Edit / Free anchors / Select acolytes of selected objects - selects all objects that are anchored together with those selected. The function does not break the anchoring links.

Edit / Free anchors / Anchor selected objects on position - does the same as the yellow icon for position anchoring, but the fact that it is also in a menu allows you to create a keyboard shortcut.

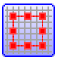
Edit / Free anchors / Anchor selected objects together - does the same as the yellow icon for relative anchoring, but the fact that it is also in a menu allows you to create a keyboard shortcut.

"*Design*" menu

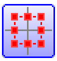


Design / Magnetic grid / Set - sets the step size for the magnetic snap grid. Leaving a value unset or set to *"auto"* defines for X or Y independently a variable step that always matches the smallest graduation of the rulers, whatever the zoom is. The angular step applies a polar grid relative to the previous point when the Ctrl key is pressed during a line drawing or other similar pointings. For more details about the magnetic grid, please refer to the chapter *"Learning to draw"*. No offense meant.

Design / Magnetic grid / Snap to grid - causes the selected object to jump to the nearest snap point of the grid values currently set. This is useful if the grid values have changed or if objects were drawn with the grid inactive.

 / 

Design / Magnetic grid / Snap and size to grid - as above, but the object will be also resized so its borders fit between the nearest snap points.

 / 

Design / Align & centre - pops up a window for selecting among all possible alignments and centring, horizontally, vertically or both. The palette is wide and should match almost all cases. In addition, let us remember that, when you move an object with the mouse, orange lines appear that you can snap for matching neighbouring objects (left hand side, top side, centre, right hand side, bottom side). The left mouse button is already pressed at that moment since you are dragging the selection frame, so snapping orange lines can be done through the Space bar or in two stages through X and Y keys.

Design / Object / Change layer / Layer N - moves the selected objects to 'Layer N'. See the zero point of the rulers (bottom left of screen) for the current layer number. The rapid data palette, when *"Layer"* mode is enabled, does the same without running through the menu, and allows you to jump from one layer to another. See hereafter the functions at the top of the *"Display"* menu to set up the visible layers and the active one.

Design / Object / Duplicate into another layer / Layer N - as above, except that the object is duplicated into the target layer and the original remains in the current layer.

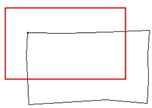
Design / Object / Open - opens a closed object by deleting the last point drawn (and consequently the last segment). If the object is an arc, then it is transformed into a semi-circle (or half-ellipse), *i.e.* 180° from the start point, with the same rotation direction. If it is a curve, then its last sector is deleted, and its geometrical properties are kept.

Design / Object / Close / By adding a point - closes an open object by adding an additional point (and consequently another segment). If the object has geometrical properties (arc or curve), then the object closure is made by adding a sector of the same nature (arc or curve sector), and its geometrical properties are kept.

Design / Object / Close / By moving last point - closes an open object by moving the end point so it coincides with the start point, which does not change. If the object has geometrical properties (curves only), then the adjustment is made by moving the last sector and the geometrical properties are kept.

Design / Object / Close / Through red cross - closes an open object by adding segments from end point to red cross and from red cross to start point. If the object has geometrical properties (arc or curve), this function does not apply. You must first convert the object into a simple polyline.

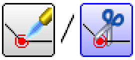
Design / Object / Degrade - distorts the selected object according to framed random values that move the XYZ coordinates. To avoid degrading the depths, just set the Z variation to 0%. The geometrical properties of the object (arc or curve) are obviously lost when degrading.



Design / Object / Link - constructs a new object between the end points of two objects of the same type, one end point having been selected in blue and the other in red. The type of the new object will depend on the existing objects and the depth will, if required, slope to join them.

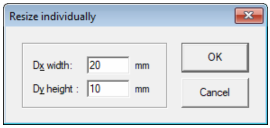
Design / Object / Weld - is an important function of the drawing, which lets you assemble independent paths into one single object. If a point is selected in red, then the function snaps the object that has an end close to that point, and fuses it with the object that holds the red point. The snapped object loses its machining data: the resulting object will take its feedrate from the one containing the red point, which is the master and will retain all of its properties. This function also applies to a set of selected objects, to weld their nearest ends together. In the latter case, you are prompted to indicate a maximum welding distance to avoid snapping objects too far. If the newly created object is almost closed, it can be closed automatically by adjusting its outgoing end.

Note that certain objects cannot be welded together, namely those that have been drawn for use with different tools, protected objects and hatch lines. Welding objects erases the geometrical properties of each object, except if they are of the same nature and can be extended, like Bezier curves. See also the reverse command *"Design / Object / Split"*, just below.

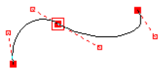


Design / Object / Split - does the opposite of the welding function by separating the object – if not protected – into two independent objects, on either side of the point selected in red. Both resulting parts lose their geometrical properties in the case of a curve. To split a Quadra-Spline or a Bézier curve at a node whilst keeping its properties, you must edit its geometry (right mouse click), click on the node so it is given the focus, then press the / key.

Design / Object / Resize each - helps you to size all the selected objects in a single operation. Here, the requested values will be applied to each object individually and not to the overall selection frame. Each object remains centered on the same position.



Design / Object / Geometry handles - displays the control points used by Beta-Splines, Quadra-Splines or Bezier curves. Selecting them with the right mouse button does the same thing, directly or through a pop-up menu.



Design / Object / Convert to polyline - erases the geometrical properties of the selected object (arc, curve or text). In the case of text, the function *"Text / Convert to polylines"* does the same but keeps the text block associated in a single selection group.

Design / Object / Define as hatching - defines the currently selected object as hatches. Objects created by hatching or pocketing are automatically given this flag. Objects defined as such cannot be welded and can be selected by the filters.

Design / Object / Create framing rectangle - draws a rectangle that matches the borders of the selection frame.

Design / Object / Create a parallel path - does the same as "Machining / Tool compensation / Create a new offset object", with fewer parameters.

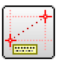
Design / Object / Create cylindrical projection - stretches the selected object so that its projection on the cylindrical workpiece corresponds to the plan view from above, for example for cutting the circle of a welded tube. Then the wrapped machining does not describe a circle but an oval shape, stretched in Y directions (angular direction of the wrapping).

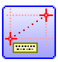
Design / Object / Add a YA cylindrical stretching - stretches the selected object up and down by a value to be specified. This function is only active with a wrapped cylindrical workpiece.

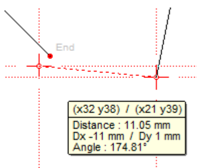
Design / Object / Replace each object with a point / At start point - replaces each selected object by a simple drilling point.

Design / Object / Replace each object with a point / At centre of symmetry - as above, except for the location of the newly created point.

Design / Object / Surface - calculates the surface covered by the closed object (or connected toolpath) that is selected. If several closed objects are selected, the result is the sum of all surfaces, even when they overlap one another. An open object has a null surface. A toolpath in "8" like a Moebius tape will have a surface that is partly positive and partly negative, and consequently a truncated result. Don't ask for too much…

 /  **Design / Object / Trace length -** displays the total length of all selected objects and indicates the number of coordinates they contain. This obviously does not apply to isolated points or groups of points.

 **Design / Object / Distance between points -** displays the distance and angle between two XY coordinates to be plotted on the board, not necessarily on objects. The result is displayed as a movable strip. After the second click, a simple mouse movement will make it disappear so you can loop to the next measurement.



Design / Object / Angle between segments - does the same thing as the previous function for measuring angles on the fly with the mouse.

Design / Object / Adjustment from two points - adjusts (*i.e.* stretches or reduces, plus rotates) the selected objects depending on the distance and angle between two points that can be outside these objects.

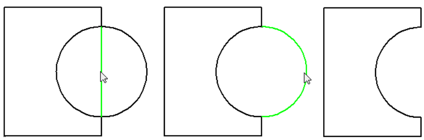
Design / Object / Display ISO G-code - displays a simple G-code programme for the selected objects to assist in checking the coordinates. This is read-only: trying to modify the code is pointless.

Design / Object / Copy tool-depth-speed - does the same as the green icon for copying machining data from the selected object. This function is also in a menu so you can create a quick keyboard shortcut (see *"Parameters / Function keys"*).

Design / Object / Paste tool-depth-speed - related to the previous function, does the same as the green icon for pasting the previously copied machining data into the selected objects.

Design / Object / Visual - does the same as the option *"Visual object (will not be machined)"* in the dialogue box for setting the tool, depth and speed, available from the green icon on the left hand side of the screen. This function is also in a menu so you can create a quick keyboard shortcut (see *"Parameters / Function keys"*).

Design / Polyline / Delete parts where clicking - is a very important function for drawing, which can help you suppress parts of objects very quickly by simply clicking on them. A part is limited by an end or an intersection, which avoids incising and deleting operations. Warning: the remaining parts are not connected automatically. You must then proceed with these connections.



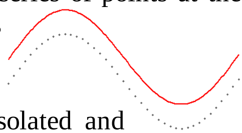
Design / Polyline / Select inner objects - automatically selects all objects that are entirely contained in already selected objects. As usual, the classical  key allows the addition of new objects to the current selection, consequently including the contour. If an object is open, the function acts as if a segment was linking its ends.

Design / Polyline / Select overlapping objects - as above, except that it filters only the objects that cross the selected ones, namely having a part inside and a part outside.

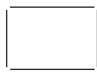
Design / Polyline / Select outer objects - as the two above, except that it filters only the objects that are entirely outside the selected ones.

Design / Polyline / Link vertices - produces a polyline from a group of selected points, by drawing a series of connected lines between the points, in the order in which they were originally drawn. The point coordinates remain unchanged. See also the reverse function, below.

Design / Polyline / Unlink vertices - produces a series of points at the vertices of a selected polyline by removing the segments connecting them. See also the reverse function, above. The major difference with the drawing icon *"Points on trace"* is that the selected object is changed in isolated and grouped points, though the icon adds isolated points without changing anything in the selected object.



Design / Polyline / Explode into segments - changes the selected object in as many independent segments.



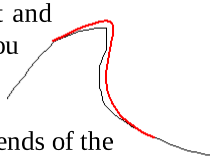
Design / Polyline / Link objects - is the reverse function of the above, since it recombines in one single polyline independent segments or isolated paths, provided that their ends match.

Design / Polyline / Set steps - inserts additional points along the trajectory of the selected polyline, by linear interpolation based on the maximum distance between two consecutive points (or maximum length of a vector), distance to be set by you.

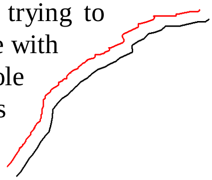
Design / Polyline / Increase steps - inserts additional coordinates along the trajectory of the selected polyline by linear interpolation based on the multiplication factor you have set. The difference with the above function is that each segment of the selected object is concerned, whatever its length.

Design / Polyline / Delete useless points - deletes all intermediate coordinates along the length of a trajectory that have no effect.

Design / Polyline / Substitute to the closest path - replaces a neighbouring path by the one selected. This function aims to redraw a part of a rough polyline without needing to split its ends, replace the part and then weld it back into the original path. For example, you can replace the jerky part of a path with a Bezier curve, drawn as closely as possible. The substituting path is a selected object which does not need to perfectly match the ends of the part to be replaced.



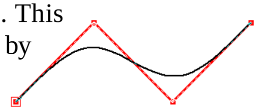
Design / Polyline / Simplify path - reduces the roughness of a jerky path, for example after having vectorised a bitmap image, by trying to find a smoother curve which complies as much as possible with the significant sharp angles. The path can be a whole selected object, or the part of an object between the points selected in red and in blue. Warning: the calculation is heavy and can be time-consuming.



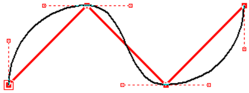
Design / Polyline / Smooth path - produces a new trajectory from the selected polyline by generating a Bezier curve that passes through its vertices, allowing you to specify by how much the curve can deviate from the original polyline, which is kept.

Design / Polyline / Smooth angles - more or less as above, except that vertices correspond to attractors of the Bezier curve and node points are the midpoints of the path segments. It is a smoothing that runs closer to the original polyline. Warning: this original polyline is not kept.

Design / Polyline / Create equivalent Beta-Spline - produces a new trajectory from the selected polyline by generating a Beta-Spline that uses the vertices from the original polyline as its control points. This is exactly as if you were plotting these vertices one by one for building the spline. Let us remember that the maximum number of points in a Beta-Spline is 256, which gives some space for unleashed creativities.

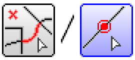


Design / Polyline / Create equivalent Quadra-Spline - produces a new Quadra-Spline from the vertices of the selected polyline.

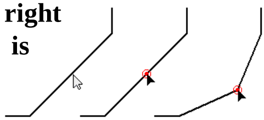


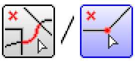
Design / Polyline / Create equivalent Bezier curve - produces a new Bezier curve from the vertices of the selected polyline.

Design / Point / Select all isolated points - selects as objects all points that are present in the active layer. This helps you check that there will be no unexpected drills in the workpiece, since the points are not always clearly visible on the screen, especially when they are covered by a path, a visual dimension or an identifier, or are simply located outside the zoom window.



Design / Point / Insert a point on a trace - interpolates a new point on the segment that you point, and selects it in red. Much faster, **the double-click with the mouse right button on any segment interpolates a point that is immediately selected in red**, so you can handle it at once for repositioning.

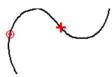




Design / Point / Delete points to be clicked - suppresses the points or the polyline vertices, or vertices in other traces without geometrical properties, that you are going to click.

Design / Point / Delete red point - suppresses the vertex selected as red point. The Del key does the same when the red point has the focus.

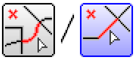
Design / Point / Set a cross mark at given distance - fixes a red or blue reference cross on the path, at a given distance from the point of this path that is selected in red. This distance may be negative for fixing the cross backwards.



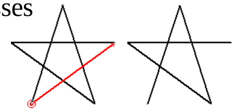
Design / Point / Change points to crosses - substitutes a cross of a size to be defined, for each isolated point selected.

Design / Point / Change points to threads - substitutes a drill-tap to be defined for each isolated point selected.

Design / Point / Change threads to points - performs the opposite operation, substituting an isolated point for each selected tap.

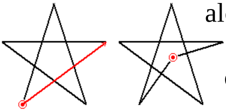


Design / Segment / Delete segments to be clicked - suppresses the segments on which you are going to click, a segment being understood as going to its end or its intersection with another object. The green highlighting shows what will be indeed suppressed.

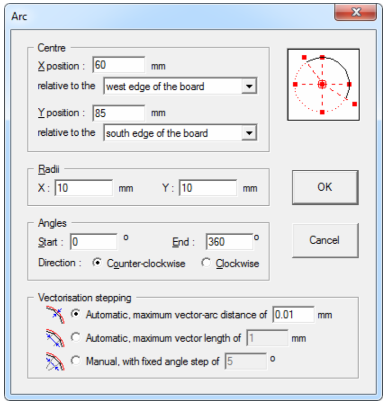


Design / Segment / Delete red segment - suppresses the segment selected in red and consequently splits the object path. The Del key does the same when the red segment has the focus.

Design / Segment / Divide red segment - inserts one additional point along the length of the red selected segment at the position of your choice, according to a given distance that you can set absolute or proportional to the vector (default being half the length). *Tip:* **the right double-click does the same thing**, more directly.



Design / Arc / Edit - opens a dialogue box allowing you to edit all features of the currently selected arc, including centre point, radius, start and end angles, *etc.* which is equivalent to editing its geometrical properties with red graphical handles, then press the Enter key for entering the circular dimensions and the stepping data.



Design / Arc / Select all arcs (open, closed, elliptical) - selects the corresponding arcs in the active layer.

Design / Arc / Select by filtering diameters - selects all arcs from the active layer depending on their diameter, between a minimum and a maximum value to be indicated.

Design / Arc / Set the diameter of all selected arcs - applies to the selected arcs a unique diameter to be indicated.

Design / Arc / Define borings - adds to the selected circles a boring path that includes the tool compensation and the plunge mode, either helical milling or basic vertical drilling. The objects are not modified, so if you need to return to original circles, then you just have to erase the tool compensation paths that this function adds.

Design / Arc / Replace circles by points - deletes the selected circles and replaces them by simple drill points at their centres.

Design / Arc / Replace points by circles - deletes the isolated points that are selected as objects (*i.e.* not a red point) and replaces them by circles being given the indicated diameter.

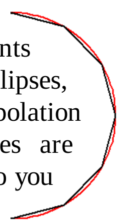
Design / Arc / Replace circles by crosses - deletes all the selected circles and replaces them with simple points at their centres.

Design / Arc / Clockwise - sets the direction of the selected or edited arc as clockwise. Opened or closed, the arc does not change its appearance but only its direction, starting and ending points being swapped.

Design / Arc / Counter-clockwise - as above, except that the new arc direction is counter-clockwise.

Design / Arc / Close - closes the selected or edited arc. The starting point remains unchanged.

Desin / Arc / Complementary - changes the selected or edited arc into its complementary arc, if open.



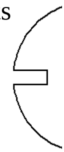
Design / Arc / Circular stepping - sets the vector stepping of arcs and ellipses, namely the angle between two consecutive segments along the arc. When milling, this stepping is reused for arcs of ellipses, or even arcs of circles if your machine does not make circular interpolation (see the machine parameters). Two automatic stepping modes are available, that reduce the angle step depending on the diameter, so you need not worry about the stepping whatever the arc size: the greater the diameter, the smaller the stepping angle becomes, and reciprocally. The result is rounded to the floor value of the following scale: 0.1° (minimum) / 0.2° / 0.5° / 1° / 1.5° / 3° / 5° / 10° / 15° (maximum). The automatic mode with maximum vector-arc distance is probably the most common in CAD applications.

Design / Arc / Set a cross mark - places a red or blue cross on the selected arc at an angle to be entered.

Design / Arc / Reset the arc from the polyline - recovers the geometrical properties that an arc may have lost after being changed into a polyline. The arc must more or less correspond to the trace, the tolerance range being short.

Tip: if you import drawings that contain drilling or boring circles but have not been imported as circles, you can select all and use this function for restoring circles where they are found. Then you can eventually replace these circles by simple points (see above) if you just want to drill to the diameter of the tool. Finally, you can replace the points by circles with a diameter to be specified, if needed.

Design / Slot / Add - allows you to place mechanical mounting slots on cut shapes. The function lets you plot directly on the existing objects: first you locate the notch point somewhere on a path, then you set the angular direction of the groove with a second click. Finally, the software asks you to validate the width, depth and direction angle.

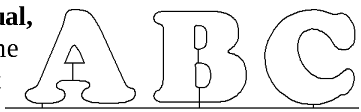


Design / Slot / Delete - suppresses a slot that has been added to a path. This slot must be pointed directly and will be highlighted in green.

Design / Slot / Change - changes the appearance of a slot to be pointed directly (width, depth, angle). It is possible to point successively several slots to be modified.

Design / Slot / Delete all - suppresses all slots that have been added to the selected objects.

Design / Slot / Change all - changes in one single operation the appearance of all slots that have been added to the selected objects.

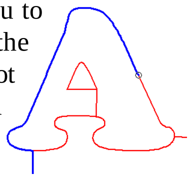


Design / Hot-wire connection / Manual, on closed contours - adds a direct linking line or double-line between two independent objects, by pointing directly on each, at an accurate position on their path. The resulting object is unique. The *"Hot-wire connection"* functions have been made for cutting polystyrene with a hot wire, generally on 2 axes where it impossible to make inactive linking movements with the tool up: the machining path has only one single point of entry into the material and one single point of exit. If the points that have been plotted do not correspond to an end, then the connection makes a round trip.

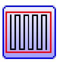
Design / Hot-wire connection / Automatic, on internal islands - adds round-trip connections between contouring objects and inner objects, for example to connect the internal island of a letter A or both internal islands of a letter B (in the latter case, both islands are connected together and one of them is connected to the contour path).

Design / Hot-wire connection / Automatic, on selected objects - adds round-trip connections between all selected objects, contours or internal islands, by using the nearest points. Once the operation has been completed, there is just one single object and consequently a unique path of the hot wire for cutting the whole set. The function asks you to position the point of entry/exit of this set, generally at an easy location for engaging the wire.

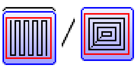
Design / Hot-wire connection / Check path - allows you to follow dynamicallly the hot wire motion along the path of the selected object. The keyboard arrows ← ↓ ↑ → make the hot spot move slower or faster, slowing down can even run backwards if necessary. The Esc key stops the checking function.



Design / Hot-wire connection / Delete - suppresses a simple or round-trip connection between two objects, from a direct pointing with the mouse.

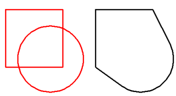


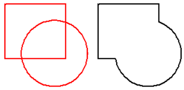
Design / Hatching - fills the surface of the selected object or connected toolpath with hatches to be defined. Please refer to the chapter *"Toolpaths"* for more details about hatching and pocketing functions.



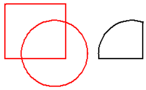
Design / Pocketing cycles - fills the surface of the selected object or connected toolpath with successive contour lines in a pocketing cycle. Again, please refer to the chapter *"Toolpaths"* for more details about pocketing functions.

Design / Overall contour / Convex - constructs a convex external contouring polyline of the selected objects. This contour has no inward angles.

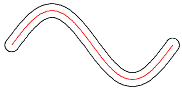




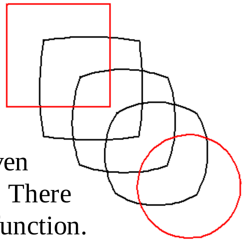
Design / Overall contour / Union - constructs an external contouring polyline of the selected objects. This contour may be open.



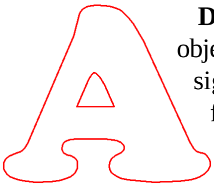
Design / Overall contour / Intersection - constructs an internal contouring polyline of the selected objects. This can be calculated only if there is one single solution.

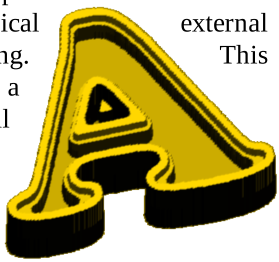


Design / Path Contour - constructs a new object by running a cutter round both sides of an open shape or the outside of a closed shape. If the selected object is closed, the contour produces two paths.

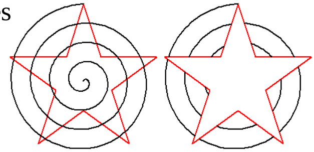


Design / Transmutation - creates the successive intermediate shapes between two selected objects, which must not be connected paths. For example, from two objects that represent topographic contour lines, this function will interpolate intermediate curves, or an object with an internal island will be given successive pockets that transform one into the other. There must be other more or less useful applications for that function.

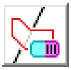




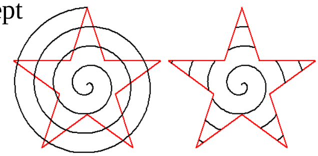
Design / Box & cap - constructs, from closed and selected objects, sets of boxes and caps, typically for obtaining shop sign letters with internal lightings, cut in a thick material for the box. The original object path defines the overall edge and is assigned a classical external tool compensation for cutting. This path is duplicated inside with an offset, giving a thickness which will create a vertical wall. This wall is also assigned a tool compensation inside for setting a deep excavation with a low floor. This vertical wall can have, in its inner side, a small shoulder so a flat cover can fit in. If the original object integrates an internal island, then the paths created are inverted inside/outside for a coherent result. Finally, the cap itself is optional and its cutting corresponds to the internal border of the wall minus a small clearance for nesting. It is transferred into an alternate layer since it will probably be machined in a thinner material with another colour or transparency.



Design / Mask / Inside - splits and eliminates the parts of non-protected objects that are located inside the object or connected path, selected in red. This function corresponds to an incision (paths are cut with new coordinates created at the intersections), followed by a deletion of the object parts that are inside the original selection. The selected object or path can be open, in which case a virtual segment is added to close it. The cut objects that are partly

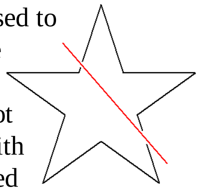


deleted lose their geometrical properties (arcs, curves or text). Please note that one of the delete icons does the same.

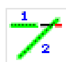


Design / Mask / Outside - as above, except that it eliminates the parts of non-protected objects that are located outside the object or connected path, selected in red, instead of inside. Please note that one of the delete icons does the same.

Design / Incise - cuts, at the points of intersection, any objects not protected that are intersected by the current selection. The paths of the selected objects becomes a blade for cutting everything it crosses. Alternatively, if no objects are selected, the mouse can be used to define where the incision should be made (simple line). The major difference with the *"Design / Object / Split"* function is that the target object can be cut anywhere, not only at an existing vertex. The incision may be multiple, with the mouse line or the incision path selected in red. Incised objects lose their geometrical properties (curves or texts, but not arcs).



Design / Trim / At selected object - cuts, at the point of intersection, any objects (neither selected nor protected) that are intersected by the selected trimming objects and deletes the remaining parts. In the case of an open shape that is intersected only once, the shortest part is deleted. However, with open shapes that are intersected more than once, or with closed shapes, the pieces between the end points and the first points of intersection are removed. The trimming selected objects are not affected. The link/trim icon in the *"Polylines"* series appears to be easier in most cases, but this function allows you to trim several objects in a single operation.

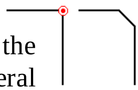


Design / Trim / Selected objects only - as above, except that only the selected objects are affected, *i.e.* they are all cut and trimmed. Warning: the sequence affects the result.

Design / Trim / Blue segment at red segment - cuts a blue segment at the point where it is intersected by a red segment and removes the shortest part of the blue segment.

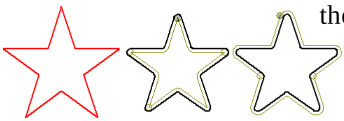
Design / Trim / Red segment at blue segment - cuts a red segment at the point where it is intersected by a blue segment and removes the shortest part.

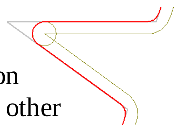
Design / Trim / Red and blue segments at intersection - cuts both a red and a blue segment at the point of intersection and removes the shortest parts.



Design / Chamfer - chamfers the vertex of a polyline to be clicked (or the point selected in red, if any) or alternatively all the vertices can be chamfered if the whole object is selected. Several chamfering methods are available, plus a filtering function for the vertices of the selected object.

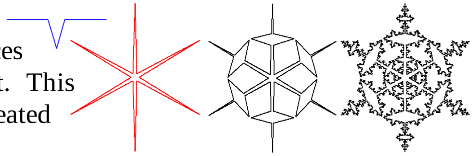
Design / Fillet / Simple, on vertices - as above, in the same conditions to be defined, inserts a single fillet radius at vertices instead of a chamfer segment.





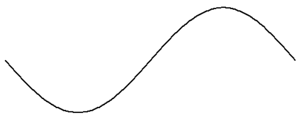
Design / Fillet / Complete, for nesting - suppresses all sharp angles of the selected object and replaces them by fillets which correspond by default to the diameter of the cutter used, so it becomes possible to assign later tool compensations inside and outside that will be able to perfectly match each other once the parts are cut. Due to its own diameter, the tool cannot cut to the deep end an inward angle, and a protruding angle on a cut edge corresponds to an inward angle on the edge of the other part that will itself be adjusted. Consequently, all angles are rounded. You can also add a little clearance to the suggested default diameter. This function has been made for cutting marquetry polygons. The original object is kept but switched to visual (not machined).

Design / Fractal - is a function for artistic purpose only. It constructs a fractal drawing by replacing each segment of an object selected in red with an object selected in blue, suitably scaled to fit, on the condition that the blue object is an open shape. Then a copy of the blue object path replaces every segment of the red object. This operation is interesting only if repeated several times.

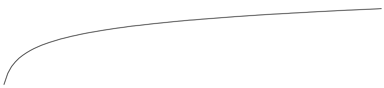


Design / Equation / Simple y=f(x) - constructs a simple plane curve based on a mathematical formula to be entered. The maths fans will immediately and delightfully be projected into the frozen and sophisticated universe of applied geometry, while the others will run away towards the next pages. Based on a few equations and much phosphorus, you can create a 2D or 3D trajectory as accurate as a polynomial. Warning: without wanting to state an academic platitude, let no one ignorant of geometry enter here. Two small examples using the very basic y=f(x):

On an empty drawing board whose dimensions are about 100 × 60 mm, call function *"Design / Equation / Simple y = f(x)"* and enter the equation Y = `20*SIN(3,6*X)` in the box at the top of the window. Set as origin point Yo = 30 mm and for definition interval for X from 0 to 100 by steps of 1 mm. Validate the whole: what comes out of it a beautiful sinusoid curve on one period, that you could have drawn much more easily by using the corresponding icon in the series of polylines and curves.



Please note that the function SIN() of the syntax analyser uses degrees and not radians. So the result was a variation of X from 0 up to 100 by steps of 1 and, consequently, a sine argument varying from 0 to 360°. The amplitude is set by the multiplication factor at left hand side of the sine, here 20 mm, and the depth is indicated by the box Zo.



Another one just for the fun: Y = `10*LOG(X)` with X varying from 1 to 100 by steps of 1, and Yo = 30 mm. If you abuse zero for the function LOG(), then a horror message will pop up to call you to order. It will be the same if you pretend to divide by zero, extract a negative square root and several other insanities which will make the real insiders shake their heads, with them is our salvation.

Keep in mind that **the functions of the syntax analyser remain available when entering directly coordinates or any numerical value**. For example, you may enter *"Abs(12.34+5.67*CosR(PI/8.9))"* instead of a dimension in a dialogue box, hoping that this big effort will help you reach the contemplative life. Unfortunately, the calculation is direct and the formula is lost as soon as the result has been validated.

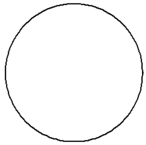
The functions that are known to Galaad syntax analyser are:

ABS (...)	returns the absolute value
ATG (...)	returns the arc tangent, in degrees
ATGR (...)	returns the arc tangent, in radians
COS (...)	returns the cosine, in degrees
COSR (...)	returns the cosine, in radians
CUBE (...)	returns the cube
EXP (...)	returns the exponential e^x
FRAC (...)	returns the fractional part
HCOS (...)	returns the hyperbolic cosine
HSIN (...)	returns the hyperbolic sine
INT (...)	returns the integer part
LN (...)	returns the natural (Naperian) logarithm
LOG (...)	returns the decimal logarithm
RAND (...)	returns a randomised value (max argument = 999)
RND (...)	returns the value rounded to the nearest whole number
SIN (...)	returns the sine, in degrees
SINR (...)	returns the sine, in radians
SQR (...)	returns the square
SQRT (...)	returns the square root
TG (...)	returns the tangent, in degrees
TGR (...)	returns the tangent, in radians

Known constant: **PI**, without brackets, no need to introduce.

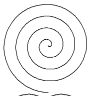
Design / Equation / Triple (x,y,z)=f,g,h(t) - constructs a curve in the space from three equations. The available maths functions are the same as those described above. Example, on a drawing board whose dimensions are 100 × 60 × 30 mm, enter the following equations:

X = 20*COS(3,6*T)   Y = 20*SIN(3,6*T)   Z = 1

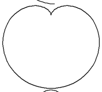


Now the variable is named T and varies in a definition domain from 0 to 100 by steps of 1. A cosine on X, a sine on Y, the whole being developed on 3,6 × 100 = 360 degrees with a main amplitude of 20, all this looks very much like the circumference of a circle with a radius of 20 mm.

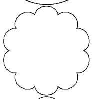
Do not forget to set the centre of your board for Xo and Yo. You can have fun entering the following equations, for a variable T varying from 0 to 1 by steps of 0.01 with Xo and Yo still at the centre of your drawing board.



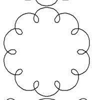
Spiral: X = `20*T*SIN(4*360*T)`
Y = `20*T*COS(4*360*T)`



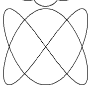
Cardioid: X = `10*(2*SIN(360*T)-SIN(2*360*T))`
Y = `10*(2*COS(360*T)-COS(2*360*T))`



Epicycloid: X = `20*COS(360*T)-2*COS(10*360*T)`
Y = `20*SIN(360*T)-2*SIN(10*360*T)`



Trochoid: X = `20*COS(360*T)-4*COS(10*360*T)`
Y = `20*SIN(360*T)-4*SIN(10*360*T)`



Lissajous: X = `30*SIN(2*360*T)`
Y = `30*SIN(3*360*(T+0.05))`

For the same price, you can find hereafter their general equations, for T being an angular variable that runs from 0 up to 360 (degrees):

Spiral: X = `SIN(N*T)*R*T/360`
Y = `COS(N*T)*R*T/360`
where *N* is the number of revolutions and *R* the max radius.

Cardioid: X = `R*(2*SIN(T)-SIN(2*T))`
Y = `R*(2*COS(T)-COS(2*T))`
where *R* is the size of the cardioid.

Epicycloid: X = `(R1+R2)*COS(T)-R2*COS(((R1+R2)/R2)*T)`
Y = `(R1+R2)*SIN(T)-R2*SIN(((R1+R2)/R2)*T)`
where *R1* and *R2* are the radii of the inner and outer circles of the epicycloid.

Trochoid: X = `(R1+R2)*COS(T)-H*COS(((R1+R2)/R2)*T)`
Y = `(R1+R2)*SIN(T)-H*SIN(((R1+R2)/R2)*T)`
where *R1* and *R2* are the radii of the inner and outer circles of the trochoid, and *H* the offset of the mobile point.

Lissajous: X = **`AX*SIN(FX*T)`**
Y = **`AY*SIN(FY*(T+`*`Delta`*`))`**
where *AX* and *AY* are the amplitudes of the Lissajous ellipse, *FX and FY* the horizontal and vertical frequencies, and *Delta* the small dephasing between both amplitudes.

This awful litany is indeed extensive. In fact, the only limit to creating 'Potatoid' curves is the imagination and perhaps the user's geometric spirit. But there is more to life than maths.

And, to end on a high note which is going to catapult us into the third dimension, a small triple equation, T varying from 0 to 600 by steps of 1:

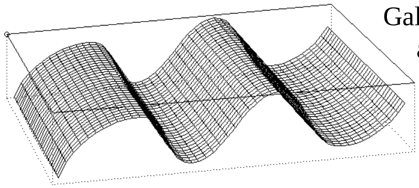
X = `20*COS(3,6*T)`
Y = `20*SIN(3,6*T)`
Z = `T/20`

Note the definition interval from 0 to 600, which will give an angular variation in the sine and cosine from 0 to 2160°, actually 6 revolutions. Do not forget to set the centre of the drawing board for Xo and Yo. At first sight, the result is a stupid circle centred in the middle of the board. Call up the 3D view of the thing, you will see that it is in fact a helix which sinks down into the board thickness, and which seems evident considering the linear variation of Z. It is up to you to play: here, all things are numbers.

Design / 3D mesh / Rectangular - constructs a rectangular 3D mesh with a cross-sectional profile in the XZ plane, that follows the XY profile of the selected object or the two selected objects. Two examples can help you understand:

Draw any curve, for example a Bezier, with the trajectory going from left to right with no overlapping parts, *i.e.* no two points should have the same X coordinate. **The curve must go continuously from a lateral border to the opposite and not double-back on itself**. On the other hand, Y coordinates can go up and down.

Got it? Now select the curve and use *"Design / 3D mesh / Rectangular"*. This pops up a dialogue box requesting the spacing of the mesh and a few additional parameters. Complete it without asking more questions and validate. You will be returned to the drawing board, then asked to define the rectangular area of the mesh. Use the mouse to drag over a large rectangle that covers most of the board and is wider than it is tall, (X > Y), but don't worry if it overlaps with the curve.

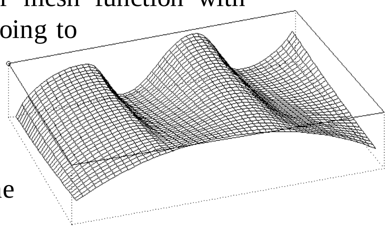


Galaad will then generate a wiremesh, the accurate XY size of the rectangle, and use the profile of the reference curve to generate the variations in the Z depth. The 3D view is automatically displayed.

Depending on the drawing of your curve and the thickness of your material, it is possible that you will get an error message stating **maximum depth exceeded**. Ignore this for now. With two shapes and three movements, you have constructed a mesh describing a 3D surface. The rest of the game is just variations on a theme.

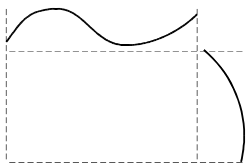
Returning to the XY view, the reference curve is no longer selected, but is still present on the board and has changed colour. Galaad understands that it is an object used to produce the drawing but not to be machined and consequently transforms it into a visual object. Anyway, it has not lost any of its characteristics and can be modified if you so wish.

Now delete your wiremesh and draw a second curve, for example an arc with a large radius, then select them both. So you have two different references selected. Bring up the rectangular mesh function with exactly the same parameters. Galaad is going to construct a new wiremesh that will make the progressive transition from one shape to the other. The overall Y position of each curve determines which one will be close to each border of the mesh rectangle.

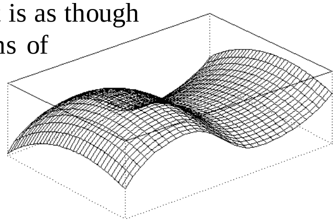


Design / 3D mesh / crossed - constructs a rectangular 3D mesh following the XY profiles of the two selected objects having perpendicular general orientations.

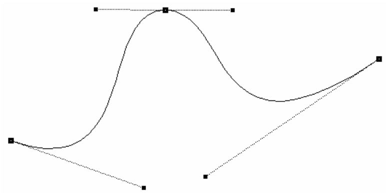
Draw two simple objects, for example again a Bezier curve and an arc with a large radius. Lay them out such that one is generally in the X direction and the other in the Y direction without doubling back, as illustrated here.



Select them both, use the function and provide the necessary parameters in the dialogue box. This time, Galaad does not ask you to define a rectangle but constructs a mesh from the two curves, using the intersection point and their lengths to define the mesh size and position. It is as though you had drawn the XZ and YZ lateral sections of your mesh. It is a bit like if you had drawn the XZ and YZ lateral profiles of your mesh. The shape of the Y oriented curve is applied to the depth first, followed by the X oriented curve to produce the overall 3D effect.



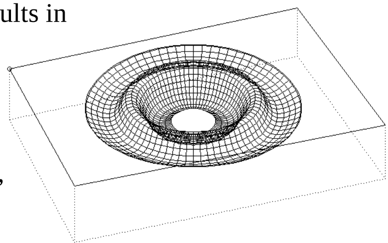
Design / 3D mesh / Circular - constructs a circular 3D mesh following the XY profile of the selected object. The path from left to right of the selected object will be running from the centre to the periphery of the mesh.



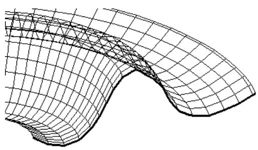
Delete everything from the drawing board and draw a Bezier curve, similar to that shown on the left, using the appropriate drawing icon. As before, ensure that the curve does not double-back on itself in the vertical plane.

Select the curve and use the *"Design / 3D mesh / Circular"* command, then provide the necessary parameters for the mesh. Finally, you have to draw two concentric circles with different radii. Make one of them small and the other large so that a decent sized mesh is produced. Forget the aperture and orientation angles which do not matter for the moment.

Galaad calculates then displays the results in the form of a circular or toroidal mesh comprising concentric and radial mesh lines, depending on your parameters. The quad view is then called automatically if it was not already active, so that you may admire the results.



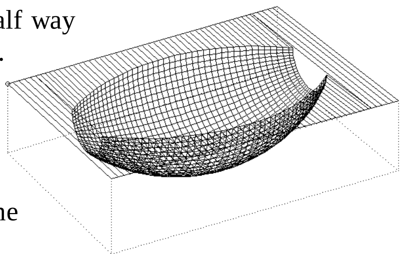
Take a cross section along any of the radial mesh lines, the original reference curve is clearly visible. The variation in depth of each radius, from the centre outwards, follows the profile of the curve from left to right.



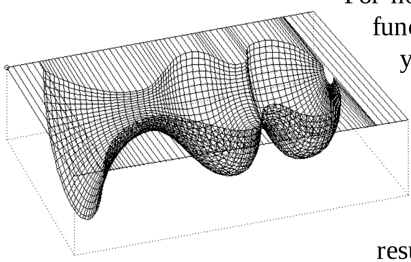
Here again, the absolute dimensions of the reference curve do not matter because they will be adjusted to match the length of the radii, which are defined by the distance between the inner and outer circle. Each concentric circle in the mesh is confined to an XY plan (the depth is constant) and consequently can be machined with a 2½D machine. Alternatively, you can define the inner construction circle as having a radius of zero, namely a point, to avoid having a hole in the centre of the mesh. **If you have selected two objects as references, the mesh will follow a progressive transformation from one to the other** (from the starting angle to the ending angle). This allows you to create partial meshes with limited aperture angles, passing smoothly from one mesh to another, following the shapes drawn.

Design / 3D mesh / Revolved - constructs a semi-cylindrical 3D mesh on the board, following the XY profile of the selected object. This type of mesh is to the cylinder what the circular mesh is to the disk. Start with something simple and generate an arc, with a very large radius, by using the icon for an arc defined by three points. You can equally use any other icon that will allow you to produce a similar result. Don't worry about being precise, it is not too important. Select it and position it somewhere towards the top of the board. Now use the function and quickly move on past the dialogue box that appears.

Position the horizontal axis about half way up the board and click to set the position. This will be the axis around which the shape will be revolved. Once again, Galaad knows how to manage the situation itself and will rotate the curve through half a revolution to produce the corresponding mesh.



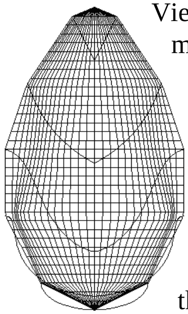
If you are unfortunate enough to trigger an error message telling you that the thickness of the material has been exceeded, don't worry, just accept it. For now it is important to understand how the function works, you can always work out for yourself any questions relating to thickness when you come to machine the thing. It goes without saying that you can also do the same thing with different reference objects, and they will produce an even more complex result. As with all the other types of mesh,



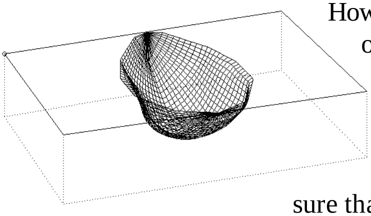
avoid any doubling back as this will produce a mesh that will be difficult or impossible to machine.

Design / 3D mesh / Sheaves - constructs a 3D mesh following the profiles of the two selected objects. This type of mesh reuses the drawing technique for sheaves between two objects selected in red and in blue. In this case, it is necessary to first set the 3D profile of each, using the plane inversion functions from XY to XZ or YZ. Galaad will just create sheaves between them, but will nevertheless add the perpendicular mesh.

Design / 3D mesh / Lofted - constructs a 3D mesh following the successive XZ profiles of the selected objects, for example the cross sections of a boat hull or an airplane wing. Granted that it is a bit more complicated, but it can be viewed as an extension of the rectangular mesh method. The aim is to draw, in XY, a series of XZ sections, which will be used as references to construct the mesh. Select each XY section, one at a time, and use the *"Design / 3D effects / Flip Y and Z planes"* command, which will convert them to XZ sections. Ensure that you are happy with their positions on the board then select them all before calling up the function and completing the ensuing dialogue box.



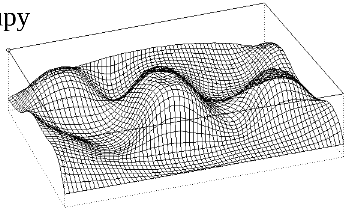
Viewed from above, the result is similar to the rectangular mesh method, but this time the area is defined by the outer extremities of the reference objects, rather than being a simple rectangle. The width of the result is determined by the width of the reference objects that were selected, whilst the height of it is controlled by their position on the board. What is not so clear from this view is how the depth and profile of the mesh have been generated, as the original reference sections can not be distinguished from the rest of the mesh. The YZ lateral view shows how the mesh changes at the reference sections.



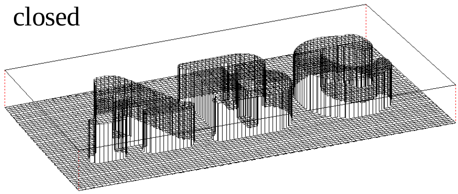
However, the 3D view shows how this method of construction works much better. The successive cross sections, originally drawn in XY, now provide the controlling XZ sections and retain their existing dimensions. Then you can make sure that the mesh will accurately correspond.

As usual, we will finish on a lighter note. Try and quickly construct a somewhat strange and rather artistic shape instead of something useful, by using several sections drawn free-hand. Note that Galaad only uses a maximum of 32 reference objects.

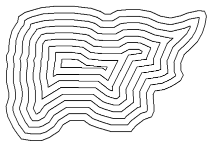
In addition, note that if two objects occupy the same Y position, there will be two different sections at the same spot, which is not possible except to make an infinite number of transformations. Galaad does not do this and will refuse to produce the mesh.

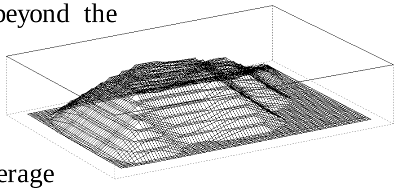


Design / 3D mesh / Extruded - constructs a 3D mesh that extrudes on Z axis the contours or XY trajectories of the selected objects. The procedure is quite simple: just draw and select closed objects which have either their internal surface or their contour extruded. These objects are already positioned and the mesh will cover them.



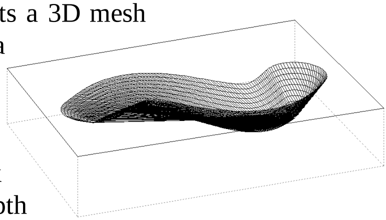
Design / 3D mesh / Contour lines - constructs a rectangular 3D mesh whose wires are given a depth that depends on the selected objects. Every curve defines an XY horizontal plane that will remain unchanged until it matches another curve with a different depth. The whole mesh must be contained inside the boundaries of the selected curves, otherwise the horizontal plan beyond the borders cannot be calculated by Galaad. The mesh lines can be either 3D (giving oblique Z variations with stretching of the intervals between lines) or follow stages corresponding to average levels by default or by excess.



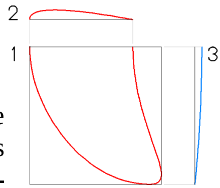


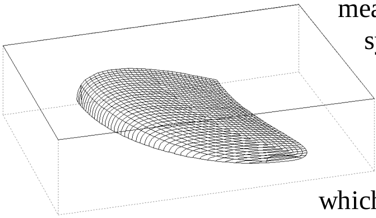
Design / 3D mesh / Resection points - constructs a mesh that is modulated on Z axis by the depths of the dispersed points, all selected together. These points can come from a topographic file with free XYZ coordinates. Each of them will attract towards its own depth the mesh in its neighbourhood, this attraction being modulable according to a smoothing function.

Design / 3D mesh / Track - constructs a 3D mesh that reproduces a milling track with a conical section along the trajectory of the selected object. The ends are rounded as if the track was machined by a tool with a conical profile. The track width is constant. The variations of depth modulate the angle of the lateral sides of the conical profile.

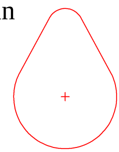


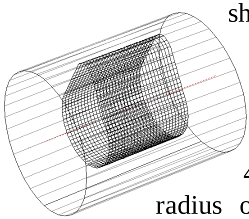
Design / 3D mesh / Airfoil - constructs a 3D mesh modelising for example a surf fin or an airplane wing. The airfoil contour and its half-profile must be selected (objects number 1 & 2 on the sketch here). Half-profile means that the airfoil is symmetrical. This half-profile will be adjusted between the extreme points of each horizontal section of the foil contour. It is also possible to define an extra curve on a vertical axis, which will define the evolution of the profile thickness from top to bottom. This curve must then be selected in blue (object number 3 on the sketch).





Design / 3D mesh / 4-axis wrapping - constructs a Cartesian mesh whose development in cylindrical milling reproduces the shape of the selected object, with an axis rotation about the red cross (so there must be one in the drawing board). This type of mesh allows you for example to quickly model a cam, perfectly machinable using a flat end cutter in a 4-axis cycle. Warning: it is very important that the radius of the cylinder to be machined corresponds to the distance between the red cross (rotation axis) and the furthest vertex, minus the minimum depth that has been entered in the dialogue box (top of the cam). If this radius is incorrect, then the machining will distort the profile.





Design / 3D mesh / Background image - constructs a matrix of points or a line mesh whose depths vary depending on the background image. This function can work on an engraving only if the machining is made with a tool with a conical profile. The lower the tool plunges in the material, the larger the hole or the track is. The result looks very good with pictures.

First, you must have a background image (see later on the functions *"Display / Background image"*). This background image must be small since every pixel will correspond to a machined line or point. If you set an interval of say 1 mm between points, then an image of 100 pixels height will already give a 100 mm result. So, looking for megapixels is pointless and a very low resolution picture is quite sufficient. For setting its size, use *"Display / Background image / Frame"* function until it is small enough. A good contrast is highly preferable.

Two options are then available: a matrix of drilled points with varying depth, or a mesh of tracks from a horizontal 3D scan. Obviously, the mesh of tracks is much faster to mill. Once the matrix or mesh has been created and selected, you can first enlarge it and re-position it on the board. Then, using the dimensioning dialogue box for the selected object (Enter key), it is possible to darken overall the image of the machined result by increasing the depth Zo, or increase its contrast by stretching the depth range Dz. A view of that result can be displayed via the function *"Display / Final rendering / Trace"* or its shortcut icon. Keep in mind that even an ultra-high resolution screen has a finite accuracy, so the final rendering will always remain rough compared to the machined reality of the workpiece. The best is to zoom in deep on important details during the final rendering (simple click, and right mouse click for zooming back out). The adjustment of Zo and Dz dimensions using Enter key becomes very important here.

Tip: you can machine a semi-opaque plastic plate with a flat-end cutter whilst inverting the machining and using a large depth range. The plate seen from the non-machined side will let the picture appear by semi-transparency when you put it in front of a source of light.

Design / 3D mesh / Change / Add perpendicular mesh - creates a framework of mesh lines perpendicular to the selected mesh, going through exactly the same coordinates. This function can be applied only onto a Cartesian wiremesh.

Design / 3D mesh / Change / Make bi-directional - converts a unidirectional mesh into one with alternate directions along adjacent lines.

Design / 3D mesh / Change / Make unidirectional - converts a bi-directional mesh into one with the same direction along adjacent mesh lines. It is also possible to use the function *"Machining / Toolpath / Define as start point / The point which is most at …"* on a Cartesian mesh.

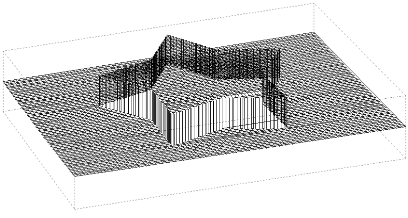
Design / 3D mesh / Change / Make continuous - converts individual mesh lines into a continuous zigzag mesh.

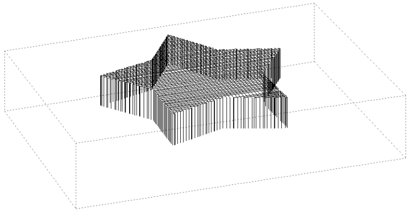
Design / 3D mesh / Change / Make individual - converts a continuous zigzag mesh into a series of individual mesh lines.

Design / 3D mesh / Change / Reduce stepover - adds new lines to the selected mesh. This addition is a simple interpolation without recalculating or smoothing the result.

Design / 3D mesh / Change / Increase stepover - spaces out adjacent lines of the selected mesh by deleting intermediate lines.

Design / 3D mesh / Change / Delete horizontal plane - suppresses the lines of the bottom or top surfaces of the mesh, to keep only the useful shape. A dialogue box lets you define the plan to be deleted, with a tolerance margin for the Z value.



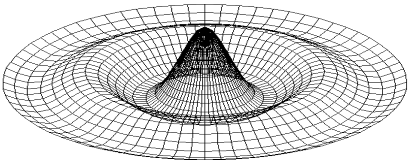
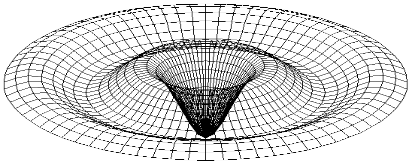


Design / 3D mesh / Change / Change mesh from background image - modifies the selected image mesh for resetting its creation data.

Design / 3D effects / Select 3D objects - selects all available objects that have a variable depth.

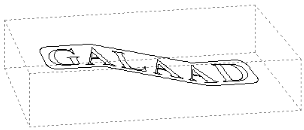
Design / 3D effects / Select points from filter - selects in *"Range of points"* mode the points or vertices situated in a depth range to be indicated. This mode corresponds to the yellow icon with the same name.

Design / 3D effects / Reverse Z trajectory - inverts the relative depth of the selected objects. On a 3D path, the hollows become bumps and vice-versa. On a group of 2½D objects having been given different depths, this places at the top those with the lowest depth, at the bottom those with the highest, and all intermediates are reversed.

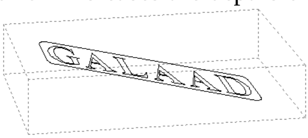
Design / 3D effects / Flip X and Z planes - exchanges the X and Z coordinates of the selected object vertices.

Design / 3D effects / Flip Y and Z planes - exchanges the Y and Z coordinates of the selected object vertices. This function is useful for creating lofted meshes.

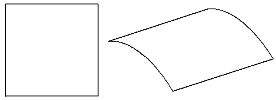


Design / 3D effects / Frame Z trajectory - sets a floor and a ceiling limit on the depths of the selected object. Any coordinate which is out of the range is minimised or maximised. Warning: for 3D objects, long lines are not vectorised; only Z coordinates change.

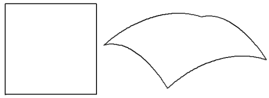
Design / 3D effects / Project onto a tilted plane - increases the depths of vertices or points on the route of the selected object according to the value of its XY coordinate. The tilted plane can be either XZ or YZ. The relative depths of the objects between them are kept.



Design / 3D effects / Project onto a cylinder - changes the Z coordinates of the selected objects for projecting them onto a cylinder, parallel to the X axis, with a diameter to be defined. The objects are segmented by interpolating new points for a smooth bending. Warning: the cylinder axis is always located at the Y centre of the board.



Design / 3D effects / Project onto a sphere - as above, except that the projection target is now a sphere, with a diameter to be defined, and centred at the middle of the drawing board.



Design / 3D effects / Apply onto background - changes the depths of the selected objects depending on neighbouring objects, for example to apply text onto an existing wiremesh. The depths of the selected object are added to the Z coordinates of their neighbours, using a weight attractor which depends on their XY distance.

Design / 3D effects / Change depths according to the blue profile - applies a new range of depths for the selected objects in relation to the curve that is selected in blue. This functions can for example compress or stretch a land mesh for enhancing the topography. If the blue selected object is a simple straight line at 45°, then the Z coordinates will remain unchanged. If this object is an arc of circle from 180° to 90°, then the low depths will be stretched and the high depths will be compressed. The blue curve represents the distribution graph. Its position and dimensions have absolutely no importance; only its out-of-scale path matters.

Design / 3D effects / Follow a continuous slope - increases regularly the Z coordinates along the path of the selected object between its point of departure and its point of arrival. The increase for the coordinate of each vertex is proportional to its distance from the start point, following the path.

Design / 3D effects / Follow blue trace slope / Along its Y trajectory - modulates the depths of points along the path of the selected object, following the relative Y coordinates of the line of the object selected in blue. Once again, the XY path of the blue object is used as a reference to apply depth variations with Y coordinates becoming Z coordinates. The difference between this function and the yellow icon *"Modify Z depth"* is that the slope runs along the path of the selected object (distance from start point along the path) and not along its X coordinates only (distance from left point).

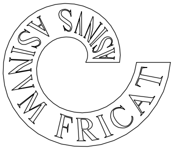
Design / 3D effects / Follow blue trace slope / Along its Y trajectory - increases or decreases the depths of points along the path of the selected object, following the relative Z coordinates of the line of the object selected in blue. This blue object is therefore supposed to have a 3D path.

Design / 3D effects / Create a 3D track - builds a 3D polyline of which the variable depth allows a conical tool to make the route of the two selected objects flush. These are considered the framing edges of the newly created object. The result is a 3D carving effect which modulates the depth according to the width of the track to be rendered. Warning: the two objects near each other must have open paths.

Design / 3D effect / Apply a 3D carving track effect - Makes the depth of the selected object path vary, if machined using a conical tool, so it gives a conical carving effect. The track becomes deeper along the path between its ends or sharp angles.

Design / 3D effects / Transform using equations - subjects the X, Y and Z coordinates to three mathematical transformation equations. The coordinates submitted to the equations are always considered as variables, but **relative to the southwest corner of the object**. This means that the overall position on the drawing board does not change the result. These equations recognise as variables the labels 'X' and 'Y' which correspond to the points or vertices constituting the selected object, it being fully understood that, in the case of multiple equations, the input values are calculated, not the resulting values after passing through the first equation. The second and third equations have nothing to do with the result of the first and second equation in their own analysis. Well, do not worry about it, this is Galaad's internal cooking. But do not forget to add the input value of the coordinates to each equation, if you do not wish to obtain a completely flat object.

For example, you must type Y = `Y+20*SIN(4*X)` and indeed not Y = `20*SIN(4*X)`, unless your tortured object will have lost all its Y thickness when coming out of the equation.



A triple equation requires three times more phosphorus (having fun...):

X = `50+(2+Y+X/10)*COS(4*X)`

Y = `(2+Y+X/10)*SIN(4*X)`

Z = `X/10`

The following little creation works very well in space, simply open the file *"3D \ MOGULS"* to see the results. We will give you the recipe: set the dimensions of a new board to 100 × 60 × 20 mm, and create a network of lines, quite closely spaced, covering the surface of the board, this is easiest by simply duplicating one. Select everything and transform them using equations, the magic formula being X = `X`; Y = `Y` (no changes, indeed, and you can simply leave these boxes blank, Galaad will understand that coordinates remain the same), and above all Z = `10+5*SIN(10*X)*SIN(10*Y)`.

❑ *"Display"* menu

Display / Active Layer / Layer N - allows you to define the layer number N as the active layer, namely the layer you are working in. All other layers remain in the background and objects on them are visible in grey but not accessible for drawing operations. Unless filtered, objects in inactive layers can be machined and are greyed or invisible (see next command about visual layers), but remain unavailable for drawing operations. No selection frame survives in an inactive layer. Furthermore, it is possible to limit snapping to the objects of the active layer. This option can be set from the advanced functions of the workspace parameters. However, in case of an emergency, the function *"Edit / Select all"*, when Ctrl key is pressed, selects together all selectable objects (*i.e.* not locked) of all layers.

Display / Active Layer / Transfer to active layer / All objects on layer N - moves to the layer currently active all objects from the chosen layer, even if they are locked. This layer N is then empty. All transferred objects are automatically selected (even those locked, one minor exception), to ease handling or to permit resending them to another layer. If some objects were already selected, then they are no more.

Display / Visual layers / Layer N - allows any background layers to be hidden if required. Obviously, the active layer is always visible, otherwise the job would become a bit difficult.

Display / Visual layers / All layers - makes all background layers visible (objects in inactive layers are displayed in grey).

Display / Visual layers / Active layer only - hides all background layers and only displays the active layer. If the active layer is changed, the view will be updated accordingly.

Display / Visual layers / Name layers - allows you to give a name and also a colour to each drawing layer. These layer names are saved with the file and will remain present in future drawings unless their own layers have also been named. After opening or initialising a new file, you may restore the last layer names and colours that have been used, through the buttons *"Previous"* at the bottom of the dialogue box. Please see also the options of the function *"Parameters / New file defaults"*, where you can decide if you wish to keep or forget the layer names and colours.

Display / Visual layers / Limit direct snapping to active layer - toggles the option for accepting the suggestions to snap the drawing plotter to positions related to the objects of the active layer (or, by default, all layers). If your drawing is heavily loaded, the snapping suggestions are too many and it can become difficult to choose the right one. This option is also available in the advanced workspace settings, but rendering it accessible in a menu makes it programmable from a function key so you can quickly change the mode (see *"Parameters / Function keys"*).

Display / Visual layers / Save these names & colours - saves the current names and colours of the layers in a working environment file.

Display / Visual layers / Restore saved names & colours - recalls previously memorised names and colours.

Display / Trace / Bold - displays all objects on the active layer using bold lines (or by default thin lines). The objects that are present in other layers always remain displayed in thin and grey lines.

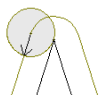
Display / Trace / Object dependant - displays objects on the active layer using a line thickness that has been drawn for each object (see the green drawing icon to define object colours and thickness). Remember that the thickness of a line on the screen has no relation to the machined result, that you can see using function *"Display / Final rendering / Trace"* hereafter.

Display / Trace / Path / Start points - displays a small cross at the start points of all objects on the active layer whether or not they are part of a larger connected toolpath. This is supposed to ease the checking of tool plunges.

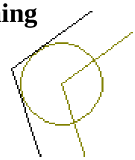
Display / Trace / Path / Link points - displays a small point at the linking points of all objects that are part of a larger connected toolpath.

Display / Trace / Path / All points - displays a small point at the vertices of all objects on the active layer. The drawing may become a bit overloaded, especially with curves, and the screen display is slower.

Display / Trace / Tool compensation / Feed-in path - displays the feed-in path of a compensated toolpath, a circle represents the diameter of the cutter used and a small arrow shows the path direction, which helps checking closed objects.



Display / Trace / Tool compensation / Feed-out path - displays the feed-out path of a compensated toolpath, a circle to represent the diameter of the cutter used.



Display / Trace / Tool compensation / Angles (roughing trajectory) - displays a circle representing the cutter diameter at all internal angles on the trace of a compensated roughing toolpath. The threshold angle is set in *"Machining / Tool compensation / Advanced parameters"*.

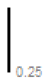
Display / Trace / Tool compensation / Angles (finishing trajectory) - displays a circle representing the cutter diameter at all internal angles on the trace of a compensated finishing toolpath.

Display / Trace / Protection - displays a small shield at the start point of all protected objects.

Display / Trace / Position anchors - displays a small anchor at the start point of all objects anchored to an absolute position.

Display / Trace / Machining pauses - displays a small cross at the start point of all objects which require a machining pause. A pause is a validation message that pops up on the screen before machining the object, whilst the process hangs in mid-air. The pause can be defined in the dialogue box for the machining data (tool, depth, feedrate).

Display / Trace / Path directions - displays a small arrow at the start point of all objects, indicating the path direction which is not always visible for closed and non-compensated objects.



Display / Trace / Identifiers / None - removes the small identifiers that are displayed at the start points of the objects (see identifiers below).

Display / Trace / Identifiers / Sequence - adds a small identifier giving the sequence number at the start point of objects.

Display / Trace / Identifiers / Tools - adds a small identifier giving the machining tool number at the start point of all objects.

Display / Trace / Identifiers / Depths - adds a small identifier giving the depth (or the depth range for 3D objects) at the start point of all objects.

Display / Trace / Identifiers / Z Changes - displays the depth changes at the vertices of all 3D objects, except if these vertices do not correspond to a change of the Z coordinate.

Display / Trace / Identifiers / Feedrates - adds a small identifier giving the machining speed at the start point of all objects.

Display / Path colour mode / Monochrome - displays all objects on the active layer in black, *i.e.* no distinction is made between different cutters. In fact, black is the default colour on a white background but can be changed from *"Parameters / Colours"* and *"Primary paths"* entry.

Display / Path colour mode / Layer dependant - displays all objects in the colour set for their layers. Layer colours are set from *"Display / Visual layer / Name layers"*.

Display / Path colour mode / Tool dependant - displays all objects in the active layer in the colour set for the cutter selected. Tool colours are set from *"Parameters / Tools"*.

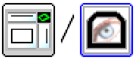
Display / Path colour mode / Object dependant - displays all objects in the active layer in the colour set for each. Objects colours are set from the green drawing icon for colour and line thickness.

Display / Path colour mode / Default tool - displays in black (primary path colour, which of course will not be black if the background already is) all objects on the active layer using the default tool and the rest in grey (secondary path colour). This colour mode is probably one of the most useful.

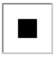
Display / Path colour mode / Default depth - displays in black (actually the primary path colour) all objects on the active layer using the default depth and the rest in grey (actually the secondary path colour).

Display / Path colour mode / Default speed - displays in black (actually the primary path colour) all objects in the active layer using the default feedrate and the rest in grey (actually the secondary path colour).

Display / Path colour mode / Black background - sets the background to black. Normal lines will appear in bright green. Please note that all display colours can be set from *"Parameters / Colours"*.



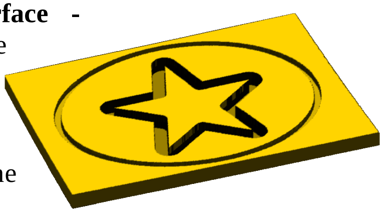
Display / Final rendering / Trace - displays a plan view of the cutter path using a line of width equal to the cutter diameter, relative to the current magnification and board dimensions. The left mouse button allows you to zoom in and the right mouse button to return to the global view. Only the active layer is concerned. Furthermore, **if objects are selected, they are displayed alone** and the rest is hidden. It is possible to change trace and background colours from the left side of the screen, in



the mini-zoom box just under the drawing icons. All this remains valid for other functions of final rendering.

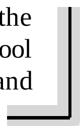
Display / Final rendering / Mesh - displays a 3D wire frame representation of the cutter path showing both the width and depth of the path relative to the material dimensions.

Display / Final rendering / Surface - displays a 3D rendered surface representation of the cutter path, showing both the width and depth of the path relative to the material dimensions. The orientation of this view reuses the one of the 3D plane view (or quad view).



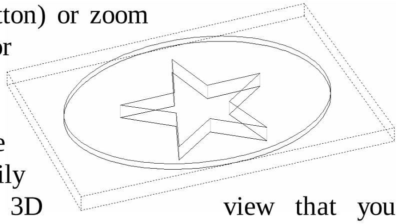
Display / Final rendering / Parameters - allows you to set up the few parameters related to 3D final rendering in mesh mode.

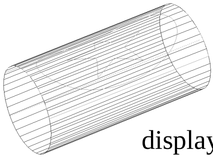
Display / Permanent final trace - permanently displays the machined trace of the objects in the active layer, calculated from the tool profile and diameter. The colour is set in *"Parameters / Colours"* and *"Dot enhancement"* entry, which also corresponds to display enhancement spots around isolated dots.



Display / Quad view - allows you to split the screen into four different views of the current drawing, *i.e.* plan (XY), front (XZ), side (YZ) and 3D view. The basic layout of the split screen can by configured in *"Parameters / Workspace / General settings"*. A double-click on the 3D view switches it to full screen, and back again. The projected paths on the top surface remain visible in this case. When clicking on the corresponding icon, temporarily pressing the Ctrl key displays a quad view with a rotary 3D part.

Display / 3D view / Plane - displays the 3D view at full screen size, letting you orientate the angle of view (left mouse button) or shift the plane view (right mouse button) or zoom (mouse wheel or + / - keys). This is for viewing only, no operations can be carried out on this board. When clicking on the corresponding icon, pressing the Ctrl key temporarily displays a rotating 3D view that you can interrupt with the Esc key. By the way, if the workpiece was defined as a cylinder, then the 3D view displays its diameter and wraps the drawing around its circumference for showing the result that will be machined. And talking about cylinders, let us say again that it is possible to select an object in blue for defining the external profile of revolution, *e.g.* a jar instead of a tube.





Display / Data palette (at top) / None - removes the data palettes at the top of the screen. See the *"Rapid data palettes"* section at the end of the *"Advanced drawing techniques"* chapter for more details about data palettes. Please remember that the palette title on the left hand side is reactive to the right mouse click, for popping a context menu where you can decide which palettes must be displayed. The small red button closes up the corresponding palette.

Display / Data palette (at top) / Layers - displays a small palette at the top of the drawing area, showing the layers used in the current drawing to allow rapid selection of the active layer.

Display / Data palette (at top) / Tools - displays a small palette at the top of the drawing area, showing the cutters used in the current drawing to allow rapid selection of them.

Display / Data palette (at top) / Machining depths - displays a small palette at the top of the drawing area, showing the machining depths used in the current drawing to allow rapid selection of them.

Display / Data palette (at top) / Feedrates - displays a small palette at the top of the drawing area, showing the speeds used in the current drawing to allow rapid selection of them.

Display / Data palette (at top) / Pen colours - displays a small palette at the top of the drawing area showing the colours used in the current drawing to allow rapid selection of them.

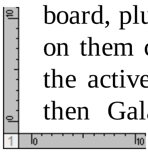
Display / Data palette (at top) / Pen thickness - displays a small palette at the top of the drawing area, showing the thicknesses used in the current drawing to allow rapid selection of them.

Display / Fast access column (at right) - displays a vertical bar to the right of the drawing window, allowing you to enter data directly for selected objects and to store objects for later re-use. See the end of the *"Advanced drawing techniques"* chapter for more information.

Display / Personal icon palette - displays a floating palette over the drawing window, where you can place your most common drawing icons for quick access. See the end of the *"Advanced drawing techniques"* chapter for more information.

Display / Margins - places a small margin around the drawing board, allowing you to check trajectories that have a certain tendency to escape. A slider sets the size, relative to the board size, which will remain memorised. Lateral margins are passive; it is not possible to draw in them.

Display / Rulers - displays the rulers to the left and at the bottom of the board, plus the lateral views. Rulers are purely visual; clicking or pointing on them calls up no functions, except the southwest square that displays the active layer number: if you click on it with the right mouse button, then Galaad saves its workspace just like if you were exiting then reopening the software.



Display / Visual grid / None - removes the background grid from the work board.

Display / Visual grid / Points - displays a visual grid of points on the work board.

Display / Visual grid / Crosses - displays a visual grid of crosses on the work board.

Display / Visual grid / Dotted - displays a visual grid of dotted lines on the work board.

Display / Visual grid / Solid - displays a visual grid of solid lines on the work board.

Display / Visual grid / Matrix - displays a visual grid of millimetric lines on the work board, which always corresponds to the smallest ruler graduations, whatever the zoom state is. Probably your best choice.

Display / Visual grid / Set step - allows you to define the distance between two lines of the visual grid.

Display / Background Image / Enabled - allows you to display or remove from the board the background image to trace from.

Display / Background Image / Open - loads a background image under the board to trace from. Zoom functions remain valid until a maximum enlargement is reached. The background image is of course passive and cannot be machined. See the *"Design / 3D mesh / Background image"* function to create a toolpath from a background image. Please note that tracing is easier with a light image (or dark in case of a black background). See below the functions for controlling the light.

Display / Background Image / Paste - puts on the board the background image copied in Windows clipboard.

Display / Background Image / Invert-Rotate - applies to the background image Cartesian transformations or a free rotation for adjusting.

Display / Background Image / Frame - allows you to frame the background image, like a rectangle to be plotted from two diagonal points.

Display / Background Image / Position - allows you to locate the background image on the board.

Display / Background Image / Dimension - allows the scale of the background image to be varied as a percentage of the original. Galaad must prepare the background image before its display, which can be a long operation. Consequently, it is recommended not to use big original images if they have to be reduced here.

Display / Background Image / Scale from two points - adjusts the size of the background image from two points whose real distance is known. The rest of the drawing remains unchanged; only the background image is resized. This is probably the best way to resize the background image accurately on your board.

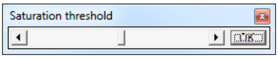
Display / Background Image / Black & white - displays the background image in black & white if the original is in colours.

Display / Background Image / Brightness-contrast - increases or decreases the overall light of the background image from two trackbars.

Display / Background Image / Original - returns to the background image as it was opened the first time, resetting all changes that have been applied since then.

Display / Background Image / Negative - inverts the light of the pixels of the background image, so it appears in negative.

Display / Background Image / Saturated - displays the background image in black or white with no intermediate grey levels, according to a settable saturation threshold.



Display / Background Image / Engrave - does the same as the function *"Design / 3D mesh / Background image"* that has been explained in detail a few pages back – so to which please refer.

Display / Background Image / Change engraving - does the same as the function *"Design / 3D mesh / Change / Change mesh from background image"* for resetting the interval between lines or the range of engraving depths.

Display / Main comment - determines whether a comment produced with the *"File / Comment"* command is visible.

Display / Visual dimensions / Enabled - displays visual dimensions that have been added to the current drawing. See also the function *"Parameters / Visual dimensions"*.

Display / Visual dimensions / Relative to red/blue cross - changes the displayed positions so they become relative to the fixed cross, red or blue. Other dimensions (distances, angles, *etc.*) are not concerned.

Display / Visual dimensions / Recalculate all - recalculates all dynamic visual dimensions that have been added to the current drawing.

Display / Visual dimensions / Delete all - deletes all visual dimensions that have been added to the current drawing.

Display / Deselect all - removes all selections (objects, segments, points), and even selections that do not have the focus. *"Remove"* does not mean *"delete"*. The shortcut key is ⇧ Esc. Changing active layers does the same, except if you use the rapid data palette.

Display / Set pointer - allows you to set the size and format for the crosshairs of the drawing cursor.

Display / Calculator - opens Windows calculator, provided that it is in the system folder. Remember that, in any dialogue box of Galaad, it is always possible to enter a maths formula instead of a numerical value.

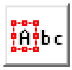
Display / Refresh - rebuilds the current drawing. The hotkey for this function is Ctrl *Space*.

"*Text*" menu

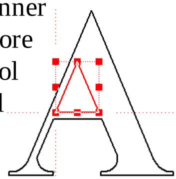
Text / Font style - opens the dialogue box for the text style settings (font, size, spacing, shading, *etc*.). *Note:* the automatic kerning controls the spacing between two adjacent letters taking their shapes and possible overlapping into account (an "A" followed by a "V", or an "L" followed by a "T", *etc*.).

Galaad provides about 50 fonts that are dedicated to engraving, including centreline fonts, for example a capital 'I' being a single line and not a rectangle. These font names have "G - " header. Other fonts with "TT - " header are those installed by Windows and we are not going to give details about how they work. By definition, a TrueType (or OpenType) glyph consists of one or several closed paths, since the filling is done on the screen or the printer. We are not supposed to get an infinitely thin line. It is possible to find TrueType fonts with centreline characters (*i.e.* without thickness) but they are nonetheless made of closed paths, meaning that for example a capital 'I' is made of two zigzag segments that overlap one another. To avoid useless machining paths, you may in this case enable the option *"Delete superimposed contour lines (thin fonts)"*. Then Galaad will keep useful lines only.

When engraving from the bottom in a transparent material, letters can be filled at a lesser depth than the contours, say 50 %, for a nice visual effect.



Text / Arrange letters - allows the letters to be selected individually and positioned independently from the rest of the text. Movement from one letter to the next is done with the help of the (tab) key or with the (shift) key to reverse direction. This command can also be found in the *"Text"* series of drawing icons.



Text / Select internal islands - automatically selects the inner paths of a previously selected text (which, of course is no more selected). This function can be used for applying a reversed tool compensation to internal islands, though the tool compensation function is supposed to do so for text objects, or making an internal hatching for an extruded text.

Text / Edit - places a selected block of text into an edit window for allowing you to both edit the text and change the alignment. This command can also be found in the *"Text"* series of drawing icons, or from a double-click on a block of text.

Text / Rebuild - recreates the selected text using its parameters. If some letters have been moved or modified, unfortunately changes are lost.

Text / Convert to polylines - converts the selected text into simple polylines and curves based on its current style. This is a one-way process and any text specific properties (style, alignment and the text itself) will no longer be available. Objects still remain associated.

Text / Split - only works after a single letter has been selected using *"Text / Arrange letters"* and splits the text into two distinct blocks at the currently selected letter.

Text / Merge - merges independent blocks of selected text into a single one. The style and paragraph settings are those of the first block of text found. A carriage-return / line-feed is inserted between each merged block.

Text / Writing direction - opens a box in which you can change the primary and secondary direction of texts. This allows some of the more exotic languages to be used in their correct orientation. This should not be confused with the direction of the paragraph itself, which can be rotated by 90° when entering the text.

Text / Increment numbers - increases the first number found in the selected block of text by one unit.

Text / Decrement numbers - decreases the first number found in the selected block of text by one unit.

Text / Column of numbers - creates a column of numbers in either descending or ascending order.

Text / Change all - partner of the previous function, replaces all selected texts by a new one to be written, in one single operation.

Text / Text file - imports text from an external text file (*.TXT). The text retains its text format and can be edited by the above commands.

❑ *"Parameters"* menu

Parameters / Workspace / General settings **-** allows you to set the general workspace parameters. See the chapter on *"Workspace parameters"* for details.

Parameters / Workspace / Password - allows you to set up and change your password in the usual manner. If by chance you have forgotten it, then please note that the password is stored in the PASSWORD.TXT file in Galaad installation directory.

Parameters / Workspace / Restrictions / Ignore - allows you to override all restrictions that have been previously imposed on the workspace, and therefore come back to full mode (restriction set is not lost).

Parameters / Workspace / Restrictions / Change - lets you define your own set of restrictions, *i.e.* customise the Galaad workspace by removing the icons and the menu functions that you do not need. See the chapter on *"Workspace parameters"*.

Parameters / Workspace / Restrictions / Level 1-3 - directly loads a set of predefined restrictions, saved in the files LEVEL-N.CUS that have been installed with Galaad but that can still be modified.

Parameters / Workspace / Restrictions / Open - directly loads a set of restrictions that has been previously defined and saved. The files that correspond to restriction sets are located in the *"CONFIG"* subdirectory.

Parameters / Colours - gives access to display colours of all visual components of Galaad and its machining module.

Parameters / Function keys - lets you customise the keyboard by assigning a menu command to each of the function keys from F1 to F12, alone or associated with Ctrl. Just click on the corresponding line then select a command in the top menus. Please note that the F10 key is generally reserved by Windows and is not customisable.

Parameters / Visual dimensions - sets the style of all visual dimensions that have been added to the drawing.

Parameters / Network / User - allows you to configure the current settings of the workstation on the local network. See the chapter on *“Using a network”*.

Parameters / Network / Workgroup - determines which workgroup the current installation of Galaad is attached to. This does not modify the Galaad installation but changes the access to central folders on the disk of the master workstation. See the chapter on *“Using a network”*.

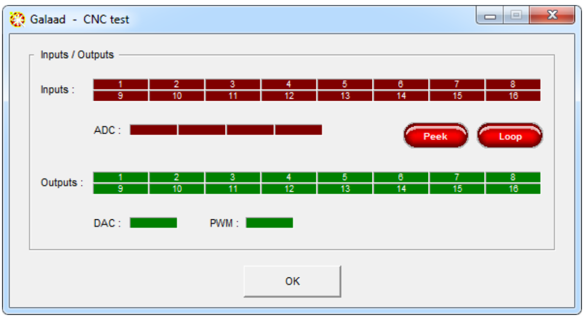
Parameters / Network / Upgrade workspace - if the PC is either an independent workstation or a networked *"Master"*, then the current environmental parameters will be saved, or if the PC is a secondary *"Slave"*, new environmental parameters will be loaded from the master workstation. See the chapter on *"Using a network"*.

Parameters / Tools - allows you to define the detailed parameters of the cutters within their own tool library. See the chapter on *"Toolpaths"*.

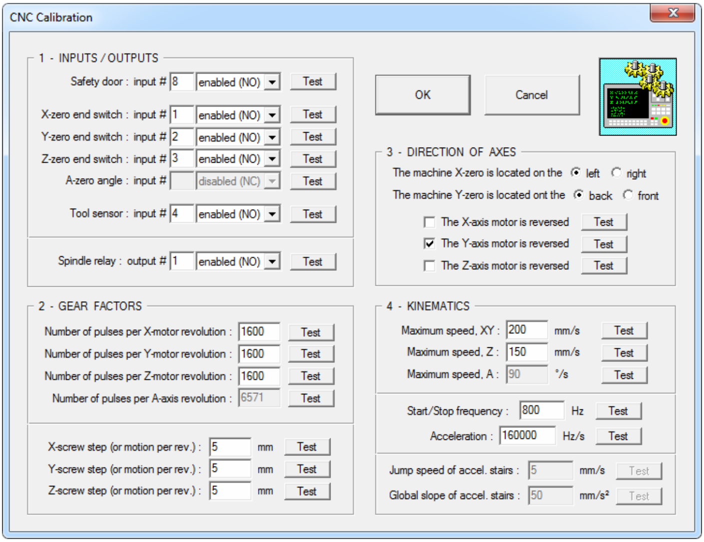
Parameters / Machine / Basic data - allows you access to the basic data for the CNC. See the chapter on *"Machine parameters"*.

Parameters / Machine / Full data - allows you access to the full details of the CNC. See the chapter on *"Machine parameters"*.

Parameters / Machine / I-O test - opens a reduced dialogue window that helps you control the inputs/outputs of the machine, without any motion. This function is useful for checking the electrical connections. See also the *"Machining / Manual control"* command.



Parameters / Machine / Calibration - opens a window for a semi-automatic calibration of your machine (inputs, gear factors, orientation, kinematics). This function is dedicated to those who have built their own machine or have integrated an electronic control that is different from the authenticated one. Galaad will save the calibrated parameters.



The first thing to be done is find the **inputs** that correspond to the safety cover, the switches for the machine zero and the tool sensor. If you have none of these, then this first phase is useless. The method for finding which input a device is connected to, and using which polarity, is quite simple: click on the corresponding *"Test"* button and trigger the contact manually (for the safety cover and the tool sensor, this is easy, but it can become a bit tricky for zero-end switches). A sub-window displays and keeps the input that has changed. The safety cover must be opened when you click on *"Test"*. The tool sensor must be released, and the axes must be away from their zero point (contacts must not be triggered).

Finding axis **gear factors** requires an access to each motor for seeeing it turn. Stepper motors generally have increments of 1.8°, *i.e.* 200 full steps per revolution. Warning: this is not always the case. Anyway, you must find the right value that corresponds to one motor revolution, taking into account the microstepping or the encoder increments for a servodrive. Try typical values like 400, 800, 1600, 3200, *etc.* by clicking every time on *"Test"* button until the motor makes exactly one revolution. Once your motors have the correct number of pulses per revolution, then you must measure very accurately the linear motion that corresponds to one revolution, if possible using a comparator or a calliper. At every new click on the *"Test"* button, the motor makes one revolution.

The **direction of axes** defines the geometry of your machine. You must indicate the location of the machine zero, *i.e.* the position of the zero-end switches for axes X and Y (zero-end switch for Z is always at the top). The direction of axes can be tested from a movement where you indicate whether the axis has moved in one direction or its opposite. Galaad will eventually determine an inversion to be applied to the motor.

At last, the **kinematics** settings, probably the most complicated to tune up, suppose that other calibrations have already been made. The first thing to set up is the maximum speed the axes can afford. Both XY axes work together, so it is the slower of the two that will be kept. When you click on *"Test"* button, the machine starts moving with a weak acceleration until it reaches the indicated speed. Galaad asks you if a motor stalled. If you click on *"Yes"*, it will keep the last speed that was validated. If you click on *"No"*, it retries with a higher speed until the motor stalls. It is up to you to check and answer: Galaad cannot know if a motor stalled due to a too fast top speed.

Once the maximum speed has been set, you must calibrate the Start/Stop frequency. This frequency corresponds to the instant speed that is applied to the motor when still, or reciprocally the final speed just before stopping. A too low Start/Stop frequency renders accelerations unnecessarily long. On the other hand, a too high frequency can block a motor which then cannot catch its acceleration ramp, or simply makes it knock at starting and stopping phases. The test consists of making the motor turn instantly at the corresponding frequency, for each axis. Galaad asks you if one motor has knocked (started too abruptly) or stalled. If you click on *"Yes"*, it returns to the previous value, otherwise it retries with a doubled value.

Corollary of the Start/Stop frequency: the acceleration ramps. If a motor accelerates too quickly, it may stall and get blocked, or just lose steps (or raise a drag-error if it is a servodrive). If it accelerates too slowly, all movements will be unnecessarily long. The acceleration of an axis must be strong enough but with no risks of blocking or stalling. This test does the same as the calibration of the Start/Stop fequency, with the acceleration slope becoming steeper and steeper. Please note that the Start/Stop frequency and the acceleration must correspond to the most critical axis, namely the one that stalls at the lowest value. These settings correspond to inactive movements, when driving manually or when the tool is above the material in an automatic milling process.

Though they are the most difficult to set up, the two last kinematical parameters are important since they determine how the machine will react when milling with the tool down in the workpiece. The test method is the same as the one used for the Start/Stop frequency and the acceleration. Please refer to the chapter dedicated to the machine parameter for understanding exactly what they are related to.

Parameters / Machine / External driver - allows you to use a milling or CNC upload programme that is not part of Galaad. See the end of chapter *"Machine parameters"*.

Parameters / Machine / Save - saves the whole set of machine parameters to a .CNC file that you can recover using the next command.

Parameters / Machine / Open - loads the whole set of machine parameters that have been saved using the previous command.

Parameters / Machine / Restore point / Create - lets you save a restore point with the machine parameters, the various settings for positions and calibrations, the cutting databases (for plasma, laser, water jet, *etc.*) and the tool library. This function saves you spending hours searching for a parameter that you might have modified and cannot remember where it is, which is changing the behaviour of the machining. The files are located in the *"RESTORE"* subdirectory of Galaad installation directory of the hard disk. Their names are composed of the recording date & time, plus an optional title. You may request the automatic creating of a restore point at regular intervals. See the next function.

Parameters / Machine / Restore point / Recover - loads the parameter files of a restore point that has been previously saved. You can select which parameters you wish to restore, and a few buttons help you manage the list of restore points.

Parameters / Machine / Post-processor - defines a file format for export, for a generic numerical controller or for an external driver that uses a particular language unknown to Galaad. All usual syntaxes are possible, provided that they are based on text format with numerical values which are decimal or hexadecimal. Some frameworks are predefined for the most common syntaxes, on which you may embroider your own format. For a description of the syntax set-up, please refer to the end of chapter *"Machine parameters"*.

Parameters / Auto save - allows you to set the period of time between every auto-save of the current drawing. The auto-save corresponds to an automatic command *"File / Save"*. This supposes that the file has already got a name. Otherwise, it is the whole workspace that is saved.

Parameters / New file defaults - allows you to define the default “new file” settings (workpiece dimensions, tool, depth, feedrate).

Parameters / Rapid depth changes - allows you to define the change in depth when using the Ctrl + or Ctrl - hotkeys on the selected objects, for adjusting coordinates.

Parameters / Automatic feedrates - opens the dialogue box which allows you to change the settings of the feeding speeds that are calculated automatically. These automatic feedrates mainly concern beginners who are not yet familiar with milling speeds, to avoid starting their career on a big heap of broken cutters. They are calculated from the material hardness family (this classification of the material is used only for this purpose), the tool diameter and profile, the machining depth and other secondary parameters. In the metal milling industry, a feedrate is normally calculated from the cutter rotation speed, its number of teeth and the standard feed length per tooth. For wood or plastic materials, the situation is different. The feed length per tooth does not apply to an engraving bit or a ball-end cutter. The automatic feedrate calculated by Galaad leads to a large under-estimation: the aim being not to optimise the tool cycle but avoiding the breakage of tools because the speed was too high. Once you have become familiar with the speeds, probably you will forget the automatic feedrates, and precisely for saving time.

The automatic speed calculation can be adapted for each family of material. You can multiply the result for feeding faster or slower.

Parameters / Save parameters - lets you save the current set of parameters (environment, restrictions, machine, tools) under a given name. The memorised sets of parameters are located in the *"CONFIG"* subdirectory of

Galaad installation directory on the hard disk.

Parameters / Load parameters - lets you fully or partially load a set of parameters that has been previously saved using the above function.

Parameters / Quick transfer / Send parameters - uploads the full set of parameters currently in use, to a removable disk or network disk, from which they can be transferred to another workstation.

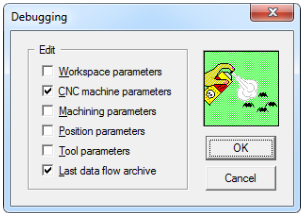
Parameters / Quick transfer / Receive parameters - downloads all or part of a set of parameters previously saved onto disk using the above function.

❑ *"Help"* menu

Help / User's manual - displays this manual in PDF format, which supposes that a document viewer such as Adobe reader has been installed on your computer.

Help / Remote intervention - calls up the integrated function for distant connection on your screen. This presumes that your computer is connected to the Internet. When calling this function, a new pop-up window indicates an ID number and a password. You must communicate these to your contact person, by telephone or e-mail, so that he or she can connect to your PC and see your screen to help you out.

Help / Debugging - creates a text file containing some or all of the parameters currently in use, together with those of the last machining operation, to assist technical support staff in case of a software or hardware malfunction. The two default options are the machine parameters and the last data flow archive. They are sufficient for tracking any problems with the machine. The resulting file GALAAD.BUG, located in Galaad installation directory, must be sent by e-mail to the technical support. *Very important:* **unless you have generated this debugging file, do not restart a manual drive or an I/O test which would erase and overwrite the last archived data flow between Galaad and the machine.** All exchanges are archived in the text file GALAAD.XON, formatted as time-tags, commands sent, feedback codes, and many automatic comments that ease the hunt for bugs. This file is concatenated with the other parameter files depending on the selected options, to generate the final file GALAAD.BUG from this debugging function.



Help / Licence key - tells the software which model of key is connected to the PC, if any.

Help / Temporary licence - allows you to enter a licence code which is valid for one month with the key unplugged. This licence can be renewed three consecutive times maximum, and the current installation of Galaad must have detected the key at least once during the past three months. The only purpose of this command is to allow the replacement of a lost or damaged key and keep on working while awaiting its return. **This function**

cannot unblock the user's licence of a Galaad installation that has no key. Please note that the identification number is not fixed and changes every time a new temporary licence is validated. Unless you have been prompted to do so, do not plug the licence key in again once this temporary licence has been activated: if the key is detected, even partially, the temporary licence will immediately be disabled.

Help / About - displays version and copyright information. This dialogue box also provides information about your installed version for potential updates, and indicates the status of your licence. The button *"Download update"* helps you update your Galaad installation without manipulating files. This obviously supposes that your computer is connected to the Internet.

❑ Shortcut icons

A promise made should mean a promise kept, so: here is the list of available icons in the top bar, with their connections to the menu commands.

 **File / New**

 **File / Open template**

 **File / Material dimensions**

 **File / Open**

 **File / Import**

 **File / Gallery**

 **File / Quick transfer / Receive drawing**

 **File / Save**

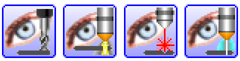
 **File / Save as**

 **File / Export**

 **File / Quick transfer / Send drawing**

 **File / Print**

 **Machining / Standard machining**

 **Machining / Standard simulation**

 **Machining / Multisided machining on 4 axes**

 **Machining / Manual control**

Edit / Undo

Edit / Redo

Edit / Repeat

Edit / Restart

Edit / Copy

Edit / Cut

Edit / Paste

Edit / Paste second-last copy

Edit / Duplicate / Add one real copy

Edit / Duplicate / In line

Edit / Duplicate / In matrix

Machining / Toolpath / Connect objects

Machining / Toolpath / Disconnect objects

Machining / Toolpath / Define as start point / Red point

Machining / Toolpath / Define as start point / Point to click

Design / Object / Weld

Design / Object / Split

Design / Incise

Design / Polyline / Delete parts where clicking

Design / Segment / Delete segments to be clicked

Design / Point / Delete points to be clicked

Design / Point / Insert a point on a trace

Design / Arc / Replace circles by points

Design / Arc / Replace points by circles

Design / Arc / Set the diameter of all selected arcs

Machining / Sequence / Set as first

Machining / Sequence / Set as last

Machining / Sequence / Mouse select successively

Machining / Tool compensation / Define toolpath

Machining / Tool compensation / Remove toolpath

Machining / Tool compensation / Support bridges / Add

Machining / Tool compensation / Set start point

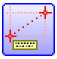
Machining / Tool compensation / Feed in point / Horizontal

Machining / Tool compensation / Feed out point / Horizontal

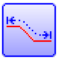
Machining / Tool compensation / Feed in & out points

Design / Hatching

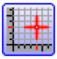
Design / Pocketing cycle



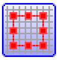
Design / Object / Distance between points



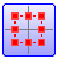
Design / Object / Trace length



Design / Magnetic grid / Set



Design / Magnetic grid / Snap and size to grid



Design / Align & centre

Text / Font style

Edit / Libraries / TrueType Symbols

Display / Quad view

Display / 3D view

Display / Final rendering / Trace

Display / Trace / Identifiers / *(on/off)*

12

0 1 1 0 0

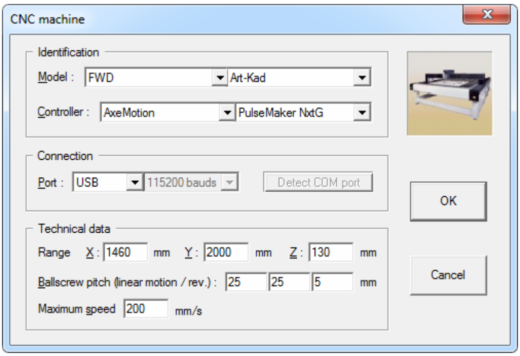
MACHINE PARAMETERS

Main parameters

Setting up the parameters of your CNC machine is one of the most important parts of configuring Galaad and must be done with the greatest care. Generally, this only needs to be done once and it is unusual to have to reset these values once the machine is performing correctly. Using incorrect values is not supposed to harm your machine, except with very special mechanical configurations. At worst, it will be happy to sulk in a corner and totally ignore you, at best it will respond a little and make the odd excusable silly move. Nevertheless, do not leave your fingers in the way, one never knows.

In order to help you configure your very first machine, Galaad provides a reduced dialogue box giving access to the main parameters only, leaving the finer details until later. It goes without saying that this only applies to full standard machines that can be found in the list of models known to Galaad. Use the menu command *"Parameters / Machine / Basic data"* to find them.

Without doubt, the most important parameter is the **model** of the machine. Look through the double-list of known machines until you find yours and select it. For the most part, all other parameters will be loaded automatically. Please note that machines are sometimes re-badged by importers.

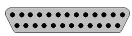


The type of **controller** is usually governed by the model of the machine. Moving directly to the frame at the bottom of this window, the physical characteristics of the machine can be set. These depend on the machine model but bear in mind that they can, and do, change with time, so check the technical data supplied with your machine (if any…), and especially the **pitch of the ballscrews**. If this value is set incorrectly, then the machine will work almost normally, but movements will not be of the correct distance and speed. Too small, a figure increases the values of the coordinates sent to the machine and consequently the size of all movements. Conversely, too large, a value reduces the size of all movements. If your machine is not actually fitted with ballscrews, nor even basic screws, then this value corresponds to the linear distance that each axis moves for one revolution of its driver motor.

The communication between Galaad and the machine generally takes place through a cable, which takes commands from Galaad to the numerical controller and returns data or acknowledgements back to the software. Most machines are connected to a **USB port** or a **COM port** (RS-232 serial port). In this case, you plug one end of the cable into the machine or its controller and the other end into the corresponding port on your computer. Few PCs are still equipped with COM ports nowadays, but these can still be present on desktop computers. If your machine requires a serial port and your computer has none, then you must add a USB/Serial converter module that will set up a virtual COM port to the PC (generally numbered beyond COM2). Real or virtual, the **available COM ports listed by Galaad have an asterisk after their numbers**. If you do not know which COM port number has been assigned by Windows to your USB/Serial converter, these asterisks can help you out. The button *"Detect COM port"* can also help you find the machine by trying all available ports until it receives a valid feedback. But this is not fully garanteed.

If your machine is driven throughout a COM port, then the communication protocol includes a **transmission baudrate**. We generally consider that this speed has been set on the machine controller and the driving software must adapt itself to the machine rather than the contrary. Do not change the baudrate needlessly. It is not really with a faster communication that you will save time.

Important: **if no machine is connected to your PC, just set the port to *"None"*.**

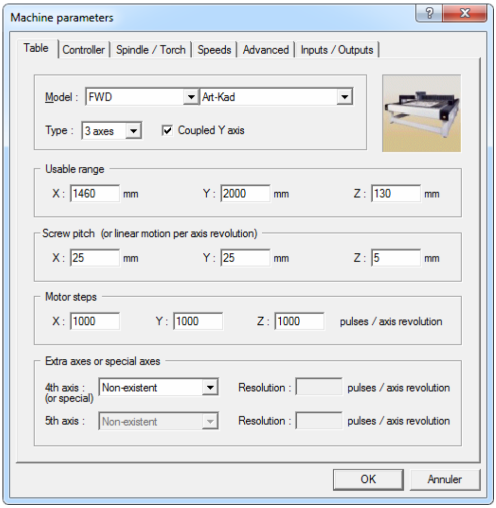
Remember also that **Galaad does not produce Step/Dir signals on the LPT parallel port** for directly driving a power stage. You must interface an electronic device that is able to produce these signals. 

The next pages refer to the complete set of parameters for the machine, which you can access very logically through *"Parameters / Machine / Full data"*.

◻ *"Table"* page

Do not proceed beyond here until you have read the information given above which will not be covered again. Check that the necessary connections have been made to your ports for both the machine and the licence dongle. The basic machine parameters have been set *a fortiori* to the full parameters described below.

As mentioned above, the complete machine parameters are accessed via the menu command *"Parameters / Machine / Full data"*. This command opens a multi-page dialogue box, which contains all the information relating to the machine, numerical controller and spindle. You will find here the basic parameters, as already discussed above, buried in a mass of more complex data. Do not panic, at least not yet, as we will go into the details of this and the penny should drop. Moving on from the machine model, set in the basic parameters, if yours is not in the displayed list, the simplest is select *"Not listed"*. Anyway, if its controller is compatible with Galaad, then all hope is not lost, it will probably work, but you will have to set up all the parameters of this beautiful stranger.



We now have to provide some additional information, this time the actual number of axes. Most machines in the list have **3 axes** XYZ, but some specific configurations can have only **2 axes**, and others have a 4th turning axis A or a double master-slave axis, or even a 5th axis B. All this is described in detail a further on.

The **usable range** represents the maximum travel of each axis, the combination of which defines the maximum working envelope of the machine. We have seen in the basic parameters the **ballscrew pitch** and the consequences of using a wrong value. If your machine has axes fitted with differently pitched ballscrews, then they must be set here. The **motor steps** specify the number of pulses that the CNC must produce for a stepper motor to make one revolution, or in the case of a servo motor the number of pulses returned from the encoder. Here again, if a value is wrong, so will be the scaling factor on the machine, generally by a factor 2 or 4 or 8, *etc*.

The bottom frame is for defining a fourth axis (and even a fifth), if exists. A fourth axis can have one of three possible configurations in Galaad: rotary axis parallel to the X axis (A axis) for cylindrical milling; rotary axis parallel to the Z axis (C axis) for use with a knife blade; and finally a simple slave linear axis, allowing there to be two actuators on any of the three main axes. It can also be an E-axis for extruding on a 3D printer, usable only with the Kay module. A fifth axis can be an orientation axis for a milling head that is tilted by the fourth axis (5-axis milling) or a second slave axis for an existing linear axis. The Galaad drawing module and its associated machining module can use only 3 axes out of 4, namely an XYZ Cartesian machining or an XAZ cylindrical machining. However, the manual control lets you jog all existing axes, even for a workpiece origin before a 3-axis process. **Only Kay and Kynon modules can drive up to 5 axes in an automatic cycle**.

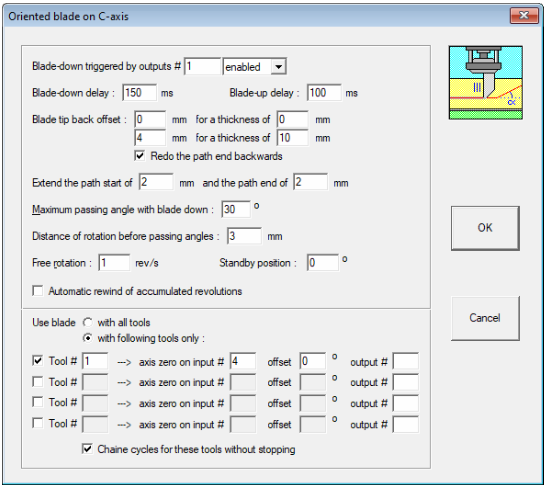
❑ Rotary milling on A axis

The current standard wants rotary axes A, B and C turn around parallels to respectively X, Y and Z axes. So an A axis is parallel to X axis and permits volume machinings by rotating the horizontal plane, which allows milling faces that are inaccessible in a classic 3-axis process, or making a simple cylindrical wrapping of the flat drawing. Please refer to functions *"Machining / Cylindrical machining on 4-axes"* or *"Machining / Multi-sided machining on 4-axes"* for more information about how a Galaad drawing can be wrapped around a cylinder or mill different lateral sides of a 3D workpiece. In addition, the Gawain turning module can use a 4-axis mill fitted with a rotary A axis for milling a drawn profile, including threadings (faces and external threadings only, since the cutter cannot reach the inside).

The configuration for a rotary A axis, or *"Lathe A axis"* is very basic: just indicate the number of motor increments per chuck revolution (and not per motor revolution). The maximum rotation speed in °/s must be set in the *"Speeds"* tab.

Steerable blade on C axis

Galaad can drive a motor that rotates a swivelling blade so it is always heading in the horizontal cutting direction. This motor can be either the third axis (if there is no Z axis on the machine) or the fourth axis. Selecting *"Orientation C axis"* as extra axis (or re-selecting it if already active) opens a sub-dialogue box that gives access to the specific parameters of this rotary blade. When the machine has no Z axis, then we can suppose that the blade is driven up and down by an electromagnet or an actuator with mechanical stops that set the high and low positions. The first parameter indicates which output triggers this electromagnet or electrovalve. The corresponding delays must be given for lowering and raising, to prevent Galaad from starting an early horizontal motion whilst the blade is not yet in low position.



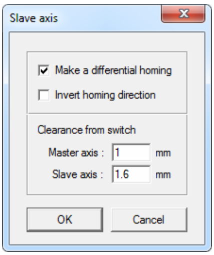
Since a rotary blade generally cannot turn for changing its apex when in low position in the material, the maximum angle allowed without rising must be indicated. If the rotation for the next vector is greater than this angle, then the cutting motion stops, the blade is retracted up, rotated towards the new cutting direction, then lowered and the motion is resumed. If the rotation is lesser than or equal to this angle, then the blade is rotated in low position. Hence, passing the angle is done with a cutting motion, then a rotation, then a new cutting motion. For keeping a continuous motion in the horizontal path, the blade can begin to rotate before having reached the turning point, consequently cheating a bit on the orientation apex near the angle. This eases chaining the cutting vectors with no stop points because Galaad will interpolate small vectors just before and after the turning point, that integrate the blade rotation. In this case, you are requested to give the distance from the turning point, from which the blade can begin to rotate. If the actual path vector is shorter, then the blade will be in permanent rotation. Warning: **the**

blade must rotate about its cutting point. There are no XY compensations if the blade is off-centre.

The rotation speed (when blade is up) can be set here, and also the reference angle (parking position). The trigonometrical 0° corresponds to the direction X+, or 3 o'clock in the aviation lingo, the direction Y+ corresponding to 90°. **When launching the process, the blade must be set to this reference angle**, and it will return to it when the cycle is completed. If the blade cannot turn infinitely, for example due to wiring or pipes turning with it, then you must enable the rewind option. Anytime the blade is lifted up, it is rotated so it is as close as possible to its reference position. Finally, this blade can be activated for one single tool only in the drawing, other tools corresponding to a classical process without the C axis.

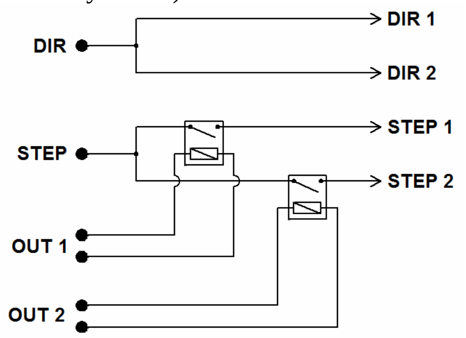
❑ Parallel axes

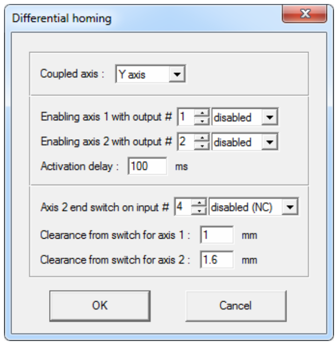
If your machine has a double axis, for example two screws or two cogs at the pillars of its mobile gantry, then several options are available. The simplest is to send the same command signals to both motors. But, in this case, if it happens that one axis drifts away from the other, the gantry will lose its perpendicularity and it will become difficult to reset it manually. This is why **Galaad offers a function for a differential homing which automatically readjusts the squareness of the gantry**. This supposes that the two motors are driveable together and separately. When resetting the machine on



its zero point, Galaad moves together both parallel axes until one of the two end-switches is triggered, then dissociates the axes, resets the master axis alone, resets the slave axis alone, and finally reconnects both axes together. For resetting the gantry accurately perpendicular after a differential homing, either the end-switches are mechanically adjusted, or both clearance values are set in Galaad parameters.

You may use the 4th axis of the controller by setting it as a slave axis X' or Y', but then you will lose a channel which, most of the time, will do exactly the same as the master axis channel. If you have only 3 axes on your controller, or if you need the 4th channel for a rotary axis, then the option ***"Coupled axis"***, at the top of *"Table"* page of the machine's full data, offers a simple solution that requires a small electronic circuit: by making the *Step* signal (or *Clock* or *Pulse*, it is the same), duplicated for the two parallel axes, pass through a couple of mini-relays or electronic switches, each being driven by an on/off output of the controller, Galaad may cut the movement of one motor and not its twin. The *Dir* signal, also duplicated, does not need to be cut. The result is the same, except that, instead of reserving an axis channel of the controller, this reserves two outputs. Anyway Galaad can deal with both solutions. In any case, a differential homing requires two end-switches connected to two different inputs, unless you use the same relay system for also cutting the link between both contacts and one single input (here, the switches should be parallel-mounted if normally opened, or serial-mounted if normally closed).





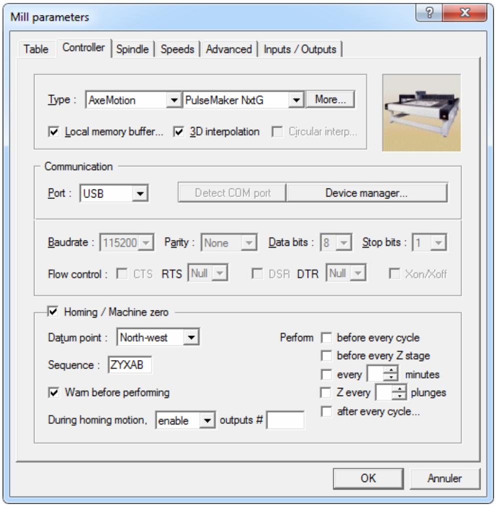
It may happen that the *Step* and *Dir* outputs of the numerical controller are unable to provide enough current each for two opto-couplers on the power stages, or that these signals cannot be parallelised downstream. A line driver integrated circuit may then become necessary.

❑ Additional axes

As above mentioned, only Kay and Kynon modules can manage up to 5 axes in an automatic cycle. The Galaad machining module is limited to 3 axes among 4, *i.e.* a process will be made either on XYZ (flat milling) or on XAZ (wrapped cylindrical milling). 3D machinings that require 4 or 5 axes must be managed by the Kay module, which is described later on in this manual. If you have a rotary tilting head for 5-axis machining, then you must indicate the number of motor increments for a 360° revolution of the head. The 4th and 5th axes must be set as independant axes A and C (or B and C).

❑ *"Controller"* page

The next page allows the CNC to be specified (integrated controller, racked controller or controller card), and its communication system to be set. In addition, the position of the machine zero point (also called the datum, home point or reference point) has to be indicated along with details of how and when a homing should be performed.

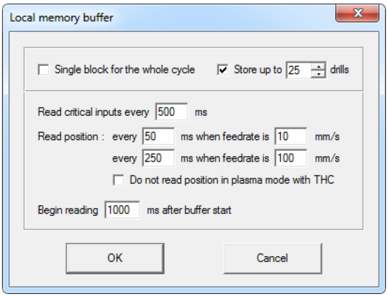


The **type** of the CNC will normally be set based on the model of machine, whose details have already been entered in the previous page. However, some types of unlisted or custom built machines can use a variety of different controllers. In this case, you must specify yourself what type drives your axes.

Many controllers integrate a **local memory buffer** which is able to store vectors and postpone their execution. This function is very important because it avoids delays between each movement and therefore provides a continuous, fast and fluid trajectory. Sending batches of vectors to the controller before it executes them breaks the synchronism between the software and the machine, so the coordinates that are displayed on the screen no longer correspond to the real movements but to the transmission. Then the software is ahead of reality, which suits it well. Using an asynchronous mode may look confusing but it is more efficient, especially when the machining path is made up of curves. Some machines such as Isel have a linear memory, with a transmission time first, the tool remaining in standby, and then an execution time, the software being now in standby. Here, it looks like some time is lost compared to the mode without buffer, which is however not the case since the transmission delay is diluted among the sendings of vectors. On the other hand, on a machine with a ring memory buffer, the transmission occurs while the vectors previously sent are executed, consequently in hidden time. On some controllers (AxeMotion, Soprolec), Galaad can read in real-time the position and, in that case, the monitoring on the screen corresponds to the motion of the axes.

You may choose to disable the local memory buffer and leave the machine directly connected to the software for supervising every single vector. This renders the movements continuously interrupted by the data transmission, and therefore the path becomes jerky. So, if your numerical controller integrates a local memory buffer, then it is always better to use it. On the other hand, if it does not, it is useless and not even recommended to make it believe something else. For sure it will not work any better.

Galaad limits the use of the local memory by re-synchronising with the machine at every tool bottom right. This makes it follow block by block the progress of the cycle and enables resuming a machining at the last tool plunge. However, it is possible to make a unique block for the whole process, provided that the local memory buffer is circular. For simple drills with or without plunge cycles, movements are much simpler and there is no need to re-synchronise at every tool bottom right. Anyway it is possible to store a series of drills in the memory to save this re-synchronisation time. If the numerical controller is able to dialogue with the computer whilst it is executing the content of the buffer, in particular for answering a read command about the status of the inputs or the position, then it is possible to monitor the critical inputs and synchronise the screen with the machine, depending on the chosen granulosity.



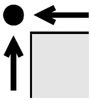
Your CNC may be a very old model that cannot move three axes simultaneously, in which case Galaad offers an alternative of interpolating only the X & Y axes and stepping the Z axis in stages, but you must then uncheck the **XYZ interpolation**. Likewise for the **circular interpolation**, if your machine is not able to produce arcs from a specific single command. These features depend completely on the actual controller and not the mechanics of your machine. Unless you have a specific reason, stick to the default values that are pre-programmed in Galaad.

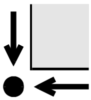
Communication with the machine is via a port to be specified: USB, Ethernet, serial, parallel or internal, the choice is wide but each machine generally offers one single possibility. If your computer is not connected to a machine, select *"**None**"* instead of a port. Galaad will make a simulation of the process with the parameters of the selected machine.

When using an RS232 serial connection, the COM ports that are available, real or virtual, are followed by an asterisk in the drop-down list. If you do not know what COM port number has been assigned by Windows for your USB/Serial converter, then these asterisks may help you out. The button *"Detect COM port"* may also help find the machine by trying all available ports until it gets a valid echo. Likewise, the button *"Device Manager"* will pop up a system window giving access to the list of peripheral devices connected to USB ports, plus the COM and LPT ports for checking their validity. On a serial port, it remains necessary to tune up the **communication protocol**. The parameters in Galaad have to be set so they match those of the CNC, rather than the other way round. Refer to the technical data for your CNC to find the correct settings. Here again, playing about with parameters brings more pain than pleasure.

The **homing** frame helps setting the position of the machine zero point. **This information is very important and determines the overall orientation of your XY axes**. It is assumed that, whatever the machine looks like, the Z axis zero point is always at the top of the travel, *i.e.* with the tool fully retracted. Yes, even in the southern hemisphere. But concerning the X and Y axes, the position of the zero point can be at 4 possible corners. An old and convenient standard wants the X axis to run from left to right and the Y axis from front to back, seen from the operator's point of view, which looks quite clear. Another more recent standard prefers the X axis to be the longest of both. It is doubtless very masculine but not always so clever for operators who might then have X and Y axes inverted compared to what they can see on the screen or what they have drawn. Consequently, it is recommended to define the X axis as the one that moves from left (west) to right (east) and the Y axis as the one that moves from front (south) to back (north). If, seen from the operator, there is an inversion of one of these axes, it is easy to rectify this inversion (see the *"Advanced"* page).

On most machines, the X axis has its zero point at west, namely to the left hand side when seen from the front of the machine. But the Y axis can have its zero point at north (back) or at south (front), depending on the model.





Simply power up your machine and give it a home run to see where the datum point is situated. Warning: the position corresponds to the tool on the table, not the movement of the table itself. If your machine has a mobile table Y, then when the table moves to front, it is in fact the tool which is moving to back, and reciprocally. **You must always consider the virtual motion of the tool on the table, not the real motion of the table**. If your machine has a fixed table and a mobile gantry, then please forget this detail. The parameter for the position of the zero point will also be updated automatically when the machine model is changed, probably not a very common occurrence. **If the axis reset is correctly performed but the manual jogging arrows and the cursor on the screen are inverted, compared to the reality, then probably the position of the machine zero that has been indicated is wrong**. The tick box, immediately below, controls whether or not Galaad should warn you before making a homing reset. For safety reasons, it is always preferable to have this feature enabled, especially when the machine is open and the homing motion is fast.

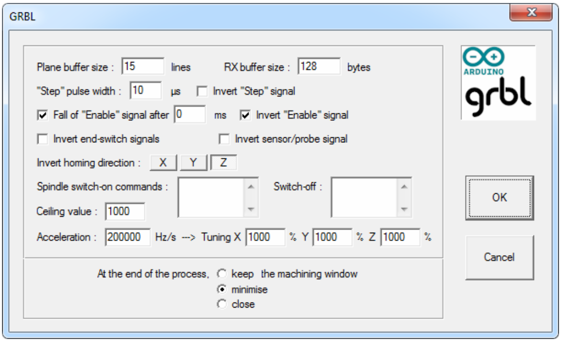
You may want to forget about the machine zero and use only floating origins. In that case, simple untick the option *"Homing / Machine zero"*. We are not going to debate here the pros and cons of working in absolute coordinates with a machine zero point. Just consider that, without a machine zero, you cannot manage fixed positions, therefore no fixed tool sensor, no tool changer, no memorised clamps. In addition, you will have to set the workpiece origin at every new process, even if the position did not vary.

When a machine is fitted with stepper motors, it is sometimes necessary for peace of mind to reset the position of the axes by performing a homing movement. Galaad does this automatically if the machine loses power as it then no longer knows the position of its own axes. However you can choose to home the axes at more frequent intervals if you so wish. In fact, if your machine is really losing steps, perhaps there are more important things you should check, and not only in Galaad.

Before we forget it, a discrete but no less important button More..., situated to the right hand side of the controller type, gives access to special parameters related to the numerical controller. Selecting any particular controller brings up the dialogue box that applies to it. These special parameters are described hereafter for the most common controllers – those which consequently generate the most questions.

❑ Arduino GRBL controllers

Many small machines for amateurs integrate Arduino controllers with GRBL firmware or more or less compatible. More or less because many variations exist on this draft standard, and not all provide the best compatibility. GRBL is a super-set of the ISO G-code for rendering it conversational and sending parameters to the controller when opening the dialogue (the $nn commands).

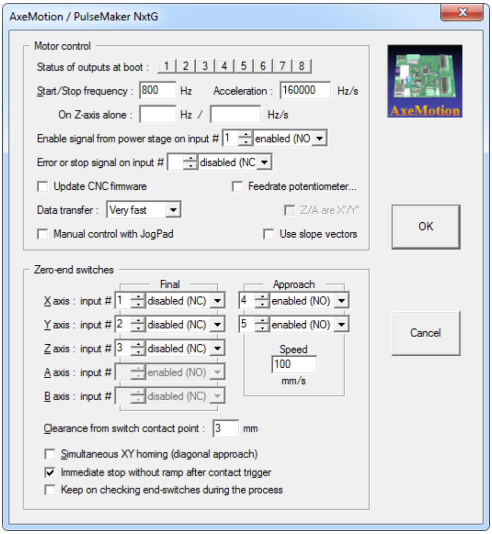


Important: if a file named **GRBL-TOP.TXT** exists in the Galaad installation directory, containing a list of $nn commands with corresponding values, then this file is sent as is and Galaad uses none of its own parameters. Every command line of the file must nevertheless be acknowledged by the controller.

With GRBL, no inputs/ouputs can be addressed directly from an identification number, but only commands for activating the spindle (you can set your own custom commands if necessary), for using the zero end-switches when resetting the axes, and for the tool measurement sensor. These inputs/outputs are assigned by the firmware and only the function is available. Well, when it is. However, you may reverse the polarity of the inputs in these parameters.

When the driving window is closed, the serial communication is cut, which re-initialises most controllers. When opening the next dialogue, the position counters are consequently reset to zero and it is necessary to perform again a homing reset. Otherwise the current position becomes the relative zero. If you wish to avoid this problem, just ask to keep the machining window or minimise it. Calling a new task will wake up the machining module.

AxeMotion controllers



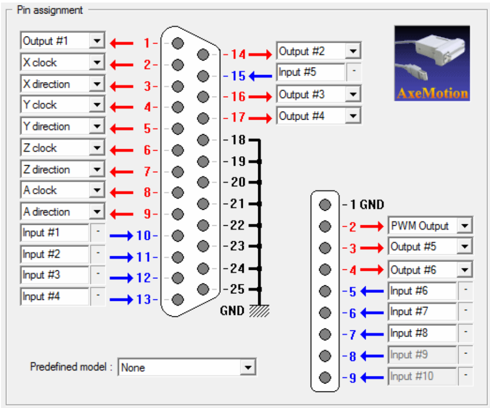
The parameters described here are shared by all AxeMotion cards, including *"PulseBox"* models which are given an extended page for assigning signals on the pins of the DB-25 connector. It is possible to preset the **status of outputs at boot** of the card, even before Galaad is active. As soon as it is powered up, the card will enable them. We will see later on in this chapter the values for the Start/Stop frequency and acceleration, along with kinematical parameters. An input can report if the **power stage** is active, or assigned to the monitoring of some **error signal**, for example a drag error from a servodrive or positive end-switches. If this critical input is triggered, then the numerical controller stops the axes and Galaad aborts the process.

A **speed potentiometer** can be mounted on the analogue input (ADC) of the *"PulseMaker"* cards. This potentiometer instantly slows down the feedrate, to the point where the axes are frozen. This is a very interesting option. The communication on the USB port is normally set to *"Very fast"* for old generation cards, but you may select a slower communication if the machine is working in a noisy environment with important sources of electrical parasites which seem to interfere with the transmission. Finally, if your AxeMotion controller integrates a jogpad on the local I2C port, then you can ask Galaad to manage it.

Here the **homing** reset is managed directly by Galaad at low level, so it is necessary to indicate which inputs are connected to the fast-approach switches (XY near-ends for large tables) and the zero-end switches, with their logical states. Warning: **a *"disabled"* input does not mean that it is not used**; in fact, this corresponds to its polarity when the switch is triggered: **switch**

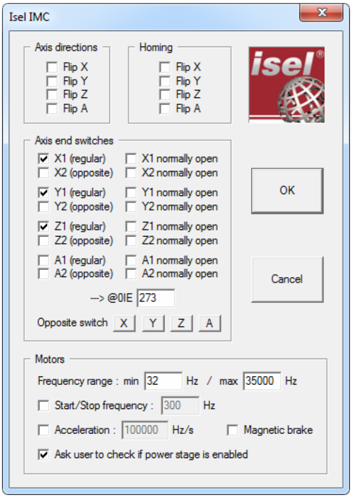
normally opened → *"active (NO)"* input / switch normally closed → *"inactive (NC)"* input. It is possible to add a little offset value from the contact points to set a machine zero that does not tickle the switches. In addition, the home runs for X and Y axes can be driven simultaneously in a diagonal motion for saving time, with final movements one by one. An option can render these inputs critical for the card which will immediately stop the automatic process when an end-switch is triggered. On a well-tuned machine, this should not happen.

For the AxeMotion *"PulseBox"* modules that advantageously replace a machine driving through LPT parallel port, the window is extended on the right hand side and there you may set up the signals that are generated on the DB-25 connector for the power rack and, for older versions, on the auxiliary connector which provides a few more inputs and outputs. Some assignment models for existing racks have been predefined in Galaad. Please note that inverting *"Direction"* and *"Clock"* (or *"Pulse"* or *"Step"*) signals has absolutely no bad consequences on your machine. The axes will simply not move. So you have the right to grope around. Warning, the *"PulseBox"* module gets its power supply from the USB port, so the available current is limited: it cannot provide more than 10 mA per output, with a total of 50 mA for the whole set.



❑ Isel-Automation *"IMC"* controllers

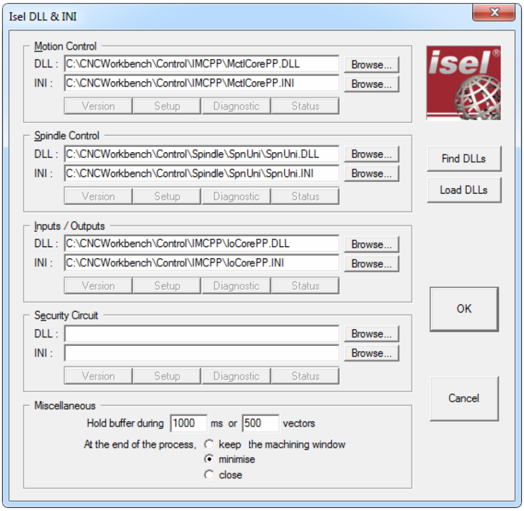
The most recent Isel machines with stepper motors are driven by numerical controller IMC4-M, IML-4 or IMC-M/P. These give access to a few specific parameters, including the possibility to **flip an axis direction** or **flip the direction of the homing** reset (both options are independent). If you select a standard Isel machine that integrates one of these controllers, the inversions will be automatically set. But for a machine with an external rack, you might have to define them yourself. The parameters for the **axis end-switches** are available for using either one of the available switches as zero point, or for changing the working polarity of a contact that is normally opened or normally closed.



The **frequency range** indicates which minimum and maximum values for motion frequency (the increments) can be sent to the motors. It is possible to set values for the **Start/Stop frequency** and the **acceleration** different from those used by default. Finally, if your machine integrates an electromagnetic brake on the Z axis, then you should indicate it here.

These machines or numerical controllers have a *"Power"* button that the operator must activate manually. Checking the status of the power stage can be disabled here, since it may not work on all controllers of these types.

❑ Isel-Automation *"CNC-API"* controllers

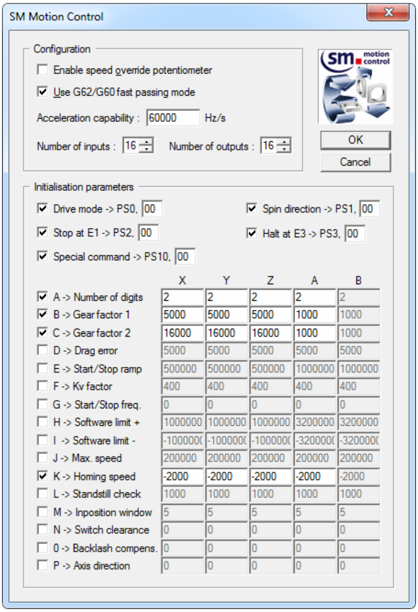


Isel machines with servodrives use dynamic link libraries (DLL) and set-up files (INI) that have been installed on the driving console at the factory. They are driven through these DLLs which consequently must be in place and configured, otherwise the machine control will be buggy. *"Motion Control"* is the only DLL that is theoretically required, for managing the axes, but it is probable that the DLL *"Inputs/Outputs"* will also be necessary, at least for controlling the spindle or the machining tool. The *"Spindle Control"* DLL is useful only for a special spindle with integrated control, not for a standard spindle that can be controlled from a basic binary output or even an analogue output. Likewise, the *"Safety Circuit"* DLL is needed only if the machine integrates a particular safety system. So it is possible to leave those two DLL settings unspecified.

The button *"Find DLLs"* allows you to start a search for the Isel factory installation, grab its parameters and apply them here, avoiding a manual search of the files. Once the DLLs and the INI parameters are validated, it is possible to check them using the button *"Load DLLs"* which will give access to several internal functions reporting the version, initialisation, diagnostic and control. But it is better not to fiddle around with the DLLs and their related INI parameters, which are supposed to have been set at the factory and probably require no operator's action. The purpose of this window is only for seeking on the hard disk the files that have been installed by Isel technicians, and to validate them (the files, not the technicians, no-one is infallible).

For an Isel machine driven through the control DLLs, **the communication port must be set to *"Internal"***.

SM-Motion controllers

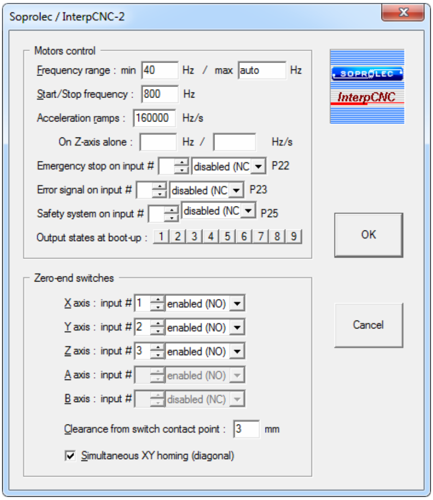


SM-Motion racks or control cards require many parameters that will be used by Galaad only if they have been enabled. If your machine has been configurated at the factory or when installing it in your workshop, then **do not modify them**, at least those related to the most advanced controls, unless you know exactly what you are doing. The whole set of parameters is sent to the machine only if the option *"Send parameters to the CNC at initialisation"* has been enabled in *"Advanced"* page (see later on, same chapter).

If your machine integrates an overspeed potentiometer, then you must enable it here, and also the number of inputs/outputs the controller provides. A maximum of 32 inputs and 32 outputs can be managed by Galaad, which should be sufficient for most applications. The passing mode G62/G60 of the SM-300/400 allows chaining vectors with a good fluidity, with the kinematics calculation. Unless you have big problems when used, it is always better to leave it active. This function is not relevant when the local memory buffer has been disabled.

The initialisation parameters are generally **preset at the factory** and do not require any changes, except parameters A (number of digits after decimal point, which must be 2 for all axes), B & C (gear factor and conversion of distance units in increments), and also the parameter K (homing speed). You should not need to change the others unless you have been requested to do so by the manufacturer or a qualified technician.

❑ Soprolec *"InterpCNC"* controllers



Developed by Soprolec, the InterpCNC cards are well known to professionals or hobbyists who build their own machines. The Interp-CNC-1 card communicates through a classical serial port RS232; the InterpCNC-2 card, able to drive up to 5 axes, communicates through a USB port using a driver DLL that is provided by Soprolec, in charge of the data exchanges with Galaad.

For the **Start/Stop** frequency and the acceleration, please refer to the section dedicated to kinematics, later on in this chapter.

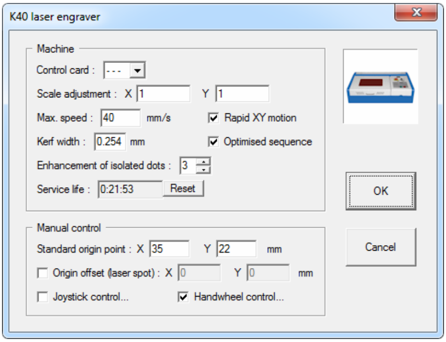
Some inputs may be used for monitoring the **emergency stop**, any **error signal** that feeds back from the power stage or positive end-switches, or a **safety system** such as a protection door. If this input is activated (or inacti-vated depending on your settings), then the CNC card immediately stops the motion and Galaad aborts the machining process. These parameters are all optional.

The **output states at boot-up** define which outputs must be activated when the card is powered on, even before it communicates with the PC.

The InterpCNC cards undertake the **homing** reset themselves, provided that you have indicated the inputs that correspond to the zero-end switches, with their trigger state (normally opened → input set to *"enabled"* / normally closed → input set to *"disabled"*). X and Y runs may be driven simultaneously in a diagonal line to save time, with a final run one axis at a time.

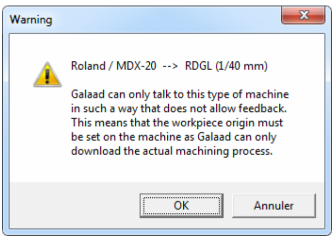
K40 controller

The K40 laser engraver has been interfaced with Galaad for a direct drive. Before going any further, let us mention that **all controller cards fitting the K40 are not compatible**. It may happen that the software cannot drive your machine. Parameters are not many and fairly clear. You can calibrate the position of the standard XY origin point in the northwest corner of the machine, so your workpieces will all be set there, or define an offset out of the workpiece with a laser flash for marking the position.



Generic controllers

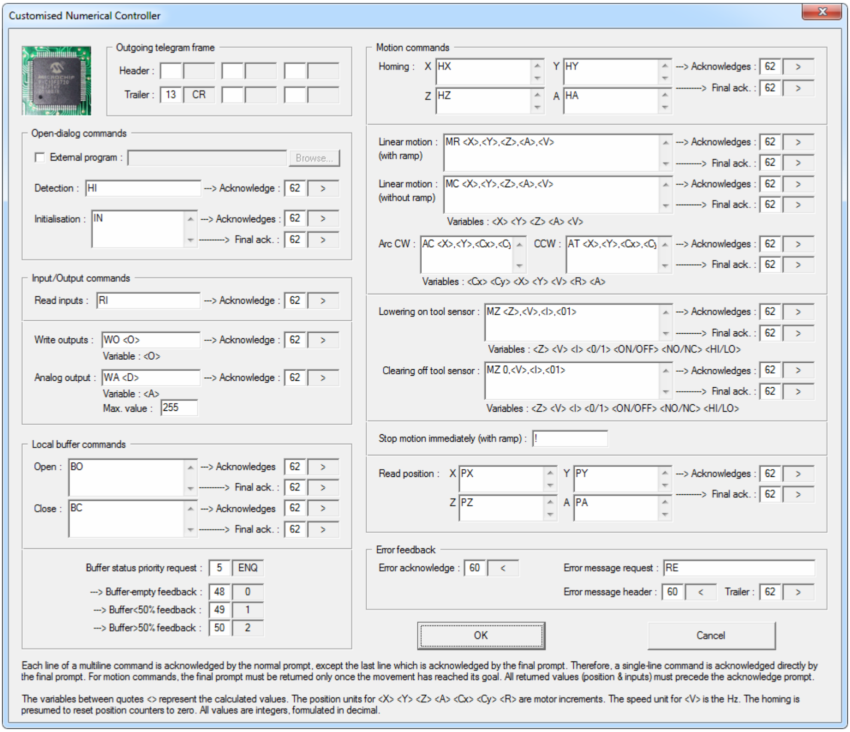
Some machines contain a numerical controller that can only **receive** commands but do not return usable data to Galaad (for example GravoGraph, Roland or Suregrave). However, it remains possible to drive it directly in unidirectional mode for automatic millings. In this case, **Galaad cannot provide manual drive functions** since this requires position feedback and specialised commands. Hence the communication is write-only, from Galaad to the machine.



But if the machine has a control panel with local manual drive functions for a workpiece origin set-up, there is hope yet. Galaad will prompt you to set the origin directly from the machine and will then send the commands that correspond to the milling process through the cable. The format to be used can be chosen among the available command languages, including the customisable post-processor.

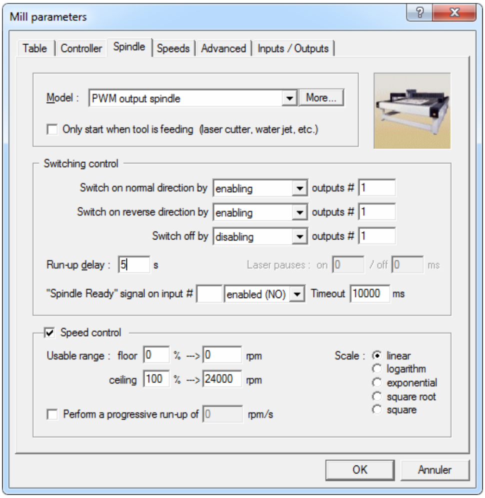
❑ Customised controller

If your machine is fitted with a numerical controller that is not listed in Galaad but uses a command language in text mode, decimal or hexadecimal, on a COM or LPT port, then it is possible to set up yourself the syntax of these commands by choosing the model *"Generic / Customised"*.



The telegrams sent to the machine can be framed by heading and trailing bytes, indicated in decimal ANSI code. The commands must be given in text, with variables between < > which will be substituted by the corresponding values when executing. The acknowledgements returned by the machine for every command line are defined as a unique byte, given in decimal ANSI code. The undefined commands will not be used. If you have a CNC on serial or parallel port that works with a decimal text language, then you should be fine, provided that you know the available commands of the controller and their actual syntax.

❑ *"Spindle"* page



The next page of machine parameters lets you define the model of the spindle that is used for milling, if any, and its control system. On a machine that has been provided by a manufacturer, generally the spindle will be the standard model. But it may be that yours is a particular model, with an alternative control that you must then set up here. If in doubt, always select the standard spindle that is associated with the machine. Please note that changing the machine model or the numerical controller model in the previous pages of parameters automatically resets the spindle to the standard model. Indeed, it is more than likely that if you change the machine, then the spindle will be changed too.

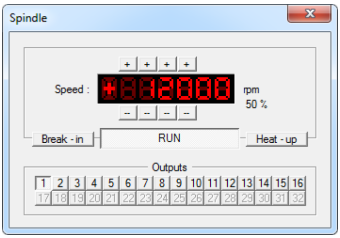
If your spindle is actually a laser cutter, plasma cutter, water-jet cutter or something of the same kind such as a liquid dispensing nose, then you should **only start it when the tool is feeding** and switch it off as soon as the tool should be lifted-up. The drawing interface, in particular the green icon at left hand side for accessing the tool, depth and feedrate, can adapt itself consequently. Please note that a full chapter is dedicated to the **plasma torch** cutting systems, later on in this manual.

Several predefined spindle models are controlled either by one or several binary outputs of the numerical controller, or by an analogue or PWM output, or through a separate communication port. You can define here its parameters, *i.e.* choose an output and a binary state for switching on and off. It is possible to **indicate several different outputs for one single command**, for example the output number 12-2 for the outputs 12 and 2 triggered one after the other. You can indicate up to four successive numbers for one digital switching. **This remains valid for all binary outputs**.

A delay can be set for the **spindle run-up**, that is the time it takes for it to reach full speed. Galaad likes to use time efficiently and will make good use of this delay by moving the spindle above the entry point of the first object to be machined, avoiding sitting idly by. But it will then be happy to wait patiently, if necessary, for the spindle to finish accelerating before plunging into the material. If the spindle is activated during the feed phase only (laser cutter, water jet or liquid dispenser), this delay occurs between cutter switch-on and feed move. In such a case, the two **laser pause** values define an extra delay after enabling (*i.e.* before starting feed motion) and before disabling (*i.e.* after ending motion). The difference with the above mentioned run-up parameter is that these two delays are not used if the spindle is a classical one that remains continuously active during the whole process. In this case they are greyed. If the spindle control system returns a signal indicating that the spindle is now ready, then you can connect that signal to an input of the controller and indicate it here. Galaad will wait until this input is triggered for considering that the spindle has reached the chosen speed. Finally, the software can manage an auxiliary engraving laser in addition to the spindle, associated with a particular tool, with its own parameters.

❑ Tachometric spindles (with speed control)

Galaad can drive the rotation speed of a tachometric spindle if it is managed by a control system through an analogue data channel or similar.

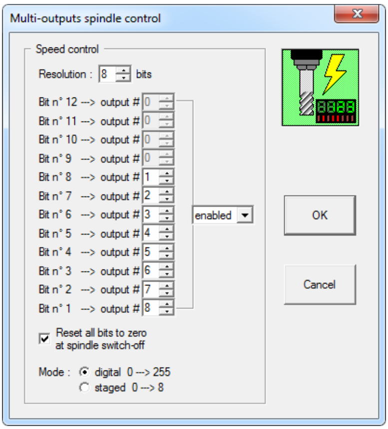


Several special parameters must be set in this case.

The spindle rotation speed is normally predefined for each tool. However, when starting the spindle in manual or automatic mode, a dedicated window will pop up to allow you adjust the speed. The control system for the speed (analogue, PWM or multi-outputs) and the spin direction does not override the binary outputs that have been set for power on/off. Consequently these outputs remain valid with such spindles. If you do not wish to use them because they are redundant with the rotation control system, then just delete the corresponding output numbers or set them to zero.

In the case of a spindle with a control system for the rotation speed, you must indicate in percentage the floor and ceiling values that can be used for the signal that will determine the RPM. The floor value corresponds to the motor starting to turn with a minimum torque. Galaad will consider that the speed range begins with that value which is not always 0%. The ceiling value corresponds to the motor at full speed, which can be reached before 100% on a variator with no feedback loop. Of course you must indicate both speeds that correspond to these values, and the range between them can be non-linear. Furthermore, the software can even drive a progressive run-up to avoid overloading the electrical power supply. This run-up is linear and can be set in RPM-per-second. It is of course subtracted from the run-up lapse.

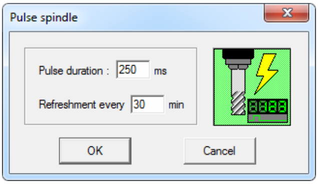
A PWM (Pulse Width Modulation) signal uses one single output bit that works in 0/1 mode, but this square signal is time-sliced according to a given frequency. The percentage of the high state compared to the total period gives a relative width for the pulse, from 0% (no high states) to 100% (full-width high state). A small integrator circuit can convert the pulses into an analogue signal, doing so with only one output bit.



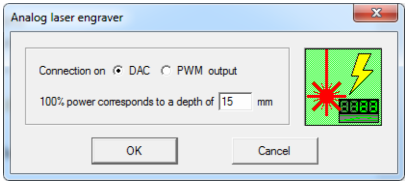
If your numerical controller has neither an analogue nor a PWM output but provides a wide range of binary outputs, then it is still possible to control the spindle rotation speed by switching several outputs that are linked to an external D/A converter. You can set the number of bits that will drive the DAC (1 minimum, but well, do not expect too much of an analogue signal in that case; and 12 maximum, which should give a fine resolution). The corresponding outputs do not need to follow each other in a sequenced list. You must assign an output number for every weighted bit that defines the analogue signal value. Galaad will then calculate the pattern of all outputs that represent the binary digits of the digital value, which then produces the required rotation speed. Instead of a D/A converter, you may also use a staged integrator device.

Special spindles

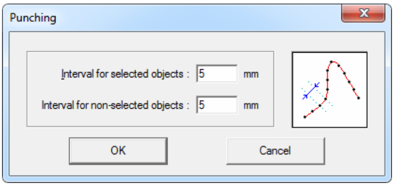
Some spindles that require particular parameters can be used with Galaad. For example, a **flip/flop pulse** spindle is started by sending a simple activa-



tion signal on one (or several) output(s) and needs to be rearmed periodically by sending again the same signal, the stop command being managed by triggering another output. That rearming command will be sent only when the tool is up so the feed process cannot be affected. You must set the renewing delay consequently. If you select an **analogue laser head engraver**, Galaad will modulate the power according to



the depth of the object, *i.e.* the Z coordinates, including along a 3D path, the maximum power of the laser (or any other system for machining) being set to a given depth that will not be exceeded. The modulation goes through the DAC or PWM output. Finally, it is possible to control a **punching or dot-drilling head** that uses the normal outputs for switching on/off the

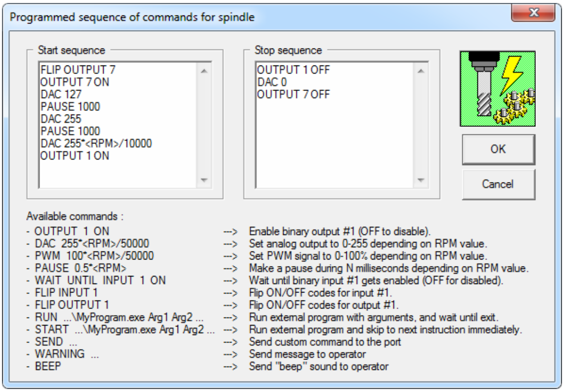


spindle, so with no specific parameters, except that this head will be activated at regular intervals along the paths, these intervals being set before every new machining process. Obviously the activation is intermittent.

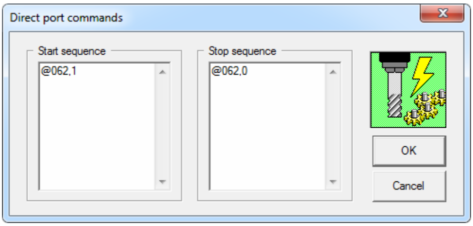
Spindle command scripts

Perhaps your spindle or your application is demanding and cannot be happy with a basic on/off command, even accompanied by several pauses and rotation speed controls throughout an analogue or PWM channel. In this case, not all is lost, you may still programme yourself the sequence of commands for switching your device on and off. Here, the spindle model should be set to *"**Programmed sequence** of commands"*.

The dialogue box for a specific set-up allows you to write the two control scripts in a very simple programming language. The software will then interpret your own instructions one by one when starting or stopping the spindle. The list of available commands is displayed at the bottom of the window. Most of them correspond to the inputs and outputs of the machine. But **you may also call up a set of external programmes** and pass them function arguments, for example the rotation speed which is available as a variable labelled <RPM>, which will be replaced by the actual speed in revolutions-per-minute, in decimal text (a maths formula can be used around this variable). You may also indicate a direct command to be sent on the communication port inside this script, using the command SEND (see also hereafter).

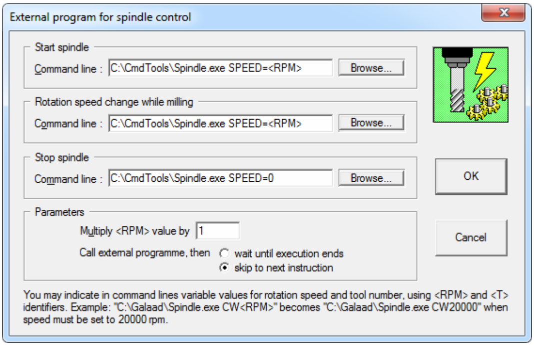


Talking about commands to be sent directly to the machine, it is possible to select as spindle model the *"Direct commands on port"*. In this case, you do not write a sequence of commands that have been defined and interpreted by Galaad, in a script language that will call the corresponding functions of the machine. Here you must totally **write in the machine command language every telegram to be sent**, knowing that Galaad will not try to understand what you are doing and will just expect a standard acknowledgement for every command it sends. These commands must be written as texts, there are no variables and of course the machine is supposed to appreciate the sense and subtleties of what it receives. You are at the helm.



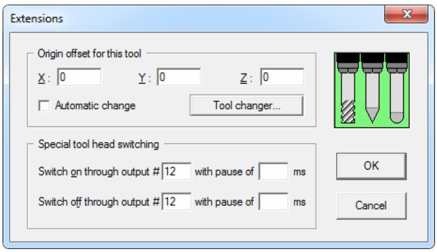
External control programme

Now, if your spindle is totally unknown to Galaad with absolutely no control channels available from the numerical controller, but was fortunately provided with its own driver programme for Windows (even in a DOS-like console mode), it should still be possible to establish the link between Galaad and the spindle through this programme.



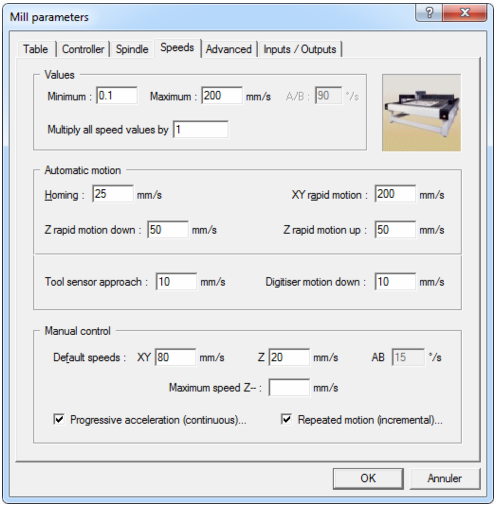
The three spindle commands available are switch-on, change rotation speed, and switch-off. For each one, you must indicate an external programme with its own arguments (of course it can be the same executable programme). Galaad will replace the variable arguments <RPM> and <T> by the actual rotation speed and the tool number, written in decimal text in the command line. **The programme called cannot access the communication port** with the machine, since it is already opened and reserved by Galaad. Using an external programme supposes that the spindle is driven by another channel that this programme is able to manage. Galaad makes these calls and immediately continues. It is up to you to deal with their execution.

Important: it is possible to define, for each tool, using the button *"More"* of its parameters, some special outputs for the spindle when this tool in particular is called. These **special outputs depending on the tool** replace the default outputs in the *"Control"* frame of the spindle parameters. If you encounter a communication problem using one tool and not the others, perhaps you should check these extensions.



"*Speeds*" page

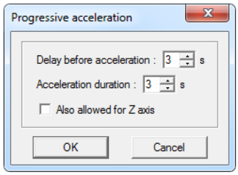
The next page of the machine parameters brings us to the set-up of the working speeds. These are generally set once for all and are not linked to any particular job. They affect neither the feedrates, that you have defined in your drawing, nor the plunge or drilling speed, that you have set in the tool library.



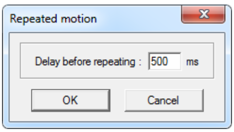
Your controller is capable of driving your machine over a wide range of speeds, but that range is not infinite. All the same, limits have to be imposed for mechanical and safety reasons and it is necessary to specify a **minimum** and a **maximum speed**, so that Galaad can ensure that they are never exceeded. Also a maximum angular speed must be set if your machine is fitted with rotary axes. If you do not know your machine limits, then let us say that it is rare that a feeding speed is less than 0.1 mm/s (6 mm/mn) or moves an axis at more than 100 mm/s (6000 mm/mn) but some large tables can move faster, and even sometimes much faster. These limits can usually be found in the technical information provided with your machine, but not always. In some cases, travelling too fast can result in a loss of the current position (at least with stepper motors in an open loop) and consequent damages on the machined workpiece and the cutter. So please do not blame Galaad immediately when you have tried to push the limits. If the speeds that are sent to the machine are incorrect, you can always change them overall by adjusting a **post-multiplier**, which is normally set to 1.

The central frame on this page sets the speeds of the various **automatic moves** that occur during the working cycle when the tool is not actually cutting the material. The **homing** speeds are quite simply the speeds, axis by axis, at which the machine will move to reset its absolute position on the zero point. For the sake of the limit switches and the mechanical stoppers, please do not use too fast a speed. The **XY rapid motion** is the horizontal travelling that takes place when the tool is retracted to a safe height above the work (the rapid plane) and moving between machining operations. The **Z rapid motion down** is the one that takes place from the rapid plane down to a point just above the surface of the workpiece. The tool then slows down to a speed that depends on the tool being used, before entering the material. Therefore this speed, together with the feedrate, is set depending on your work and the cutter being used. The **Z rapid motion up** covers all movements where the cutter is retracting away from the material. Finally, there are the **tool sensor approach** and the **digitiser lowering** speeds which, as their names suggest, cover the speed at which the cutter approaches the tool sensor for measuring its length during the workpiece origin procedure on Z axis, and the speed at which a digitising probe approaches the workpiece, if they are fitted to your machine.

The two values for **manual control** set the initial positions of the speed control sliders that will be displayed in the machine control windows. Pre-setting here avoids moving speed sliders anytime you open a manual drive window, whatever the mode, manual jogging or workpiece origin setting.



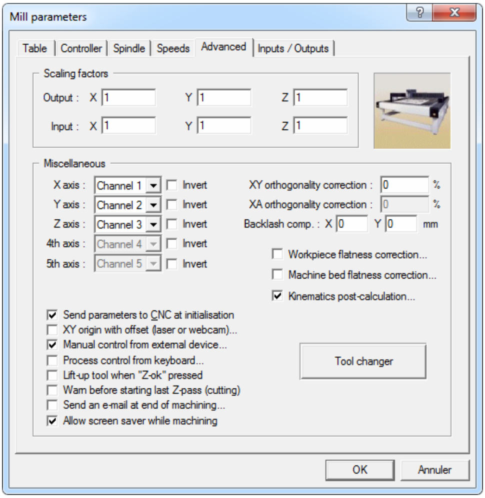
If your numerical controller offers a speed override function (real-time control of the speed), then Galaad may **accelerate motion** progressively when an axis is moved manually. This can be useful to undertake both long movements and accurate approaches without having to change the speed slider all the time. Ticking this progressive acceleration checkbox pops up a small dialogue box that allows you to set the delay before acceleration (start speed remaining the slider speed) and the run-up delay (motion accelerates slowly until speed is doubled). Another possibility: the **repeated motion in incremental mode** (when the manual jogging is not continuous and unlimited) lets you reproduce



the same movement after a brief delay if the motion key is maintained pressed. A small beep is emitted for punctuating the movement. Remember that, whatever the speed sliders display, you can still **move axes manually at slow speed** using the right mouse button, by pressing the Ctrl key on the keyboard or by pressing the button #2 on the joystick (default setting).

❑ *"Advanced"* page

Now we come to the spare room that contains a hotchpotch of machine related parameters that do not have anywhere else to go, which does not make studying them any easier. It is now recommended that you get a compass before venturing alone into this dark hole. But there again, that is the purpose of this manual, so follow the guide and stick to the path.

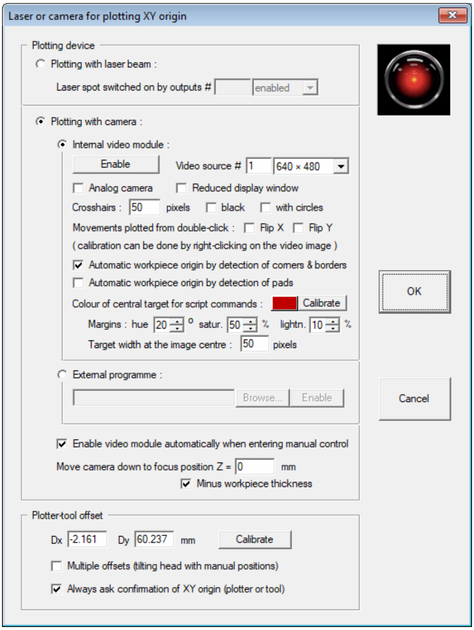


Right at the top of this window are the **scaling factors,** whose values will increase or reduce the magnitude of all distance commands sent to the controller or returned from it, for each axis. We have already seen similar factors for the speeds, but these work in one direction, Galaad → machine, because these values do not have to be returned as only the software can change them. These scaling factors are linked very closely to the pitch of the ballscrews and the resolution of the motors, as set in the *"Table"* page, and ultimately all distance values are rounded to the nearest motor step (or half-step, quarter-step, *etc.*). Unless you want to correct a micro-error, all scaling factors should be left at value 1.

Descending further into the cave, you will find on the shelf at the left hand side the features to **reassign the axes**, for example for inverting the X or Y axes if the machine orientation seen from the operator's point of view does not match the reality (then set X on channel 2 and Y on channel 1). You may also **invert** the direction of an axis. Warning: the purpose of inverting an axis is not to change the position of the machine zero, but just to make the motor turn in the correct direction. For making the screen match what the machine movements, you must first set the position of the **machine zero** in the *"Controller"* page (see a few pages before in this manual).

Many available options in this page display a text followed by suspension dots. This means that, if you enable the option, a dialogue box will pop up for setting the sub-parameters. If the option is already active, then you must double-click or simply press Ctrl while clicking on it. **If you validate a sub-window, you must also validate the parent window**.

If your numerical controller receives **initialisation parameters**, you can decide whether these parameters must be sent or not. Do not disable this option unless your machine is fitted with a dashboard giving access to internal settings.

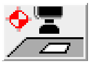


The **XY origin with offset** allows you to validate the workpiece origin on XY axes (using green buttons) at the current position of a pointing device instead of the tool itself, for example a laser spot for a cutting table with a plasma torch, or a video camera for a printed circuit engraver. Please see the chapter *"Advanced milling functions"*, section *"Plotting the origin with a video camera"* for more information about the use of a webcam. Together with Galaad is provided a small visualisation module which, unfortunately, will not work with all webcams, but you can replace this with another video display programme and simply specify its command line. Laser spot or webcam, the aim must be accurately parallel to the Z axis, and special care must be taken to the calibration of the **plotter-tool offset**, otherwise you will have wrong workpiece origins. Here a button gives access to the calibration procedure, which consists of these four stages: marking with the tool end a position on the workpiece or directly the machine bed, activating the spindle to drill a small hole there; validating this tool position with the corresponding green button; moving the XY axes so that the laser spot or video cross accurately match the centre of the hole; and finally validating the plotter position. Galaad will memorise the XY offset in its parameters.

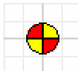
But let's get back to the webcam. Using the video window, you can double-click on a given point in the image to send the machine to it. Distances are calibrated using a ruler placed horizontally in the field of view, with the zero point well positioned under the aiming reticule. Right-clicking on a distance scale indicates the actual distance. The software can then convert the pixels into mm.

The webcam must be perfectly Cartesian, *i.e.* the aiming reticle must correspond exactly to the X and Y axes. The quality of its mechanical assembly is therefore important. And so is the lighting.

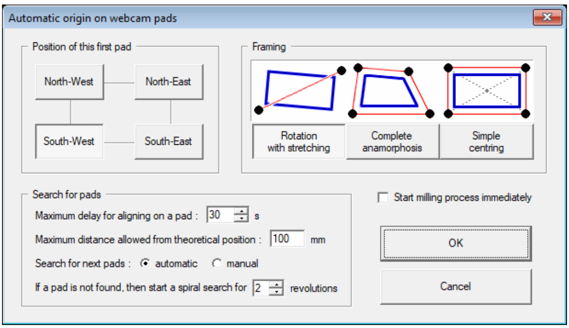
Galaad can use the camera to perform automatic workpiece origin by edge detection. If this option is active, a specific button appears in the workpiece



origin window. Simply place a corner of the workpiece in the field of view and click on it. The software detects it, moves the pointer over it and validates the origin. If the workpiece origin is detected using test patterns (light or dark discs on an inverted background), Galaad adds a *"Machining / Webcam reference points"* function. You must have selected one or two points, or circles, or crosses, or a rectangle to use its four corners as a reference. The selected objects are replaced by targets on the drawing board, which can be seen in the machining window view. The same button above will launch a test pattern detection procedure



with different options. A diagonal by detecting two test patterns on the part,

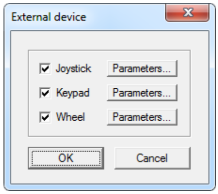


adjusting the orientation and stretching the lines to be machined; a complete anamorphosis with lines stretched so that the targets on the drawing match the test patterns; or simple centering. A number of search parameters are available.

The webcam can also be used to detect the presence of a colour in the area at the centre of the visual field, typically to check the presence or absence of a tool with an automatic changer, in order to avoid an attempt to pick up an empty tool (not too serious) or to put it away in a slot that is already occupied (more annoying). Simply place a specific coloured background under this slot.

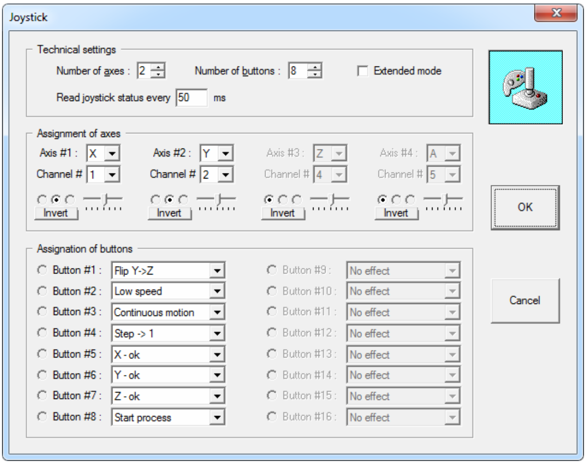
A CAMCHECK command is available in the tool changer scripts, stopping execution if the expected colour is not the right one. The colorimetric margins are adjustable, but it is best to have a colour that is clearly distinct from the tool holder cone.

❑ Manual jogging from an external device



If you wish to jog your axes in manual mode from an external device (a joystick or gamepad, a numerical keypad, an encoder wheel), which is an excellent idea, then you must tell Galaad that it must read the data coming in from that device or keyboard. Of course you may activate several devices all at once.

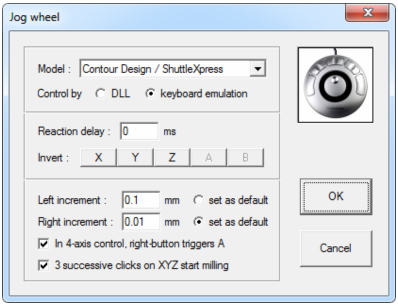
If you have a **joystick** or a **gamepad**, both being equivalent seen from Windows, then Galaad is supposed to react to your pushings on the stick. The device must have at least two axes for jogging the machine. Analogue models are suitable: the small sliders help you calibrate accurately the thresholds that trigger the motion of each axis. The buttons available on the joystick can also be assigned functions to be chosen in the drop-down lists.



You may also use an **external keypad** (or more simply the main keyboard) and assign a function to each key. The set-up is very basic: click on a button, free or already occupied, then press a key on the keyboard or the keypad, then select in the drop-down list the function to be associated. The manual drive will enable this function when you press the programmed key, provided that it is usable at that moment. The option for the **process control**

from keyboard offers a similar window with fewer possibilities. The functions for supervising the machining cycle are not many since, at that particular moment, while Galaad is working hard, the operator is away at the canteen.

For an accurate approach of a workpiece origine, the **encoder wheel** is an interesting device. Remember that the mouse wheel moves the last axis that



has been jogged by an increment of 0.01 mm (if the resolution is sufficient), which is a basic but no less efficient handwheel. Galaad also manages some control devices that have been developed for the video editing but can be adapted successfully to the manual control of a CNC machine. The very first function of a handwheel consists of moving the axes by small increments, for completing the approach to a workpiece origin. The movement direction is linked to the wheel rotation direction, which it is possible to reverse for each axis. The number of buttons varies a lot from one handwheel to another (one single for the *PowerMate*, fifteen for the *ShuttlePro2*). The functions associated with the buttons are not settable. Generally, these handwheels provide a keyboard emulation and every action virtually corresponds to a key press. The usable models are as follows:

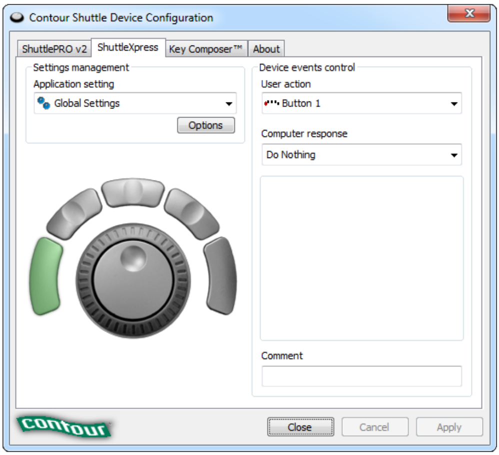
- *Contour Design / ShuttleXpress*, efficient and cheap, fitted with a handwheel at its centre for managing the incremental movements of the active axis, with a fast motion ring to left or to right for the continuous move. If your numerical controller integrates a real-time control function for the speed (AxeMotion, SM-Motion, TechLF), then the movement will be modulated according to the position of the ring. The two buttons beside the centre wheel allow you to choose the increment (their values are settable here). The three buttons at top select the active axis among XYZ. The position of the active axis at the bottom of the screen is framed by a white rectangle to avoid

misunderstandings. A double-press on one of these buttons validates the workpiece origin as if one of the green buttons *"X/Y/Z - ok"* was clicked. With a speed-controlled spindle, the handwheel and the lateral buttons also adjust the rotation speed, the three buttons validating the window. When machining, these three buttons correspond to the step mode, the pause and the stop. The ring then corresponds to the overspeed and the wheel to the spindle rotation speed.

- *Contour Design / ShuttlePro* is the big sister of the previous one, much more richly endowed in buttons. The functions for the wheel, the ring and the lateral buttons are identical to the *ShuttleXpress*, but the five buttons above control five axes XYZAB. The four buttons at the top call the command buttons of the manual drive, and the four buttons at the bottom emulate the Space, Tab, Esc and Enter keys, also usable in the dialogue boxes. The installation procedure for these handwheels is described hereafter.

Installation of a *ShuttleXpress* handwheel

When the *ShuttleXpress* or the *ShuttlePro* handwheel is plugged to the PC and its driver installed, an icon appears at the bottom right of the screen in the System Tray. Click on it and call the function *"Open Control Panel"*. The window shown here is opened. Now click on button *"Options"*, then on *"Import settings"*. Then look for a file named SHUTTLEXPRESS-LAN.PREF in the hard disk directory where Galaad has been installed. Click again on the button *"Options"*, then on *"Change target application"*. Now look for the file LANCELOT.EXE, again in the disk directory where Galaad has been installed. Then close this window *"Contour Shuttle Device Configuration"*. For Kay and Kynon modules, if you are using them, repeat the operation with the files SHUTTLEXPRESS-KAY/KYN.PREF associated to programmes KAY/KYNON.EXE that have been installed.



To finish looking at the various options, it is possible to automatically **bottom right tool when Z-ok pressed**. After you have moved the tool down so it touches the workpiece top surface or the machine bed, and you click on the green button *"Z - ok"* for validating that new Z origin, then the tool will be lifted up to the new retraction point, which is the workpiece top surface minus the retraction height. This eases the next operations on axes because you do not have to lift Z up manually, for example to move XY horizontally towards a new origin point.

The option for **warning before starting last cutting Z stage** induces a machining pause, tool up, when it is going to proceed with the last Z stage that will free the cut parts. This is not an invitation to put your fingers in the machine workspace. Better use the support bridges in the drawing, located among tool compensation functions.

After the machining cycle, you can request **sending an e-mail**, and even at different phases of the process. The technical data for accessing the mail server must be indicated in the sub-window.

At last, for monitoring from afar what happens on the computer, it is possible not to **allow the screen saver while machining**, temporarily. The settings of your Windows screen saver are not affected by this option, but will simply be disabled as long as the machining cycle is in progress, and consequently that screen saver will not be called. Once the end-of-cycle message has been validated, the screen saver is restored as it was.

❑ Mechanical corrections

To compensate for a mechanical error caused by the X and Y axes not being accurately at a right angle to each other, you can use the **XY orthogonal correction** factor, after measuring the error. The following example explains how: use the machine to cut out a large rectangle from any old cheap material, using a cylindrical cutter. Take a set-square and test the corners,

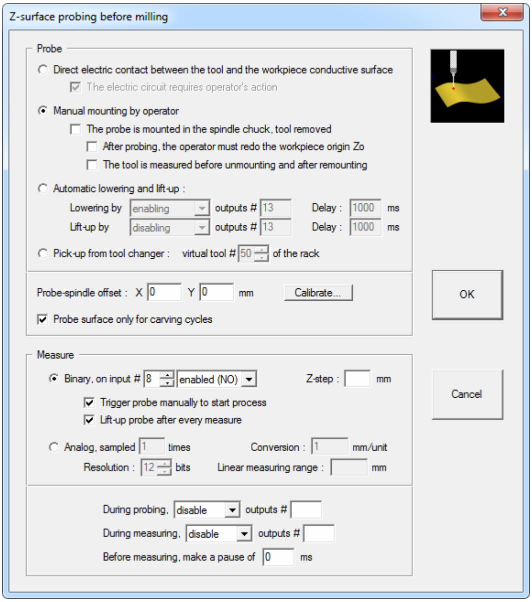
looking for a gap, or just daylight, between the square and the material. If you see a gap, then it is because the X and Y axes on your machine are not strictly perpendicular. In other words, it machines a parallelogram where you expected a rectangle. Before grabbing a very big and vengeful hammer, please consider the possibility that Galaad might be able to cure the problem using some alternative medicine.

Accurately measure the **gap G, between the square and the material**, as far from the corner as possible, and also the **height H, of the rectangle at that point**. It does not matter if you measure H along the board or the square unless the gap is gigantic, in which case your legal adviser should perhaps write to the manufacturer. Bring G back to a percentage using the little prorating rule of three $G' = G \times 100 / H$ and simply enter the resulting value G' into the text box marked "*XY orthogonality correction*". Now machine again the same rectangle. If the problem has become worse, then the correction has been made in the wrong direction. Simply reverse the sign of the correction and make it negative, *i.e.* enter **-** G' instead of G'. This time, it should be better. An alternative method to check for any error involves measuring the two diagonals of the rectangle that you have just cut. We will spare you the maths lesson, as it is slightly more complicated and best left for devotees of applied trigonometry, we salute them all.

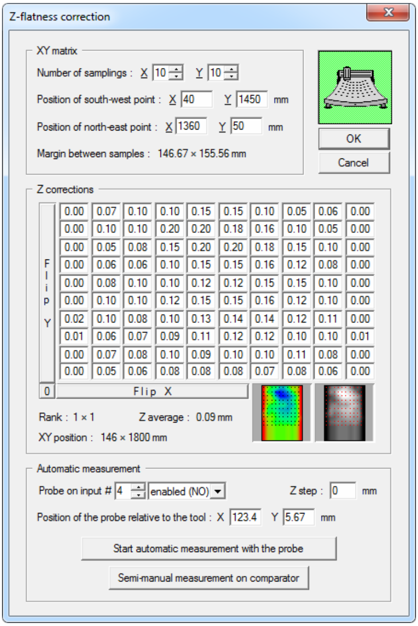
The **XA orthogonal correction** does exactly the same as above for rectifying the error of alignment between the linear X axis and the rotary A axis.

Another great classic of machine tools, the **backlash compensation** for X and Y axes is possible with Galaad, though that type of correction does not make much sense in numerical control since the axis remaining still is not blocked by a brake and will therefore be free to move inside its mechanical clearance. Let us also argue that this function does not mix well with the rapid and fluid machinings of curves where the kinematics calculation helps to achieve good chaining dynamics for the series of motion vectors. The dynamics will be irretrievably broken by the instant offsets that are needed by the backlash compensation when an axis is reversing its direction of movement, which is somewhat frequent. If your machine requires backlash compensation, then it is up to you to make a quantum choice between speed and positioning accuracy, knowing that it is always better to fit the machine with quality transmissions. The clearances to be corrected on X and Y axes are given in absolute values, and there is no backlash compensation possible for the Z axis. In addition to the above-mentioned limits, the spindle weight and the plunge pressure render it even less useful.

The **workpiece flatness correction** is an advanced function of Galaad that allows you to make engravings at constant depth even if the workpiece top surface is not perfectly flat or when its thickness varies, for example a printed circuit made of epoxy. Before machining, Galaad can carry out a set of Z measures with a probe for mapping a topographic matrix of the surface and then apply field tracking corrections. Many options are available at the top of the sub-dialogue box for managing the probe. The simplest device can be a basic electrical contact between the tool and the workpiece, for example a crocodile clip on the tool, connected to one of the input plugs, and any type of contact with the workpiece, connected to the other plug. When the tool touches the top surface, then the circuit is closed and the input is triggered. This is perfect for engraving a copper plate or any other electrical conductor. Of course you must make sure that the circuit cannot be interrupted by something else like losing the clip or the wire connected to the surface, in which case both the tool and the plate will need replacements. A sub-option asks you if this probing phase requires an intervention from the operator, for example to avoid starting the spindle while the clip is still locked on the tool, which could result in a big tangle. Another simple probe can be an accurate mechanical switch mounted beside the spindle on the Z axis, or even inside the chuck instead of the tool. In this case, Galaad must know, to avoid performing operations with the tool that the sensor might not appreciate at all. You may also use a retractable sensor with plunge-down and bottom right commands under software control through an on/off output. This allows you to chain the probing and the machining phases immediately with no operator's intervention requested. Finally, the sensor can also be a virtual tool in the rack of the tool changer, consequently mounted in the spindle, but automatically.



If the probing sensor is not mounted in the chuck at the location of the engraving tool, then the sensor-spindle offset must be calibrated, so Galaad can probe at the right locations. It is also possible to avoid probing when there are only cutting paths and no engraving jobs. The measurement itself will generally be managed by a binary input that is triggered when the sensor is pressed, or through a laser sensor that directly feeds Galaad with the Z variation. For a binary sensor, it is possible to set up a safety procedure by requesting that the probing session is launched only when the sensor is pushed manually by the operator, which enables checking that the electrical circuit works and is connected to the right input. This, because a probe that plunges down without stopping at the workpiece top surface can quickly become a probe to be replaced.



In the same vein, the **machine-bed flatness correction** considers the whole machine surface defects once and for all. On a small-sized machine, surfacing the bed using a flat cutter with a big diameter is quite enough for obtaining a very good flatness that corresponds to X and Y axes. Machinings on a bed that has been surfaced by the machine itself should be no problem concerning the Z variation. But on a big table, or if the bed that supports the workpieces cannot be surfaced for any reasons, then the software can apply a calculation for correcting these variations of the workpiece plane. This supposes that the part to be machined follows the variations of the machine bed. The function is based upon a matrix of points located at regular intervals on the machine bed, every point having a measurable Z position. You can indicate the number X and Y of points to be probed, and the general position of the matrix, *i.e.* the lateral margins around it. The software will help you probe the altitude of every point of the matrix, by positioning itself the probe or the comparator, taking into account the offset between the probe and the spindle. If your machine is fitted with a probe sensor, then Galaad can fill up the matrix all alone by measuring every point. Otherwise you must mount a comparator and drive the Z axis manually until the needle returns to the reference position. Once a measurement has been validated, the software lifts up the Z axis, moves it to the next XY point and plunges down to the previous position. Here, you just have to gently move the Z axis up or down so the comparator needle returns to the reference and Galaad records the relative altitude.

The matrix of flatness points is used for readjusting the Z coordinates of the machined paths, with a progressive smoothing function so the result is perfectly adapted to the variations of the bed.

❑ Kinematics post-calculation

The kinematics calculation is an important function that manages fast machinings. It aims to ensure an optimal motion fluidity at high speeds whilst avoiding losses of steps of a stepper motor, or drag errors in case of a servodrive. When this function is enabled, the feedrate is no longer a constraint that is imposed for every movement, but an objective to be reached as far as possible, the major requirement being to prevent a motor from overriding its electromechanical limits. The kinematics calculation can be activated only on machines that have numerical controllers integrating three functions:
- A local memory buffer (see that parameter) that is able to store the motion vectors before they are executed. The software then works in asynchronous mode and tries to maintain a high level of buffer fill-up.
- Corollary of the above, a high bandwidth so the transmission of the vectors is faster overall than their execution. If the path is completed faster than its own transmission, then the buffer gets empty and then, at some point, the machine will be thrown into a fast trajectory, to be brutally interrupted by the lack of new coordinates not yet transmitted, resulting in knocks or losses of steps.
- The possibility to execute movements with acceleration ramps and movements at constant speed (see hereafter). Other types of movement exist, that Galaad uses when they are available, generally on very advanced CNC systems.

If one of these three functions is lacking, then the kinematics calculation cannot work. The bandwidth, closely related to the communication port as well as the protocol and the command language, often represents the crucial point of a fast machining. If the number of vectors to be transmitted is high for a short distance of the path, for example in a curve with a very fine resolution, then the motion speed may render the transmission time longer than the execution time for the machine. Conversely, a long straight line requires only one coordinate to transmit, which is very brief, and much time for executing that movement, which lets the transmission get ahead of the curve. So, the local memory buffer is in charge of absorbing the variations of the gap between the transmission and the execution along the machining path. But, even with a very large memory buffer, the transmission must be faster overall than the execution.

The kinematical parameters are split into two main parts, each of them corresponding to a type of movement that the machining process can perform. Let us take this opportunity to do some theory about vector dynamics. Sure you were missing it.

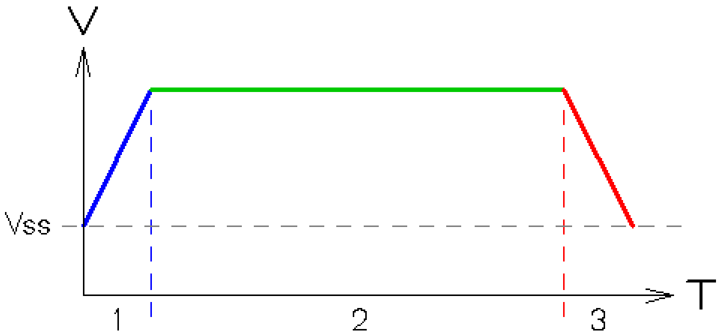
The first type of movement a numerical controller can undertake is the vector with acceleration ramps. A vector is a basic movement from one point to another at a given speed (or more accurately the movement from the current point to a distant target point). For example, in a 3-axis machining, a vector will have relative distance components X, Y, Z and a velocity component V. The power of a motor is not infinite. Coupled to an axis, it has an inertia that thwarts its speed changes. It cannot jump from immobility to a high rotation speed, nor reciprocally jump from a high rotation speed to immobility. Consequently, the numerical controller imposes three phases for performing its movement between two points:

1 - An acceleration phase, enabling achievement of the speed demanded. The motor does not start from a null-speed but a floor speed named *"Start/Stop Speed"*. The numerical controller makes the motor jump instantly from stillness to that start/stop speed and then increases the RPM according to the acceleration ramp that its power and its inertia can afford.

2 - Once the cruising speed is reached, the RPM remains stable. The motor does most of the movement to the target point at that speed.

3 - Before arriving at the target point, the controller triggers a braking ramp that makes the motor slow down and stop softly on the target point. The deceleration is generally symmetrical to the acceleration.

The closer the points are to each other, the shorter the movement. So if, furthermore, the cruise speed is high, it can happen that the motor jumps directly from phase 1 to phase 3, *i.e.* begins to brake even before having completed its acceleration but without reaching the speed. The controller manages all this. In fact, the speed curve describes a trapeze: a first ascending slope (acceleration ramp), then a stable flat, and lastly a lowering slope (braking ramp).



The two parameters of a movement with ramps are then the start/stop speed and the acceleration slope. The numerical controller generally uses neither the metric nor the imperial systems, but rather the motor increments. So the speed is actually a frequency, set in Hertz. We are talking about a ***"Start/Stop frequency"*** which is translated into a speed for each axis. If these have different gear factors, then the frequency will correspond to the most critical axis, namely the weakest motor of the heaviest axis having the highest gear factor. Please put all this in a jar, shake strongly and it will generally output a frequency of about ½ to 1 revolution per second, at least on a more or less classical machine. For example, if your motors have a resolution of 1600 pulses/revolution, the start/stop frequency will typically be between 800 and 1600 Hertz.

The start/stop frequency indicates the instant speed for pulling away from immobility, and also the last speed before stopping the motor. A bad setting can have important consequences on the behaviour of the machine. If the value is too low, then the acceleration and braking ramps will be a bit longer and it will sound like the movements are too soft, which is not very harmful. But if it is too high, then the motor will have to make too big a jump for catching its acceleration ramp, and reciprocally it will turn too fast just before stopping. At best, the machine will knock and, at worst, the motors will lose steps (the position will drift away gradually) or, for a servodrive, raise a drag error. **A good ear listening to mechanical noises is a significant asset for fine tuning a start/stop frequency and more generally the kinematics**.

The second parameter of a vector with ramps is the **acceleration** slope, namely the time for increasing the speed. Since the deceleration slope is symmetrical for decreasing the speed at the end of the vector, the acceleration value is the same for the braking. As for the start/stop frequency, the distance is encoded in increments. An acceleration in increments per second square is then quantified in Hz/s and will be common to all axes, like the start/stop frequency. If the value is too low, then the speed increase and decrease will be too long, with too soft movements. But if it is too high, then the motor may fail to follow the the slope and will stall on the way up, resulting in a position drift or a servodrive error. The acceleration slope is supposed to be calculated for the working conditions of the most critical axis, therefore the weakest motor having the heaviest load, taking into consideration the torque curve of this motor relative to its speed. In other words, this is not simple and, here again, experience and a fine ear are often more precious than a calculator for finding the good acceleration value. By the way, Galaad's calibrating function may help you fine tune these settings.

The vectors with ramps are used for all types of isolated movement between two points. Isolated means that the speed at each of these points is slow or null. All driving movements in manual or semi-automatic modes, with no exceptions, use vectors with ramps, including the probing, the plunge down to the tool sensor or the tool changer. All these movements are considered isolated even if they are part of a chained sequence. In automatic machining, the positioning movement with the tool cleared, the tool lowering down to the contact with the workpiece top surface, the vertical drilling plunge in the material, the tool bottom right to clearance height, and the parking are also isolated movements, consequently using vectors with ramps. Lastly, when the tool is down in the material and feeding horizontally, if a movement is performed between two sharp vertices on which the tool passes at slow or null speed, for example the sides of a rectangle, then here also the software will use a basic vector with ramps, even if it is not isolated.

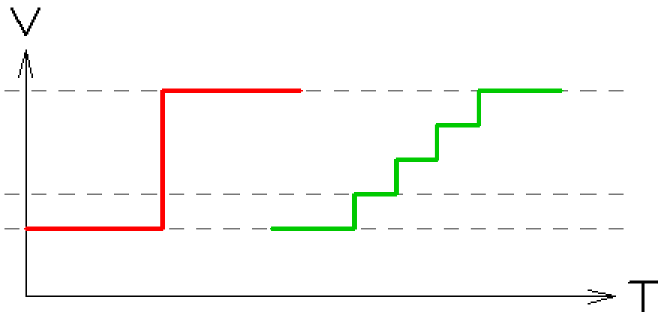
Many machines of an old type or having limited speeds can perform only movements with ramps. Consequently, on a curve trajectory, there will be an acceleration, a cruise speed and a braking for every small vector that makes up the curve. This means that the machine spends most of its time accelerating and braking, and probably never reaches the target speed, except for long movements. The result is slow and jerky, especially when the cruise speed is high. Obviously, if the numerical controller integrates a circular interpolation command, at least the arcs of circles will be performed in one single movement with an acceleration and a braking at the beginning and the end of the arc instead of every vector forming that curve. But this supposes on one hand that the arc itself is isolated, with no direct chaining before and after with the rest of the path, and on the other hand that resolutions and gear factors of X and Y axes are strictly identical, otherwise the circular interpolation is ineffective. Anyway, this type of interpolation is irrelevant for a non-circular curve such as an ellipse, a spline or a Bezier curve. In fact, the best way for getting a good path fluidity whatever the curve type and the axis features are, would be chaining the vectors without slowing down between them, which leads us to look for something other than the vectors with ramps for moving fast enough along a continuous trajectory.

The second type of movement Galaad can manage, provided that the numerical controller accepts it, is the vector at constant speed. Here, there are no acceleration and braking ramps; the speed remains unchanged from the beginning to the end of the vector. The advantage is, several vectors can succeed to one another with no slowdown between them. But this creates new problems, the first being the fact that, for reaching the cruise speed, the axes must accelerate along the curve and not forget to brake at the end of the path, and also a calculation formula must be applied for maintaining an overall consistency in the speed changes.

The vectors with constant speed do not have main parameters such as the start/stop frequency and the acceleration, since their speed does not vary. On the other hand, for managing them correctly, the software needs more complex parameters since it is no more possible to try them separately in a simple manual jogging.

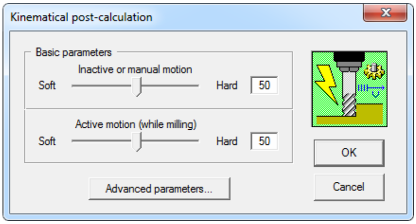
Many systems for overall path management are based on the angle between two consecutive vectors, this angle determining the speed reduction at their passing point. But this solution, that looks simple and efficient, actually works only in the most optimal cases. The local angles of a series of consecutive vectors reveal nothing about the chaining dynamics, which may vary a lot depending on whether the overall curve is wide or tight. A U-turn performed along a half-circle made of 180 vectors following one another with an angle of 1° between each of them is obviously not run at the same speed if the arc has a radius of 100 mm or a radius of 1 mm.

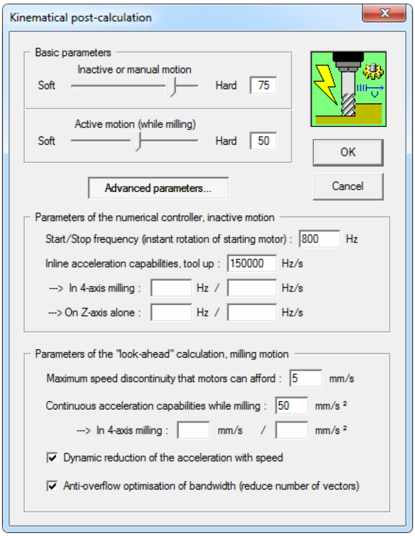
The Galaad function for calculating the kinematics is arithmetical and not geometrical. It considers every single vector separately. In fact, even if we have a global view on the path in a motion using several axes, a motor does not want to know what its dear colleagues are doing in the meantime. The only important thing to know is whether each motor can deal with the speed change it has to face, depending on the power it can provide and the inertial load it must carry. Above all, the calculation is a search for the discontinuities of the rotation speeds. There are two kinds of resulting parameters, that look more or less like a start/stop frequency and an acceleration: first, what speed change can a motor accept without stalling? Second, how much delay does a motor require for absorbing a speed change before being able to cope with a new one? The purpose of the calculation is to replace too high a step, by a series of small ones that the motor can climb or descend without stumbling on it. A big change of speed then becomes a succession of affordable little changes, spread over time. Now we have only to define the height of the small stages (the maximum speed discontinuity that the motor can deal with) and their width which, once combined, determine the overall slope of this set of steps. The more the motor can handle an important speed change, the less there will be intermediate steps. And the shorter the delay for absorbing a speed discontinuity, the stronger the acceleration will be.



Indeed, adding intermediate mini-vectors greatly increases the number of coordinates to be transmitted, which is a bandwidth killer. Hence it is tempting to set a high value for the maximum discontinuity. But the purpose is, above all, to keep the motors inside their own range of efficient work. The bandwidth will have to cope with that, otherwise it will become necessary to reduce the maximum feedrate, considering that the machine cannot follow. Nevertheless, please note that a motor torque is not constant. Particularly on stepper motors, there is a brutal stall of the torque with the speed, generally about 10 to 20 revolutions per second, due to the charging time of the coils, which can be reduced by increasing the voltage. So the parameters for speed discontinuity and for acceleration capability are not constant, unlike those for start/stop frequency and acceleration for the vectors with ramps which are applied around the null speeds when the torque is maximised. It is up to you to find the best compromise for the usual feedrates of your machinings.

Setting the kinematical parameters can be done by activating (or double-clicking on) the option *"kinematics post-calculation"*. The underlying dialogue box offers a possibility for setting easily from two sliders, one for the vectors with ramps, that correspond to the inactive movements when the tool is up or the manual jogging, and the other for curved paths with fast vector chainings. The more a slider is on the left hand side, the more the motion it controls is soft. Conversely, the more the slider is on the right hand side, the more the motion is hard, until it reaches the risk of losing control and beyond.





For accessing the fine tuning of the calculation, click on the button *"Advanced parameters"* at bottom, which displays the complete dialogue box. Here, you can indicate in the middle frame the actual start/stop frequency and acceleration values (for movements with ramps if you have understood the demonstration), and in the lower frame the values for the **maximum speed discontinuity** that the most critical motor can carry out without stalling, and the **continuous acceleration capabilities** along a path. These two values are to be set respectively in mm/s and mm/s² because the calculation applies to coordinates and real speeds, even if the numerical controller uses increments and frequencies. The start/stop frequency is sent as a general parameter to the numerical controller when initialising the dialogue and, thereafter, Galaad does not want to hear about it anymore. On the other hand, the speed discontinuity and the continuous acceleration capability are used for its own calculations concerning every new path.

These two values for the *"look-ahead"* calculation when milling can be rendered variable with the option that manages a **dynamic reduction of the acceleration with speed**. This reduction makes sense with motors that lose torque when they are gaining speed, even though this loss of torque is generally not linear. If that option is enabled, then you should consider that the acceleration value (and this only) that has been set applies to a feedrate of about 20 mm/s (1200 mm/mn) and will progressively decrease when the speed is higher (but the acceleration is not increased when the speed is slower).

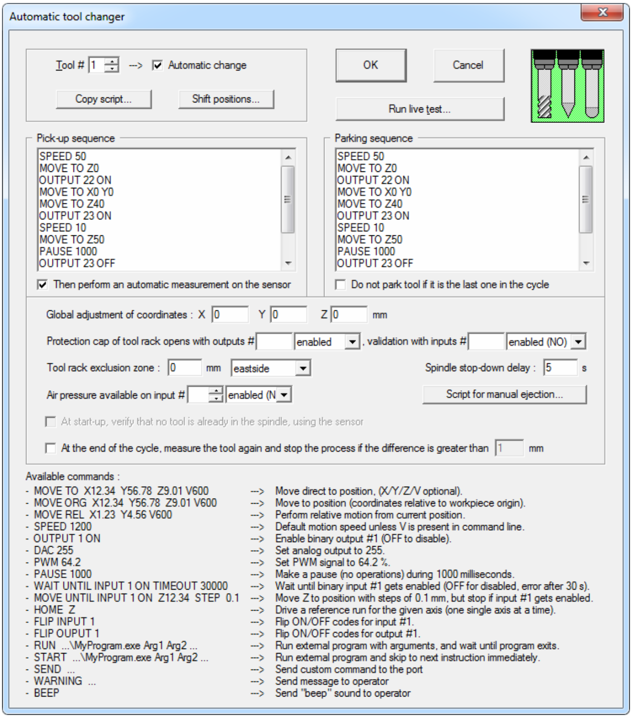
The *"look-ahead"* kinematics calculation dramatically increases the number of vectors by interpolating new ones that are used as intermediate levels for overcoming important speed discontinuities. Therefore, it can become necessary to reduce overall the number of vectors to be transmitted. This may prevent a traffic jam on the bandwidth, which would result in drying out the local memory buffer. The **anti-overflow optimisation of the bandwidth** will be in charge of filtering mini-vectors that are considered non-significant and consequently will not be transmitted. A discarded vector actually makes a very small angle with the previous vector and runs along very short distance, say a few tenths of millimeter, that distance being adjusted depending on the feedrate. This aims to lighten the curves with too high a resolution that would lead to a risk of stalling the transmission capabilities. A saturated bandwidth is characterised by a machining that suddenly becomes jerky though it was fluid just before.

As a conclusion about the kinematics calculation, let us say that it is a rather complex function that requires much more accurate settings than simple sliders varying between *"soft"* and *"hard"*. Please keep in mind that **the energy needed for accelerating or slowing down a movement is proportional to the square of the speed**: feeding twice faster requires four times more power; feeding three times faster requires nine times more power. The kinematics calculation cannot perform miracles, and the stepper motors are running against important falls of torque when turning fast. Don't ask your machine for the impossible, even with Galaad.

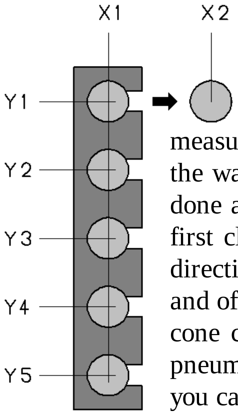
Automatic tool changer

To conclude these pages of advanced parameters, a last button at bottom right allows you to open the set-up of an automatic tool changer.

A tool changer, whatever it looks like, rack, barrel, or anything else, always uses a sequence of automatic commands for moving axes, switching outputs, waiting for inputs to toggle on/off, *etc*. Galaad offers the ability to **programme for every tool a pick-up sequence**, *i.e.* the process that will load the tool from where it was parked and mount it on the spindle, and a **parking sequence**, *i.e.* remove it and store it back in the rack. If the positions should be identical, it is more than probable that the set of input/output commands will differ. But it is up to you to set up these processes according to the technical information you have.



Manual jogging will help you find the right positions, inputs and outputs. Do not forget that the *"Refresh"* and *"Loop"* buttons allow you to check the inputs, and clicking on the green boxes of outputs will change their state. A good preliminary manual drive is worth any teach-in sessions. On a classic tool changer with a linear rack, the procedure consists of bringing the XY axes vertically above the tool cone and noting the accurate position. Then lower the Z axis until the cone is engaged in the chuck, of course having made previously sure that the jaws are open, and note the Z position. After having closed the chuck by



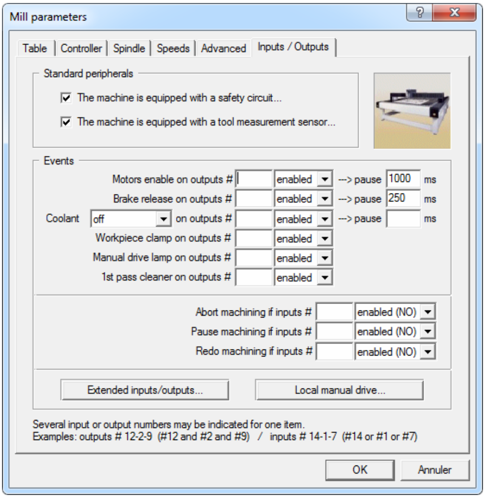
triggering the output that controls it, you can move the X or Y axis horizontally so the tool is extracted from its clamp with a sufficient margin, and again the clearance position is noted. Now you just have to measure the distance between two neighbouring clamps. By the way, this can be done by simply jogging the axes. Having done all this, you should have the absolute XY position of the first clamp, the clearance position (X or Y depending on the direction of the rack), the offset (X or Y) between the clamps, and of course the output number and polarity for triggering the cone clamping jaws, whatever the mechanical technology is, pneumatic or electric. Once these informations are recorded, you can easily write the scripts for picking-up and parking any tool that is present in the rack. A button for copying these scripts and another one for shifting all coordinates in one operation should help you out.

For every tool numbered, you must indicate whether it belongs to the changer by ticking the checkbox *"Automatic change"*. If this checkbox is active, then only you can access its two scripts. The button *"Run live test"* will help you give it a try. Warning: the pauses are sometimes long when there is a mechanical action, especially with pneumatical items that have some inertia. And do not forget to insert in your scripts a prior bottom right of the tool before sending it to a new XY position, just in case there could be some obstacles in its way. Galaad does not like to impose movements and consequently does not decide them alone. You can also write a script for manually ejecting a tool off the chuck without moving the machine. But there is often a button for this beside the spindle.

The linear racks are sometimes topped by a protection cover which prevents the tool cones from receiving chips or dust. You can integrate the command for opening this cover, which may also protect the tool sensor, in which case you must indicate it in the parameters of the tool sensor (see the page *"Inputs/Outputs"* hereafter). A linear rack will generally be located on the table side, with an exclusion zone that forbids Galaad to operate automatic movements except when executing these scripts. Manual jogging is possible inside the exclusion zone, but a stop point is set at the borderline. Another security, the tool measurement sensor can enable checking if one of them is already present in the chuck. It is a bit time-consuming, but quite efficient. In the same way, a measurement of the tool at the end of the cycle can help checking whether it is still alive or a part of it is missing.

"*Inputs/Outputs*" page

The last page of machine parameters gives access to a special control of the inputs and outputs for some applications that use peripheral devices. Galaad agrees to trigger binary outputs on demand according to logical events of the machining process (an output is considered binary if it has only two possible states, on or off, with no intermediate states such as an analogue or PWM output). The software can also react when a binary input is triggered.



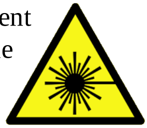
Do not forget that **you may indicate several outputs** (4 maximum) and in some cases several inputs for one single function. To do so, you must separate their numbers with spaces or hyphens. If you specify several outputs, then all will be triggered when the corresponding function is activated. If you specify several inputs, then Galaad considers that the function is triggered when at least one of them is in the indicated state (a logical *"or"* is applied). For one single function, it is not possible to specify several outputs or several inputs in different logical states.

The top frame allows you to manage standard peripherals such as the safety devices for the operator, or the tool measurement sensor. If you wish to modify their underlying parameters, then double-click on them.

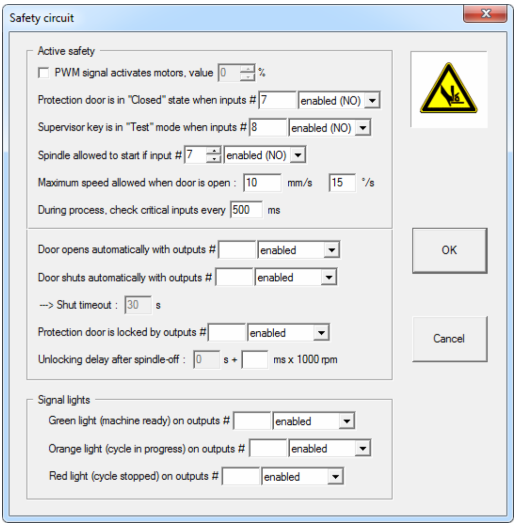
According to official standards in force, the mode of operation for a **safety circuit** must be fully independent from the software and even from the computerised numerical controller. Opening the door of a safety enclosure or pressing an emergency stop button cannot alone notify a programmed process. Safety circuit action must be immediate and direct by stopping the tool and all machine movements, the simplest being of course a clean cut-off of their power supply. The operator should not be confronted with an active spindle without an intermediate cover or screen, and an axis involved in an automatic movement is potentially dangerous. The real importance of protective equipments can never be overemphasized, including for hobby CNCs. **Even a low power machine can seriously injure an operator**. The applicable regulations are not flexible, but they do not ask for the impossible. Safety is

important. Protect your eyes; protect your hands; protect your colleagues. Do not shunt the human protection systems, even though they can sometimes look cumbersome. If your machine does not require an enclosure, then at least please think about wearing **protective glasses**.

It is becoming easier and easier, nowadays, to find class 4 laser sources that can act as a computerised pyrography device or equivalent. Working without an efficient eye protection with such a system is dangerous. A simple diffused reflection on a slightly reflective material can be sufficient to cause a skin burn and moreover irreversibly damage the retina. **Never work with a laser source unless you have the appropriate protection equipment**!



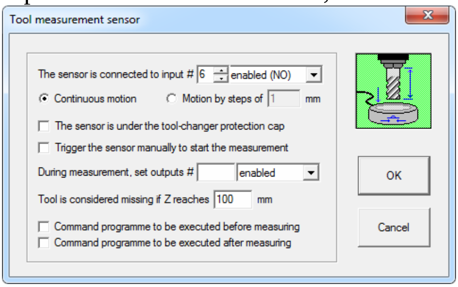
The fact that the safety system reacts autonomously does not forbid a polite dialogue with the software that supervises the automatic process. Consequently, that software is supposed to know in which state the protection equipments are and it can even talk to active safety devices. If the protection cover is open, Galaad will refuse to switch the spindle on and to launch the automatic process. So does it check this status before sending a sensitive command to the machine, and warns the operator if the cover remains open. The safety regulations may allow a lower safety level when the supervisor's key is in *"Test"* mode, provided that your machine has one and assuming that the machine can then just perform harmless calibration and verification movements. These two states, cover and key, correspond to a



couple of inputs that you must set: their number and polarity. If there is no key, then just indicate nothing.

Opening and closing the safety cover can also be controlled by two outputs and a closing delay before starting the automatic cycle. Some protection enclosures are also fitted with an electromagnetic or pneumatic safety lock that Galaad can activate before starting the spindle or the automatic process. The unlocking will be completed after switching off the spindle, plus a pause delay that lets the motor stop. That pause delay can integrate a fixed part plus a variable part that depends on the rotation speed when switching off. Finally, you may indicate three outputs which correspond to a signalling tower that reports the general status of the automatic cycle: green light for the machine being stopped, waiting for a cycle or cycle over; orange light for a cycle in progress; red light for a stopped cycle. If there is no signalling tower, then these outputs can be used as well for switching a beacon light, a horn, a cello suite, it's up to you to decide.

If your machine is fitted with a **tool measurement sensor**, which is highly recommended for making life easier, then you must tell Galaad on which input that sensor is connected, and of course its polarity. Let us remember that here once again, an *"enabled"* input is triggered when the contact is normally opened (N.O.) and a *"disabled"* input is triggered when the contact is normally closed (N.C.), *"disabled"* absolutely not meaning *"unused"*. If an input is not linked to a switch but to a peripheral device that sends a signal, please conclude nothing from the high or low states and rather use the function *"Parameters / Machine / I-O test"* to know more about it.



The descent of the tool on the sensor can be continuous, in which case Galaad sends one single movement command down to the low end of the Z axis. This movement will be interrupted by the trigger of the sensor. But not all controllers have such a command. Then Galaad will drive a stepped descent, each step being followed by a read command for checking the state of the input, until the sensor is triggered. A dichotomic search of the trigger point will then be launched for finding quickly the accurate position. You must set the step length: too short, the descent will be long; longer than the sensor stroke, you might become a good client for your provider of cutters. Or for your provider of tool sensors if your cutters are solid. Do not forget that, before each descent on the sensor, you can enter a rapid Z movement for pre-positioning. By the way, before and after having launched a tool measurement procedure, you can execute command scripts according to your needs.

On machines that are fitted with an automatic tool changer made of a linear rack, it can happen that the measurement sensor is located under the protection cover of the changer. Galaad should be informed to avoid doing unfortunate operations. Of course, you have indicated in the parameters of the tool changer the output that triggers the opening of the cover, because if it is integrated in the scripts, Galaad will not find it alone. You may set here an output to be triggered during the tool measurement procedure. This output can also correspond to the changer cover or any electrical requirements for operating the sensor. A security option allows you to manually press (and of course release) the sensor to start the descent, with a waiting message on the screen. If the sensor does not work or has been wrongly parametered, then at least the tool will not smash it. Your cardiologist recommends that option, even though your manicurist disagrees.

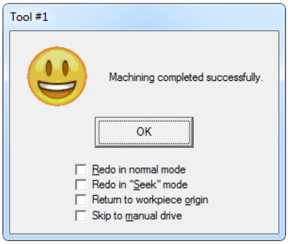
Back to the *"Inputs/Outputs"* page of the machine parameters, the frame *"Events"* helps you trigger outputs correlated to events such as the **activation of axis motors** after initialising the machine, with a possible pause just after. The output will be reset before exiting the machining module. Powering the motors can also **release the Z axis brake**, but the parameter is independent since it can happen that both outputs have opposite polarities. Galaad is often in a hurry to move axes, so releasing the brake may also require a pause to temper his ardour for getting to work.

Perhaps your machine is fitted with a **tool cooling system**, connected to a binary output which will be activated either continuously from the beginning to the end of the machining cycle, or only when the tool is down in the material and interrupted every time it is lifted up, or when you request it, in which case a message will ask you about it at every new process. But nothing prevents you from installing a basic switch beyond the output so you can decide during the cycle whether you want the coolant on or off, it being fully understood that, if it is on, the output will in any case be switched off at the end of the process. And also the intermittent mode remains valid then.

If you have a **workpiece clamping system**, for example a pneumatic vice, a vacuum table or any device that can keep the workpiece locked on the machine bed, then that system will be switched on with no interruptions from the beginning of the workpiece origin set-up to the end of the last machining cycle, the workpiece being completely finished (or the cycle aborted). Galaad can also **switch on a lamp** for lighting up the table during the manual jogging, which can ease the approach of the workpiece datum, and this lamp will be switched off when the machining cycle begins. Please do not overdo this gadget, a simple switch mounted on the machine is also not too bad. Lastly, you can set a **vacuum cleaner** to be switched on during the normal passes and switched off after, if your ears do not appreciate much a vacuum cleaner that works continuously with the spindle. It is obvious that the above mentioned items are purely indicative and of course you may use other more esoteric devices on the same actions.

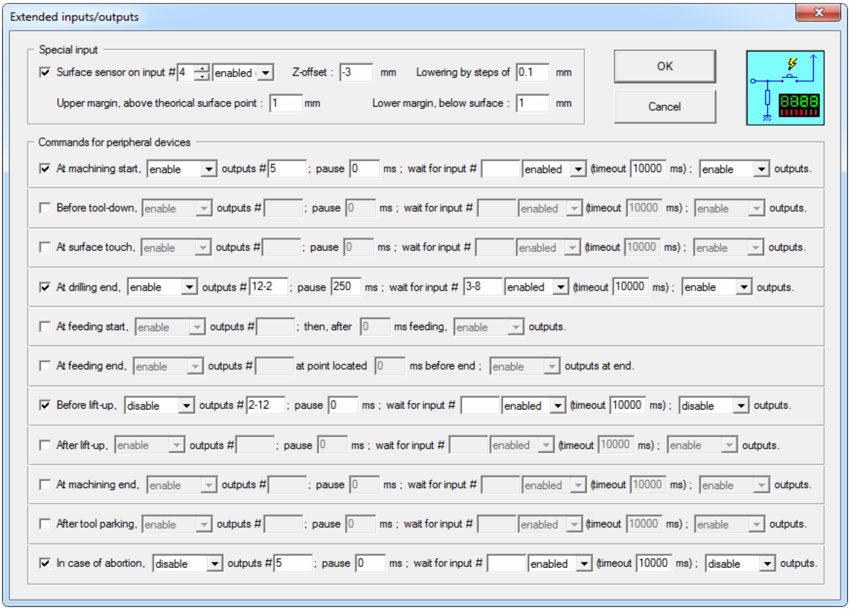
The bottom part of the frame allows you to automatically trigger a stop of the machining cycle if an input changes its state. In this case, Galaad manages a simple stop with a spindle switch-off and sending the tool to its normal parking position. But this stop is not always immediate, because the numerical controller may not consider the input critical, *i.e.* triggering a rapid braking procedure. This depends on the capabilities and the specific parameters of the controller. Some can deal with this, some cannot. Since it is not possible to overload the machine with repetitive messages permanently asking for the input statuses, Galaad reads these inputs only before making the tool plunge down into a new path. If, at that moment, a stop or pause input is triggered, then only the function shall take effect. There cannot be any reading of that input when the tool is feeding in the material, and an immediate reaction is therefore impossible. If your numerical controller integrates a settable critical input, for example an error signal, then you may reuse this input here to inform the software that the machining cycle is already stopped and it is now useless to go on sending motion commands to the machine. If you wish to have a less programmed stop but more efficient and possibly more abrupt, then you should integrate a small circuit that powers-off the machine directly. You can also request a simple **machining pause** when an input is triggered. In this case, the machining will not be aborted but will be resumed as soon as the input returns to its initial state. The operator will not even be bothered by Galaad popping up a message on the screen. This is just a local pause, managed locally.

When the automatic cycle is completed, a last input can be checked for **reiterating the process**, for example when the operator is pressing some *"Start"* button without having to use the keyboard or the mouse. When the cycle-end message appears on the screen, then this input is read in a loop (actually twice per second so pressing the button must be a bit long). If it is triggered at this moment, then Galaad interprets it as the equivalent of a click on the option *"Redo"* followed by a validation of the message.



❑ Extended inputs/outputs

Galaad manages more input/output related events than appear on the standard parameter page. When you click on the *"Extended inputs/outputs"* button, a large window appears on the screen:



At the top of the window, you can set up a probe configured especially to find the Z position of the material to be machined before every tool plunge. This can be helpful for milling or drilling workpieces with a top surface that is not perfectly horizontal. In addition to the classical parameters of the surface probe, you may indicate a **Z offset** value from the trigger point, which allows you to lift the sensor higher above the tool if necessary. The lowering will be continuous unless you indicate a stepping value or if your machine does not integrate a function for moving until an input is triggered. The **upper margin** determines the Z position (in relation to the workpiece origin Zo) from which the search for the contact begins. There is no point in entering too high a value that would slow the process. The **lower margin** indicates the maximum Z position for stopping that search if the contact is not found (still in relation to the workpiece origin Zo). If the top surface was not detected at the workpiece origin plus this value, then the process is stopped and the operator called up, who can then decide to continue from the lower Z point, without searching below, or abort the whole cycle. In brief, in relation to the workpiece origin point Zo, the search for the top surface is performed between the upper margin and the lower margin.

The main frame is intended for automation technicians. Its lines display repetitive events of the machining process, on which you can insert output triggerings, wait-states related to inputs, or simple pauses. For example, if you want to activate a coolant system only when the tool is drilling the material and stop it when it starts the horizontal or 3D feeding, and this coolant is connected to the output number 2 enabled but the software must wait for a validation of the coolant system on the input number 4 disabled after an instability delay of 50 milliseconds, then you should validate the two following event lines:

At surface touch, enable outputs #2;
pause 50 ms, wait for inputs #4 disabled (timeout 1000 ms);
enable outputs.

The last argument *"enable outputs"* means that the output #2 must remain in its active state at the end of the event. It could be possible to stop it for getting just a pulse. For stopping that drilling coolant, it would be:

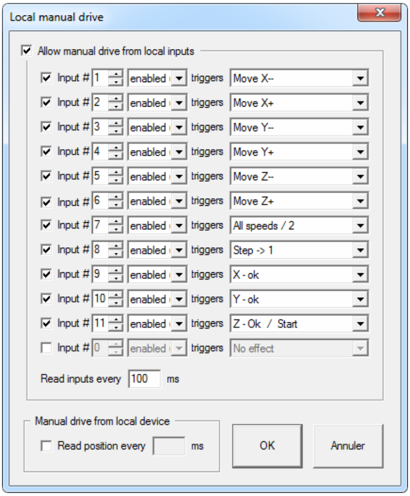
At drilling end, disable outputs #2;
pause 0 ms, wait for inputs #0 disabled (timeout 0 ms);
disable outputs.

An unused input or output gets the number 0 or remains empty. The words *"inputs"* and *"outputs"* are plural when you may indicate several, separated by hyphens or spaces.

The two lines *"At feeding start"* and *"At feeding end"* add intermediate triggering points on the path. Their positions will be calculated automatically depending on the feedrate. This can be used for dispensing glue or paint when the liquid flow must be stopped some time before reaching the terminal point of the path.

Local manual drive

If your machine offers a dashboard that is connected to the binary inputs, then you may assign functions to these inputs and Galaad will permanently scan inputs when in manual control mode. The usual commands are for moving the axes and miscellaneous control items for all manual operations.



Once again it is better to begin with a short trip to the function *"Parameters / Machine / IO Test"* so you can take note of which buttons correspond to which inputs and which states. This manual control is available only on numerical controllers which can start a movement that will be stopped when triggering an input defined as critical, *i.e.* interrupting any motion currently in progress. If the controller does not provide this function, then no movement can be linked to an input. Other functions will remain usable.

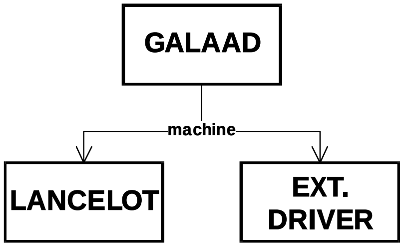
This local manual drive is a bit eclipsed by the extended possibilities that a joystick or an external numeric keypad can offer, since these have buttons or keys that can also be programmed from Galaad, as mentioned in the previous pages. But you can nonetheless link these functions to the few buttons that are sometimes available on a machine or a controller front panel. Finally, if your machine includes a dashboard with local motion control, Galaad can ask for the position of the axes at regular interval for keeping the display of the coordinates coherent with the reality.

External driver

As you have understood it, Galaad integrates the whole process chain, from drawing the object with its related toolpaths, to supervising the automatic machining cycle, through setting the workpiece origin point. However, many machines not known to the software cannot be controlled directly by it in real-time mode. Galaad can still be used for drawing an object ready for machining, but a different software package will have to be used to download the machining file, or control the automatic milling process, or both.

To facilitate the connection, Galaad will start the external process directly from the command *"Machining / Standard machining on 3 axes"* or the corresponding icon. In this case, the current drawing is automatically exported in the format required by the external driver, which is immediately started as a new Windows task. Please note that this driver programme can equally be an application running under a Windows console session.

In fact, the two modules Galaad and Lancelot represent the drawing part and the driving part of the same software. You may use Galaad as a CAD-CAM application with any other CNC driver module which will be called automatically via the *"Machining / Standard machining"* command.

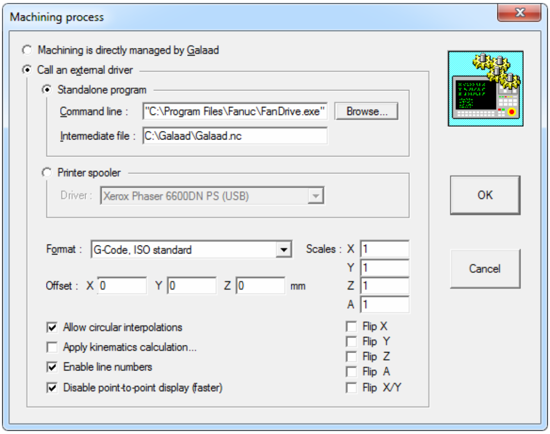


Alternatively, it is possible to call up the Galaad machining modules Lancelot (specialised in 2½D processes) and Kay (3D) from another CAD application, if it permits this. The Lancelot programme just has to be called up with the mill file name as an argument, and the standard name of the file format in brackets, if the extension is not significant. Example:

"**C:\Galaad\Lancelot.exe**" **C:\…\MyDrawing.plt**
"**C:\Galaad\Lancelot.exe**" **C:\…\MyDrawing.xyz (plt)**

Calling up an alternate machine driver is not absolutely direct but goes through Lancelot as an intermediate module that defines milling parameters and manages tool cycles. In fact, it is Lancelot and not Galaad itself that will chain to the external CNC driver once the intermediate file is generated.

To configure an external driver programme, use the command *"Parameters / Machine / External driver"*. The dialogue box offers the choice between the module Lancelot, internal driver for Galaad, and an external module which can also be a virtual printer driver.



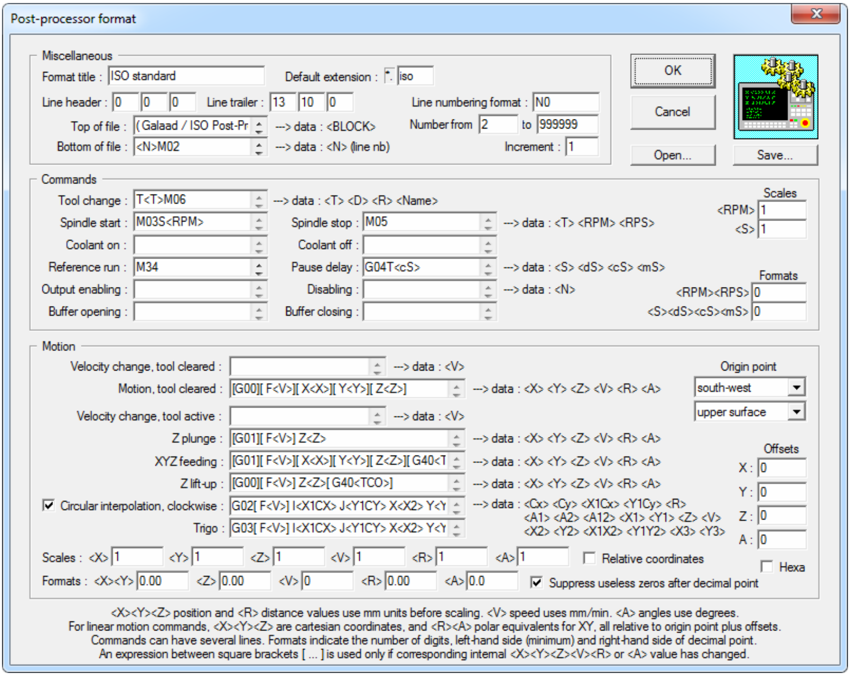
When you launch a machining cycle through an external driver, an **intermediate file** will be created in the **format** specified and the external programme will be called up with all its arguments as specified in the **command line** text box. Galaad will then continue to operate independently while the external driver gets on with its work. If you do not set a command line, the intermediate file will nonetheless be generated but no external programme will be called to chain the process. If you do not indicate an intermediate file to be transferred to the external driver, then you will be prompted to give a name every time you start a new process. Since this file is automatically generated by Galaad, it is probably not useful to save it under its own name. Instead, you should overwrite it when starting a milling process. The important thing is the drawing file, not the intermediate file for the driver. You may also indicate neither command line nor intermediate file, in which case no driver will be called up and you will be prompted to give a file name **every time**. Please note that the file name (and its complete path) is automatically added at the end of the command line. Consequently it is pointless to enter it twice.

The formats for the file sent to the external driver are those among the export functions that are the most usual. Of course you can use the post-processor whose syntax is entirely definable in Galaad (see hereafter).

The other parameters help you set the coordinate system and the directions of the axes. It is possible to accelerate the generation of the intermediate file by disabling the path displays, which may slow down the process depending on the performance of your computer.

❑ Post-processor format

If you are happy with none of the available export or external driver formats, you still have the possibility to define for yourself your own format from the function *"Parameters / Post-processor"*, which opens a generously endowed dialogue box.



Here you can find everything needed for describing any format, provided that it is in text mode and that the encoding of the numerical data is decimal or hexadecimal. The binary formats, which would raise many problems related to the data size, are therefore excluded. Fortunately, they are never used for file transfers, at least in the area of numerical control.

Even if the standard format remains the ISO G-code, on which you can graft several variants your machine might need, you can nevertheless create your own alien encoding with an exotic syntax. By the way, Galaad provides a few frameworks based on ISO, Isel-NCP or HPGL, that you can pick up and modify using the button *"Open"* at the top right. On the other hand, you can memorise your modified format under your own name, using the button *"Save"*.

Since the format is based on text, you must indicate the codes of the constant characters that will be used as **line header** and **line trailer**. These codes are set as decimal values representing an ASCII code, for example 2 for <STX> *(Start Text)* and 3 for <ETX> *(End Text)*. However, the most common remains a line with no header codes (set 0 for no code) and 13 for <CR> *(Carriage Return)* then eventually 10 for <LF> *(Line Feed)* as line trailer. Every line produced by Galaad will begin and end using these control bytes, with no exceptions.

It is possible to number the lines. In this case, you must set the **line numbering format**, the zero indicating the number. For example, if you set *"N0"*, the lines will have the heading character 'N' followed by the line number with just the number of digits needed. If you set *"N0000"*, then the number will have 4 digits filled with heading zeros, for example *"N0012"* for the line number 12. Of course, if the number overruns the size, then it will not be truncated: the number of zeros set here is a floor, not a ceiling. If you enter nothing in this box, then there will be no line numbers. Warning: the **top of file** and the **end of file** are not numbered, and these only. In the first one, you must add your own line numbers (you may set as many lines as necessary, with a maximum of 1000 characters for the total). This block will be sent exactly as you have defined it. But the end of file block may eventually receive normal line numbers that follow the previous ones, see below how.

Very important: the variable data must appear in the form of labels predefined by Galaad, inserted between symbols **<** *(lesser than)* and **>** *(greater than)*. For example, for the end-of-file block, the only variable data that Galaad suggests is the line number **<N>**. If the line counter is at 123 at that moment, for producing a line *"N123 M02"*, you must set the parameter **<N> M02** in which the variable **<N>** will be replaced by the pre-formatted line number, in this case *"N123"*. This remains valid for all boxes or blocks that accept variable data. **The available data labels are indicated at the right hand side of each box to be completed**. Other example, for a linear motion command to a target position, if you indicate *"G0 F**<V>** X**<X>** Y**<Y>** Z**<Z>**"*, then Galaad will replace **<V>** by the velocity value, **<X>** by the coordinate X, *etc.* The spaces that you have set are preserved, even if you enter too many of them (except those heading or trailing the line). **No syntax validity checking**

is done. It is your own format, you indicate what you want as you want. The software does not interpret it.

Almost as important: you may set between square brackets an expression containing one or several variable data. In this case, the whole expression will be produced only if at least one value inside the brackets has changed. For example, if you do not want to repeat the feedrate unless it has changed, then the line of the previous example becomes *"G0 [F**<V>**]X**<X>** Y**<Y>** Z**<Z>**"*. If the speed has been changing since the previous call to a command using it, then the expression *"F**<V>** "* will be retained with its trailing space. If the speed did not vary, then Galaad will act as if the line was *"G0 X**<X>** Y**<Y>** Z**<Z>**"*. The same applies to coordinates.

The available variables are as follows:

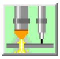
- *Tool change:* **<T>** for tool number, **<D>** for tool diameter, **<R>** for tool radius, **<Name>** for tool name.
- *Spindle start/stop:* **<T>** for tool number, **<RPM>** for rotation speed in revolutions per minute, **<RPS>** for rotation speed in revolutions per second.
- *Pause:* **<S>** for seconds, **<dS>** for decisecond, **<cS>** for centisecond, **<mS>** for millisecond.
- *Enable/Disable output:* **<N>** for output number.
- *Speed change:* **<V>** for velocity (see the scaling factor).
- *All movements, active or inactive:* **<X>**, **<Y>**, **<Z>** for Cartesian coordinates (absolute or relative depending on the mode that has been chosen in the option *"Relative coordinates"* at bottom right), **<V>** for motion velocity, **<R>** and **<A>** for polar coordinates (absolute or relative).
- *Circular interpolation:* **<Cx> <Cy>** for Cartesian coordinates (absolute or relative) of the arc centre; **<X1Cx> <Y1Cy>** for coordinates of the centre relative to the arc start point; **<R>** for arc radius; **<A1> <A2>** for the arc start and end angles (in degrees of angle); **<A12>** for the arc opening angle (in degrees); **<X1> <Y1>** for Cartesian coordinates (absolute or relative) of the arc start point; **<V>** for feedrate; **<X2> <Y2>** for Cartesian coordinates (absolute or relative) of the arc end point; **<X1X2> <Y1Y2>** for coordinates of the arc end point relative to the start point; **<Z>** for Z coordinate (absolute or relative) of the arc end point.

A bit less important, but nevertheless: the **scale** boxes allow you to individualise the multipliers to be used for each coordinate or for the speed, and you may use relative (incremental) coordinates, considering that absolute coordinates are in fact relative to the workpiece origin point, according to the corner and the plane that have been defined. The **format** boxes indicate the number of digits of the variable data and also the decimal separator for real numbers.

13

0 1 1 0 1

PLASMA TORCH

Many features in Galaad are dedicated to the management of a plasma cutter table, including additional tools. In the machine parameters, if you select a plasma torch instead of a classic spindle, then the software changes some little items of its look here and there: the green icon at left hand side for  setting the tool, depth and speed is then replaced by a simple selection of the plasma function (cutting or engraving) and a few commands in the top bar are redrawn.

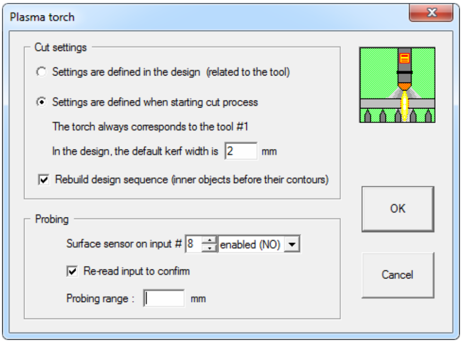
Here, we are in a pure and simple 2D design. The machining depth no longer exists; the feedrate is set by the operator when launching the cutting cycle; the torch is the only tool available, or almost. Almost, because it remains possible to perform an engraving job with the torch set to a reduced current or using an engraver pen, pneumatic or piezoelectric. The paths drawn are either assigned to the cutting job (tool number 1), and then they are assumed to have a tool compensation attached to them since the cutting kerf has a given width, or to the engraving job (tools number 2 or 3 depending on the type of the engraver). The machining window is also going to lose many features that become pointless.

❑ Constants of use

As we have seen in the previous chapter, each type of milling spindle has its own set of parameters. In the case of a plasma torch, they can be numerous and sometimes very complex, depending on the machine and the numerical controller that fits it. Since the modes of operation may vary, we can consider three major possibilities:

1- the cutting tables with Soprolec controller;
2- the cutting tables with ThunderCut controller;
3- the other cutting tables with other controllers.

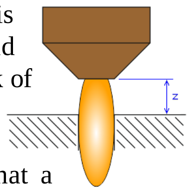
Setting the origin Zo is not managed in the same way as milling with an approach to the contact with the metal sheet then validation with the green button *"Z-ok"*. The cutting quality requires a repeated probing along the cycle because the Z coordinate of the sheet surface may vary. Hence, **the machine must be fitted with a probe for measuring automatically the Z position of the surface**, throughout the duration of the process. The most common is an assembly of the torch on a vertical slide rail with a stopper at the bottom, the torch being pushed down to that stopper by a spring but still able to move back up until it triggers a sensor. It is all planned in Galaad for calibrating it, we are going to see this later in this chapter. Another important probing parameter is the **operating range**. It is not always useful to probe again the sheet top surface just a few millimeters away from the previous probing. You may indicate a range for the last probing done, to avoid losing time. Lastly, for calculating the cutting compensations, Galaad must know the **kerf width** that corresponds to the grooves of material that are removed by the fusion. This width can be readjusted when launching the cycle, depending on the operator's settings and the cutting charts. But the calculation of the basic compensation is done in the drawing screen and the software applies a default offset.



If your machine is fitted with a numerical controller that does not provide any functions for plasma torches, then the set-up stops here. Thanks for reading.

THC

Among those managed by Galaad, the numerical controllers Soprolec *InterpCNC-2* and ThunderCut *PMK3* integrate a Torch Height Control (named THC in the next pages). This function is an important feature of the plasma cutting process, that helps keep a torch-sheet distance stable enough, based on a direct feedback of the field tracking. **A correct cutting height is essential for obtaining quality kerfs**. If too high, the torch is going to scatter the beam, overheat the material around, and the cut edges will be slanted, narrowed at bottom, with a risk of extinguishing the stretched electrical arc. If too low, the torch is going to leave slags and the upper edges will be narrowed, with a risk of collision. Add to this the fact that a metal sheet may curl – especially a thin sheet which dilates faster because its caloric mass is less – and everything is ready for making the cut go wrong. However, on a thick sheet, hence rather flat, laid down-and-out on the machine with no raised edges, the THC becomes less significant. We will come back in detail on the different settings, a bit later on, same chapter.



The field tracking is in fact a simple **measurement of the arc voltage**: the more the torch gets close to the sheet, the more the voltage decreases, until it reaches a null value when both are in contact. Then there is no more arc. Most torch boxes provide measurement plugs for that voltage, which generally varies between 80 and 140 V. It is necessary to first isolate galvanically and divide this voltage for receiving a usable analogue value with no risks of interference for the numerical controller. A plasma torch is already producing enough electrical noise, no need to add more. You should use a good quality isolator-divider, which will divide the voltage by 15 approximately if your analogue input has a range of 0 to 10 V, or divide by 30 if your analogue input has a range of 0 to 5 V. Hence, for an arc voltage of 150 V, the ADC input (*Analogue-Digital Converter*) of your numerical controller will indicate a value of 100 % that will be transmitted to the software. Regardless of the actual resolution of the ADC, what does really matter is the reliability of the measurement.

It is not Galaad which measures in real-time the arc voltage and manages the Z axis for rectifying upwards or downwards. **The reaction loop is purely local**. The software just activates and frames it. Even with a THC correction acting on the Z position, we are still in the 2D area, at least in theory. This is why the controller must integrate a specific function that links the analogue input to Z movements that are added to the coordinate received. Here, Galaad can deal with two operating modes:

1- The Soprolec mode uses a typical voltage value, provided by a chart in a database that is installed with the software and that you may access. The probing is useful only for triggering the torch at the correct height, but then the controller will target the analogue value that has been transmitted by the software. Advantage, the probing accuracy is less significant; drawback, if the voltage value that has been picked from the database does not or no longer correspond, then the whole cutting cycle will be affected.

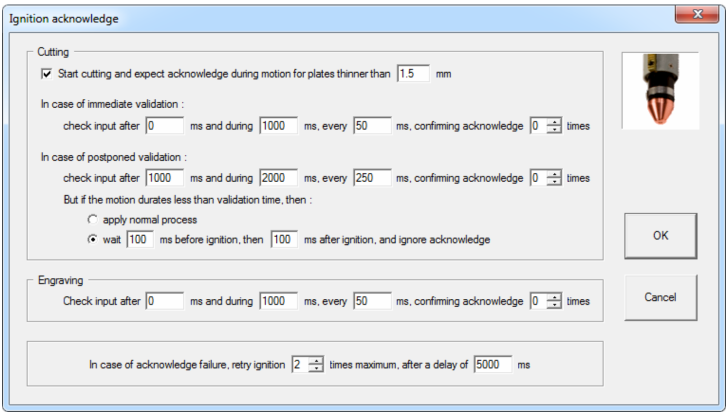
2- The ThunderCut mode considers that the probing gives the correct cutting height and performs a quick sampling of the arc voltage, measured during the first millimeters of the horizontal path. The 2 extreme values are rejected and the controller takes the average of the 32 remaining measurements, which becomes the target. Advantage, if the nozzle is worn out, the current wrongly set or anything else, then the sampling remains valid; drawback, if the probing has been made on a slag or the sampling on a slope, then the cutting height at the beginning or the average voltage are incorrect for the paths until a new probing is performed.

As we will see, Galaad incorporates advanced management of the THC. The purpose is to follow a varying ground, so we can consider that **a path that covers a small surface does not require a Z correction** since it does not go far away from the probing point. In addition, the cutting job on thin sheets is often performed with high speeds. But, as we have seen before, speed induces kinematical constraints, on which we are not going to spend more time except for arguing that the cutting speed will not be fulfilled everywhere. The problem is, when the movement is slowed down, the melted metal being ejected downwards, there is less of it under the torch and the voltage increases, which the controller will interpret wrongly as a raising distance from the surface. Then it is going to rectify downwards, with a risk of collision. So it is necessary to **cut the THC when the speed gets under a percentage of the nominal value**. Of course this percentage is settable in the software.

During the cutting job, the parts of the paths where the THC is enabled are displayed on the screen with a blue halo.

❑ Ignition acknowledgement

When the torch is triggered, there is a little delay before the arc is steady. Since the movement will begin after that, Galaad will wait until the corresponding input gets the signal. But it is actually a bit more complicated, as we will see by clicking on the small neighbouring button *"More…"*, which pops up a new subordinate window for more ignition parameters:



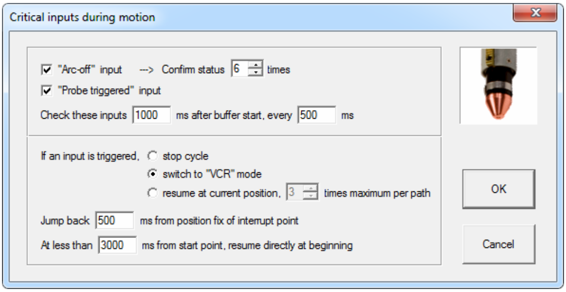
If the software waits too long for the start signal, then the arc is going to dig a big hole at the ignition point, especially if the sheet is thin. But if it does not wait long enough, the torch may move to the cutting height before the arc is stabilised, with a risk of extinction. To resolve this dilemma, Galaad suggest that two cases be managed, discriminated by the thickness of the sheet: **immediate validation** or **postponed validation**. The first case is the simpler: the torch is triggered, then there is a time lapse to be defined, during which the acknowledgement is ignored because there can be noisy feedback when the pilot arc is active, then the input that receives the signal *"Arc OK"* is read periodically during a maximum time interval, and eventually the signal is confirmed. Then only, the Z axis is moved for reaching the cutting height and the horizontal feeding can begin. For example, if you set the line *"check input after 250 ms and during 1000 ms, every 50 ms, confirming acknowledge 2 times"*, Galaad is going to trigger the torch, then wait 250 ms, then enter a reading loop of the acknowledgement input every 50 ms. It must get 1 good feedback plus 2 confirmations, so 3 times in a row a reading that validates the ignition. If a reading is negative, then the counter is reset to zero: the first good reading and the confirmations must follow each other with no change. Finally, if after 250 + 1000 ms, the acknowledgement has not been validated yet, then it is a failure. At the bottom of this window, you can decide what the software should then do. We may simply wait a while because it is possible that the failure be due to an overheating of the torch that the control box secures, and repeat the attempt several times, after which the cycle stops and the operator is warned by an error message on the screen (plus an e-mail if you have set it).

If the sheet is too thin and it is not possible to wait motionless for the acknowledgement, then we must opt for a postponed validation. In this case, Galaad will trigger the torch and start moving the axes without waiting more than the pauses that are present in the database. The validation of the ignition will then be done while the cutting job has begun. If this validation is not obtained, then the torch is switched off, sent back to the start point and the procedure is looped. It can happen that the path is too short so the validation cannot be done. In this case, it is possible to ignore the acknowledgement and replace it by a simple set of pauses before and after ignition.

The same line for the immediate acknowledgement is available for the engraving mode when the current is reduced. The risk for holing the sheet being very unlikely in this case, only the immediate validation makes sense.

Critical inputs

Trigger the torch, validate the ignition in different conditions then follow the Z variations using a THC cover most of the needs, but some other problems may still occur, that we are going to learn by clicking on the small button *"Critical inputs"* in the torch parameters. The first problem is the **sudden extinction of the arc**, generally when it passes over a hole in the sheet. The software response is not urgent here. Galaad just needs to be informed before the end of the path that it has been interrupted. So the input can be monitored with a low frequency, about one second, to avoid overloading the transmission. As long as the validation of the ignition has not been completed, the input *"Arc off"* is not checked, because it is usually the same. This input can require a confirmation, in which case the procedure will be a first reading then the eventual confirmations, all at once, for considering that the arc is indeed off and that it was not just some noisy feedback.

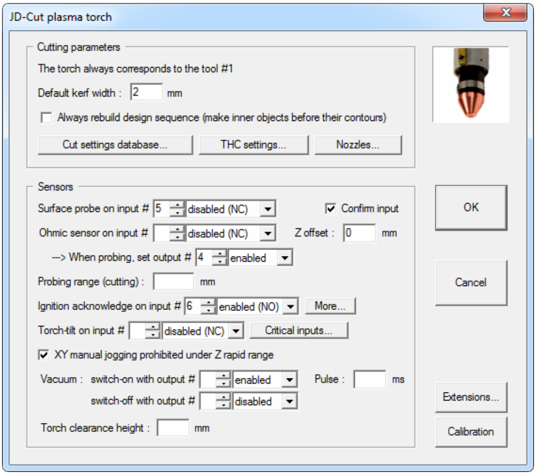


The input that corresponds to the **surface probe** can also be read for security reasons, when there is no THC. If triggered, it is better not to insist and stop the process. But there is a chance that the input *"Arc off"* already reacted since it does not wait until the torch has moved back up.

In case of a critical input trigger, the software is going to stop the motion and switch off the torch. You may choose what must be done then: stop the cycle and send an error message in full and final settlement, or switch to the mode *"VCR-Seek"* so the operator can retrieve manually the resuming point, or also resume alone at the estimated point of interruption, with a maximum number of times so the software does not enter an infinite loop. Since the detection of the critical input is not immediate and the machine must follow a braking ramp, it is possible to go back a little to find the interruption point. Galaad will convert the time lapse in distance according to the cutting speed. At last, if this interruption occurs shortly after the starting point, the resuming can be a simple restarting from zero.

Soprolec controller

The numerical controllers InterpCNC-2 are integrated in the cutting tables of various manufacturers (Air Liquide Welding, JD-Cut, Lincoln Electric, Phenix Technologie, FabTec) and some handmade machines. The parameters are many and spread over several windows and sub-windows. Once again, do not forget that you must validate the stacked parent-windows for saving what you have modified in the offspring.



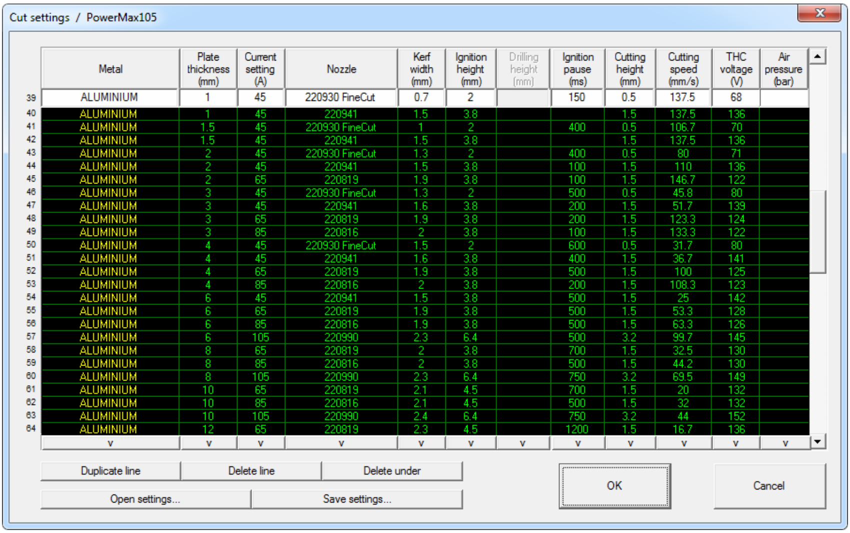
It is possible to pre-drill thick sheet metal. In this case, the software will initiate drills one after the other, then come back to start cutting. Galaad can also control continuous cutting on square or rectangular tubes with rounded edges. To avoid slowing down too much at the edges, you can cheat by multiplying their radius. The rotary movement will be extended around the corner.

For the probing, it is possible to use an **ohmic sensor** which has the advantage of not pressing the sheet, which can distort the measurement. The drawback is that such sensor is subject to interference from water on the sheet or a variation of the electrical resistance due to the material and its oxidation state. The software can manage both the recoil sensor of the torch and the ohmic sensor for more security. In the latter case, a small fixed value for the Z calibration allows you to add or subtract to the Z coordinate that the probe defines. Since that ohmic sensor will probably not appreciate too much the 150 V of the arc, it is safe to put in an isolator relay, driven by an output during the probing. The option for confirming the input corresponds to a re-reading after the probing plunge has stopped: if the input has in fact not been triggered, then Galaad redoes the probing.

Let us skip the ignition acknowledgement, already seen. An input can be assigned to a **torch-tilt** sensor, for stopping any movement if the torch collided with an obstacle. Since there is a reaction delay, and even more when it must inform the software, it is better to cut the main power supply to the machine when the torch is tilted. It is much safer and costs less in case of a crash. Let us skip also the critical inputs, already seen too.

While calibrating the torch recoil sensor, it is possible to define a rapid Z plane. When the torch is plunging towards the surface, it can begin with a fast movement down to a fixed height plus the sheet thickness. The slower probing starts here. That height can also be reused for locking XY movements when jogging the axes manually, which could damage the torch due to a collision after a wrong operation: if the option **XY manual jogging prohibited under Z rapid plane** is enabled, then the operator will be allowed to move the XY axes only when the torch is above this critical height.

Above the Soprolec torch parameters, several buttons give access to different functions, the first being the **cut settings database**, which opens a sub-window featuring a data chart:


Cut settings / PowerMax105

	Metal	Plate thickness (mm)	Current setting (A)	Nozzle	Kerf width (mm)	Ignition height (mm)	Drilling height (mm)	Ignition pause (ms)	Cutting height (mm)	Cutting speed (mm/s)	THC voltage (V)	Air pressure (bar)
39	ALUMINIUM	1	45	220930 FineCut	0.7	2		150	0.5	137.5	68	
40	ALUMINIUM	1	45	220941	1.5	3.8			1.5	137.5	136	
41	ALUMINIUM	1.5	45	220930 FineCut	1	2		400	0.5	106.7	70	
42	ALUMINIUM	1.5	45	220941	1.5	3.8			1.5	137.5	136	
43	ALUMINIUM	2	45	220930 FineCut	1.3	2		400	0.5	80	71	
44	ALUMINIUM	2	45	220941	1.5	3.8		100	1.5	110	136	
45	ALUMINIUM	2	65	220819	1.9	3.8		100	1.5	146.7	122	
46	ALUMINIUM	3	45	220930 FineCut	1.3	2		500	0.5	45.8	80	
47	ALUMINIUM	3	45	220941	1.6	3.8		200	1.5	51.7	139	
48	ALUMINIUM	3	65	220819	1.9	3.8		200	1.5	123.3	124	
49	ALUMINIUM	3	85	220816	2	3.8		100	1.5	133.3	122	
50	ALUMINIUM	4	45	220930 FineCut	1.5	2		600	0.5	31.7	80	
51	ALUMINIUM	4	45	220941	1.6	3.8		400	1.5	36.7	141	
52	ALUMINIUM	4	65	220819	1.9	3.8		500	1.5	100	125	
53	ALUMINIUM	4	85	220816	2	3.8		200	1.5	108.3	123	
54	ALUMINIUM	6	45	220941	1.5	3.8		500	1.5	25	142	
55	ALUMINIUM	6	65	220819	1.9	3.8		500	1.5	53.3	128	
56	ALUMINIUM	6	85	220816	1.9	3.8		500	1.5	63.3	126	
57	ALUMINIUM	6	105	220990	2.3	6.4		500	3.2	99.7	145	
58	ALUMINIUM	8	65	220819	2	3.8		700	1.5	32.5	130	
59	ALUMINIUM	8	85	220816	2	3.8		500	1.5	44.2	130	
60	ALUMINIUM	8	105	220990	2.3	6.4		750	3.2	69.5	149	
61	ALUMINIUM	10	65	220819	2.1	4.5		700	1.5	20	132	
62	ALUMINIUM	10	85	220816	2.1	4.5		500	1.5	32	132	
63	ALUMINIUM	10	105	220990	2.4	6.4		750	3.2	44	152	
64	ALUMINIUM	12	65	220819	2.3	4.5		1200	1.5	16.7	136	
	v	v	v	v	v	v	v	v	v	v	v	v

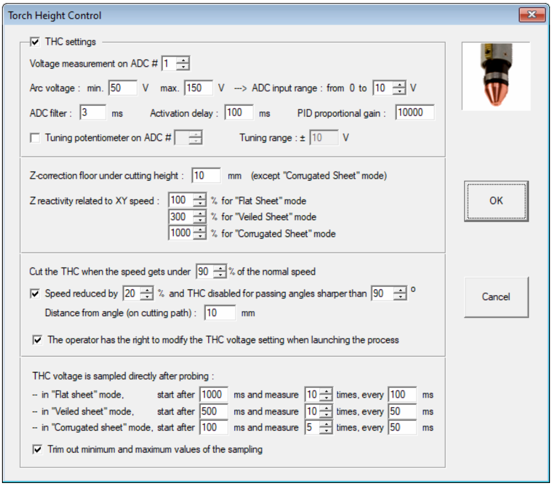
Duplicate line | Delete line | Delete under | OK | Cancel

Open settings... | Save settings...

This database is installed along with the software. It contains settings for cutting jobs depending on the material (steel, inox, aluminium), its thickness, the current and the nozzle to be used. The operator provides information about the two first items and Galaad suggests the next on the same line. This database is not frozen. You may modify the values at your convenience for adapting them to your machine, your torch or your nozzles. However, we suggest that you save your own settings into a file with an alternate name: if you need to reinstall the software, you will retrieve the data by default (on the other hand, updating Galaad does not affect your personal data).

In a few words, the **current setting** is the value you must set on the torch box before launching the cycle, unless this setting is under software control through a DAC or PWM output or an RS-485 port. The **nozzle** depends on what consumables you have at your disposal. The **kerf width** is the resulting cut groove, whose value will let Galaad rectify the default kerf of the drawing. The **ignition height** is set relative to the sheet surface. The **drilling height** is used only by Air Liquide Welding and Lincoln Electric machines, which are fitted with torches that require an intermediate transfer point for managing the pilot arc. The **ignition pause** is the time lapse immediately after the torch ignition, before the acknowledgement procedure begins. The **cutting height** is the optimal torch-sheet distance for a quality cutting job. The **cutting speed** is the XY feedrate along the paths. The **THC voltage** is the target value that Galaad will send to the numerical controller when the THC mode is enabled, correcting the height by moving the torch up or down so it sticks to that voltage. The **air pressure** is the compressed air setting the torch must be provided with.

Still in torch parameters, the button for **THC settings** gives you access to the THC internal functions. You must indicate here which analogue input receives the arc voltage, isolated and divided, and its validity range. Galaad will use this for converting the ADC value into Volts, and also reciprocally for sending the THC instruction to the controller. The activation delay and the PID proportional gain are internal data for the Soprolec card, which is best not to modify. These two settings are given here for granting access to specific applications. If your machine is fitted with a **Z tuning potentiometer** for adjusting the cutting height during the process, then you must indicate here to which analogue input it is connected, and its variation range, on the basis that the median position corresponds to a null correction. This potentiometer is supposed to rectify the deviations of a tired nozzle. Do not forget to wear good welding goggles if you follow the cutting job close to.

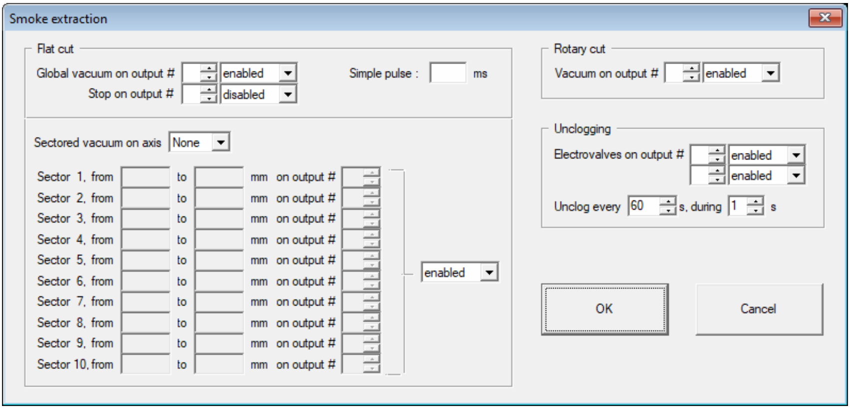


The **Z correction floor** defines a maximum offset between the cutting height as measured by the probe, and the lowering that is driven by the THC. We can consider that this offset cannot increase infinitely downwards. But this floor is not applicable when the operator chooses a corrugated sheet. The **Z reactivity** related to the XY speed sets the reactive power of the THC correction. On a veiled sheet, and even more on a corrugated sheet, it will be much more responsive than on a flat sheet, so you can define a field tracking with a soft or hard reactivity.

The THC may be switched off when the real cutting speed falls below a minimum percentage of the theorical cruise speed, to prevent the torch from plunging when it slows down in tight turns, the lack of metal increasing the arc voltage, which is wrongly interpreted by the controller as a raising distance from the surface. For managing sharp angles, generally during the tool compensation overruns, the speed can be reduced and the THC switched off at some distance away from the angle. This also avoids leaving the THC active, when the angle overrun in a triangular cusp causes an overlap, with the torch crossing again the path it has just traced, causing a brief increase of the arc voltage and an attempt for correcting immediately, which will make the Z axis plunge uselessly.

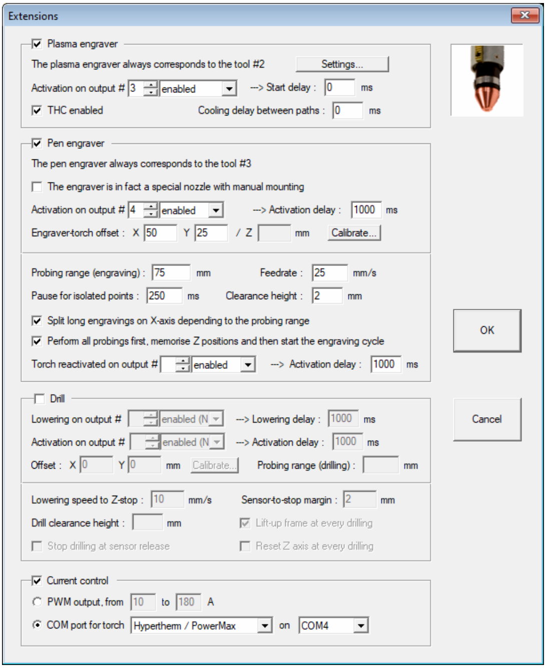
Still at the top of the parameters dialogue box concerning the Soprolec torch, the management of the **nozzles** by Galaad is detailed later on in this same chapter.

At the bottom right, the *"Vacuum"* button gives access to output management for a compartment hoover, to avoid vacuuming in a vacuum.



Here you can define an output for triggering the hoover (a large relay is required), possibly via a device with arming, as well as a specific output for 4-axis rotary cutting suction, since it is rare to rotate cylindrical parts immersed in water. The table can be divided into sectors along an axis, with an output triggered for each sector when the torch is operating in that sector. Finally, Galaad can control a device for **unclogging** the filters, by switching solenoid valves at regular intervals.

Bottom right again, a button *"Extensions"* gives access to some related functions, and in particular the **engraving devices**. Two possibilities: a **plasma engraver** directly on the torch, or a **pen engraver** mounted on an actuator beside the torch and working independently. The plasma cut always corresponds to the tool number 1 in the drawing. The engraving with the plasma torch corresponds to the tool number 2. As with the cutting job, a database for the engraving settings is available for the three usual metals and depending on the nozzles to be used. Switching the torch to engraving mode may require an additional output, with its own pause delay after ignition.The THC can also be used in engraving mode. Lastly, you may impose a cooling delay between paths, to avoid ignition problems of the pilot arc due to an overheating.

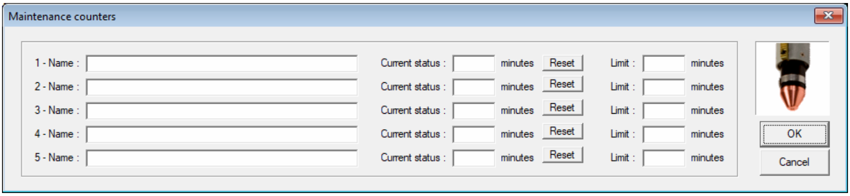


The pen engraver is a device to be mounted beside the torch, with a vibrating hard tip driven by a piezoelectrical or compressed air oscillator. It corresponds to the tool number 3 in the drawing. The pen is lifted up and lowered by an actuator that requires a motion delay. It is the Z axis which moves up and down at every new path, but the actuator is lifted up when the probing must be redone, depending on the distance from the previous probing and its operating range (which can be different from the one used for the cutting job). The torch remaining the reference tool, the XY offset of the pen relative to it must be calibrated. There are no databases for the pen engraver and the feedrate must consequently be set here once and for all, and also its own clearance height (if it is not indicated, then Galaad reuses the one for the cutting job). Finally, when the path to be engraved is an isolated dot, then the software needs to know how long it must remain still so the dot is engraved.

In addition to the torch (tool #1 in the drawing), engraving with the torch (tool #2) and engraving with a vibrating tip (tool #3), the software can manage a drill head with drills as tools #4 and following, to be integrated into the tool shop. In this case, you draw the points or paths and Galaad will place them between engraving and cutting in the sequence.

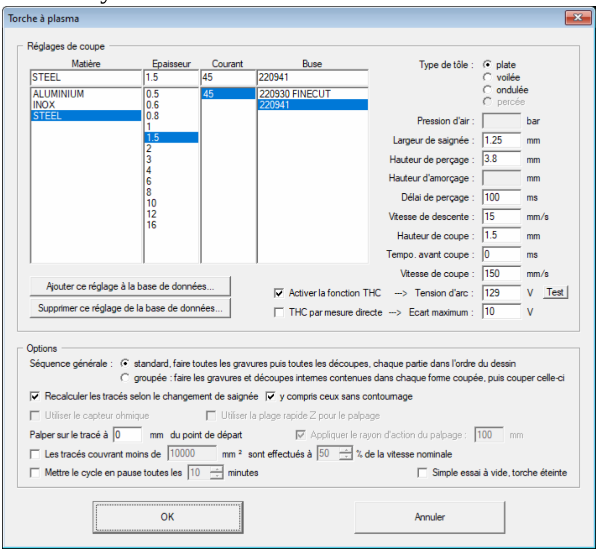
At the bottom of this window, the **current control** allows Galaad to drive directly the amperage without having to set it manually on the torch control box. This control can use either a PWM output between 0 and 100 %, with the current range to be indicated, or a command from an RS-485 port.

The *"**Maintenance**"* button helps you manage machine maintenance. You give a name to a maintenance operation and set a time limit in minutes. Once this time limit has been exceeded, a message will appear at the start or end of the automatic process, to remind you of the work to be done.



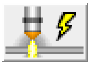
Last button at the bottom right of the parameters dialogue box of the Soprolec torch, the **calibration** allows you to measure the offsets between the torch on one hand, and the additional devices such as the laser spot or the pen engraver on the other hand. Lower the torch down so it touches the surface, validate its position, draw a circle around its nozzle, retract it up, bring the device accurately at the centre of the circle and validate it. That is all.

When lauching the cycle, the screen displays a window that allows you to choose the **cut settings** (or the engraving, or both). You just have to select the **material** and its **thickness**. The database is going to help Galaad set all the rest. But you can choose also the **current** and the **nozzle** to be used, or change any resulting value on the right hand side. Do not forget to indicate if the metal sheet is **flat**, **veiled**, or **corrugated**, so the software will consequently apply the corresponding THC reactivities and eventually keep the pilot arc. The **kerf width** depends on the nozzle and the current. If it differs from the default



width that has been defined for the drawing, then the compensated paths are rectified using a simple offset, unless you disable the corresponding option at the bottom of the window.

An **important function** for the setup is hidden behind the small, barely noticeable ***"Test"* button** on the right. When this function is active, the THC will not be applied to the first trace to be cut, but Galaad will take ADC measurements after probing, the Z height then being deemed correct, and then report back. This allows you to check that the voltage value in the database corresponds to the case in question: metal, thickness, current and nozzle. Whether using this test mode or on your own initiative, if you modify the cut settings in the right-hand column, then you can **add them directly to the database**. You can also delete a setting that you consider unnecessary.



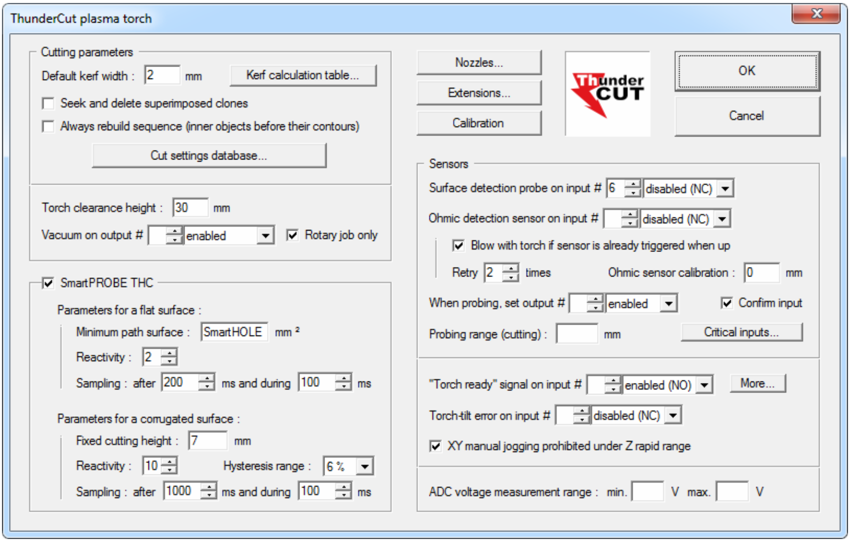
Tip: in basic manual control (*i.e.* not setting a workpiece origin), when you request the torch to be switched on using the control button opposite while pressing the Ctrl key on the keyboard, an **arc voltage measurement** window opens. Take the torch to a metal sheet and ask for a cut to a target position, say 10 cm away. Galaad will take care of the probing, ignition and feed while measuring the voltage through the ADC. The final report can help you fine-tune your THC. A sub-window offers a **cutting test** with variable heights and speeds.

Among the few cutting options, it is possible to disable the rapid plane for probing if you think there is a risk of collision, in which case the probing will start from the top of the Z axis. You may also disable the probing range, and then it will be performed at every new path. Also, the paths that cover a small surface can be executed at slow speed relative to the normal cutting speed, which may induce a THC switch-off.

If there is an engraving job to be performed with the plasma torch, some similar settings and options will be displayed, either in a reduced window if there are no cutting paths, or in the same window, just extended.

❑ ThunderCut controller

The ThunderCut cutting tables are fitted with an AxeMotion controller that integrates the THC function. Here again, the parameters are many and spread over several windows and sub-windows. Do not forget that you must validate the stacked parent windows for saving what you have modified in the child sub-windows.



In addition to the kerf width and the surface probe already seen above, it is possible that Galaad **seeks and deletes the clones**. In this case, when validating the drawing, the software will search for the overlaid copies. Time-consuming for the CPU, this calculation will be long on loaded drawings. It is not very useful if you are drawing with Galaad or if you have used the same filtering function when importing. The option for **rebuilding the sequence** is going to put the inner cuts before the outer cuts that surround them, to avoid letting a part of the workpiece fall out before its islands are made.

The torch clearance height can be set here, in which case the operator will not be allowed to modify it. If you enter no value, then it may be set classically in the machining window. An output for switching the **vacuuming** can be defined, possibly just for the cylindrical cutting jobs on 4 axes. That output is triggered at the process start and released at the process end.

The **SmartProbe** functions define how the THC is going to be used. When starting the cycle, the operator must select the type of metal sheet: flat (or almost) or corrugated. On a flat sheet, it is possible to disable the THC on paths that cover a reduced surface, since in that case the torch does not go far away from the probing point. If you indicate no surface value, then the operator will be allowed to set it. The **reactivity**, in a range from 1 to 10, gives the velocity of the Z correction by the THC. Too weak a value induces a soft reaction, with a risk of extinguishing the arc if it is moving away from the sheet; too strong a value induces a hard reaction which will make the axis jump over small slags or wrong measurements. The **sampling** is performed just after the feed start, on a brief duration. It is probably better to keep the default values. On a corrugated sheet, there are no variable cutting heights picked up from the database, but a fixed value to be set here. The **hysteresis range** is then definable, which gives a threshold for activating the THC correction to avoid oscillations.

In addition to the surface detection by the torch itself, it is possible to use an **ohmic sensor** that offers the advantage of not pressing the sheet, which can distort the measurement. The drawback is that such sensor is subject to interference from water on the sheet or a variation of the electrical resistance due to the material and its oxidation state. The software can manage both the recoil sensor of the torch and the ohmic sensor for more security. In the latter case, a small fixed value for the Z calibration allows you to add or subtract to the Z coordinate that the probe defines. Since that ohmic sensor will probably not appreciate too much the 150 V of the arc, it is safe to put in an isolator relay, driven by an output during the probing. The option for confirming the input corresponds to a re-reading after the probing plunge has stopped: if the input has in fact not been triggered, then Galaad redoes the probing. If the ohmic sensor is already triggered though the torch is in an upright position, then we can infer that it is wet and the software will briefly activate the torch, which will blow a compressed air jet in the nozzle without igniting the arc.

Let us skip the ignition acknowledgement, already seen. An input can be assigned to a **torch-tilt** sensor, for stopping any movement if the torch collided with an obstacle. Since there is a reaction delay, and even more when it must inform the software, it is better to cut the main power supply to the machine when the torch is tilted. It is much safer and costs less in case of a crash. Let us skip also the critical inputs, already seen too.

When calibrating the torch recoil probe, it is possible to define a rapid Z plane. Then the probing procedure begins with a fast motion down to a fixed height increased by the sheet thickness. The actual probing, slower, then starts from that height. This height is also usable for locking the XY movements in manual mode, which might damage the torch if colliding with something during an improper handling: if the option for **prohibiting XY manual jogging under Z rapid plane** is enabled, then the operator will be allowed to move the XY axes only when the torch is above that critical height. Lastly, you can define a measurement range for the electrical arc, which values will be used for displaying the Volts at the bottom of the screen.

Back to the top of the dialgue box, the button for accessing the **kerf calculation table** pops up a small chart for correlating the current, the kerf width and eventually the PWM value if the amperage is under such control. At last, another button gives access to the **cut settings database**, which displays a chart window:

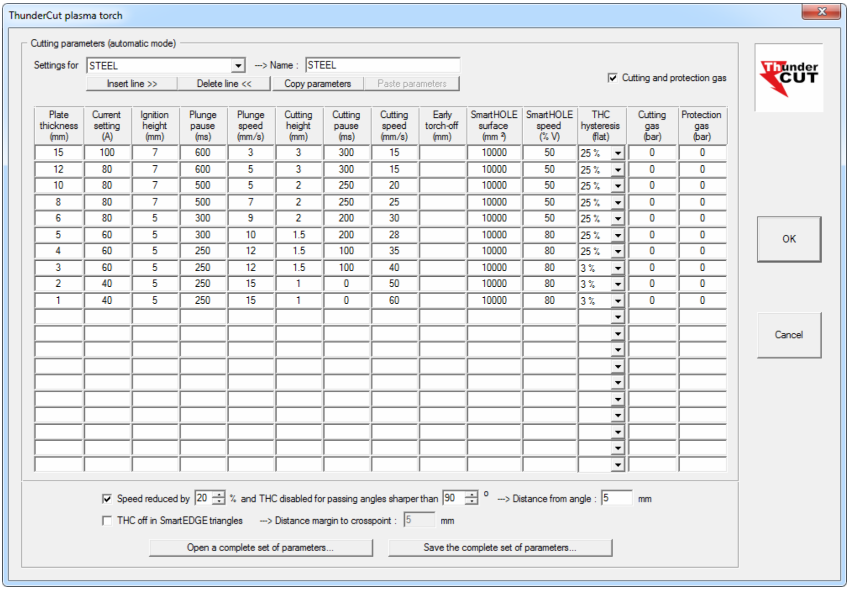


Plate thickness (mm)	Current setting (A)	Ignition height (mm)	Plunge pause (ms)	Plunge speed (mm/s)	Cutting height (mm)	Cutting pause (ms)	Cutting speed (mm/s)	Early torch-off (mm)	SmartHOLE surface (mm²)	SmartHOLE speed (% V)	THC hysteresis (flat)	Cutting gas (bar)	Protection gas (bar)
15	100	7	600	3	3	300	15		10000	50	25 %	0	0
12	80	7	600	5	3	300	15		10000	50	25 %	0	0
10	80	7	500	5	2	250	20		10000	50	25 %	0	0
8	80	7	500	7	2	250	25		10000	50	25 %	0	0
6	80	5	300	9	2	200	30		10000	50	25 %	0	0
5	60	5	300	10	1.5	200	28		10000	80	25 %	0	0
4	60	5	250	12	1.5	100	35		10000	80	25 %	0	0
3	60	5	250	12	1.5	100	40		10000	80	3 %	0	0
2	40	5	250	15	1	0	50		10000	80	3 %	0	0
1	40	5	250	15	1	0	60		10000	80	3 %	0	0

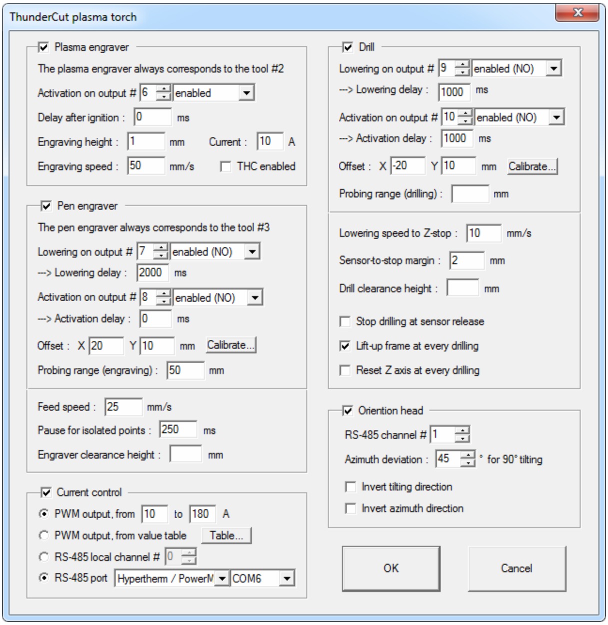
The software can manage up to 10 different materials, but only three are preset: steel, inox and aluminium. For each of them, you can associate up to 20 chart lines indicating, for different sheet thicknesses, the typical cutting values. The default database is installed with the software, but these lines are not frozen: you may modify the values as you like for adapting them to your machine, your torch or your consumables. However, it is better to save your changes to your own file: if you need to reinstall the software, you will return to the data by default (but on the other hand, updating does not alter your personal data).

In a few words, the **current setting** is the value you must set on the torch box before launching the cutting process, unless this setting is under direct control through a PWM or RS-485 output. The **ignition height** is set relative to the sheet surface. The **plunge pause** is the time lapse after triggering the torch and before checking its acknowledgement, if any. The **plunge speed** corresponds to the movement between the ignition height and the cutting height. The **cutting height** is the optimal torch-sheet distance for a quality cutting. The **cutting pause** occurs immediately after the plunge, just before starting the horizontal feed. The **cutting speed** is the XY feedrate along the paths. The **early torch-off** is the distance from the end point for switching the torch off. The **SmartHole surface** is the surface covered by a path, under which the THC will remain disabled unless that surface is indicated in the general parameters of the parent window. It is assigned a specific **SmartHole speed**, that will not consider the normal cutting speed which applies to larger surfaces. The **THC hysteresis** gives the THC correction threshold to avoid oscillations. It does not apply to corrugated sheets which use a unique setting in the parent window. The **cutting and protection gas** remembers the operator about the settings for the gas inputs, if they are part of the process.

The THC may be switched off when the real cutting speed falls below a minimum percentage of the theorical cruise speed, to prevent the torch from plunging when it slowers down in tight turns, the lack of metal increasing the arc voltage, which is wrongly interpreted by the controller as a raising distance from the surface. For managing sharp angles, generally during the tool compensation overruns, the speed can be reduced and the THC switched off at some distance away from the angle. This also avoids leaving the THC active, when the angle overrun in a triangular cusp causes an overlap, with the torch crossing again the path it has just traced, causing a brief increase of the arc voltage and an attempt for correcting immediately, which will make the Z axis plunge uselessly.

At the top right, a button *"Extensions"* gives you access to related functions, and particularly the **engraving and drilling systems**. Two possibilities exist for engraving: a **plasma engraver** directly from the torch with reduced current setting, and a **pen engraver**, mounted on an actuator beside the torch, working independently. The plasma cutting always corresponds to the tool number 1 in the drawing. The plasma engraving corresponds to the tool number 2 and the pen engraving to the tool number 3. These are constants that you cannot change. Switching to engraving mode with the torch requires an additional output, with a specific pause time after ignition, an engraving height, a speed and a fixed current (no database in this case). The THC remains available.

The pen engraver is a device to be mounted beside the torch, with a vibrating hard tip driven by a piezoelectrical or compressed air oscillator. It corresponds to the tool number 3 in the drawing. The pen is moved vertically by an actuator which requires a movement delay. It is the Z axis which moves up and down at every new path, but the actuator is lifted up when the probing must be redone, depending on the distance from the previous probing and its operating range (which can be different from the one used for the cutting job). The torch remaining the reference tool, the XY offset of the pen relative to it must be calibrated. There are no databases for the pen engraver and the feedrate must consequently be set here once and for all, and also its own clearance height (if it is not indicated, then Galaad reuses the one for the cutting job). Finally, when the path to be engraved is an isolated dot, then the software needs to know how long it must remain still so the dot is engraved.



At the bottom of that window, the **current control** allows you to manage the amperage directly from the software so there is no need for the operator to set it on the torch box. This control can use: a PWM output between 0 and 100 % with the range of current to be indicated, or the same PWM output using a mapping table, or a command throughout an RS-485 channel of the numerical controller, or a direct command from an RS-485 port of the PC.

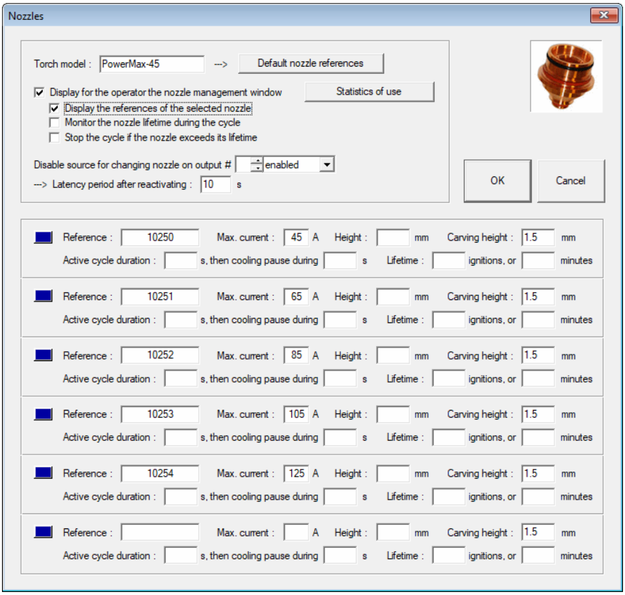
ThunderCut machines offer as an option a **classical drill**. Every drill bit must be accurately defined in Galaad, it being fully understood that the three first numbers are reserved for the torch, the plasma engraver and the pen engraver, even if these are not present. Hence the drill bits begin with the tool number 4. The drill is mounted on an actuator so it can be moved lower than the torch, using an output for that actuator and another one for activating the motor, which can be the same for both functions. The probing is undertaken as for the torch with the drill bit end, the input for the recoil sensor being shared with the torch. The actuator is not lifted up when probing. On the other hand, the drill motor is off at that moment, even if the only risk is to drill a hole, which by the way remains the purpose for that job. The plunge speed is fixed, to be set here, and also the margin between the sensor trigger and the upper mechanical stop. So Galaad knows the recoil distance for the drill after the sensor is triggered and until it presses the upper stop for drilling efficiently. The plunge depth corresponds to the material thickness.

If your torch is mounted on a **tilt & turn head** under the control of a couple of servomotors that are driven by an RS-485 channel of the numerical controller, then Galaad enables the mode *"orientation head"* in the drawing, which allows you to define, for each path, a tilting angle so you get oblique cut edges towards the inside or outside, still subject to the kerf compensation. The calculation will be performed internally and dotted lines will be displayed so you can visualise where the oblique path matches the bottom of the workpiece, provided that the thickness indicated is correct.

When launching the cycle, Galaad displays a window that allows the operator to choose the **cut settings**. You just have to select the material, indicate its thickness and the type of surface (flat or corrugated). The corresponding parameters are automatically picked from the database and displayed in the right hand column, because it is the action centre.

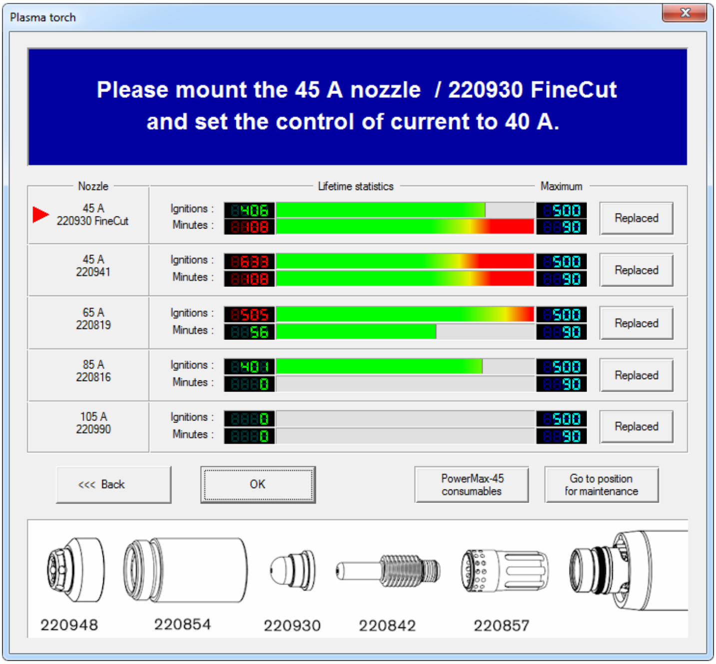
❑ Nozzle management

Whatever the controller you are using, Soprolec or ThunderCut, Galaad can assist you in tracking the ageing of your cutting nozzles and other consumables. When starting a cutting job, the software displays a message

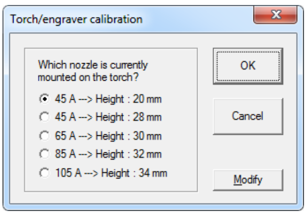


asking for the mounting of one of your nozzles and monitoring statistics such as the duration of use and the number of ignitions. That message shows at the bottom a button *"Consumables"* with the torch model you have set. If you click on that button, then you get on the screen its references and image. Example: you have indicated as torch name *"PowerMax-45"*. The button becomes *"PowerMax-45 consumables"*. If you click on it, Galaad will seek, in the subdirectory \CONFIG of the software, the image POWERMAX-45 REFS.BMP and, if found, displays it on the screen. Under that button, at the bottom of the message window, you can also have the images of the nozzles. For example, if you enter as a reference a nozzle *"220930 FineCut"*, when launching the cutting job, Galaad will seek the image NOZZLE-220930FINECUT.BMP still in the subdirectory \CONFIG and, if found, display it at the bottom of the window when the nozzle is selected. The spaces are ignored in the file name, as well as the case.

For each nozzle, in addition to the **reference name**, you may indicate a **colour** that will be the background of the flashing message to the operator, to facilitate the selection of the correct nozzle, and a **maximum current** that will help pre-selecting it in the cut settings. The **coolant pause** will stop the torch after one or several paths to let it cool down if necessary before resuming the process. The **lifetime** in number of ignitions and duration is the base for the statistics that are displayed just before starting the cycle. Galaad points out in red the overruns but will not prevent the use of a nozzle beyond its lifetime. Up to you to manage your business.

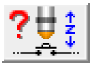


Still for each nozzle, the **height** value looks pointless at first sight since the probing is performed with the torch touching the sheet surface, whatever the nozzle can be. But in fact, with a pen engraver, the Z calibration depends on the nozzle that has been mounted. Galaad considers that the nozzle number 1 is the reference and therefore it is the one you must use for calibrating the probe-pen Z offset. If, later on, you have mounted another nozzle, when probing before engraving, the software will use the calibration by adding the height of the current nozzle and subtracting the height of the nozzle number 1. So, if you are starting an engraving job with no associated cutting job, then you will be prompted to indicate which nozzle is mounted under the torch.

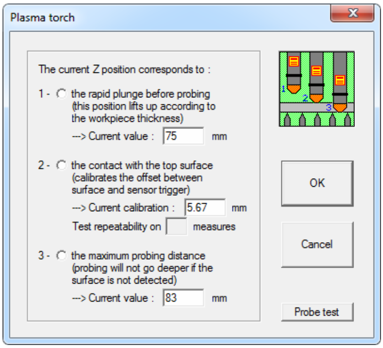


Calibrations

When mounted on a vertical slide rail, the torch becomes the probe directly with its nozzle and a recoil sensor. It is obviously necessary that the sensor should be calibrated, namely measure the Z distance between the contact with the sheet surface and the trigger point, such as a milling tool sensor. **This calibration is accessible only from the basic manual control** (not in the workpiece origin window). The process is simple: bring the torch down in contact with any surface, firm enough not to move when the nozzle pushes it. Be very accurate, for example by trapping a thin paper so you can find the position where the torch touches the surface without beginning to move back up in its slide rail.

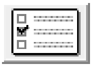


Now click on the calibration button. If you have configured a pen engraver or a drill, indicate which one you are currently calibrating. But let us consider that it is indeed the torch which is being calibrated. A dialogue box then asks you what the Z position corresponds to. Click on the radio-button ***"the contact with the top surface"*** and validate. The software is going to note the Z position and then lower the torch until its recoil triggers the sensor, note the trigger point and memorise the offset between both coordinates. From now on, moving the torch down until its sensor is triggered and subtracting the offset will be sufficient for finding the position that would make it gently touch the surface, and consequently the coordinates for the ignition and the cutting job. In the same way, you may lower the Z axis down until it reaches an intermediate position that you can calibrate as being the **rapid plunge before probing**. When performing a surface probing, Galaad will move the torch down quickly to that Z position **minus the sheet thickness** and will start the probing plunge from that point. That Z coordinate is reused for forbidding the XY manual jogging when the Z axis is underneath, provided that you have enabled the corresponding option in the parameters. Finally, you may lower the torch until it touches the machine bed and use the third calibration option ***"maximum probing distance"***. The software will set it as a floor and, if a probing goes underneath, then it will consider that the probing has been made in a hole and the process will be stopped.

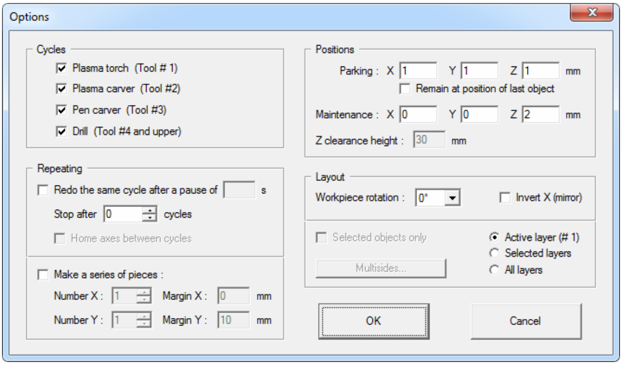


❑ Workpiece origin and related functions

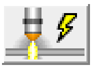
The workpiece origin is simplified when Galaad works in plasma mode. There are no more tabs at the top of the window and the machining parame-



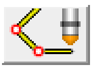
ters are all gathered together in a dialogue box which can be accessed from the first command button at the top left. You can choose the **cycle** which is about to start, if there are several of them among the cutting, engraving and drilling jobs. Galaad begins with the engraving, then the drilling, and the cutting is the last.



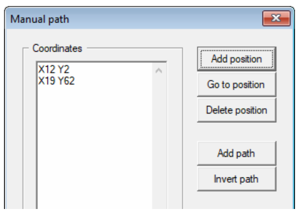
The classical options for repeating the cycle are present, and also the parking and maintenance positions. The **Z clearance height** of the torch can be set here, but if a value has been indicated in the parameters, then it is imposed and the operator cannot change it, the edit box is greyed out. The other parameters for the layout and the filtering are classical, already seen in the machining advanced functions.

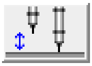


The torch ignition button is at your disposal, but be careful with this. Even if the software is cautious and pops up a confirmation message before activating the torch, to prevent an assistant-operator from an electrical shock. A plasma torch is not a harmless toy; please always think about your own safety and that of your colleagues, particularly concerning your eyes, your hands and your lungs.



The direct plot button allows you to **record XY points** on the table for building a polyline that will then be executed in its entirety with probing and torch activated. This aims to detach a part of the skeleton of the workpiece already machined without using a disc cutter nor make a new drawing.



  When in basic manual control, some additional buttons let you operate the actuator for the engraving pen or switch the vacuum cleaner on, if they exist.

There is no green button Z - ok for the workpiece origin in plasma mode, even if you can still move the Z axis as you like. The workpiece origin Zo is validated by the probing operation for every path or almost, depending on the probing range that has been set in the parameters. So you do not have to worry about it, at least if you have calibrated your probe. You can also use the restricted or full display mode with the small button <<< / >>> at the top right of the workpiece origin frame. The function for **testing the XY workpiece origin** with the cyan button allows X Y - Test you to check that a cutting job will fit in an already machined workpiece part. The torch will not be activated and the coordinates used correspond to the laser spot (or the video camera) to facilitate the tracking. The **double-origin X2Y2** remains valid for a plasma cut and even more useful that the metal sheets thrown on a table are rarely positioned against Cartesian stops. So you can simply validate the southwest corner as XY origin, then the southeast corner as secondary origin for adjusting.

Note: **the *"VCR-Seek"* function is very useful in plasma mode** for resuming an interrupted process due to the loss of the arc or any other incident. Please refer to the chapter *"Advanced milling functions"* for more information about it.

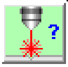
14

0 1 1 1 0

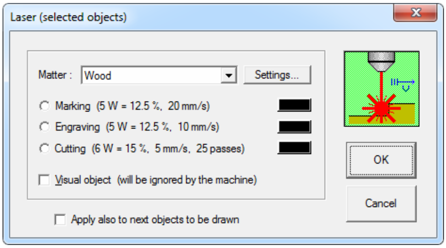
SPECIAL APPLICATIONS

❑ Laser engraving and cutting

If you have selected a laser head as spindle model, then Galaad is going to modify part of the drawing window and a bit more the machining window.

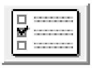


The green icon for choosing the tool, the depth and the feedrate is changing its look. The underlying dialogue box also changes, so that only three modes become available: **marking**, **engraving or cutting**. You may associate a colour to each of them for identifying easily the objects on the drawboard. The laser power

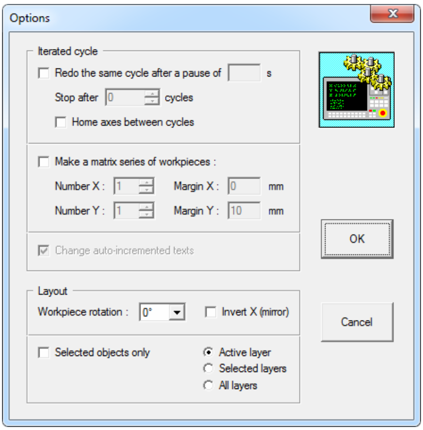


and the number of passes are related to a **type of material** for which you can add or modify the existing data by clicking on the small button *"Settings"* at top right. **The maximum power of the laser is assumed to have been indicated** in the *"Spindle"* page of the machine parameters (beside the model line). According to this, each mode is assigned a relative power – which will be converted to a percentage of the maximum power – and a feedrate. To prevent the material from burning or even catching fire, it is better to operate successive passes instead of feeding slowly.

The workpiece origin is simplified when Galaad works in laser

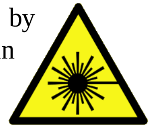


mode. There are no more tabs at the top of the window and the machining parameters are all gathered together in a dialogue box which can be accessed from the first command button at top left. The tools no longer exist, neither therefore the overall sequence, but Galaad will always begin with the marking, then the engraving, and the cutting is the last. The options for **repeating** the cycle remain present.



In the spindle parameters, it is possible to select an analogue laser engraver, whose power will be modulated directly by the depth of the objects, with a fixed maximum value. In this case, the laser mode with its user interface is not enabled and the classical milling functions remain displayed. You still set the depths and speeds, even if the tool concept is no longer significant.

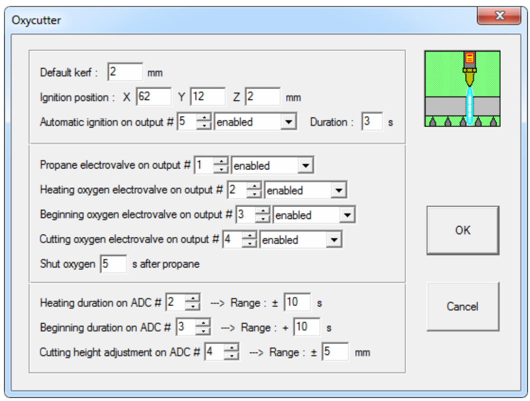
Very important reminder: **a laser is dangerous**, and even more because its use looks easy. But a simple reflection of the beam on a surface or even shimmering dust can cause irreversible damage to the eye by burning the retina, and as a consequence a black spot for life in the field of vision. It is very easy to find class 4 laser diode sources with a power of several Watts, that can be used as pyrography device or small numerical cutter. Working with a laser supposes that your machine is fitted with certified protection equipment or that you **wear good quality protective goggles**. For testing them, just send the laser on a border (because they will be damaged there) and check that the beam does not pass through by putting a paper sheet behind. If the paper is marked, even a little bit, then you can consider your glasses are an illusory protection.



A waterjet cutting system, seen from Galaad, does not differ from a laser system. The changes to the user interface and the general operation remain identical.

❑ Oxycutting

Some specific parameters are available in the software for managing an oxycutter. As for the plasma torch, the user interface of the machine control is simplified and a default kerf must be indicated, which will be rectified eventually when launching the cutting process. The use of an ignition device is possible, at a given XYZ position, by a pulse on an output. A gas electrovalve and three



oxygen valves can be driven. If the numerical controller provides several analogue inputs, it is possible to plug in potentiometers that give the delays for heating and for adjusting the cutting height, according to range values to be defined. Like for the cutting jobs using a basic plasma torch, the operator can set working heights, feedrates, *etc.* while building his own database.

❑ Liquid dispensing

If most usual Galaad applications consist of milling a workpiece by removing material during the paths drawn, the software is nonetheless capable of controlling a system in charge of adding it at the same locations. It is even possible to deal with both cases one after the other, for example by engraving tracks and then injecting ink or paint. Of course, it that particular case, it will be necessary to have an injection nozzle beside the milling spindle, with an XYZ offset fully calibrated between them. The extensions of the tool parameters allow you to manage such a situation easily, by assigning different outputs to the triggers of the spindle and the nozzle. In addition, a script for an automatic tool change can consider the respective Z heights of both items. It is also possible to assign differents tools to different injectable tints, still with the extensions in the tool parameters. Simply, each electro-valve or pump must be connected to the corresponding output.

But there is no compulsion to perform a milling process before injecting the liquid. Another similar application with Galaad consists of dispensing glue or an equivalent in the paths of a drawn workpiece, with no preliminary machining. The extended inputs/outputs allow you to closely manage the dispensing cycle, by waiting a little after the opening of the valve or the injection pump, and shutting off this valve a bit before reaching the final point of the path, so the remaining liquid can flow out. This is probably not necessary for all liquids; some will be very happy with a classical trigger at the path start & end points.

❑ Other intermittent tools

We have already seen that, in the machine parameters, *"Spindle"* page, it is possible to choose an engraving or cutting system that is activated only when the machine is moving along the path. That type of machine is often fitted with two XY axes, the Z coordinate being manual. But who can do more can do less, and Galaad will be equally happy to drive a machine with 2 axes only. In addition, the workpiece origins will be easier since the approach on the Z axis is often the toughest.

The system for activating and stopping the cutter remains a classical control, generally submitted to the triggering of a binary output (on/off) that drives downstream a power device. The simple checkbox *"only start when tool is feeding"* determines the operating mode, either an activation when launching the machining cycle with a stop at the end, or an activation at the start point of every path with a stop at the end of that path, when the classical milling spindle would move down and up. Anyway, if your machine is fitted with a Z axis, then it remains possible to make both operating modes live together, plunge to the contact then to the depth, activation of the cutter, inactivation before bottom right.

The **laser pause** parameters are available only for an intermittent cutting system. It can become necessary that Galaad begins the horizontal feeding motion along the path after the tool is completely active. If there is a small inertia, then you must indicate a delay (in milliseconds) in the *"on"* edit box. Galaad will trigger the output and then will hold on before starting the feeding motion. Conversely, the *"off"* delay sets a pause to be completed at the end of the path after switching off the tool, before moving the head to a new path or to the parking location.

If that tool integrates a switch-on control function with a ***"Ready"* feedback signal** returned to the supervisor system, then the best is to connect that signal to an input of the numerical controller so Galaad can read it. Just set this input as *"Spindle ready signal"* in the spindle parameters. In that case, Galaad will activate the tool, will hold on during the *"on"* pause, then will wait until the *"spindle ready"* signal is triggered, before beginning the cutting or engraving motion. See the extended inputs/outputs for advanced controls.

❑ 3D printing

A frequently asked question about Galaad concerns the control of 3D printers. Altough the purpose is also to create a mechanical workpiece with a 3-axis Cartesian machine, we are here in a completely different area that does not have much to do with machining. Anyway, Galaad is not a 3D design software, even if it can execute a 3D machining.

However, the Kay module, specialised in machining files, is able to open G-code files for 3D printing with coordinates that correspond to unwinding a wire on an A or E axis, without making a graphical interpretation of a workpiece rotation. So is it usable for driving a machine that is fitted with a printing head, provided that it can talk to its numerical controller. A milling machine can eventually be adapted for printing, but keep in mind that the mechanical needs are much different: a mill requires a good resistance against the cutting force, with a strong body, and on the other hand, a 3D printer just requires motion speeds that the inertia of a heavy mill cannot afford. Reusing a mill for printing is tempting, but please be aware that the machine will not be able to work as fast as a real 3D printer.

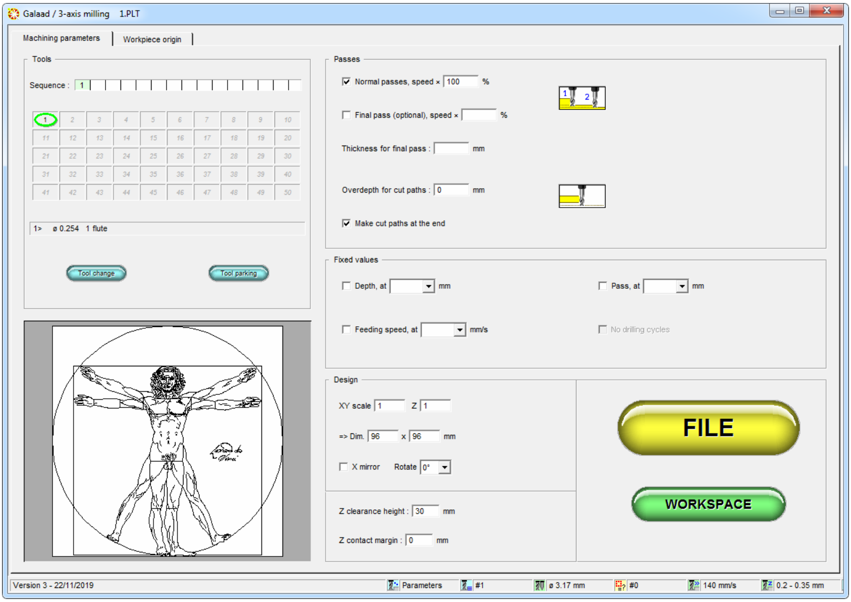
15

0 1 1 1 1

"LANCELOT" STANDALONE USE

Galaad and Lancelot

You have doubtlessly noticed that the installation of the Galaad suite added on your Windows desktop a folder that contains several cousin icons, with different names, including an icon *"Lancelot"*. You may also have purchased only the basic *"Kay"* licence for the machining, without dongle, in which case the modules Lancelot and Percival remain at your disposal. Anyway, please double-click on that icon *"Lancelot"*.

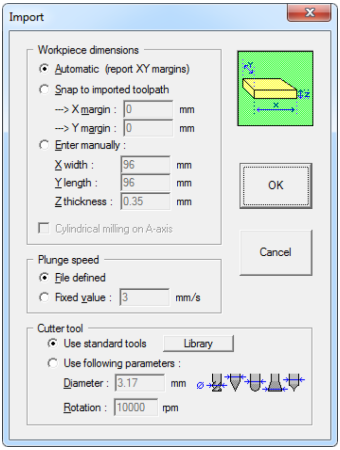


This immediately opens the Galaad window that asks for the machining parameters, but it is covered by a file selection dialogue box which closely resembles the one used for the imports in the drawing module. If you just have the *"Kay"* licence and are opening this user's manual here, then you will nevertheless need to trek through the chapters *"Learning to mill"*, *"Advanced milling functions"* and even *"Machine parameters"*. Please take the time these chapters need, we will wait for you by the coffee machine.

As you may have understood, Lancelot is the standard machining module for Galaad and both work closely together. But this is common-law partnership and, just as Galaad can be unfaithful by entrusting the machining process to a third-party software, Lancelot can also become the recipient for masterpieces made by another creator. Both programmes are technically independent, which allows you to launch a machining process and go on drawing another workpiece, provided that you can bear working with the noise.

So, Lancelot can be used alone. The file is then imported directly for the machining and you see it in the previews. Now you just have to configure it, and indeed it is there that the **fixed values** (especially depth and feedrate) may become useful. For the rest, namely the workpiece origin and the supervision of the machining process, you know the music since you have read the previous chapters.

The main problem with a file import, that we will see again with the Kay module, is that no formats contain standard information about the raw dimensions of the workpiece, and particularly its thickness. Nevertheless, **the raw dimensions are essential for setting the workpiece origin**, and this is why you are prompted to indicate them. The dialogue box that sprouts out as soon as you open the file agrees very fully with this. You may let Lancelot reframe the drawing with margins to be entered, or indicate yourself the raw dimensions, including the thickness. Please note that the drawing can also be wrapped around the A axis for a cylindrical machining, provided that your machine is fitted with a rotary 4th axis.



The speed for the plunge into the material is not in the fixed values, so you may enter it here. Lastly, the toolset can correspond to the standard rack in Galaad which is also accessible from Lancelot, but you can also indicate in this window a diameter and a rotation speed. The diameter will help compensate the lateral approach of the X or Y origin, and the default rotation speed if automatic. These settings remain accessible by clicking on the button *"Workspace"* at the bottom right, through the option *"Import parameters"*. The other options give a direct access to the parameters or to the specific functions of Lancelot, normally reachable from the Galaad module that perhaps you will not use, depending on the licence you have.

Note that **Lancelot can be launched directly by an application other than Galaad** with the automatic opening of a file that is passed as a programming argument, *i.e.* the actual input expression supplied to a function. The extension of the file name, if it is standard, gives the information about the format. Otherwise it is necessary to pass as a next argument the standard extension between brackets. Examples:

C:\Galaad\Lancelot.exe "C:\My Works\MyFile.plt"

C:\Galaad\Lancelot.exe "C:\My Works\MyFile.xyz" (plt)

The quotation marks are used to integrate in a single block a file name including spaces, without splitting it into two arguments. Please go to the chapter *"Technical matters"* at the end of this manual for more information about the various possible arguments of the command lines for the different modules of Galaad suite.

❑ Lancelot and Kay

The Kay module does the same as Lancelot, *i.e.* driving manually the workpiece origin and supervising the automatic machining, but with more specialised formats. Let us just say that the major difference between Lancelot and Kay is that **Lancelot takes only the active part of the machining files** (tool down in the material) and therefore uses the inactive speeds and the drilling cycles that have been defined in Galaad, with its toolset. On the other hand, Kay drives the machine with the file as it is without changing anything, including the inactive movements. Kay is a 3D machine driver for 3 up to 5 axes; Lancelot is a 2½D and 3D driver for 2 or 3 axes (XYZ or XAZ).

16

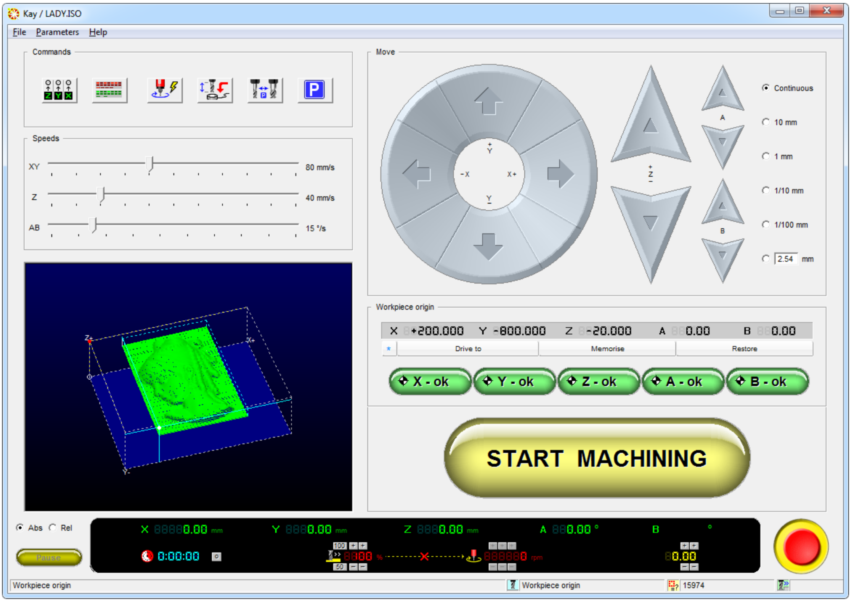
1 0 0 0 0

"KAY"
3D CNC DRIVER

❑ General

The Kay module is a component of Galaad software that can be used independently, either with the Galaad standard licence, or with its own limited licence. **Kay is a CNC machine driver**, *i.e.* it simply loads CAM files under ISO G-code, Isel-NCP or MasterCam NCI format, from which it supervises the automatic milling process once the workpiece origin has been set using the same functions as the standard machining module of Galaad.

Kay is specialised in the milling of 3D toolpaths, from 3 to 5 axes. 4th and 5th axes are presumed to be normalised, *i.e.* rotary axes A and B according to the current standard. The manual drive applies to all axes that are available on the machine.



At start-up, Kay displays a classical manual control window for workpiece origin set-up, and immediately asks for a file to be machined. If you do not select a file, it can be used as manual drive module.

The file name can be passed as a programming argument with its access path, in which case the file will be opened automatically. Hence Kay can be directly integrated at the end of a CAD-CAM-CNC processing chain.

If you use a software application that can create the toolpath under a format known to Kay, and this application can call up an external machine driver, as Galaad does, and give it the name of the file that has been created, Kay will automatically open this file. The file extension then indicates the format. If this extension is not known to Kay, just add it as an argument next to the file name, between brackets. Examples:

→ **"C:\\Galaad\Kay.exe"**

calls Kay without giving it a file to open;

→ **"C:\ Galaad\Kay.exe" "C:\ CadCam\File.iso"**

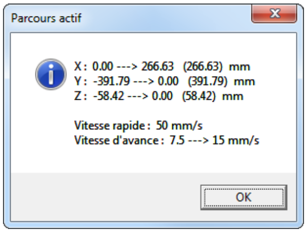
calls Kay and asks it to directly open the file under ISO format;

→ **"C:\ Galaad\Kay.exe" "C:\CadCam\File.xyz" (iso)**

calls Kay and asks it to open the file, the format being indicated immediately after the name.

❑ Coordinates and file origin

As soon as the file is loaded, Kay succinctly displays the volume that is used by the toolpath on XYZ axes with extreme values, and the speeds for feed and rapid movements. This is purely indicative information, which you cannot change.

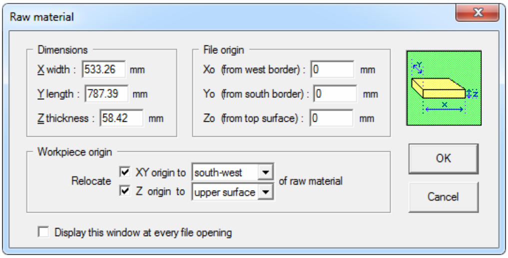


It is important to keep in mind that the formats of the files that Kay reads contain no data about the dimensions of the raw material, information which is essential for a milling process with Galaad. These files provide only vectors and arcs that define the active and inactive toolpath, with corresponding speeds and tool numbers. **It is impossible to determine either the shape or the dimensions of the raw workpiece from the file data.** The automatic drive will therefore start from an origin point that must be defined on the machine bed. **The origin point that is set with Kay corresponds to point (0,0,0,…) of the file.** This point may even be completely outside the toolpath, depending on the coordinate system and the offsets that have been configured by the application that produced the file.

It is common practice to find the zero point XY (0,0) at the southwest angle of a rectangular workpiece, the zero point Z being generally located either at the top surface of the workpiece or at its bottom surface. **You must set the position of the file zero point yourself from the module that has been used to create it.**

In case of a 4-axis machining job, the standard location of the Y origin point is always at the centre of rotation of the A axis. This standard applies to Kay graphical display. If your file did not use this, the display will not represent the toolpath correctly, nevertheless the machining process will remain correct, provided that you have set the Y origin at the corresponding location in the file. In addition, it is possible to indicate the Yo-A axis offset, if any. Otherwise, Kay's coordinate system complies with the geometrical standard with ascending values for X from west to east (left to right), for Y from south to north (front to rear), and for Z from bottom to top.

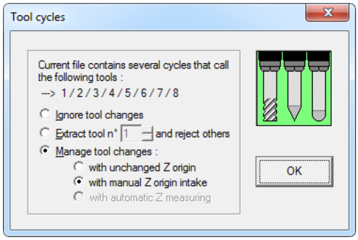
However, it is still possible to **change the file origin point** once it has been opened in Kay. The *"File / Dimensions & origin"* command opens a dialogue box in which you can manually specify the Cartesian dimensions of your raw material, and locate the origin point inside these dimensions, including a fixed position at an XY corner of the rectangle and the Z point at the top or bottom surface. This dialogue box can be displayed at every file opening if you so wish. A checkbox makes that option accessible.



There are more commands available in the *"File"* menu to manually shift, reverse or rescale coordinates. You may also **filter part of the file** by indicating start and end points of the toolpath. All vectors and arcs located before or after will be ignored but commands that concern speed, tool change, *etc.* will remain active.

Multiple tools

When opening the file, and if it contains several successive tool cycles, for example for roughing and finishing, you are prompted to define how the **tool changes** must be managed. The list of the tools detected in the file is displayed at the top of the dialogue box, and several options are then available:



1 - The tool changes are simply ignored. In this case, Kay will consider that only one single tool is used to machine the whole path, the one that is mounted on the spindle when starting the process. The file is then machined in one go with no interruptions.
2 - Only one tool cycle in the file is to be processed. The code lines that refer to this tool are the only ones that generate machining motion, and all others are ignored. It is therefore possible to redo an exclusive cycle.
3 - The tool changes are managed by Kay. Several sub-options are then possible:

3.1 - The tools are ringed or have all the same length under the spindle. Mounting a new tool does not require setting a new Z origin. In this case, Kay stops the machining when it comes to a tool change command in the file sequence, and prompts the operator to mount the tool that is going to start. As soon as the message is acknowledged, the cycle is resumed.

3.2 - The tool change requires a new manual set-up of the Z origin. In this case, the message for mounting the new tool is displayed, and then Kay returns to the manual jogging window. Clicking on the yellow button for launching the process resumes the cycle.

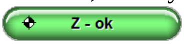
3.3 - The tool change requires a new set-up for the Z origin, but this can be done automatically by the tool sensor. In this case, as soon as the message for mounting a new tool is acknowledged, Kay operates an automatic tool measurement which sets the new Z origin, and immediately resumes the cycle.

3.4 - The machine is fitted with an automatic tool changer that has been programmed in the parameters. The interruption is then limited to the parking of the tool in its rack, the picking of the new one and eventually its measurement on the sensor. Then the cycle is resumed.

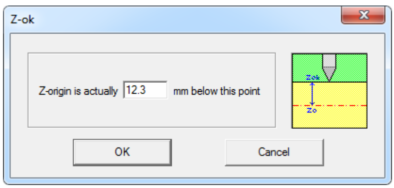
Workpiece origin

As soon as the file is loaded and the active range message validated, Kay starts communicating with the machine and lets you take control of it. Setting the workpiece origin is operated classically from the manual jogging and related functions, including the green buttons for validating the current positions of the axes. The manual control aims to **tell Kay where the file zero point is physically located on the raw workpiece**. To do so, just move manually the tool end to the coordinates of the reference point, axis by axis. For manual jogging and origin validation functions, please refer to the chapter *"Learning to mill"*, section *"Workpiece origin"* which will give you all details plus a few tips.

You can enter a **numerical position** to be reached by clicking on the axis coordinates at the bottom of the screen, or by pressing the X, Y or Z keys.

But the purpose is not only make the machine move along its axes. Although Kay is supposed to know permanently the position of the tool end, it does not know any better where the machining path is located in the machine volume, and this is what you must indicate. Now you are going to approach X Y and Z one by one or together, validating for each axis the target position. For example, you set the Z axis at the point on the raw workpiece that corresponds to the zero of the file, then you click on the green button for **validating the Z position**: 

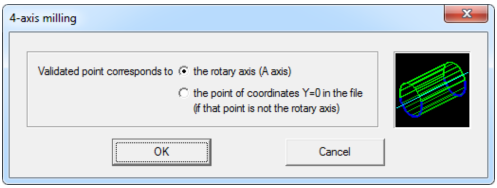
With Galaad standard machining, the point of coordinate Zo in the drawing is always the workpiece top surface, and its thickness is known. With Kay, it is possible that you cannot move the tool to the point of coordinate Z=0 of the file, simply because it is located inside the material. If it is at the bottom of the workpiece, then the simplest is to approach the machine bed or the bottom of the clamp. But this is not always possible and the Zo point can also be somewhere inside the workpiece thickness, at a known but unreachable position. Consequently, when validating *"Z-ok"*, Kay asks you to state if the file origin Zo is below the point you have approached. The Z coordinate then jumps to the workpiece origin position.



Simply follow exactly the same steps for all other axes, including rotary axes if necessary, by physically approaching the point on the raw workpiece presumed to correspond to the zero point in the file, and click on the other green buttons. It is not necessary to move the axes to the origin point before starting the process. If you have a fixed XY clamp and all your toolpaths refer to the same coordinate system, *i.e.* start from this point, then you just have to set the XY origin once and for all when you prepare your very first workpiece process. Incidentally, it is possible to **memorise** origins and **recall** them later. The axis positions for the Z axis and for any potential rotary axes can also be memorised, the last position used remaining valid until it is moved.

Important: if your machine is fitted with a **tool sensor**, please refer to the chapter *"Advanced milling functions"*, section *"Automatic tool measurement"* for detailed information. The method remains the same here, but **using a tool sensor supposes that your machining files have their Zo origin located either at the bottom of the workpiece (fixed sensor) or the top surface of the workpiece (mobile sensor)**, since Kay cannot know its thickness. It is the same for an automatic tool changer.

In a 4-axis machining, *i.e.* if your machine is fitted with a rotary axis and the current file contains rotary coordinates A, then the validations *"Y-ok"* and *"Z-ok"* can define either the position of the rotary axis on the Y and Z axes,



information that is necessary for calculating the tangent speed and a correct display, or the position of the points of coordinates Y=0 and Z=0 in the file, if these are not located on the rotary axis (this may happen though it should not). When you click on the green buttons, then Kay asks more about the validated position.

Note that the keyboard space bar *Space* stops a display update that is in progress. This can be useful with very big files on a slow computer, to avoid unnecessary screen updates.

❑ Parameters

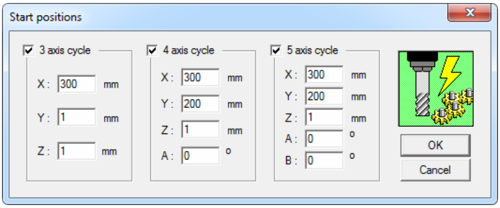
We are not going to repeat here how you can define and set up your machine. Please refer to the chapter *"Machine parameters"* if you need information about it. **Setting the machine parameters is exactly the same from the main module Galaad, and any changes that are made from a given module apply to all**. In Kay, changing one or several CNC parameters resets the communication as soon as the changes are entered. In the same way, the *"Parameters / Reopen communication"* command avoids closing then restarting Kay in case of a break in communication, for example a machine temporary switch-off. The initialisation process is then restarted.

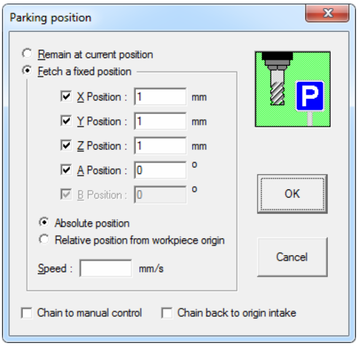
The **calibration** functions concern the machine tuning and the tool sensor. Please refer to these functions in the previous pages of this manual.

It is possible to initiate a **system shutdown** once the milling process is achieved. In this case, Kay is automatically closed after it has sent a system shutdown command to Windows. Depending on its format and its generator programme, the toolpath file may or may not contain commands that switch a **coolant** device on and off. Obviously, Kay switches off the coolant at the end of the process, whether it is complete or not, even if this switch-off was not specified in the file. In all cases, the output that corresponds to the coolant system must have been defined in the machine parameters.

The **warm-up** moves the machine along a slanted ellipse at fast speed, so the three axes are heating up to the working temperature. A **repeated process** can be defined, so the cycle will be restarted after a given delay, and once the previous cycle has been completed without errors. The workpiece origin remains the same. A **serial process** can also be configured, to get a matrix of workpieces on the machine. It is necessary to indicate the number of work-pieces per row and column, and the origin offset from a given workpiece to its closest XY neighbours. As for the coolant, the **spindle command** can be disabled or forced independently from what the file indicates. The **counter** allows you to follow the cycles that have been applied to one or several workpieces to get the overall time spent for machining. Unlike the Galaad module, Kay provides only one single counter.

The **tool start position** is important if your machine is cluttered by various obstacles such as clamps, a rotary axis, *etc*. A file produced by a well-mannered CAM application generally begins with a rapid XY movement to the start point of the first path. But the coordinates of this file are relative to the workpiece origin point and not the machine zero. The very first movement is responsible for ensuring the convergence. And here, we never know if, between the current position and the start point of the first path, the tool is not going to hit some barricade. Either you manage things every time so the tool is close to the start point when launching the cycle, to prevent a collision, or you indicate a start point in absolute coordinates which will always be reached before beginning a cycle. That point can be defined for each type of machining, 3, 4 or 5 axes. If your machine works in floating origin mode (*i.e.* no machine zero), then you have read this paragraph uselessly. Sorry about it.



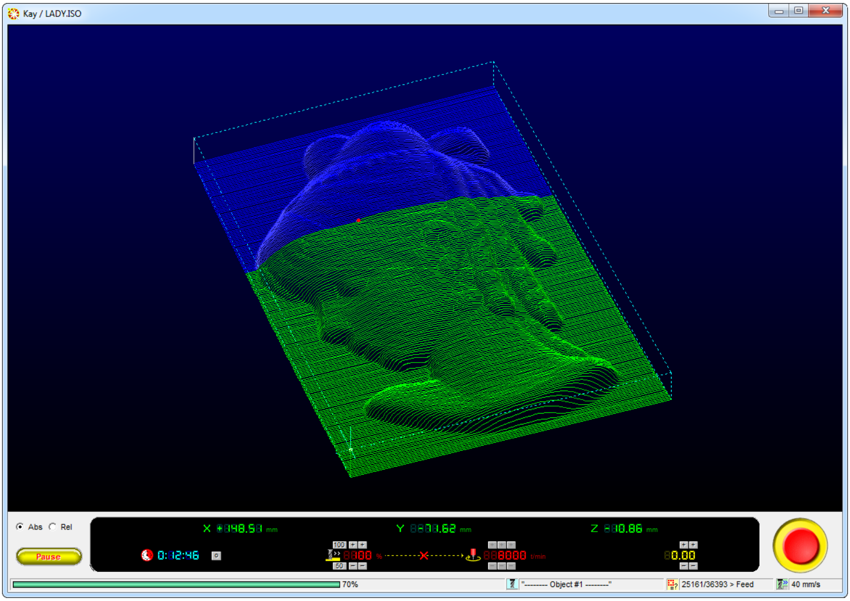


Conversely, at the end of a cycle, the machine moves automatically to the **tool change position** if another tool must be mounted manually in the spindle before resuming the process, or to the **parking position** when the workpiece is completed. You can find here the same settings as Galaad. An **interruption position** can also be indicated, corresponding to what the machine must do when the milling cycle has been stopped for any reasons.

The **import scales** allow you to adjust the scaling factors for all coordinates, plus the speed scales that can be a bit complicated when a linear axis and a rotary axis are working together, since they do not use the same units. The **ISO inputs/outputs** are an add-on for interpreting the G-code so the on/off outputs can be controlled from M commands, and eventually waiting loops related to inputs, the standard being what it is. The M0 and M1 codes for calling the operator can be ignored. The **distance and speed units** concern the screen display. It is up to you to set the import scales and calibrate your machine according to them. Last of all, the **assignment of axes** indicates, in order, which coordinates are used when opening a file, the default being logically XYZAB. For example, for inverting A and B axes in a given format, just reset it to XYZBA.

Milling process

Like all other driving modules in the Galaad suite, the big yellow button starts the machining process defined in the file, from the origin point previously set. When validating the cycle start, you may still decide to leave the spindle and the coolant off.

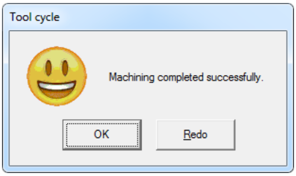


Keep in mind that the space bar Space stops the automatic process, the same as when you click on the emergency button at the bottom right of the screen. A bit more civilised, the Pause button at the bottom left of the screen allows you to interrupt the machining process as soon as the tool is retracted above the workpiece. In this case, the interruption is not immediate, and it is still possible to resume the process where it has been interrupted.

This can be useful to clean the tool or workpiece during the machining, with no risks of marking the material. However if the whole toolpath is one single active block, the pause will have no effect since it will correspond to the end of the complete cycle.

Like the standard machining module Lancelot, Kay displays at the bottom of the screen some buttons that allow you to adjust the cutter feedrate and rotation speed of the spindle if it is under control. It is also possible to shift the Z origin, *i.e.* change the toolpath depth whilst the machining cycle is in progress. This offset is valid only for the current tool cycle and will not apply to any subsequent cycles.

Once the automatic process is complete, all tools having been called in, the spindle is switched off, the last tool is sent to its parking position and a final message warns that the workpiece machining is finished, even if you have gone off to lunch.



If you have purchased only the Kay licence, the current chapter may look succinct or even incomplete. Please remember that this licence also gives you access to the Lancelot module, which is Galaad's standard machine driver and has already been described in the chapters *"Learning to mill"*, *"Advanced milling functions"* and *"Lancelot, standalone use"* of this manual. Consequently, it is highly recommended to read these chapters for more details, not only about Lancelot, but also about the general machining process. Most functions that are described in these chapters remain valid for the Kay module. In the same way, the chapter *"Machine parameters"* remains valid for the use of the Kay module, even if certain specific parameters only refer to machining with Lancelot, for example the flatness correction of the extended inputs/outputs.

It is worth reminding that Lancelot is called up directly by the main 2½D drawing module Galaad. It can drive only 2 or 3 axes but can still supervise a 3D milling, including with external files. Simply, the rapid movements when the tool is retracted up are filtered and redefined by Lancelot, which applies the maximum depth stages that are assigned by Galaad tool parameters. Furthermore, Lancelot allows you to control extended inputs/outputs for processes related to special peripherals. On the other hand, the Kay module drives from 3 to 5 axes but drives the machining process with the file as it is, with no addition of passes, which it is presumed to have already been defined by the CAM application that generated the file. The kinematics calculation is the only change that Kay is allowed to operate to the paths, modulating the feedrates to prevent any off-road trips.

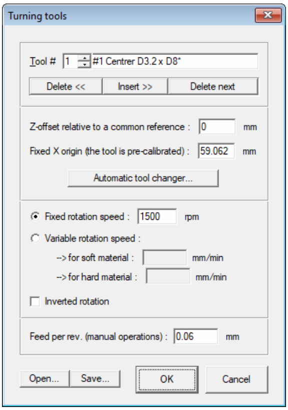
❑ Turning

Kay is not just a milling driver. It can also open a turning file, help set the workpiece origin and supervise the cycle. The lathe parameters are shared with the Gawain module (see hereafter). They are not mixed with those of the milling machine.

Turning mode is accessed either via the *"Parameters"* menu, or by passing the "Turning" argument on the command line. The main screen changes slightly (you switch to 2-axis ZX) and provides a set of semi-manual commands that enable you to perform lateral facing, diameter setting, preparatory

boring, slicing, threading or chamfering or rounding. We will not go into detail here about each function, which is up to you to discover and try out unloaded and then on worthless workpieces.

Semi-manual or in an automatic cycle, **threading** assumes that, even without an encoder, your chuck **motor speed is under software control**. On some controllers, you can use an optical fork sensor connected to an input, giving a 0° top-start for multi-pass threading. Similarly, the speed of rotation can be linked to the X diameter.



To ease changes, tools can be pre-calibrated with a given Z offset and X length in relation to a common reference (tool or other). These settings can be saved. Kay can also manage a tool turret with a location table or even an automatic changer.

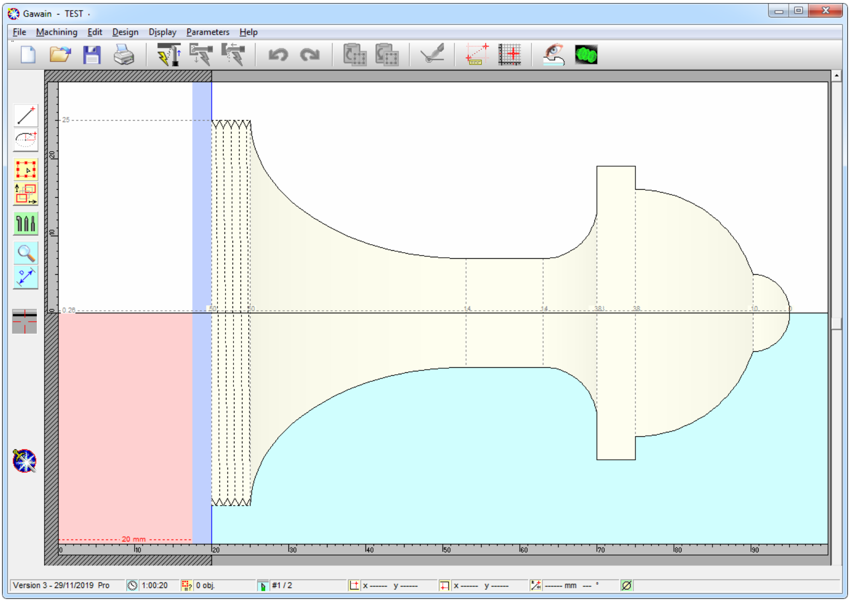
17

1 0 0 0 1

"GAWAIN" TURNING CAD-CAM

Drawing module

Turning applications with Gawain closely resemble milling, but are somewhat simpler, especially for the drawing, and more complicated for the machining. This module is not as powerful as Galaad, but workpiece drawings are much less complex, with just a single external profile, typically chess pawns or chair rods, with in addition the turning of the inner and lateral sides. The use of the Gawain module assumes you are already familiar with the use of Galaad, as well for the drawing as for the workpiece origin and the automatic machining. If this is not the case, please go back to chapters *"Learning to draw"* and *"Advanced drawing techniques"*, which will tell you more about the methods and functions. Drawing with Gawain is as intuitive and should not cause any problems at first sight.



According to the geometrical references of turning standards, "**Z**" **is the feeding axis** and "**X**" **is the diameter axis**, which might induce some confusion about positions and dimensions when drawing on an XY Cartesian plane. Galaad (generally) and Gawain (particularly) comply with the official standards. If you are familiar with milling applications but not turning, then you will have to remember that the X and Y axes of the drawing become respectively Z and X. But the general use is unchanged, so this should not be your biggest problem.

The display of the workpiece profile on the drawing screen shows the **turning path at the top**, where you draw your lines and curves using drawing icons like in the Galaad CAD module, and its **symmetrical twin at the bottom**, area which remains inactive (no need to click here). The whole display represents the complete workpiece once machined. So you draw the half-profile of the workpiece, the opposite side being displayed passively. For a better overview of the machined result, sharps angles of the profile are enhanced with dotted lines that link to the symmetrical path. The Z positions of these sharp angles are displayed near the rotation axis, the default origin being located on the right hand edge of the workpiece.

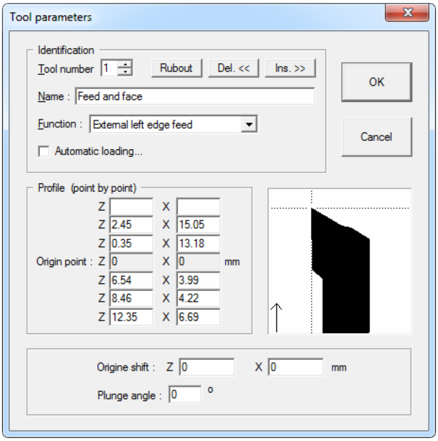
You will soon notice that there are much fewer drawing icons in Gawain than in Galaad. It is obvious that drawing functions that create dots or closed shapes are useless in a turning application, since the purpose is here to **draw parts of the profile using lines, arcs and curves**, the whole drawing being a single continuous path from one lateral end to the other one. In fact, it is possible to draw several different profile sub-sets along the feed path, without connections. Empty spaces between shapes will simply not be machined, so the workpiece remains raw material at these locations. Therefore you have the right to make a non-continuous path.

Since there is only one profile, **the drawing entities must be connected** to build a single path. Gawain understands that separate shapes whose ends meet make one path. Galaad contains a connection function which is useless here. Gawain considers that two entities do not build one path if the distance between the nearest ends is greater than $1/10^{th}$ of a millimetre. In this case, the turning process will retract the cutter to move from one end to the other. Along the same lines, the drawing **sequence** has no importance. When machining, Gawain will ask you to indicate the feeding direction of your choice, *i.e.* from left to right or from right to left. Also the **speed** parameter disappears. One single feedrate can be set directly in the machining window, for the whole process (roughing, finishing, threading).

Tools

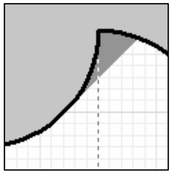
The concept of **tool** remains valid in Gawain, even if turning cutters have little relation with Galaad's machining tools. Turning tools are not drilling, engraving or milling cutters, but passive blades that erode a turning work-piece, like potter's hands.

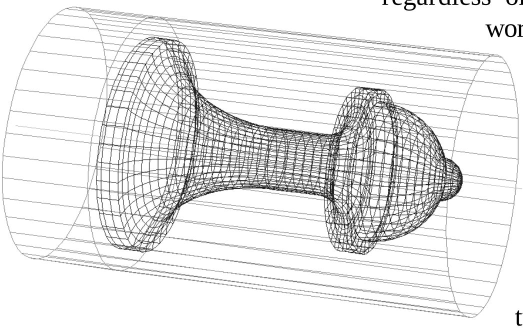
The profile can be machined in successive passes using different tools. So it is necessary for each object drawn to be assigned a tool whose technical parameters have already been set. It is possible to define a library of a maximum of 50 tools, but one single toolrack. And like Galaad, it is necessary that, first of all, you define the parameters of the tool you are going to use.



Hence, you must call up the function *"Parameters / Tools"* which gives you access to the technical data of each tool. **The changes you apply to one tool are memorised when you select a new number**, the whole being saved definitively when you click on *"OK"* button for closing the window, the *"Cancel"* button erasing all the changes you just made. The tool **name** is purely indicative. It will be displayed in the messages and selections when machining, but has no incidence on the process. Conversely, the **function** is very important: **it tells Gawain who does what in the process, and particu-larly the external, internal or lateral sides** (see later on in this chapter).

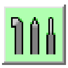
The cutting **profile** and its origin offset allow you to use a calibrated tool-holder and tools accurately measured. The cutter profile can be defined roughly with two sets of coordinates around its **origin point**. This point corresponds to the tool end that touches the workpiece, for example the sharp angle of an oblique blade. When setting the workpiece origin, it is this point that will be the reference. Every couple of ZX coordinates defines a new segment on the tool profile. Consequently, it is not possible to draw a really complex profile, but basic cutters remain definable very easily. In the case of a sectioning cutter, only the Z width of the tool can be defined. Such a cutter is considered to be a simple bar with a flat end, and therefore with a rectangular profile. This type of tool is rarely used in a turning profile since it just cuts the end of the already machined workpiece. When you ask for a final display of the trace actually machined, Gawain calculates and shows any **conflicts** between the originally drawn path and the application of the cutter profile. As usual with Galaad, this is purely indicative: you can still decide to ignore the warning and start a turning cycle regardless of these overlaps, even if the workpiece profile may therefore be altered. On the other hand, **3D displays**, with wiremesh or surface rendering, represent the workpiece as it has been drawn, regardless of the tool shape and therefore without the possible conflicts. These two views are printable as they appear on the screen, like the normal display.





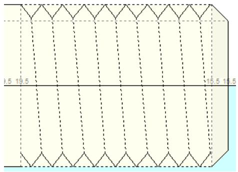
❑ External, internal and lateral sides

As we have just seen, the function *"Parameters / Tools"* allows you to define each tool and link it to an external, internal or lateral job. Hence, assigning a tool to an object, using the green icon, determines what type of machining is going to correspond to it. The display also takes this into acount; an object for the internal profile will not come into conflict with an object for the external or lateral profile.

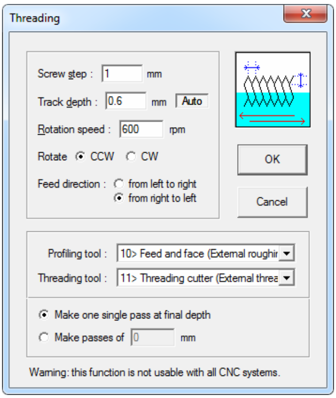


When turning, if you have drawn with tools assigned to different sides, the corresponding functions in the machining menu and their shortcuts in the top command bar are no longer greyed out. Of course, the approaches differ, and particularly for retracting and extracting the tool when turning internal sides. Note that only the profiles for the external side can also be machined using a 4-axis mill.

❑ Threading



With Gawain, it is possible to integrate in the workpiece design horizontal or conical threadings, regardless of the fact that your lathe will or will not be able to execute them. **If the chuck motor of your lathe is not indexed or its speed not under control, then the turning can produce only workpieces perfectly symmetrical about the rotation axis, which actually exclude threadings**. Indexing the motor obviously supposes that the numerical controller that supervises the lathe can operate a movement subject to an external synchronisation, a function not so common. A typical indexation consists of mounting on the motor an encoder or an optical disk that returns the accurate count of the pulses to the controller card, through a frequency divider or multiplier. Threading without indexation remains possible, provided that your lathe is fitted with a very accurate spindle variator or integrates its own closed loop for controlling the rotation speed.



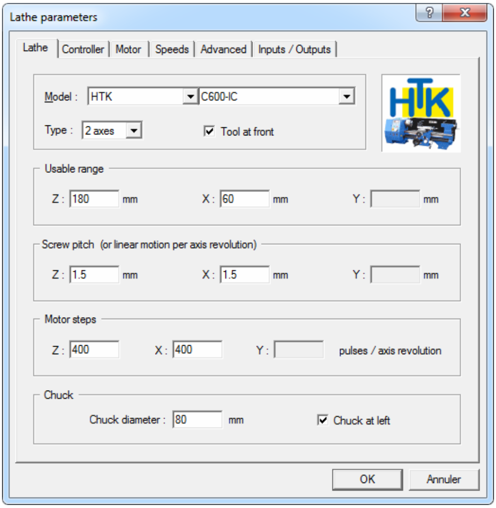
The last icon in the series of lines inserts a threading object in the drawing. The double-click or the right mouse click on a threading summons the editing dialogue box for its technical data. You may indicate **any screw pitch** and any thread depth, even non-standard. A small button *"Auto"* will make the direct conversion from the pitch to the depth according to the usual calculation formulas. The **rotation speed** can be set here for each threading object, as well as the spindle **direction of rotation**, provided that the motor variator can turn clockwise. You may also choose the feeding direction which, combined with the direction of rotation, will determine a screw pitch on the right or on the left.

The **profiling tool** is the one that will be used during the finishing pass for setting the workpiece to the correct dimension before threading. Finally, the pitch depth can be reached after several **successive passes**, on condition however, that your lathe also integrates a 0° synchronisation on the motor.

Lathe parameters

If it can reassure you, the technical data of your turning machine seen from Gawain are much simpler than the corresponding machine parameters seen from Galaad. But Gawain can deal with the parameters of a lathe and a 4-axis mill for machining a profile in a rotary milling, including threads. We will see later on how that can be done.

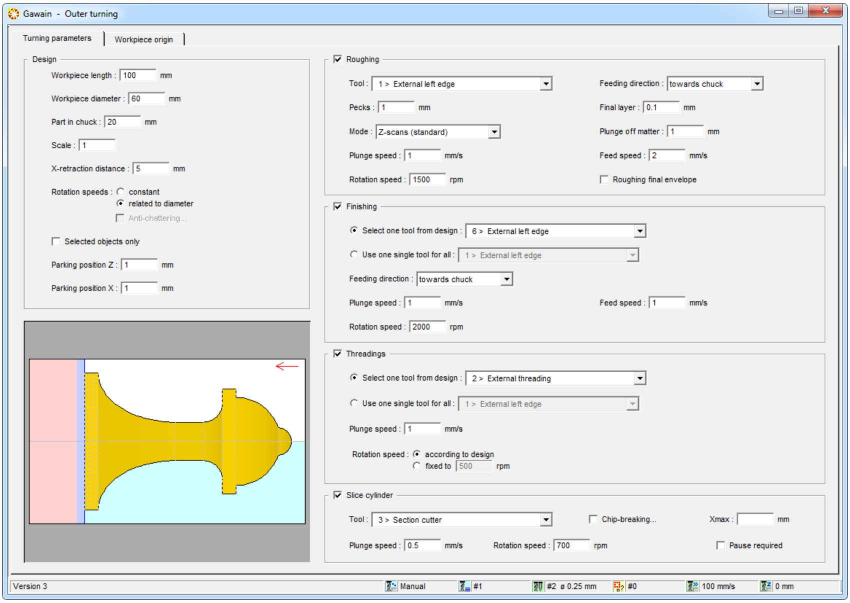
The function *"Parameters / 2-axis lathe"* displays the settings of your machine. The listed lathes are not so many, compared to mills, but **Gawain can control all numerical controllers already available for milling**. This allows you to reuse the axes of your mill for turning at low cost, by mounting on the table a very basic turntable made of a motorised chuck and its opposite mobile tailstock, the turning tools being fixed on a clamp on the Z carriage of the machine. By the way, the set-up asks you whether the lathe is eventually fitted with a 3rd vertical Y axis (the Z axis of a mill). If yes, then the manual jogging will help you adjust the tool height on the rotary axis.



The mechanical parameters are classical. You must nevertheless indicate whether **the chuck is at left** and if **the tool is at front** (which is the standard), so the jogging becomes coherent between the screen and the machine. The parameters in the other pages are identical to those of the mill and have the same scope. Only a few of them do not appear because they are not relevant when turning. Consequently, please refer to the chapter *"Machine parameters"* for more details. In the *"Controller"* page, a small discrete option at the top right says if the lathe controller is able to operate threadings.

❑ Guided tour of the launch pad

Once the workpiece has been drawn, it is time for a test of your lathe in real conditions, calling up *"File / External turning 2-axes"*.

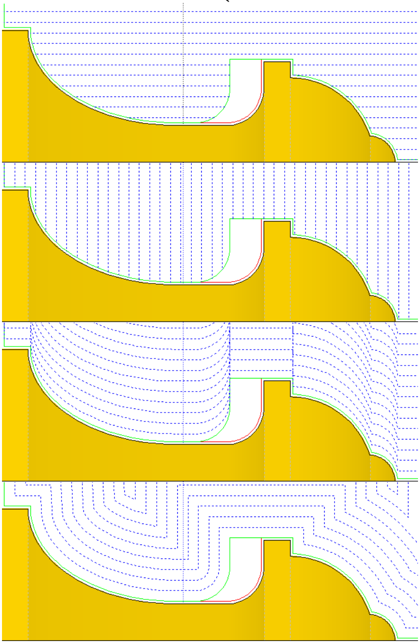


The turning parameters page is more crowded than the one for milling, but its content is not so complicated. On the left hand side of the window, the miscellaneous parameters are general and not related to the current job, except the dimensions of the raw cylinder that you can change here without modifying the drawing. The **part in the chuck** defines the Z length of the workpiece that is sunk in the chuck, accessible in the internal turning mode but not in the external turning (displayed in pink). The **retraction distance** indicates how far the tool must be cleared out for its positioning movements. This distance is relative to the external diameter of the raw cylinder. If your lathe is fitted with a variator, then the **rotation speeds** can be either constant or variable depending on the X position in the material, so the tangent speed is regular. In the latter case, the rotation speeds that are indicated correspond to the external circumference of the raw cylinder, and the software accelerates the rotation when the tool gets closer to the axis. Obviously, the spindle variator must be under control and its parameters calibrated.

On the right hand side, the process is split into four stages: roughing, finishing, threading, slicing, with their own parameters. You must indicate which one you wish to operate. These stages will be chained with no interruptions if there are no tool changes, which would be a bit surprising.

1 - Roughing:

The tools in the drawing correspond to the finishing pass and the threading cycles. The drawing does not tell which tool is roughing the workpiece, and you must select it, together with the roughing parameters. With a classical flat cutter (basic tool for slicing), then **the software automatically calculates the lateral offset of the tool** to avoid machining the vertical or oblique sides. Four modes are available for performing the roughing passes: the classical **Z scan** executes simple horizontal passes in stages on the feeding axis, with a little variant which can limit the tool back retraction whenever possible, for saving time; the **X scan** operates partial slicings vertically on the diameter axis towards the centre of the cylinder; the **X envelopes** follow the drawn objects and gradually get closer to the finishing profile; lastly, the **ZX envelopes** calculate for every pass an offset trajectory relative to the finishing profile. The two last modes make both Z and X axes turn together in oblique movements.



Plunge speed and **feed speed** correspond to the movement for penetrating the material along the X axis, and to the lateral or oblique movement along the defined pass. The option for **plunging off material**, valid only for Z scans, tells the software that, for machining the lateral side of the workpiece, opposite to the chuck, it can make the tool plunge beside the raw cylinder and then come back laterally in the workpiece, which avoids plunging in the material. Beware of the collisions with the mobile tailstock.

Once all the passes have been executed, the option for a **roughing final envelope** makes a final path that follows the finishing profile, leaving just the **final layer** around the profile drawn, so the jagged edges that have been created by the lateral or vertical passes are removed.

2 - Finishing:

The remaining material thickness to be removed by the finishing pass is given by the value for the **final layer**, to be set in the roughing parameters. If you are roughing with scan passes, the stepping effect will increase that thickness here and there, unless you have executed a final envelope. The parameters for the finishing pass are few: you must either select a tool used in the drawing, that will execute its cycle on the corresponding objects, or ignore the drawing tools and select one single tool that will machine all objects. This last option is of course not appropriate if you have feeding directions westbound and eastbound, except if you filter the selected objects. On the other hand, if you are using the same tool for roughing and for finishing, both passes will be chained automatically, with no recovery of the origin.

3 - Threading:

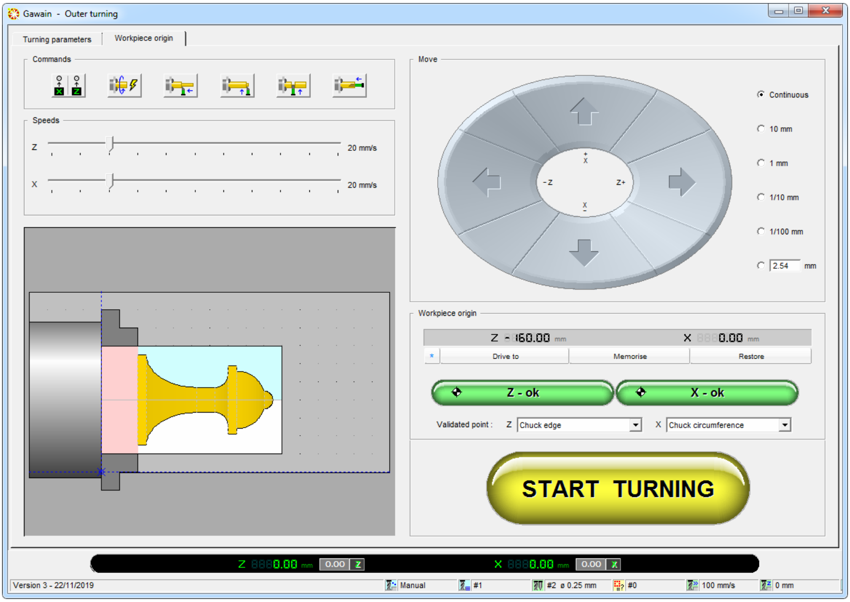
The threadings can be done only after the drawn profile is completed, with the finishing pass in charge of dimensioning the external (or internal) side of the threading. The profiling tool that has been indicated in the drawing corresponds to the tool which will execute that preparatory work. As for the finishing pass, you can either select a tool assigned to a threading in the drawing, which will then execute its cycle on the corresponding objects, or ignore the threading tools that have been set in the drawing and select a single tool that will machine all threadings. The rotation speed for each threading can be overwritten here. The feeding direction is set in the drawing.

4 - Slicing:

In the dimensions of the workpiece, you may have indicated a Z position for slicing with the corresponding tool. In this case, and if you enable the option, that will be done, with an eventual chip-breaking cycle and a maximum X position for overriding the rotation axis. A pause can be set if the whole cycle is based on automatic tool changes.

❑ Workpiece origin

Once the operating phase and its parameters have been carefully set, the next page of the turning window lets you define the workpiece origin for the tool that is going to start its cycle. As soon as the window is displayed, Gawain opens the dialogue with the numerical controller, eventually resetting the axes.



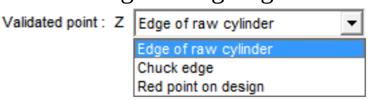
The big circular button and the sliders allow you to **move Z & X axes** at a given manual speed. The keyboard arrows obviously correspond, and also the joystick or any jogging device that you have configured. Like all manual control modules in Galaad's suite, it is possible to apply a continuous movement mode, or incremental by steps to be defined. If your machine is fitted with a 3rd vertical Y axis, then you can also set the height of the tool carriage, for aligning it on the rotary axis. Please refer to the chapters *"Learning to mill"* and *"Advanced milling functions"* for more details on the subtleties of the manual control.

Gawain adds, at the bottom of the screen, some buttons for resetting to zero, switching to relative coordinates and back to absolute coordinates. You can mark a position manually and then offset from an accurate value relative to that position. The relative coordinates are displayed in yellow instead of green.

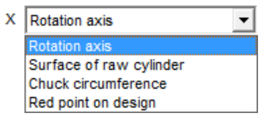
Setting a workpiece origin on a lathe is an operation more complicated than on a mill. Where the milling is happy with a reference point XY, generally not varying as long as the workpieces are stuck in a clamp or two stoppers, plus a Z approach that can be helped by a tool sensor, the turning requires that you approach a Z or X position at every tool change, these positions being themselves subject to variation. The art of turning is an art of the origin recovery. Classically, the operation consists of **making the cutter edge touch a reference position on the workpiece or the chuck, and validate that position whilst indicating what it stands for**.

For the Z axis (feed along the workpiece), we can choose to position the edge on the **right hand edge of the raw cylinder**, tailstock side, on the **lateral side of the chuck** (*i.e.* the base of the workpiece if it is correctly embedded), on the **red-marked point** in the drawing, or on a **point to be defined**, relative to one of the three previous references. Once the tool's lateral edge is positioned at the chosen origin, click on the button 

, which will make the coordinate of the axis jump to the workpiece origin position. Do not forget to select in the combo list under the green button which reference point you have just validated. Important detail: for a slicing or roughing tool with flat end, you must imperatively approach its left hand edge (chuck side). It is this edge that is used as tool reference for the origin and the automatic process.



For the X axis (diameter of penetration into the workpiece), you can choose to approach the **rotation axis**, assuming that the raw workpiece is not yet clamped, or the **surface of the raw cylinder**, supposing that it is perfectly cylindrical and centered on the rotary axis, which is not always evident, or the **chuck circumference**, its diameter giving the position of the rotary axis, or a **red-marked point** in the drawing. As for the Z axis, once the active edge of the tool is positioned at the chosen origin, you must click on the button

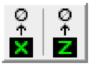


for validating the coordinate, without forgetting to select in the combo list under the green button whose reference point you have just validated.

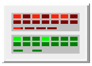
Of course you may set Z and X in any order. The last position used is memorised relatively to the left hand edge of the workpiece (chuck side), and relatively to the rotary axis, considering the raw dimensions of the workpiece. So you will have to redo the origin set-up only if you change the tool.

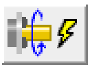
Semi-automatic commands

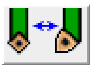
Before launching the automatic process, let us make a little detour through the command buttons at top left of the window.



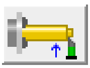
The first button corresponds to the classic command for **resetting the machine zero point**, already seen in the milling mode. This supposes that your lathe works in absolute coordinates from a fixed position, defined by axis end-switches.


This button opens the **inputs/output control** window, in case you need to take control.

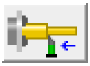

That one starts or stops the **lathe motor**. If there is a variator, then a window helps you manage the rotation speed.


That one controls the **tool change**, automatic if you have a changer, or simply send to manual changeover position.

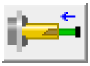
That one sends the axes to the **parking position**. When pressing Ctrl key simultaneously, you can set that position directly.


The next button drives a **lateral preparatory work** from the current machine position on the workpiece, an operation which is somewhat close to the slicing but allowing a repetitive cycle. The plunging value may or may not reach the rotation axis, and even overrun it: Gawain just moves to the indicated X position. The plunge can be repeated from a

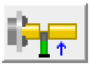
starting Z position to the current position. Note that, once it is started, you can nonetheless stop the process using the space bar Space on the keyboard.



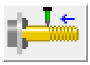
That button allows you to **reduce the diameter** of the raw workpiece. This operation consists of feeding lateral passes along the Z axis, sinking the tool step-by-step in the cylinder from the current position. The indicated data are used as they are with no validity checking in relation to the drawn workpiece or the origin already defined. It is up to you to verify your settings before proceeding.



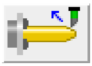
That one offers the possibility to perform a **preparatory boring** at the end of the workpiece, tailstock side, from the current position of the axes. You must mount a reaming cutter on the carriage that support the tools, oriented towards the chuck with its leading edge located on the correct side relative to the rotation direction. The software will drive a Z- feeding movement towards the chuck at the feedrate, and will reiter the cycle until the X feed is reached, with or without passes. It is possible to smooth eventual jagged edges appearing at the bottom of the pit by setting a finishing pass on the X axis.



That one allows you to **slice** the workpiece at the current Z position. You must indicate the maximum X depth for the penetration, which can overrun the rotation axis at your own risks, and also the speed for plunging in the material. Again, keep in mind that there are no validity checkings here. It is possible to execute a chip-breaking cycle for this operation. You can also ask for the slicing to be repeated at regular intervals.



That one makes it possible to trace a **thread** from the current Z position. The speed of the chuck motor must be under the precise control of a speed variator. A multi-pass cycle is possible depending on the numerical control model, provided that an inductive or optical fork sensor is connected to an input giving the 0° top-departure.



That one allows you to make a **chamfered or rounded shoulder** on the edge of the blank, at the current position.

❑ Automatic process

Once the turning parameters and workpiece origin have been validated, you just have to click on the big yellow button that starts the turning process. The Enter key is the keyboard equivalent. A last message prompts you to confirm the process launch. It is time to switch on the lathe motor if it is not automatic. Validating this message starts the cycle immediately.

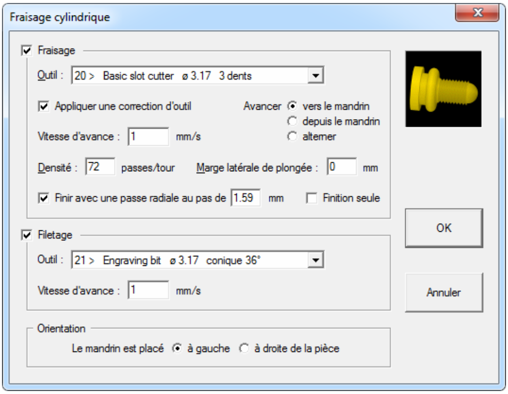
The tool path corresponds to its leading edge. In the case of a slicing or roughing cutter with a flat end, remember that the coordinate is the one for its left hand border, usually chuck side, with an automatic compensation offset on the opposite border. Hence the software displays a wide cursor that represents the tool in its full width.

In the end, and if everything went well, a new bishop for your chessboard is now machined. It might be more complicated for the horses.

Turning on a 4-axis mill

Although Gawain is dedicated to the turning process only, it gives access to the mill parameters. If this mill has a 4th rotary A axis, then the software can ask Galaad's machining module for a small service, that is to execute the machining of the drawn profile by milling, using the command *"Machining / Cylindrical milling 4-axes"*. This is the only gateway (one-way) between turning and milling.

The specific parameters of this cylindrical milling are displayed in a dialogue box when opening the machining module. In case of need, it can be recalled with a little **double-click on the preview area** at the bottom left of the window. The parameters look like those of the classic turning, but more concise because the machining is going to reuse the tool and pass settings that are valid for the cylindrical millings launched from Galaad's drawing module. The turning tools have nothing to do with milling tools, so the software needs to know which tool is about to be used for the machining. This tool will generally be a cylindrical flat-end cutter. **Gawain will automatically calculate a lateral compensation offset for fulfilling the drawn profile**. The machining will feed successive XZ passes (these axes corresponding to a mill) with an incremental rotation of the A axis, the lateral margin for the plunge allowing the tool to move down out of the material, tailstock side. A perpendicular finishing pass can be added, milling rotary slices with XZ being still.



Threading is fully enabled in 4-axis machining since the indexing angle of the rotary axis is known permanently. As for the turning, the milling path will complete its job at the accurate dimensions of the drawn profile. Then you will have to mount an engraving bit with a conical end that corresponds to the thread gorges.

18

1 0 0 1 0

"KYNON" MOTION PROGRAMMER

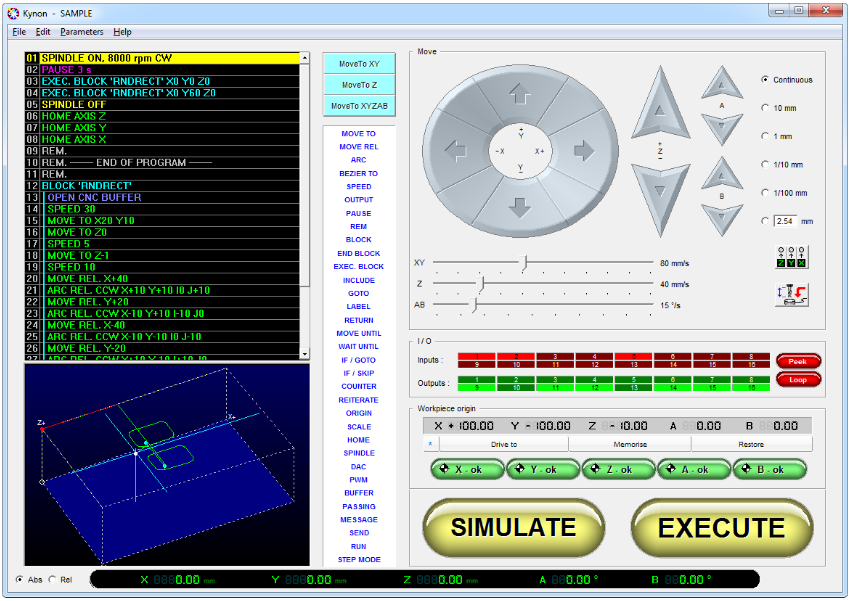
❑ Automaton for axis controls

If the main purpose of Galaad software is to perform milling – and possibly turning – operations, the CNC machine can roughly be defined as a more or less complicated set of motorised axes plus a numeric controller, able to communicate with a specialised software application. Hence we have several linear or rotary axes and input/output signals, the whole set being under the full tight control of your computer. Consequently, this gives the computer the ability to **drive movements, wait time lapses, react depending on input signals, and switch connected peripherals on/off**, be they electrical or electromechanical.

In addition, those who become interested in Galaad's **extended inputs/outputs**, for special applications such as glue dispensing or engraving on a variable Z surface, may have noticed that the application is able to consider external signals and pauses during automatic machining processes. In such particular cases, Galaad remains no less focused on creating and driving a 2D or 3D path, the management of electrical signals being a secondary function which is closely framed by the fact that this geometric path must be followed. The main Galaad module cannot deal with all possible cases, and particularly conditional jumps, loop sequences or interactive messages to the operator. Even including external controls, **Galaad's movements remain a graphical path** with a start point, an end point and a unique sequence with no possible dynamical redirections.

Most Cartesian machines that have been built for automatic milling can also become compatible with other types of more exotic applications, with or without a milling spindle and a cutter tool, such as **automatic pick & place** or, more generally, motion that interacts with the workpiece. Furthermore, the sequence of handling or a similar operation cannot always be displayed as a path design, but rather as a **list of operations** that must be performed in a given order, which may vary during execution. It is for these reasons that the Kynon module has been added to the set of main functions of the Galaad software, but here graphics are no longer the main purpose.

The Galaad standard **licence** grants access to the use of the Kynon module and all additional programmes, though it is also possible to purchase a restricted licence for the use of Kynon alone.



At start up, the Kynon module displays the window shown above. For experienced Galaad users, the right hand side of the window should not be a total surprise. Indeed you can find here the classical buttons for **manual motion of axes**, with corresponding speed sliders and motion mode selectors, continuous or stepped. Those who have become familiar with the maze of functions of the manual control will also recognise the double panel of binary **inputs and outputs**. The bottom of the page provides classical control objects that allow you to set a **workpiece origin**.

It is not useful to enumerate again and again the subtle functions that appear at the right hand side of the screen; they have already been much commented in the previous pages. Help us avoid converting too many trees into paper, this manual is guilty enough in that matter. If you are reading only this chapter and are not familiar with manual drive from Galaad software, probably because you have purchased only the Kynon licence, then please jump back to the previous sections, as follows (if we dare to say):

- Chapter *"Advanced milling functions"*, section *"Moving axes"*, which will tell you how to **jog axes manually** at a given speed, in continuous or stepped mode, with the mouse, keyboard, joystick or a handwheel.

- Chapter *"Advanced milling functions"* still, now section *"Manual drive"*, to learn how you can **read inputs and switch outputs** from a mouse click.
- Chapter *"Learning to mill"*, section *"Workpiece origin"*, which very logically describes how you can **define a workpiece origin** for an automatic process. In fact, Kynon can ignore this origin point if there is no workpiece in classical terms and that all coordinates are based on the machine zero.
- Chapter *"Machine parameters"* last of all, which should tell you all the possible ways to set up your CNC configuration and especially the gear factors and communication port.

The right hand side of the screen is now presumed to be known and the machine correctly defined, but you may still play with the parameters. At start-up, Kynon first opens the dialogue with the machine, possibly resetting the axes with a homing movement to the machine zero. It then becomes possible to play with the buttons. It is worth mentioning that you may **enable/disable an output** by clicking directly on its corresponding number with the mouse. This should help you check the reactions of the peripheral devices that you have connected to the machine. Concerning inputs, no real-time reading functions are possible, but you can **display input states** by clicking on the *"Refresh"* button, or make a **cyclic reading** by enabling the *"Loop"* button (click again to disable). If you have connected a peripheral that produces a signal to be monitored and you wish to check the number of the corresponding input or the correct electrical wiring of the device, this cyclic reading may be very helpful.

Kynon can drive from 1 to 5 axes depending on the capabilities of your machine. Classically, XYZ axes are considered linear axes and AB are considered rotary axes that are presumed to be parallel to XY. It could be the case that your machine is different and does not match this configuration, however this should not block your use of the Kynon module. A special set-up of the machine gear factors may help you solve this problem to obtain coordinates that correspond to reality, even if the units displayed no longer make sense.

❑ Programming technique

The left hand side of the screen displays a zone that is supposed to contain the sequence of instructions to be executed – in other terms the programme – and just below, a preview of the path, if any. If anyhow the axes of your machine are not Cartesian, then you can get rid of this preview thus making more room for the programme itself, by unticking *"Parameters / 3D view"*.

The body of a Kynon programme may be displayed in capitals or lower case, colours or monochrome. The top of the *"Parameters"* menu helps you select all this. Operators who already have programming language skills, even the basics, should not have big problems becoming familiar with Kynon. Also, absolute beginners have not been forgotten: learning this programming technique is very easy and close to BASIC language, which is the simplest ever invented and was made for beginners, as its name says *("Beginner's All-purpose Symbolic Instruction Code")*. Therefore you should not encounter huge problems creating a Kynon programme, and then… well, let us just say that there are no good programmers, only old programmers. And unfortunately we know what we are talking about.

In classical sequential programming, and Kynon matches this model for the most part, **the sequence begins at the top and instructions are executed one by one downwards**, until the last line of the programme is reached. However, there are some instructions which let you break that sequence by executing jumps forward or backward. Kynon also offers the possibility to define blocks of macro-instructions, over which the sequence will skip regardless of what they contain, but can be called and recalled at several stages of the process. This will be described later.

A basic Kynon programme consequently has a beginning point at line #1, followed by a more or less long series of operations that will be executed one by one until the last line is reached. Once this end line has been executed, the process is completed. There are no limits concerning the number of lines.

There are four major groups of instructions in a Kynon programme, which will be detailed later with the usual examples:

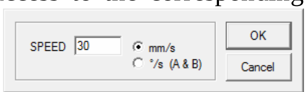
- Motion commands send one or several axes to an absolute position or perform a relative movement at a given speed. Arc commands are also available, the circular interpolation being valid only in the XY plane. This group of instructions also integrates the homing, which resets the machine to its zero point, or conditional movements which stop when a binary input state is changed. All motion commands apply to coordinates and therefore to axis positions. These commands are: **SPEED**, **MOVE TO**, **MOVE REL**, **ARC**, **ARC REL**, **HOME**, **MOVE UNTIL**, **ORIGIN**, **SCALE** and **PASSING**.

- Switching and pausing commands change output states or freeze the programme for a given length of time. Unlike Galaad, which is able to trigger multiple outputs simultaneously, it is not possible with Kynon to switch a whole block of outputs with one single command. You must define a sequence, even if the time between every instruction is limited to the communication delay with the machine, which is generally very short. These commands are: **OUTPUT**, **DAC** *(Digital-to-Analogue Converter)*, **PWM** *(Pulse Width Modulation)*, **SPINDLE**, **PAUSE** and **WAIT UNTIL**.

- Jump commands, fixed or conditional, break the linear process sequence by skipping instruction lines forwards or backwards. These commands are: **GOTO**, **GOSUB**, **LABEL**, **RETURN**, **IF / GOTO**, **IF / SKIP**, **COUNTER**, **BLOCK**, **END BLOCK**, **EXEC BLOCK**, **INCLUDE** and **REITERATE**.

- Management commands allow you to add a neutral comment line, modify the programme execution mode in the machine, interact with the operator or call external software. These commands are: **REM** *(remark)*, **BUFFER**, **MESSAGE**, **SEND** and **RUN**.

Kynon programme editor is purely passive: unlike classical text editors, **you do not type the language instructions yourself**. The list of available commands is displayed in the central column of the window, you simply click on them to pop up a dialogue box that gives access to the corresponding command parameters. This should at least avoid syntax errors, which are quite usual with a classical text editor.



When the active line (displayed in negative colours) is empty, clicking and validating an instruction will add it at this position in the programme, by shifting down the following instructions. If the line already contains an instruction, either the instruction you have clicked on is identical, in which case it will be a simple change of its parameters (it is even simpler to double-click directly on the line in the programme), or it is not identical then the new one will be inserted just before the existing one, which will be shifted down. Therefore, if you wish to insert a new instruction, which is identical to the existing one, you must first **insert an empty line** using the Ins key. Conversely, you can **delete the active line**, whether it is full or empty, using the Del key. All consequent instructions are immediately shifted up.

Warning! When you write a programme with Kynon, **you are still in manual drive mode**. Keyboard arrows and / (PgUp / PgDn) keys or Home / End keys do not move the active line but actually move the axes of the machine. Moving the active line of the programme can be done **only with the mouse**. Consequently, the programme body provides little interactivity, but this is not a classical development with many changes. Building a Kynon programme – which is generally short unless you have a rich imagination – is closer to a line-by-line lesson in that it is related to the manual drive, as we will see in the next pages.

❑ Motion commands

The main command in this group is the **MOVE TO** instruction, which moves the axes to the indicated position. This instruction will generally follow a **SPEED** instruction, which defines the corresponding motion speed. The **MOVE TO** instruction does not require the coordinates of all axes; one coordinate is sufficient to define a movement. Obviously, if the position to be reached is the same as the current position, nothing happens – which is always better than an accident along the way.

<u>Very important:</u> **the position coordinates are always *relative* to the origin point of the whole path**. If your application does not need an origin point and is happy with a simple machine absolute origin, then you can simply set all origin positions to zero. In such case, the machine zero point will be the actual origin for the path. Exceptionally, you have the right to indicate a **negative origin**. Kynon is more permissive than Galaad in that matter, and let us hope that so will be your machine.

If you have entered a series of motion commands that define a fully correct path but at a wrong location, do not panic: there is no need to redo everything, you can simply shift a set of lines using the *"File / Shift"* command. Block macro-functions can also help in such cases (see later on).

The question of actual position has no sense with the **MOVE REL** instruction, which uses only relative values. The sequence is identical, but then the origin has no more importance since all movements are driven relative to the previous point. Keep in mind that Kynon checks the absolute position of the next point to be reached and will stop if you are trying to escape from the valid range of an axis.

The **ARC** and **ARC REL** instructions describe an arc of circle on the XY plane, which starts from the current position towards another XY target position around an IJ-centre. Depending on the instruction you have used, coordinates are indicated in absolute (*i.e.* relative to the total path origin point) or relative coordinates.

The **SPEED** instruction defines the motion velocity for the **linear axes** on the actual path (and not the velocity of the axis that carries the longest movement like most numerical controllers read). If you move a rotary axis A or B with a linear axis X, Y or Z, the speed applies to the linear motion and the rotary axis will obviously be synchronised. If the rotary axis moves on its own, then the speed becomes a **tangential velocity** around the rotary axis, unless you have previously indicated a **fixed angular speed** in °/s. Consequently, the speed will be greater when the position is closer to the axis centre. For an A axis, the arc radius is the YZ distance from the current position to the origin point; for a B axis, the XZ distance applies. If you want a rotation of the A or B axis whilst the current YZ or XZ position is (0,0), then Kynon will not dare to divide by zero and therefore the speed will be the maximum value the numerical controller can perform. This particular case should never occur if your path has been validated.

The **MOVE UNTIL** command starts a movement to a given position, movement which is **stopped if an input state changes**. Kynon immediately updates its position registers, but it is up to you to execute the following sequence with absolute or relative movements, depending on your application.

The **HOME** instruction allows you to drive an axis to the machine absolute zero, which will reset its position. Each axis can be driven individually. It is quite usual to shift the Z axis up first to avoid problems.

The position of the **path origin can be changed while running,** using the **ORIGIN** instruction, which allows you to relocate all subsequent coordinates. Here, *"subsequent"* means *"executed after"* and not *"below in the programme"*, especially if there are jump instructions. On the same subject, it is possible to change the coordinate scaling factors while the programme is being executed, by using the **SCALE** command.

Finally, the **PASSING** instruction, eligible only for certain numerical controllers, allows you to chain vectors with no deceleration before the end point and no re-acceleration after the start point. This mode obviously assumes that you are using the **BUFFER** command (see next pages) so that there are no transmission delays between vectors, which would cut the benefit of the chaining. If your machine does not read this passing mode instruction, it will simply be ignored.

❑ Switch & pause commands

In addition to axis motion, Kynon can address outputs and check inputs. The main command in this group is the **OUTPUT** instruction which sets a given output in a high (active) or low (inactive) state, the default state being generally inactive. It is up to you to connect the peripheral devices of your custom application to the corresponding outputs.

Please note that, unlike Galaad, it is not possible with Kynon to specify multiple output numbers that could be switched in one single operation. You must set the switching sequence line by line.

The **DAC** instruction sets the signal level of the machine **analogue output** number 1, if it exists. The resolution is 8 bits, which allows you to set values between 0 and 255, from 0 to Vmax. On the same topic, the **PWM** instruction sets the frequency of the machine PWM signal, again if it exists, and the pulse width in percentage (*i.e.* between 0 and 100%). These commands allow you to drive an analogue device such as a rotation or dispensing control.

Though the purpose of the Kynon module is not focused on milling processes, the **SPINDLE** command is available with the same external calling and speed setting capabilities as in Galaad. The instructions therefore include rotation speed and direction arguments that may not apply to your machine, depending on its type.

The **PAUSE** command allows you to suspend the process during a given time period. The unit is the second, and you may naturally use a decimal value. In direct mode, the basic time unit of Kynon is the millisecond and it is not possible to set a smaller value. In buffer mode (asynchronous upload to local memory), this depends on the capabilities of your numerical controller.

At last, the **WAIT UNTIL** instruction suspends the process until an **input** state changes (with a possible timeout lapse). The operator can manually jump to the next instruction without waiting for the input to be triggered, by pressing the Enter key. In such case, no message is displayed on the screen but the current instruction line blinks so the operator knows that the process is suspended.

Jump commands

In any civilised programmable sequence, it must be possible to break the linearity of the process and skip instructions, imperatively or depending on certain conditions. Obviously, Kynon provides such possibilities with its jumping instructions.

Before we go further, please keep in mind that it is always better to use the **LABEL** instruction to **define a landing line for a jump in the programme**. Nevertheless Kynon accepts to jump directly to a line whose number is known, and is even courteous enough to shift the winning number when new instructions are inserted above this line, but there are at least two serious reasons to use labels: firstly, reading the programme is much easier; secondly, you do not need to check the changes of line numbers to keep your jumps correctly targeted. You define a label, *i.e.* a line name, then you can add a jump to this line whatever its number is and however it evolves. It is obvious that **the label line is neutral**, *i.e.* nothing happens when it is executed, such lines are just fixed references in the programme sequence.

Jump instructions can be either **imperative**, *i.e.* Kynon executes the jump whatever, or **conditional**, which can become much more interesting. The two imperative jump instructions are **GOTO** and **GOSUB**, that probably most BASIC programmers know. The first makes a **one-way jump** whereas the second makes a **temporary jump** until the programme encounters a **RETURN** instruction, which allows you to define round trips to specific sections of the programme, especially in conditional mode.

In the same way, the conditional jump instructions are **IF / GOTO** and **IF / GOSUB**. The conditions are either an **input change** from/to low/high state, a **loop counter** (see the next pages) with a value that reaches a threshold, an instruction to be executed by the numerical controller that overruns a **predefined timeout lapse**, or a **message to the operator** which must be acknowledged by *"Yes"* or *"No"* (condition being true when affirmative).

Slight variation, the **IF / SKIP** instruction allows you to undertake **conditional jumps** depending on an input state, but this instruction differs from the above because it can be uploaded to the local memory of the machine, depending on the controller type, for example Isel C-10, C-142 and i-MC. The number of lines to be jumped can be negative (backward jump), positive (forward jump), or null (conditional wait loop).

Kynon can manage numbered counters, for example when repetitive loops are created with an internal **GOTO** instruction that exits the loop. Simply insert in the programme, if possible before entering the loop, a **COUNTER** that will be initialised to a given value, zero or otherwise. Then, in the loop, add an incrementing or decrementing command for this counter, for example *"Counter #3 + 1"* which means that the data in the counter number 3 will increase by 1 every time the loop is executed. This instruction will obviously be followed by a conditional jump related to the counter level, if the corresponding value has overstepped a given threshold. In this case, the programme will have executed N times the loop before jumping to the next instruction lines.

Similarly, the **REITERATE** command allows you to **repeat several times** the last N lines of the programme. This looks like a loop with an internal counter, but that instruction can be stored in the machine local memory, depending on the controller type (still Isel C-10, C-142 and IMC4). It has little interest if you do not have such a machine or do not use the upload to the local memory.

❑ Macro-commands

To ease reiterations, improve the modularity and consequently the readability of the programmes, it is possible with Kynon to define **programme blocks**, that will be called up from another part of the programme. For example, you define a given procedure that contains movements, output switching, *etc.* and you wish to **reuse** this procedure without modifying it and without adding a complicated set of jump instructions.

Kynon can define a sequence of commands that begins with a heading **BLOCK** instruction with a block name, and ends with an **END BLOCK** instruction. When the process reaches the block header line, it **jumps directly to the end of the block** without executing the inner lines. The block is considered a neutral set when executing, so you can locate it anywhere in the programme body, before or after a call to this block. To execute the inner lines, it is necessary to call up the block with an **EXEC. BLOCK** command. Furthermore, and here it becomes really interesting, you can **assign a coordinate offset** to your call.

For example, you define a series of XY movements which draw a rectangle with rounded corners, using linear and circular movement instructions, and the start point of this rectangle is (0,0). All movements are written inside a block that is named for example ROUNDRECT. Hence **ROUNDRECT** becomes an instruction like any other, that you may insert in your code and give it XYZAB coordinates according to your wishes. If you put the instruction line **ROUNDRECT X10 Y20 Z30** somewhere, Kynon will execute all lines inside the block named *"ROUNDRECT"* and will shift its XYZ coordinates by (10,20,30), *i.e.* all inner coordinates are added these values. The main purpose is obviously multiple calls to **ROUNDRECT** with different coordinates. But this can also be used to simply make the reading of the programme easier. Please note that all or part of the position arguments XYZ *etc.* are optional. If you do not indicate any, the corresponding coordinates when executing the block lines will be the actual coordinates of the inner lines with a null-offset.

To close this section, it is worth mentioning that **a block can contain an inner call to another block** and so on, to define more and more elaborated macro-blocks. In the same way, a block can call to itself, but it is better to avoid this unless it contains a conditional jump to the block end line, so you do not have an infinite recursive loop (in fact, Kynon is able to stack a maximum of 256 successive calls, this maximum value also applies to all **IF / GOSUB** instructions, which still leaves a rather comfortable margin for most applications).

And finally, the **INCLUDE** instruction allows you to **call up an external file** which can either be another Kynon programme, and this would define a super-block, or a 2D or 3D vector path file. In the same way, a **coordinate offset** applies to this external file to relocate it. This can be interesting for integrating a whole path without having to encode the sequence of movements.

Programme management commands

Some not so easily classified instructions can help supervise several process steps, hereafter and helter-skelter:

The **REM** *(Remark)* instruction, known to well brought up programmers, allows you to add a **comment** into the sequence for the next generation who will have to decrypt the programme and try to understand how it works – if it does. Obviously, this instruction is neutral and consequently has no effect when executing.

On the other hand, a message can be sent to the operator during the process execution, using the **MESSAGE** instruction which opens a window and displays the attached text. The operator can then either acknowledge it by pressing the [Enter] key (or clicking on *"OK"*), or abort the process with [Esc] (or clicking on *"Cancel"*). As long as the message remains unacknowledged by the operator, **the process is hanging**.

With certain numerical controllers, it is possible to **send instructions to the local memory for execution later on**. Either the machine starts executing already loaded instructions whilst receiving the next ones (ring buffer), or the machine first receives all the instructions then executes them (linear buffer). This depends on the machine you are using. Kynon provides an *"Upload"* mode in its menus that uses the local buffer if it exists. But this does not prevents the use of buffer management commands inside the programme itself. Hence you can **execute locally** some sequences, for example the chained movements in passing mode (mentioned above in the motion command section).

Since certain numerical controllers may integrate some instructions that are unknown to Kynon, you still have the possibility to handle them yourself using the **SEND** command which will simply transmit the corresponding telegram on the communication port without trying to understand it. Kynon will nevertheless add data frame codes according to the machine dialogue protocol.

Finally, you may execute **external software**, even in console mode, using the **RUN** instruction which will call up this executable programme and either skip to the next line or wait until its execution is terminated before resuming the current process.

Manual drive teach-in

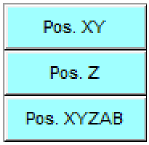
A major feature of Kynon module is the fact that you can directly drive the main functions of the machine (linear or rotary movements, output switching, input checking) to build up a programmed path step-by-step. So the process looks like the manual digitising function of Galaad, except that here you can access the supplementary commands of the machine, and that up to five axes are available. On the other hand, the origin point becomes important.

Two concrete cases may occur, depending on your application: either your process handles a workpiece or a restricted volume which is located somewhere in the available machine space (operations on workpieces), or it applies to a general space with no local reference point (pick & place handling).

In the first case, it is important to **first define the origin point** of the programmed path. Classically, you just have to move the axes to the corresponding position and validate them with the *"XYZ… - OK"* green buttons. Then all coordinates indicated in the programme body for absolute movements **MOVE TO**, **ARC**, **MOVE UNTIL** and calls to programmed macro-blocks or external files to **INCLUDE** will be considered relative to this origin point. *In extenso*, **position (0,0,0,…) will correspond to the origin point**, which can be readjusted later so you do not have to shift the programmed coordinates. If you have skipped this step and have already begun to encode movements, simply position or reposition this origin (it is not too late) then call up the *"File / Shift"* function to change all programmed coordinates in one operation. The offset value for each axis will obviously be the position of the new origin point minus the position of the old one.

Concerning the second case (no path reference point), it is necessary to **validate the point (0,0,0,…) as the path origin**. Hence programming movements will be made in absolute coordinates from the **machine zero point**, which is set by the homing reset. If you notice that this machine zero point has drifted away, which should never occur, either you need to readjust all programmed coordinates with the *"File / Shift"* function, or you can indicate a path origin point that is not (0,0,0,…).

The instructions for relative movement **MOVE REL** and **ARC REL** are obviously not concerned by these considerations about the path origin point.



A manual path teach-in can be done very easily, once the origin point has been correctly validated. Simply move the axes directly to the target position, using the manual drive buttons, then **click on the blue buttons** at the top centre of the Kynon window.

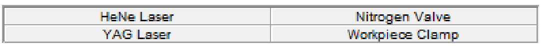
For example, you move the three axes XYZ to a given position that you wish to reach in two steps, then you click successively on *"Pos. XY"* and *"Pos. Z"*: the programme will memorise a movement to the XY position, then a Z movement to the actual point. These buttons automatically insert the corresponding instructions at the active line of the programme.

Important: remember that the **Galaad module allows you to export drawing paths to Kynon programme format**. This can be very helpful when creating paths based on full graphics.

Last of all, you can enable/disable all outputs under control by clicking their numbered rectangles, and use the *"Refresh"* button or the *"Loop"* button (cyclic refreshing) to check input state changes.

❑ User-defined buttons

In addition to numbered outputs, it is possible to **define custom buttons that correspond to your peripheral devices**. These buttons are displayed at the bottom left of the Kynon window. The *"Parameter / User buttons"* function pops up a dialogue box in which you can give each button a title, active and inactive colours. Clicking directly on a user-defined button will enable you to disable one or several outputs that are classically defined, or will make a system call to an external executable programme.



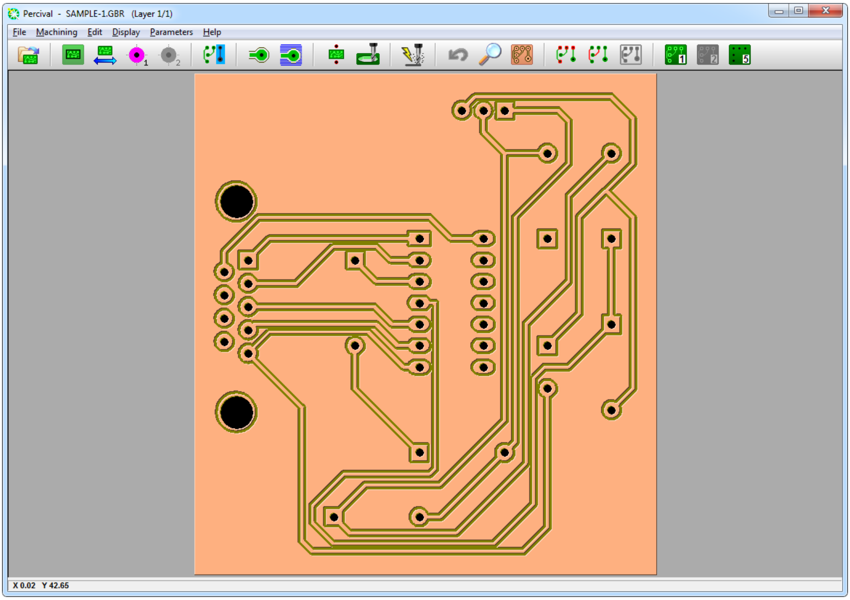
19

1 0 0 1 1

"PERCIVAL" PRINTED-CIRCUIT ENGRAVER

General features

The Percival module has been added to the Galaad suite for allowing the owners of a CNC machine to transform **Gerber** and **Excellon** files, produced by electronic CAD software, to drilling and engraving toolpaths that isolate copper pads, tracks and ground planes. This rapid prototyping module is fully integrated into Galaad, sharing the tool library and the machine parameters. In addition, and still in the philosophy of Galaad, it can directly call up the milling module without requiring heavy manipulations of intermediate CAM files, so the machining process is the immediate next operation.



Percival is not an electronic router software, and not even a manual printed-circuit design software. Its capabilities are few compared to Galaad; it provides no drawing functions, and consequently focuses simply on the preparatory works on the isolation engraving and the drilling of a circuit that has already been designed.

❑ Gerber files

The standard file format concerning electronic CAD is the GBR format (or GRB, or many other possible extensions depending on the software that produces them). This format was intended for Gerber Scientific Instruments photo-plotters. Photoengraving circuits using light-printing techniques leads to specific considerations that appear in the format. Light-printing is performed by an optical head that focuses the light beam on the circuit after travelling through a diaphragm, at the locations where the copper should be preserved. The diaphragms generally have predefined size and shape, the simplest being a basic circular disc. The pads may have more exotic shapes, but the tracks are printed using simple circular diaphragms of given diameters along the connecting path.

A Gerber file therefore contains **aperture numbers** that are either predefined in the optical library of the photoplotter, or referenced in the file itself. Light-printing instructions are very simple: the flash head can be **moved, with the diaphragm shut**, to a given XY position (movement without light-printing), or **with the diaphragm opened** (light-printed track), or even be sent to a position and the **diaphragm being opened then shut** to light-print a fixed point (pad). This makes a total of three positioning instructions, plus the number of the aperture that is currently used. The Gerber RS274-X format offers more geometrical data about the diaphragms used, the macros, the polygonal surfaces, and sometimes the drills.

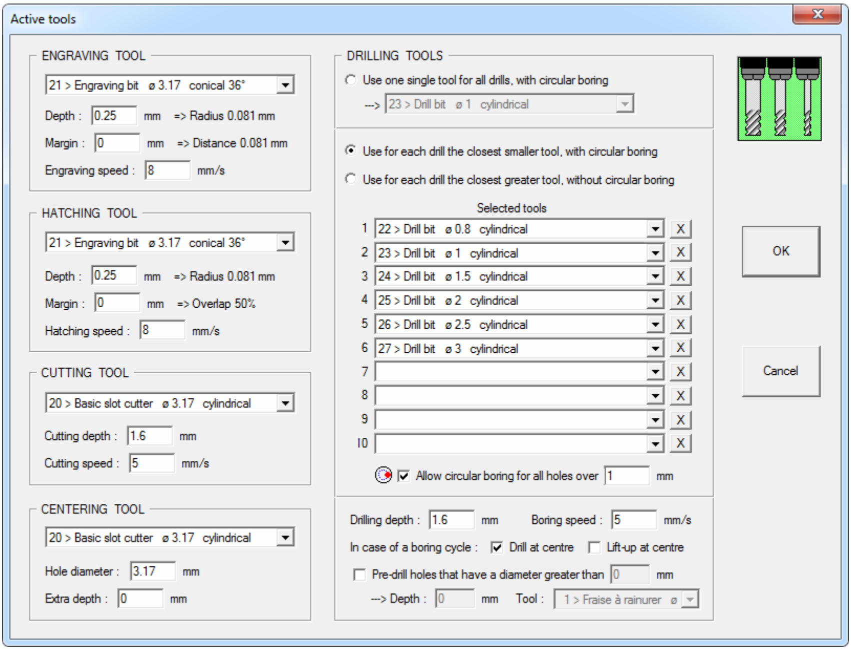
❑ Excellon files

Another file format concerns the **drilling process of pads** in a printed circuit, this format has been elaborated for Excellon Automation multi-drills. Like Gerber files, Excellon files contain XY drilling coordinates and tool numbers whose diameters correspond to the holes. These DRL (or EXL) files unfortunately do not always contain the drilling diameters, so it is necessary to fetch them in attached files that are not standard, or indicate them manually once the file has been loaded. **An Excellon file cannot represent a printed circuit, not even its pads, but only the drills**.

Furthermore, there are actually different Excellon formats, one using real coordinates (XY numerical values are formatted and indicated in immediately usable units), the other in coordinates with no trailing zeros on the right hand side, which may generate position errors if the file does not contain header information about the numeric framing format. To avoid this problem, Percival lets you select both formats under two distinct entries, which allows you to directly choose the correct numerical model. If your file looks wrong, load it again using the other available Excellon format, and all being well things should look better.

❑ Tools

The very first thing you should do after installing the software is to define the tools that will be used for drilling, engraving and cutting the circuits. The tool library is shared with the Galaad drawing module. It is accessible in Percival from *"Parameters / Tool library"* (see chapter *"Toolpaths"*, section *"Tool parameters"*). This allows you to define all available tools but, for indicating which ones among the whole set will be used for machining the circuits, you must then call up the function *"Parameters / Selected tools"*.



The tools are assigned to five types of tasks, the first of them being the **isolation engraving** for the circuit pads and tracks. It is obvious that the engraving depth must be at least equal to the copper layer so the tool reaches the underlying epoxy. The software will calculate the toolpath by offsetting it outside the circuit border, of a distance that corresponds to the appearing radius, measured at the engraving depth. The margin allows you to add something to the calculated offset. It can be negative for getting closer to the circuit if necessary. The same tool, or another tool, can be assigned to **hatching** areas that must be completely cleared of the useless copper. The hatches will be spaced by the tool radius, for a recovery of 50%, plus the eventual margin.

The **cutout** of the card will probably be done using a special cutter for milling the epoxy, a material whose dust is very abrasive. It is necessary to indicate the cutting depth, assumed to correspond to the card thickness. Setting the thickness from the function *"File / Dimensions"* updates the value that is set here.

The **centering** drills and bores the holes that allow you to put in fixing pins when the card is reversed for engraving the opposite side. The depth of these holes will be the card thickness, plus an eventual overdepth for driving the pins into the bed.

Things get a bit more complicated with the **drilling** which offers two possibilities: either you are using one single tool for all drills and this will reduce your round trips to the machine for changing the tool, or you set up a list of available drill bits so Percival will choose, for each series of holes, the tool whose diameter is the closest by default, adding a small boring, or by excess without boring (which in fact gives us three possibilities). The small buttons  let you eject a tool out of the list. Since the drill bits are generally not made for feeding horizontally, and particularly in the epoxy, you can limit the circular boring that finishes the hole only from a minimum diameter. As for the cutting job, the drilling depth is assumed to correspond to the card thickness, plus an eventual opening margin.

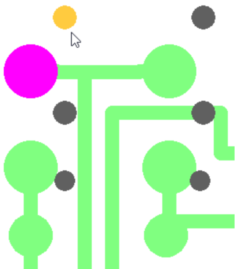
❑ Automatic functions

When opening a Gerber file, Percival searches in the new circuit or the new layer for the track nets that are connected to no pads at all. These nets can represent a cutting contour, a ground plane, a text, a reference point, *etc*. **If the net found is completely closed and surrounds the whole circuit, pads and tracks, then the software considers that it must be a trimming path for cutting the card**, since the cutouts do not have a specific definition in the Gerber format. Percival then asks you to confirm that the red-displayed path indeed defines a trimming path. If yes, the path will be transferred into the layer #6, containing cutouts. The inner trims are not detectable automatically, but you can filter them manually, as we will see later on.

If you have left enabled the function *"Machining / Copper planes / Detect when opening file"*, then the software is going to seek all track nets that are connected to no pads, and neutralise them. Neutralise means that tracks appear in brown and will be considered when calculating the isolation contours. Percival will make no differences between a real copper-plane made of a hatching of thin tracks, and a network that draws text or position crosshairs. It just filters all sets of tracks that are connected together and that touch no pads. The copper-planes are neutralised simply because the copper will be removed only around the active circuit, saving the rest of the surface. If you leave a copper-plane made of thin hatch tracks, then the calculation for the isolation might be much longer, every track of the plane being fully isolated and then submitted to a search for all possible collisions. Such a copper-plane defined by a thin hatching can requires hours of calculation, even on a high-performance computer. **It is recommended to use G36/G37 polygonal surfaces of the Gerber RS274-X format, whose isolation calculation takes only a few milliseconds**. For the texts and crosshairs, a function allows you to engrave them at centreline instead of making isolation contours. The function *"Edition / Re-enable"* undoes the neutralisation, and also the right mouse click on a neutralised track followed by *"Re-enable track net"*. You can neutralise yourself parts of the circuit; several functions are available for that.

❑ Adjusting layers

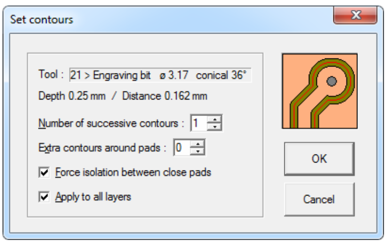
It can happen that successive circuit layers do not match, depending on how the Gerber files have been produced. It is even more common with a drill file because Gerber and Excellon are independent. You may invert or rotate the current layer with the functions *"File / Flip"* or *"File / Rotate 90°"* and their shortcut icons. **If the active layer is the layer #1, then all layers are inverted or rotated at the same time**. For other layers, including drills and cutouts, only the active one will be concerned. This allows you to apply overall or limited operations.



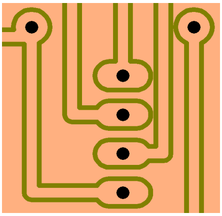
When two layers do not match one another, you can plot a pad with *"File / Reference pad / Define"* (or right mouse click on the pad and corresponding command). Then jump to the other layer, call up *"File / Reference pad / Adjust"* and plot the pad or drill that should match the reference pad but does not (or right mouse click on that pad and corresponding command). The active layer will be automatically relocated so both items match each other.

❑ Isolation contours

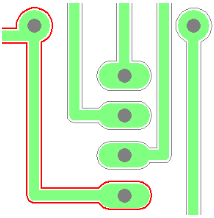
Isolating active pads and tracks requires a preliminary offset calculation, depending on the diameter of the engraving tool head, and managing the collisions when the paths are close to one another. *"Machining / Contours / Calculate contours"* (or the corresponding shortcut icon) is Percival's main function. From the geometrical data in the file, it is going to create the isolation paths. The distance from the toolpath to the border of the pad or track depends on the selected tool, its diameter, its profile and eventually the engraving depth if the profile is conical. You can add successive contours to improve the isolation, around all networks or only around pads.



When the offset calculation is completed, Percival displays the circuit with the toolpaths in thin yellow lines. You may magnify part of the screen using

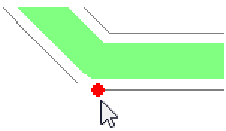


the **mouse wheel** or with the zoom icons, the **mouse centre button** allowing you to drag the zoomed view. It is also possible to visualise the isolated circuit as it will be left after the milling process, via the function *"Display / Final rendering"* or the corresponding icon, the zoom remaining active. This final rendering helps you check visually that the isolation contours are valid, especially some networks that might remain in contact because the tool was too large and could not find its way between close items.



In some particular cases, it may happen that the calculation of the isolation contour of a pad and track network remains open. Most of the time, the contour will actually be made up of two paths that complement or overlap each other but could not be welded together. Since an isolation is naturally supposed to be a closed contour, **the software displays in red the paths that remained open**, *i.e.* whose ends do not match, the purpose being to catch your attention on these. You can nonetheless disable that function from *"Display / Open contours"*.

Some manual operations on isolation contours are available. The function *"Machining / Contours / Delete selected contours"* allows you to click directly on a calculated path to be completely removed. The function *"Machining / Contours / Add segments"* lets you plot manually new isolation lines that the calculation might have missed, for example between two neighbouring pads that were too close to each other and that you wish to trim a bit. **Plotting with the mouse left button snaps automatically the ends of the open contours,**



whereas plotting with the mouse right button remains free. Reverse operation, you can remove isolation segments by using *"Machining / Contours / Delete selected segments"*.

Many functions are available from the **mouse right button**, when a pad, a track, a drill or a cut path is clicked.

❑ Limitations

Percival suffers from some limitations that you should keep in mind when working with this module:

- The isolation of copper networks requires a contour calculation around every active circuit item (pad, track, surface), then a search for possible collisions with neighbouring contours, and lastly a re-assembly of the collided items, removing residual segments. When calculating the overall contour of a copper-plane made of thin hatch tracks, this process can be extremely long. Let us insist on this: it is better that the copper-planes are defined by a simple contour polygon with G36/G37 surface encoding in the Gerber RS274-X standard, which any electronic CAD software is able to produce nowadays.

- Percival can read pads that have been defined by macros for non-classical shapes, but it cannot read parametric macros. These macros use redefinable coordinates that are given as variables **$n** passed as arguments, instead of numerical constants. They are rarely used, but it can happen.

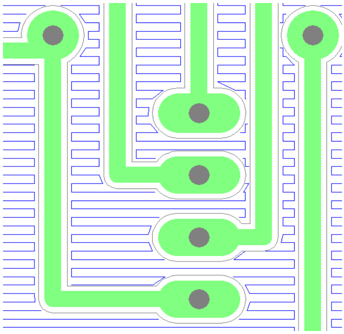
- The calculation with floating point real numbers does not have an infinite accuracy, and the roundings that are used for speeding up the process can lead to isolation paths that do not close up. The open paths are displayed in red to catch your attention. Generally, the isolation is completed but the contour is made of two or more open paths that overlap one another.

- Last of all, when the tool is too large at the engraving depth for being able to pass between two pads or tracks, there are no mir acles and the isolation is not made. Reduce the depth or use an engraving tool with a sharper head.

❑ Hatching empty zones

Engraving contours is sufficient to isolate tracks and pads, but you may want to **remove the remaining copper** from the card surface. This allows you to make sure there are no copper chips that have been left, which would create shortcuts. Furthermore, the printed circuit looks much cleaner when the copper is totally removed. Since your machine is happy to work while you enjoy a cup of tea and read your two hundred e-mails, there is nothing wrong with increasing the number of paths to engrave. The result is always better, even for a simple circuit prototype.

The hatching function can be called up only if the track and pad contours have already been calculated. This is because hatches are not supposed to touch the actual border of the active copper islands but need a contoured path. The hatching tool is the same as the contouring tool.



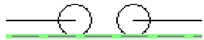
The hatching density can be set in the selected tool's parameters. It corresponds to the distance between two consecutive hatches. The default value is the tool radius at the engraving depth (for a conical cutter), which gives an overlapping ratio of 50%. But you may obviously choose a custom distance according to your needs. Percival can link hatches together to define a zigzag path whenever possible without milling a shortcut through a contour. Hatch ends approach the contours at about 10% of the hatching distance. This leaves a very small margin to avoid eroding the active copper. Logically, the hatchwork is performed before the contouring process, to obtain a better finishing of tracks borders. You may also hatch only the zones where pads are aligned at regular intervals. The software will detect these alignments.

The hatching is Cartesian, in horizontal or vertical direction. If the option *"optimal"* has been selected, then the software calculates both directions and keeps the one that will give the shortest cycle. So it is longer to calculate but faster to execute on the machine.

Here again, the final view shows the engraved circuit. If hatchwork has been defined, only the remaining copper of tracks and pads appears on the screen, unless your tool diameter was too small compared to the calculated distances. Please keep in mind that **the tool library is common to both Galaad and Percival**, and must correspond to what you actually have in the tool rack attached to your machine.

Card cut-out contour

The layer #6 contains the paths for trimming the card. These paths can have been detected when opening a circuit layer, or plotted manually, and a tool-compensated path is calculated automatically. The possible operations in this layer are very few: you can change the offset side (outside or inside), remove the cutting path, or add support bridges. These functions are available by pointing the path concerned and clicking on it using the **right mouse button**.

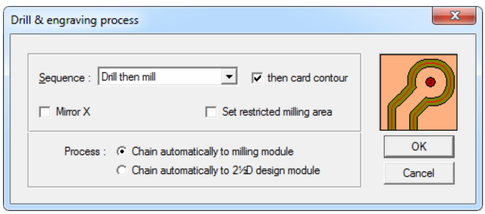


The support bridges avoid trimming the card completely but leave some linking points with the raw workpiece. You must locate them yourself, directly on the calculated offset toolpath. Warning: they will be removed if the tool compensation must be re-calculated (tool or side changed).

Milling

The display on the screen does not represent the ultimate goal, at least not with Galaad. Now we must do the business, that is wake up the machine for milling the copper and drill the epoxy.

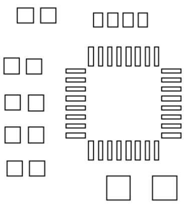
The command *"Machining / Mill"* calls up a final dialogue box which allows you to set the sequence of tasks to be completed, add depth and speed data, and then send the resulting work to the machining module, or to the drawing modules if you need to add something else.



The data relative to the sequence, tools, depths and speeds will be retrieved by Galaad or its machining module Lancelot, which can also chain to an external driver if this is how you are working. Then you just have to mill and close Percival.

Dispensing weld-paste

Two additional functions may help you weld surface-mounted components, which require dispensing weld-paste on the pads concerned. The first possibility consists of milling a simple stencil sheet that will be the template for dispensing the paste manually with a scraper blade. The function *"Machining / Cut stencil for SMC pads"* lets you choose which series of pads on



the circuit will be concerned, with a few cutting parameters (tool, depth, feedrate). The automatic chaining to the machining module Lancelot is direct, and it will be very happy to cut the stencil. Be careful with the choice of the tool, with the material and its clamping on the machine bed. Milling a thin plastic or metal sheet require some experience to avoid tearings.

The other possibility requires an electric or electropneumatic pump for ejecting the paste directly onto the printed circuit. Instead of machining, Lancelot will drive the dispensing nozzle that is mounted on the Z axis, and will control the attached pump through an output that activates a motor or an electrovalve during a settable duration. You may configure a typical dose for the paste, proportional to the surface of the pads. But the dose thus calculated for every series of pads will nonetheless remain under your full control. The dispensing cycle will just bring the nozzle to the centre of the pad, lower it down to the contact (depending on your workpiece origin set-up for the Z axis), then trigger the output that drives the pump for the given duration, and lastly lift up the nozzle for reaching the next pad. Remember to avoid dispensing paste at the centre of a drilled pad.

Surfacing the varnish on pads

In the same way, Percival offers to chop the varnish on the surface of the pads of your choice. The easiest way is to try it. Of course, you will need to set the Zo origin very accurately.

20

1 0 1 0 0

"GAHERIS" BITMAP IMAGE ENGRAVING

For the same price, the Galaad suite offers a small module for engraving images with your milling or laser machine. Parameters are shared and chaining to the machining module is automatic. The icons for framing, orientation and lighting are self-explanatory.

The machining icon gives access to the engraving settings: **dimensions** of the surface to be engraved with mesh interval or point spacing, and trimming if the surface shown does not match the image.

A choice is then made between the **conical tip** or the **pyrographic laser**. In the first case, Gaheris creates a matrix of points or scan lines whose Z depth is modulated according to the whiteness or darkness of the pixels. In the second case, either simple on/off flashing can be obtained, in which case the image can be switched to black/white without intermediate grey levels, by thresholding or error diffusion, or power modulation to mark more or less of the surface under the beam.

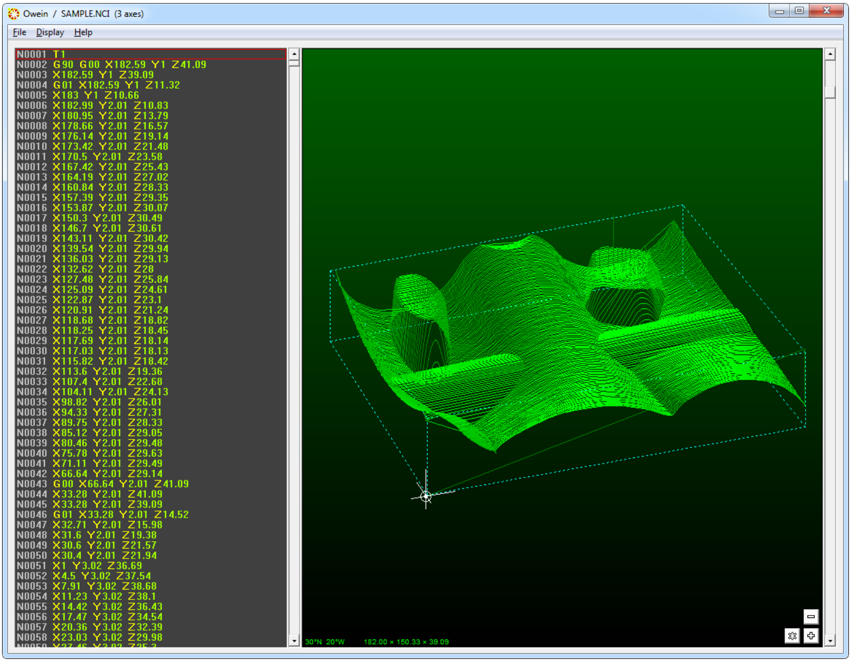
Gaheris also offers the option of producing a direct G-code file without using Galaad's marvellous machining module. That is your right, after all.

21

1 0 1 0 1

"OWEIN" GRAPHICAL TOOLPATH BROWSER

Small add-on utility, the Owein module is able to display 3D toolpaths that are encoded in ISO G-code, Isel-NCP or MasterCam-NCI files. It accepts from 2 to 5 axes, be it fully understood that the 4th axis (A) and the 5th axis (B) are classical rotary axes respectively parallel to the X axis and the Y axis, according to the current standard, turning around the linear support axis and thus defining the zero point. Obviously, XYZ axes are assumed to be linear and Cartesian axes of a direct orthonormed system.



Owein allows you to **directly modify coordinates** in the file that is displayed under ISO or NCP format at the left side of the screen. Just double-click to edit a line of code. Changes are incorporated into the display, and you may also save the modified file. Some side functions like general offset, inversion, and scaling are available, and you may remove all lines of code before or after the active one. This remains the kind of exercise that should be reserved for experts who are used to direct programming on a numerical controller display.

22

1 0 1 1 0

FIRST AID

This chapter is about the questions and issues usually raised about Galaad's different modules, with the corresponding answers or remarks.

Drawing

Galaad does not detect the licence key

- First of all, check that the dongle key is plugged into a USB port. If that key was provided before 2017, then it requires installing a driver so Windows can communicate with it. Installing Galaad from the CD also installs that driver silently and you do not have to worry about it. But if you copy a Galaad installation from one computer to another, the driver is not copied since it is something linked to Windows, not to Galaad. In that case, you must install yourself the driver from the software that is provided by the key manufacturer, also available from the Galaad website. Just validate its successive pages as they appear. Last, call ***"Help / Licence key"*** and check the type displayed.

Impossible to save my parameters or my current drawing

- You must make sure that, in your Windows user's session, Galaad is allowed to save files in its own installation directory on the hard disk (parameters files). And also, for drawing files, in the directory where you choose to save them. If you have installed Galaad somewhere in the tree *"C:\Program Files\..."* or *"C:\Programs\..."* on the hard disk, then bad luck, these directories are write-locked in a Windows basic user's session, and even worse with Windows 10 which can even lock the access to the administrator. Galaad is more a workshop software than an office software and therefore does not comply with Microsoft's *"best practices"* for the installation target, so it can be copied easily between workstations.

I have forgotten the password that protects the restrictions

- Don't panic. Use Windows Explorer and find the file PASSWORD.TXT in the hard disk directory where Galaad is installed, double-click on it and your password will appear in clear text in Windows NotePad editor.

A DXF file cannot be imported

- Galaad offers two filters for importing DXF files. The first one makes a direct interpretation and the second one via an external converter. DXF format is complex and varies within different AutoCad versions. In such conditions, it is difficult to guarantee the accuracy of the reading for older, present and future versions. The best is to try generating the file from the software that produced the drawing by using another version of the DXF compatibility. The binary DXF format, rarely used, is not read by Galaad. Neither are the DXF 3D files that contain facets or surface functions.

A PostScript (EPS or AI) file cannot be imported

- The EPS format can contain three types of graphics: 2D vector coordinates, bitmap images, and font glyphs. Galaad will not consider the last two. It can read only vector graphics, namely made of coordinates in relation to segments and curves. An image is made up of pixels with no geometrical meanings. In most cases, an EPS file which cannot be imported into Galaad is in fact a file that contains an image. When importing it into a drawing software that accepts images and zooming on paths can confirm this by showing the pixels as big squares.

A G-code ISO file displays unexpected large circles

- Due to a breach in the standard, there are two conventions for the circular interpolations G2/G3 in ISO G-code, which tarnishes the very notions of standard. One of these conventions, the most widely used, carried away by GE-Fanuc and Siemens, considers that I and J coordinates of the centre are always relative to the arc start point. The other convention, used by NUM, considers that I and J coordinates are absolute by default, or relative if the mode has been set to incremental coordinates. Both conventions are incompatible and it is very difficult to detect which one has been used by the software that generated the file. So it is up to you to choose one and try the other one if it does not give a good result.

Impossible to apply a drawing function onto an object

- Check that your object is indeed in the active layer and has not been protected (protections are visible with *"Display / Trace / Protections"*).

Impossible to obtain a tool compensation on a path

- It can happen that you have in your contour path some very small recoils of the trajectory, which may in some cases interfere with the offset calculation. Zoom in the path for checking that there are no little spines, sharp and useless and, if yes, remove them.

❑ Manual control and automatic machining

Galaad does not detect my machine

- The small button *"Detect"* in the machine parameters, *"Controller"* page, helps only for identifying an RS232 serial port, on which the machine has been connected. This detection does not apply to other types of ports, especially USB ports which do not need to be identified. In addition, some machines that communicate through a serial port do not always pass the detection test. Keep in mind that the COM ports that Windows recognises are given an asterisk beside their number, in the combo-list of available ports. Generally, there are only one or two recognised ports that you can try for real. With a few exceptions, the ports COM1 and COM2 are the physical serial ports of the motherboard, and the ports COM3 up to COM256 correspond to USB/Serial converters. So it should be easy to find out.

Galaad cannot drive my machine

- If your machine does not react at all when the software tries to open the dialogue with it, then we can suppose that either the indicated port is not the right one, or the communication is failing, or the selected machine is incorrect. First, check the machine model or the numerical controller model that you have configured in Galaad. If it looks good, then check that your cable is indeed plugged in and the machine is powered on. Some RS232 cables have an end for the PC and an end for the machine, and some machines do not communicate anymore when their protection door is open. If it is a serial port transmission, then check the baudrate.

The homing movement goes toward the opposite direction

- If your machine is an Isel with numerical controller IMC or IML, then you can reverse the direction of the homing (see chapter *"Machine parameters"*, section *"Isel-Automation IMC& IML controllers"*). If your machine is fitted with a CNC card with settable end-switches, either your axis is reversed (see *"Advanced"* page of the machine parameters), or you have indicated a wrong input number for the end-switch of the axis that goes towards the bad direction, or you have indicated a wrong polarity for the switch (*"enabled"* instead of *"disabled"* or the contrary), so the software tries to offset slowly from the switch, thinking that it is triggered. You should use the function *"Parameters / Machine / I-O test"* for checking the input numbers and polarities of your end-switches.

The homing movement does not stop

- If you are using a CNC card, then check that you have parametered the right input number for the corresponding end-switch. Call up the function *"Machining / Manual unblock"* for approaching the switch while monitoring the inputs that trigger after every movement. Otherwise, or if the checking does not say more, just check the switch itself and its electrical wiring.

The manual jogging is reversed

- The XY position of the machine zero, indicated in the *"Controller"* page of the machine parameters, does not correspond to the real position. If you have a mobile table, do not forget that the movement to be considered is the appearent movement of the tool on the table, and not the real movement of the table itself. If the inversion concerns the X and Y axes, then swap the assignment of the channels in the *"Advanced"* page of the machine parameters, or swap the motor connectors. Warning: swapping the motors may require also swapping the corresponding inputs for the end-switches.

The automatic machining runs above the workpiece

- Check that, when your have set the tool touching the workpiece top surface and validated the green button *"Z-ok"*, you have indeed selected *"Location on workpiece: upper surface"* and not *"lower surface"* in the combo-list below the green button. It is a very common mistake, fortunately with no danger. The opposite situation (touching the machine bed whilst indicating that it is the workpiece top surface) is more harmful. Take a quick look at the chapter *"Advanced milling functions"*, section *"Defining the workpiece origin"*. This cannot hurt you.

The machined result is out of scale

- Several parameters should be checked, and especially the ballscrew pitches and the motor resolutions. Some machines built in series can have different screw pitches depending on the year, and unfortunately Galaad cannot list all possible combinations. Also check that the scaling factors, input and output, are set to 1.

My machine drifts when milling

- First, try to redo the machining at very slow speed. If the offset persists, then you might have a mechanical issue, for example the workpiece not properly clamped, an axis with some backlash or a slippery motor-screw coupling (this case happens more often that we think). If the offset appears only when the feedrate is high, then you should reduce your kinematical settings. Please refer to chapter *"Machine parameters"*, section *"Kinematics post-calculation"*, and also the chapter *"Menu functions"*, function *"Parameters / Machine / Calibration"*. Keep in mind that the energy needed increases with the square of the speed: machining twice as fast requires four times more power. Speed is indeed a question of money.

The screen and the machine are not synchronous

- When the numerical controller is fitted with a local memory buffer, then Galaad sends the vectors beforehand and the machine stores them in this memory before executing them. This workmode produces very fluid motion dynamics in curves, but it desynchronises the software and the machine. The screen corresponds to the bursts of vectors sent. The software ignores the actual position of the axes at that moment.

❑ Printed circuits

The drills do not match the pads

- The drilling files and the circuit files use different formats that talk to different machines. Percival searches for a minimum correlation between holes and pads, but if your circuit contains a majority of undrilled pads or tracks that are used as oblong pads, or macro-pads, it will not find that and you must operate manually (see below).

The layers do not match

- If your files use different references or orientations, you must rotate or reverse the added layer and then use the functions for choosing a reference pad and adjusting on the reference.

The calculation of isolation contours is endless

- You have on your circuit a ground-plane made of thin hatches which require a very big calculation time for each, and hours will be needed to complete the result. Percival prefers the ground-planes that are defined by surface polygons.

23

1 0 1 1 1

TECHNICAL MATTERS

Command-line arguments

A relic from the ages of text consoles, it remains possible with Windows application software to give a starting programme some specific arguments. Obviously, no arguments are attached to the default icons associated with Galaad modules. However, you may add some indications, provided that you have a special workspace configuration, for example a computer for CAD/CAM tasks and another one for machining sessions under the operator's control. These arguments may be passed manually from the *"Run"* command of the Windows *"Start"* menu, or permanently integrated into the properties of a Galaad shortcut icon, be it an original or a simple copy. An argument is indicated in the command line after the programme call, with an intermediate space character. Several arguments may be passed one after another, and their order has absolutely no impact. If an argument has to contain spaces, you must put the whole expression between double quotes *"… …"*; also, all arguments are read regardless of upper/lower case characters.

All Galaad modules accept specific arguments. Of course the first usable argument is a file name to be opened as soon as the programme starts. The automatic links between GAL files and the Galaad module, or between KYN files and the Kynon module, are based on such arguments. For example, double-clicking on a GAL file makes Windows run Galaad and give it as argument the path and name of the file that has been double-clicked. If you do it manually, just start Galaad with the following command line (this is an example):
"C:\Galaad\Galaad.exe" "F:\Galaad Files\Drawing.gal"
This is going to start Galaad and ask it to load automatically the file *"**Drawing.gal**"* located in directory *"**Galaad Files**"* of the disk unit *"**F**"*. If a drawing is already open and not saved, Galaad will display it and will ask you if you want to save it first, unless you have enabled the option for multiple files.

In addition to the file to be loaded at start-up, the other arguments that are accepted by the **Galaad drawing module** are as follows:

- **MINIMIZE** starts the software in the taskbar (nothing on the screen).

- **MAXIMIZE** starts the software in full-screen mode.

- **RESET** removes the current drawing and opens Galaad with an empty board. This is useful only if the current drawing file has been damaged and blocks Galaad's normal opening.

- **CUSTOM** starts the software in the restricted mode that has been defined.

- **CONFIG:xxx** starts the software with the parameter set *"xxx"* that has been previously saved by the command *"Parameters / Save parameters"*.

- **ORIGINX=123.45** directly sets the workpiece origin to the value indicated for the X axis. The argument is available for each axis.

- **AUTOMILL** opens Galaad and immediately starts the machining module. This one takes over, skips the normal stages and does not ask the user for anything. Machining parameters are the default ones; the workpiece origin is the last one validated; messages to the operator's attention are not displayed, except the very last one before the actual launch of automatic process, which the operator can only confirm or abort, and which allows them to check the displayed toolpath that is going to begin (see below the AUTOSTART argument for Lancelot & Kay which is the logical continuation of this AUTOMILL argument).

- **AUTOTEXT** instantly modifies the existing text of the current drawing, whatever its style and mode, in a straight line or written along a curve. This argument must be followed by two others that indicate which block of text should be replaced and the text it should be replaced by. For example, if the drawing contains *"Old text"* that must be replaced by *"New text"*, the command line will be: **…\Galaad.exe AutoText "Old text" "New text"**. Here it is important to take care with the order of the arguments (text to be replaced immediately after the AUTOTEXT argument, and the new text immediately after the text to be replaced), and the upper/lower case of characters. If the text that is found does not exactly match the text that is given as argument, it will be ignored. You may put several successive *"**AutoText**"* arguments for multiple replacements. Also, if there are several occurrences of the same text in the drawing, the first found will be replaced, which is one less for consequent replacements.

- **AUTOQUIT** closes Galaad as soon as other arguments have been applied. This one is complementary to *"**AutoMill**"* or *"**AutoText**"*.

- **AUTOIMPORT> DIM:100x75 TOOL:3 DEPTH:0.5 FEEDRATE:20 MIRROR:X FILE:C:\GALAAD\IMPORTS\TEST.DXF** immediately imports an external file and resets its workpiece dimensions, tool, depth, feedrate, and X, Y or Z inversions. The values for the tool, depth and feedrate will replace those of all objects in the file. You may ignore some of the arguments, and the spaces between them are optional. If you are using the spaces, then you must set the whole string between quotation marks *"…"* so the argument *"**AutoIm-**

port>…" is seen as a single block. Do not forget the colons between the labels and the corresponding values.

- AUTOEXPORT> FORMAT:ISO FILE:C:\GALAAD\IMPORTS\TEST.ISO immediately exports the current file (which can also be opened just before using the GAL file argument or the ***"AutoImport>…"*** argument), under the indicated format. See the default extensions for specifying the format.

The **machining modules Lancelot and Kay** also accept a direct file name, from allowed import formats. The file extension determines the import filter. If this extension is not standard, it becomes necessary to add its standard extension between brackets as the argument immediately after the file name. Example: **…\Kay.exe …\Drawing.nc (iso)** to open the file ***"Drawing.nc"*** under the ISO import format. Other arguments that the Lancelot and Kay modules recognise are as follows, and are also valid for the **Kynon programming module**:

- AUTOSTART skips all intermediate stages without asking the operator anything, exactly like the ***"AutoMill"*** argument of Galaad, above mentioned. Machining parameters are the default ones; workpiece origin is the last one used; messages to the operator's attention are not displayed, except the very last one before the actual launch of the automatic process, which the operator can only confirm or abort. The module closes as soon as the process end message has been acknowledged by the operator, or if the process is aborted.

- AUTOSKIP allows you to avoid all intermediate messages for the operator before starting the process. See also the above argument ***"AutoStart"***.

- AUTOUPLOAD starts the machining process exactly like the above mentioned ***"AutoStart"*** argument, except that the machining path is sent to the local memory of the machine with no run command (*"Upload machining to the controller memory"* mode of the machining options).

- AUTOSTORE does the same as the ***"AutoStart"*** argument, except that the upload goes to the local disk drive of the numerical controller.

- AUTOTOOL makes an automatic measurement of the tool on its sensor as soon as the initialisation dialogue with the machine is completed. Of course the tool is supposed to be already mounted on the spindle.

- AUTOPARK1 sends the tool to its parking position as soon as the initialisation dialogue with the machine is completed. The variant ***"AutoPark2"*** sends to the tool change position.

- OPERATOR locks the access to the machining parameters. The operator can only set the workpiece origin and start the automatic process.

- ORIGINX=123.45 sets directly the workpiece origin to the value indicated for the X axis. The argument is available for each axis.

- MANUALM opens the Lancelot module on its manual jogging window for the milling machine.

- MANUALT opens the Lancelot module on its manual jogging window for the lathe.

❑ Integration into an automated chain

Galaad does not need external software from creation of the drawing to the automatic machining process. Except for the specific case of machines or spindles that cannot be driven directly and consequently require a call to an external driver, it is therefore not necessary to include Galaad or part of it in a heterogeneous processing chain.

However, and once this has been made clear, it is still possible to find room for Galaad in a wider set, in which case it will do its best to cooperate with its workspace environment. Obviously, the commands that should be passed to the software can only be carried through the command line, whose arguments have been mentioned above. Considering that it is a programme that starts Galaad at a given moment in an automatic process chain, the four main things that can be expected from Galaad are as follows:
1 **-** load a given GAL file name;
2 **-** modify blocks of text in this file;
3 **-** start the machining process and skip the operator's dialogue stages;
4 **-** close itself once the process is completed.

As seen before, there exist some arguments that Galaad can understand which may help you build that sequence, stages 3 and 4 being merged into one single argument. Please refer to the arguments above described for more details. It is obviously not possible to tell Galaad to create or modify pure graphic items from the command line, which would be very difficult, requiring too many parameters. It you wish to create a drawing automatically and start its machining process with no operator interference, then the best way to do it is to create a file in a standard format that Lancelot or Kay can read, and call them up with the corres-ponding file name as an argument, followed by an ***"AutoStart"*** argument that will launch the automatic milling process and will even close the module once the cycle is complete.

If Galaad's machining task is performed by an external driver, it will be opened automatically, provided that the corresponding arguments have been passed to the software.

For a more complicated process that requires advanced input/output controls, pause delays or others, the Kynon module should do the job since it is also able to load and execute immediately any programme that has been generated by external software, using the same ***"AutoStart"*** argument. The format of Kynon programme files is text type, and the syntax is not very complicated, looking very much like the instruction list on the screen. It is important to use this syntax and particularly the spaces, because bad instruction lines will be purely ignored.

As a conclusion, anything that can be done by Galaad modules under the control of an operator can also be done automatically using command line arguments. These may be a bit difficult to handle by a user from the Windows desktop, but much easier to integrate in a calling application that encapsulates Galaad in a superset, including a batch process.

❑ Message-transmitted commands

For programmers, there is another possibility of directly calling up some machine driving functions, based on Windows inter-applications message system and limited to Lancelot and Kynon. When these are in manual jogging mode, they are listening to WM_COPYDATA messages that you can post through the function *SendMessage()* of Windows API. The structure COPYDATASTRUCT encapsulates the data to be transmitted to the module, its address being given by the argument LPARAM of *SendMessage()*. In this structure, the value *dwData* must be 26912021, the pointer *lpData* targets the ANSI string of the command to be executed, and the value *cbData* gives the size of the string, including the terminal zero.

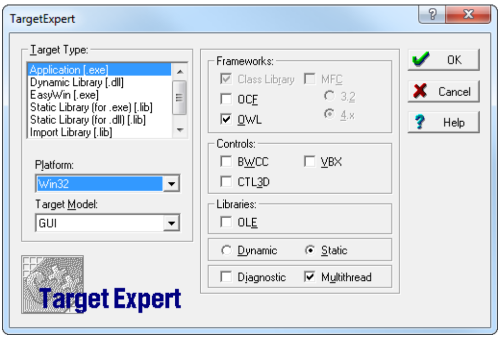
For recovering the *handle* on the Lancelot or Kynon window, you can use the API *FindWindow()*, considering that the module *ClassName* is the programme name with its full path, for example *"C:\GALAAD\KYNON.EXE"* (all in upper case). You can also use the window title, but it is subject to variations. The commands to be transmitted are those you can find in the tool change scripts. If the calling application has indicated the *handle* of its own window in the argument WPARAM of *SendMessage()*, then Kynon will return the result as another WM_COPYDATA message, acknowledging the command once it is executed. The returned data structure is identical, the data buffer *lpData* now containing the text *"OK"* or *"ERROR"*.

The small programme COPYDATA.EXE which is installed along with Galaad allows you to send commands to the module. The first argument is the name of the target module, and the second argument (between quotation marks) is the command to be executed. Example: **COPYDATA KYNON "MOVE TO X12.34 Y56.78 V600"**

❑ Interface with Windows

This section is for those users who have a solid understanding of the control mechanisms within the Windows environment. The various technical details described below can help when trying to resolve the possible problems associated with either loading or using Galaad.

Remember that Galaad is a coherent collection of files, comprising modules created with the Borland C++ 5.02 compiler, and intended exclusively for use on the **Windows 32 bit** platform in graphic mode, that is Windows 95, 98, ME, NT-4, 2000, 2003-Server, XP, 2008-Server, Vista, Seven, 8, 10 and all the successors that Microsoft will be happy to endow us with in the future. **These 32-bit modules work perfectly in the standard**



encapsulations of the 64-bit systems (XP-64, Vista-64, Seven, 8 & 10). You do not need to change or parameter anything. The only restriction is the write-access in the directory where the software is installed, from the user's current session.

As already mentioned in the chapter on installation, **Galaad does not add any files** to your computer outside its installation directory, with the exception of a couple of shortcuts placed on the desktop and in the *"Start"* menu. These shortcuts, GALAAD.LNK, LANCELOT.LNK, *etc.*, are placed in the usual subdirectories of C:\WINDOWS or C:\WINNT and are of the classic type, associated to the executables. Two new keys are added to the Registry, in HKEY_CLASSES_ROOT\GALAAD and HKEY_LOCAL_MACHINE\GALAAD to associate *.GAL files with Galaad. Nothing else is done to the registry. Should you ever remove Galaad and delete its folder, then you can find and remove its registry keys if you really want to, but Windows will not be bothered if you forget.

The executable files of Galaad suite are compact and self contained, created entirely using the Borland *Object Windows Library*, without the use of exotic VBX, OCX controls or other adornments. Some DLLs that have been developed by AxeMotion, Isel-Automation, Soprolec and others are added for driving their numerical controllers. The core module of Galaad is the programme GALAAD.EXE, which manages the drawing and working environment. For convenience, the machining and manual control module has been created as a standalone executable programme, LANCELOT.EXE, which is automatically called up by Galaad, who passes it all necessary arguments and files. This enables the machining to be run as a background task while Galaad returns to perform further drawing work, if the noise (which is not always in the backgound) allows it. The other modules are independent and work autonomously, except the drawing module for turning GAWAIN.EXE which also uses the module LANCELOT.EXE for driving the lathe. Shared space.

Windows undertakes all of Galaad **file management**, including use on a network, and therefore you can use long file names and any legal Windows features available in this area. The file extensions of Galaad 3 are *.GAL, the library objects are *.GLI and the templates are *.GLT. Double-clicking on a GAL file causes Galaad to be started, and the file opened and displayed.

To prevent worries about portability, **communication** with the machine uses the standard stream control functions of the Windows API, (*CreateFile(), ReadFile(), WriteFile(), etc.*), rather than special functions that access the ports directly. This is to provide compatibility with the latest and future versions of Windows. However, in the computer area, it may be wise not to predict too much.

A little reminder: if your machine is fitted with a **special spindle,** that is started or stopped with codes other than the default ones built in, Galaad can substitute codes provided by yourself. It is also possible to use an external DOS or Windows programme just for the spindle. Please refer to the chapter *"Machine parameters"*, section *"Spindle"*.

❑ TrueType and Galaad fonts

Galaad 3 uses standard TrueType fonts or compatible fonts having the Windows *flag*, TRUETYPE_FONTTYPE. These vector fonts are constructed from two types of geometrical objects, polylines and Q-Splines. The hatching or clearing out of the interior is undertaken automatically by Galaad, which detects the external contours and any islands. Specifying Italics uses the corresponding TrueType font, however, if there is no italic version available, you can always incline the characters yourself and specify the angle of slanting. Alternatively, if the italic version exists, that angle will be added to the current slanting.

You may find TrueType **fonts that have no pen thickness**, though Windows cannot use fonts with simple lines (for example a capital 'I' drawn from a single line, actually a rectangle with no width but nevertheless a to-and-fro movement). In such case, Galaad is able to get rid of the overlaid vectors, which saves time when milling, even if the vectors do not match absolutely.

The old fonts from Galaad 1 & 2 are still available to provide simple line characters that do not exist as TrueType fonts, or equally fonts where the hatching is already done. These fonts, approximately fifty of them, are installed in the FONTS sub-folder. Three extensions are used: GLF for *Galaad Light Font* (simple line fonts, very useful), GOF for *Galaad Outline*

Font (thick contoured fonts and possibly hatched) and GSF for *Galaad Special Font* (special fonts such as Braille).

❑ Parameter files (and other files)

In addition to the executable modules, we can find many files in the software installation directory. So that you can copy thoses that may be useful to you, the simplest is to describe what they are used for, in alphabetical order of their extensions. All these parameter files are text type and so are they editable with Windows NotePad, but it is better to change their content only if you know what you are doing…

- **GALAAD.CNC** *(Computerised Numerical Control)* contains all machine parameters.

- **GALAAD.CUS** *(CUStomised workspace)* contains the restrictions that you want to apply to the workspace.

- **GALAAD.FLT** *(FLaTness data)* contains the matrix of the Z-plane measures for the machine bed.

- **GALAAD.INI** *(INItialisation data)* contains all workspace parameters (except the restrictions). Warning, this file also contains the name of the current drawing and its directory (but not the drawing itself). If you copy it to another installation that works on another drawing, there will be a confusion. In addition, the recent file names are stored here.

- **GALAAD.IPC** contains the plasma torch cutting database for the Soprolec InterpCNC-2 controller.

- **GALAAD.JDC** contains the plasma torch cutting database for the JD-Cut machines.

- **GALAAD.LAS** contains the cutting and engraving database for laser works.

- **GALAAD.MAC** *(MAChining data)* contains the machining parameters overall: forced values, scales, iterations, 4-axis milling, *etc*.

- **GALAAD.NET** *(NETwork data)* contains the network parameters.

- **GALAAD.OBS** contains the definitions of roadblocks on the machine bed.

- **GALAAD.OXY** contains the oxycutter setting database.

- **GALAAD.PIN** contains the objects of the fast access column (at right).

- **GALAAD.PLC** *(PLasma Carving)* contains the database for simple engraving using the plasma torch (with reduced current).

- **GALAAD.PLS** *(PLaSma data)* contains the plasma torch cutting database for a basic controller without THC function.

- **GALAAD.PPF** *(Post-Processor File)* contains the syntax and the parameters for the post-processor.

- **GALAAD.PRT** *(PRinTing)* contains the printing settings.

- **GALAAD.PTN** *(Plasma Torch Nozzles)* contains the nozzle parameters for the plasma torches. It does not contain their statistics of use.

- **GALAAD.STA** *(STAtistics)* contains the statistics of use for the tools and the plasma nozzles, plus the main statistics for the machinings.

- **GALAAD.THC** contains the plasma torch cutting database for the Thunder-Cut machines.

- **GALAAD.TOO** *(TOOl library)* contains the tool parameters.

- **GALAAD.UDC** *(User-Defined Controller)* contains the syntax and the parameters of the customised generic numerical controller.

- **GALAAD.XON** contains for debugging the exchanges between the software and the machine of the very last machining or manual control.

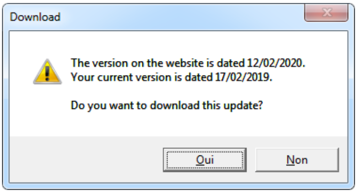
- **GALAAD.XYZ** contains the calibration data (tool sensor, laser pointer, torch recoil sensor) and the absolute positions on the machine (current origin, memorised origins, parking) and also the last settings used for the plasma torch, the laser, the oxycutter or the hot wire.

The modules Gawain, Kay, Kynon, Owein, Percival use these same extensions for the same types of parameters. If these files do not exist for them, it is simply because they are shared with Galaad, for example the file GALAAD.CNC which is used for all modules that access the machine parameters. You cannot set up a machine in Galaad and another one in Kynon. In that case, you must have two different installations which will coexist peacefully on the same hard disk. You just have to identify clearly the shortcut icons on Windows desktop.

The current drawing file is **GALAAD.CUB** *(CUrrent Board)* and it is a simple GAL format, which is binary and not documented. If Galaad freezes at start-up, then possibly that file has been damaged. Yet just delete it manually after making a copy elsewhere. The file **GALAAD.CMB** *(Current Machining Board)* is the one transmitted for machining and therefore exactly the same as long as the drawing remains unchanged.

Downloading updates

Galaad evolves regularly. If bug fixes have become rather rare over time, some new functions are still being added here and there, at random and in accordance with suggestions. These developments can be of interest for you, especially since the update in the same version is free and easy when you have a direct access to the Internet: call up the function *"Help / About"* and click on the button *"Download update"*. If your Galaad workstation is not connected but you can still access the Internet from another computer, then just grab the update file GAL-ENG.ZIP (version in English) and then copy it into the installation directory of the software and restart Galaad. It will detect the presence of the file and will be in charge of the update. **Updating the software does not affect your parameters** nor your current drawing board.



24

10100

A FEW TIPS FOR BEGINNERS

Reading many Internet forums dedicated to CNC machinery, it appears that many persons of good will rightly hesitate before embarking on the purchase, the modification, or even the build-up of a machine, fearing that the consequences of a mistake become expensive because difficult to correct. This chapter is a bit off-topic in the user's manual of a piece of software that provides a large number of parameters for calibrating your machine, and will do its best for compensating possible defects. The purpose of these lines is not to settle the few items presented as definitive and condescending answers, but rather make you ask yourself the right questions or, at least, ask yourself some useful questions.

- Mechanical structure

Before getting excited about a type of machine, think about what you intend to do with it. The requirements in rigidity are not the same depending on whether you are going to mill expanded polystyrene or tempered steel. Strong rigidity means heavy structure, and heavy structure means big inertia. If you want to machine quickly, choose for preference a light machine with an aluminium profile frame. If you want to machine metal with greedy passes, you will not machine quickly and inertia is less important, as long as the assembly is solid. Obviously, if you want to machine metal at high speed, then you will have to put a price on it, to have both good rigidity and high driving power. It is possible to consider machining a material harder than the machine body, provided that every milling stage removes a small amount of material. But in general, it is still better to have a more rigid structure than the workpiece to be machined.

- Bending

The critical point in a machining effort is the tool that feeds in the workpiece. The resistance of the material strains the elements of the structure in a chain, which will eventually distort, causing a loss of precision. The first one concerned is obviously the Z axis. In theory, the longer the stroke, the thicker the parts can be machined, at least if you have a suitable tool. But the lower your spindle goes, the further it moves away from its guiding rails that remain at the top on the gantry, and the more your axis will tilt when the workpiece resists the toolfeed. Choosing a long Z axis is a miscalculation if you do not need it. However, you can compensate by using a very thick machine bed to raise the workpiece so the machining point is brought closer to the gantry.

- *Guiding rails*

As long as the machine body does not warp under the power of machining forces, at least at reasonable feedrates, the quality of an axis can be considered to depend essentially on its guiding rails. Nature being fundamentally hostile to any kind of intelligence, a guided carriage will always take maximum advantage of the small amount of clearance allowed for trying to get some contorsion. If the rail supports only one skate, the longer the skate is, the less freedom it has. If there are two skates, the further apart they are from each other, the more limited the twisting force will be. It should also be noted that poor parallelism between two guiding rails of the same axis can cause a rapid wear of the skates. A fine ear, possibly armed with a stethoscope, a dynamometer and foil sheets are the best friends of your rails.

- *Transmissions*

The only thing requested from a transmission is to have no backlash. There are still mechanical designs whose guiding quality is partly based on the transmission, for example a double screw on a wide axis. This is equivalent to admitting that the guiding rails are not optimal or that the body is bending. On a perfectly rigid structure whose axes cannot twist, the positions of the transmission points do not matter. Instead, focus your efforts on the rigidity of the structure and the quality of the rails. Also, if you use V-screws or toothed belts, do not rely too much on the software for managing a reversed backlash compensation. A classical manual milling machine can easily accept such a backlash compensation on one axis, provided that the other axis is blocked by a mechanical brake. Such blocking does not exist in numerical control. When a transmission has some backlash, its axis will oscillate by the value of this backlash when it is supposed to be stationary while other axes are moving, depending on the forces involved in the machining. This erratic backlash cannot be compensated. If an axis has some backlash, it is much better to fix it by a mechanical action.

- *Coupling*

Aligning perfectly a motor on a screw is mission impossible. You must insert a flexible system that will absorb the misalignment: either a mounting of the motor to the chassis on silentblocks, or a cardan or homokinetic joint, or two pulleys and a toothed belt. In the latter case, keep in mind that the larger the diameter of the pulleys, the less the angular error caused by the elasticity of the belt or its backlash on the pulleys results in a linear error on the screw. But the more the inertia increases.

- *Accuracy*

We can sometimes hear promises of 0.01 mm accuracy with a machine made with a light aluminium structure. Let us not dream too much. At best, you are actually told about the linear resolution of the motors, which has nothing to do with mechanical accuracy. Unless you mill polystyrene (and again, given its elasticity), you can only expect to reach a machining accuracy of 0.01 mm with a heavy-duty machine, made of cast iron and reinforced on all diagonals, whose feet are sealed in the concrete slab of the workshop, equipped with high-end guides and transmissions. Be realistic about mechanical performance.

- *Motor resolution*

The accuracy of a machine depends solely on its mechanical design. The movement fineness of the motors does not contribute to the accuracy. If your transmission has too high a ratio, *i.e.* one motor revolution produces too long a linear movement, for example with toothed belts, or racks and pinions, or cables, then increasing the resolution of the motors is not the solution, since the torque remains unchanged. On a stepper motor, the torque even tends to decrease when increasing the microstepping mode. And the torque curve of a stepper motor is drops sharply above a certain speed. It is therefore appropriate to stay below this critical speed. A low transmission ratio requires a high rotation speed, which may exceed the torque drop point. If you want a fast machine, better use long screw pitches with powerful motors that do not rotate too fast. Increasing the resolution is mostly used for reducing vibrations, because the rotation in microstep mode is smoother. However, pushing the resolution further is generally useless.

- *Power electronics*

The threshold for the torque drop on a stepper motor is related to the charging time of its coils. As the rotation speed increases, the coils have less time for getting charged. If this charge is too slow, then it will be incomplete and the accumulated energy will be reduced, causing the torque to fall. For making a coil charge quickly, a high voltage is required. The speed of a stepper motor is therefore linked to the voltage of its power drive. But be careful then, handling high voltages is dangerous. Think first about your safety. In the same vein, an inadequate power supply will inevitably cause losses of steps or drag errors from a certain speed when several motors need power simultaneously or in resonance. As far as possible, it is preferable to have a power supply for each axis, and largely oversized.

- *Control electronics*

The performance of a CNC system is not measured in KHz on its output signals, but in its ability to process the maximum number of vectors in a minimum of time. In addition to an optimised command language, it is important to have a high bandwidth so the transmission always runs faster than the machine. When this is no longer the case, there comes a time when the machining gets jerky, because the numerical controller has no more motion vectors in its local memory buffer. A USB port does not always guarantee high bandwidth; it depends on how it is managed, the exchange protocol and the transmission mode. In addition, a large buffer size helps to absorb the fluctuations of the gap between the transmission and the execution of the vectors. On most machines, a small movement and a long movement require as much time to be transmitted. But the long movement will take time to execute and the transmission will then be able to take advantage of it for getting far ahead, provided that the size of the buffer allows it. This will prevent a critical situation when, on the other hand, a large number of small motion vectors have to be transmitted, their transmission being slower than their execution on the machine.

- *Motion dynamics*

Fast machining requires big technical – and therefore financial – resources. Keep in mind that, if the energy required to move a given mass increases in direct proportion to that mass, it also increases with the square of the velocity. Therefore, driving an axis twice as heavy at the same speed with the same acceleration certainly requires twice as much power, which must be taken into account with long ballscrews or those having a large diameter. But, on the same machine, moving twice as fast requires four times more power; moving three times faster requires nine times more power, *etc.* It is easy to move an isolated axis at a very high speed, as long as you apply accelerations and braking ramps long enough. In automatic machining, the feedrate is supposed to be more or less constant. It is therefore out of question to spend most of the time in endless acceleration and braking phases that will cause the average speed to drop and make the required cruising speed completely inaccessible anyway. *Festina lente*.

- *End-switches*

If your machine works using absolute coordinates, with an invariable zero point (sometimes called *"machine origin"* or *"machine zero"* as opposed to the workpiece origin), it is necessary that it can be set with the best accuracy as possible. A lack of reliability will affect all fixed positions reused by the machine. Microswitches with tabs such as the Cherry DB1 or Saia-Burgess XC are fine, provided that they are facing the mobile carriage rather than being mounted laterally in its path. A reliability of the click point of about 0.01 mm is then easily achieved with a microswitch of good quality. It goes without saying that the switch must be protected by a mechanical stop that will prevent it from being destroyed by crushing if the carriage does not brake fast enough or if a configuration error prevents it from stopping when the contact is triggered. If your switches have too short a stroke, simply bend the tab to increase it and thus avoid knocking on the mechanical stop when the homing movement is fast. With an almost infinite lifetime, inductive sensors also offer excellent reliability, but their use does not always allow a braking distance after the trigger. For the tool sensor, a simple vandal-resistant switch such as the ITW 76-95 is sufficient for obtaining an excellent reliability.

- Spindle

It would be a pity to have a rigid machine, well-guided and fitted with solid transmissions, for moving a low-quality spindle. And unfortunately, it is very easy to find poor spindles whose bearings quickly become loose, in addition to any weakness they may have already when leaving the factory. A bad spindle will give bad results and will wear down your tools faster. Do not neglect it in your budget. Choose spindles with high-quality bearings, such as ceramic, and if possible including four bearings.

- Cooling

The faster the spindle turns, the more the tool heats up by friction. And the more the tool heats up, the faster it will wear down. Tools are expensive: if you mill at high speeds, or if you mill hard materials using deep stages, then please feel sorry for them and think about fitting your machine with a coolant device: watering or spraying, venturi chilling system, even a simple flow of compressed air can be enough to lower the temperature significantly.

- Software

If you are reading this user's manual, then you probably already have a Galaad licence. Excellent choice, nothing more to say.

25

11001

GLOSSARY OF TERMS

The aim of this glossary is to provide a guide to the meaning of certain words used in this manual, which may not be apparent to an inexperienced user.

2½D: motion generated by the simultaneous interpolation of any 2 axes with the third one being stationary (engraving or cutting application).

3D: motion generated by the simultaneous interpolation of 3 axes. A toolpath whose depth varies is presumed to be 3D.

Arc: circular or elliptical curve, opened or closed.

Axis: motor-actuator assembly enabling either linear or rotary movement of the spindle or the workpiece.

Axis X: horizontal linear axis, X+ moves from west → east.

Axis Y: horizontal linear axis, Y+ moves from south → north.

Axis Z: vertical linear axis, Z+ moves from low → high.

Axis A: rotary axis parallel to the X axis.

Axis B: rotary axis parallel to the Y axis.

Axis C: rotary axis parallel to the Z axis.

Baudrate: units of speed for serial data communication (bit-per-second).

CAD/CAM: Computer Aided Design / Computer Aided Manufacturing.

Cartesian coordinate: position defined by its linear distance from the origin along the X, Y and Z axes.

CCW: Counter-Clockwise.

Chip-breaking cycle: drilling operation where the bit drills down to a given depth stage, then retracts slightly up for breaking the swarf spirals, before descending and drilling a further step. The cycle is repeated until the required final depth is reached.

Chip-clearance cycle*:* drilling operation where the bit drills down to a given depth stage, then retracts completely out of the hole for deburring the swarf, before descending and drilling a further step. The cycle is repeated until the required depth is reached

Circular interpolation*:* simultaneous movement of two axes, resulting in a circular arc.

Clearance height*:* a safe vertical distance above the surface of the workpiece, that the tool retracts to before making any rapid lateral moves.

CNC: Computerised Numerical Controller.

Contouring*:* a compensated toolpath, usually around the inside or outside of a closed shape, such that the centre line of the tool is at an offset distance of half the diameter from the trajectory of the shape.

Cut-out*:* machine an object at a depth equal to or greater than the thickness of the material.

Cutter: see *"Tool"*

CW: Clockwise.

Digitise*:* operation where the points on the surface of an existing object can automatically be obtained by using a suitable digitising probe.

Drill*:* operation where the cutter descends into the workpiece to make a hole. (See also *"Plunge"*).

Encoder*:* precision measuring device mounted on a servomotor to provide feedback to the CNC as to the exact position of an axis.

Feed*:* motion whilst cutting material either horizontally or in 3D.

Feedrate*:* speed of motion whilst cutting material either horizontally or in 3D.

Finishing: a second machining pass following a roughing operation, to remove any residual stock and provide an accurate finish (see *"Roughing"*).

Homing: special movement where the machine resets the position of the axes to a fixed zero by sending each one, individually, to find the built-in reference switches.

Inactive movement: any movement when the tool is retracted above the workpiece, including the one for pulling up.

Initialisation: operation to establish communication with the CNC.

Island: material within a pocket that is to be left intact.

Linear interpolation: simultaneous movement of two or more axes, resulting in a straight line.

Machine origin: see *"Machine zero point "*.

Machine zero point: absolute zero position (0,0,0) of the machine, determined by in-built reference switches.

Manual control: the CNC is controlled directly by Galaad to determine the position of the workpiece origin.

Override: real-time increase or decrease of the final speed (plunge or feed).

Pass: a single machining operation, (roughing, finishing or cutting out) of the drawing.

Plunge: the phase where the cutter descends into the workpiece to the required depth, before making any lateral moves. (See also *"Drill"*).

Pocketing cycle: a series of moves, each at a fixed depth, to clear out material from a defined area.

Polar coordinate: position defined by its radial distance from the origin and the angle from the X axis.

Port: port on computer to which the communications cable to the CNC is attached.

Post-processor: programme or function that produces a dedicated machining code for the CNC in the required format.

Profile of a tool: cross sectional view of a cutter (cylindrical, conical, hemispherical etc.).

Rapid movement: non-cutting motion where the tool is clear of the workpiece, made at the rapid speed of the machine.

Resolution: the smallest movement the machine can make.

Retraction: vertical movement when the tool is pulled up above the workpiece.

Roughing: initial course cut to remove the bulk of the material rapidly, whilst still leaving a small amount of stock material to be cleaned up with a finishing pass. (See *"Finishing"*).

Snap: a jump to the nearest position meeting the snapping criteria.

Spindle: device for holding and spinning the cutting tools, usually driven either electrically or pneumatically.

Stage: an intermediate Z pass, repeated by levels until the final depth is reached, due to the fact that the tool has a limited cutting height.

Stock: material remaining after a roughing pass, which will be removed by the finishing pass.

Surfacing: operation for cleaning up the top surface of a piece of material so that it is accurately parallel to the XY plane of the machine.

Tool: any cutter, including slot drill, end mill, router, engraving tool, *etc*.

Tool centre: vertical axis around which the tool rotates.

Tool cycle: complete machining operation for one tool.

Tool sensor: device for automatically measuring the length of a cutter mounted in the spindle.

Tool shank: parallel portion of the tool that does not cut and is held by the collet in the spindle.

Visual: refers to an object used in the construction of the drawing, but not machined.

Workpiece origin: reference point on the workpiece to which all positions and moves are referred.

Printed in Nice, France by

facimprimeur

www.facimprimeur.fr