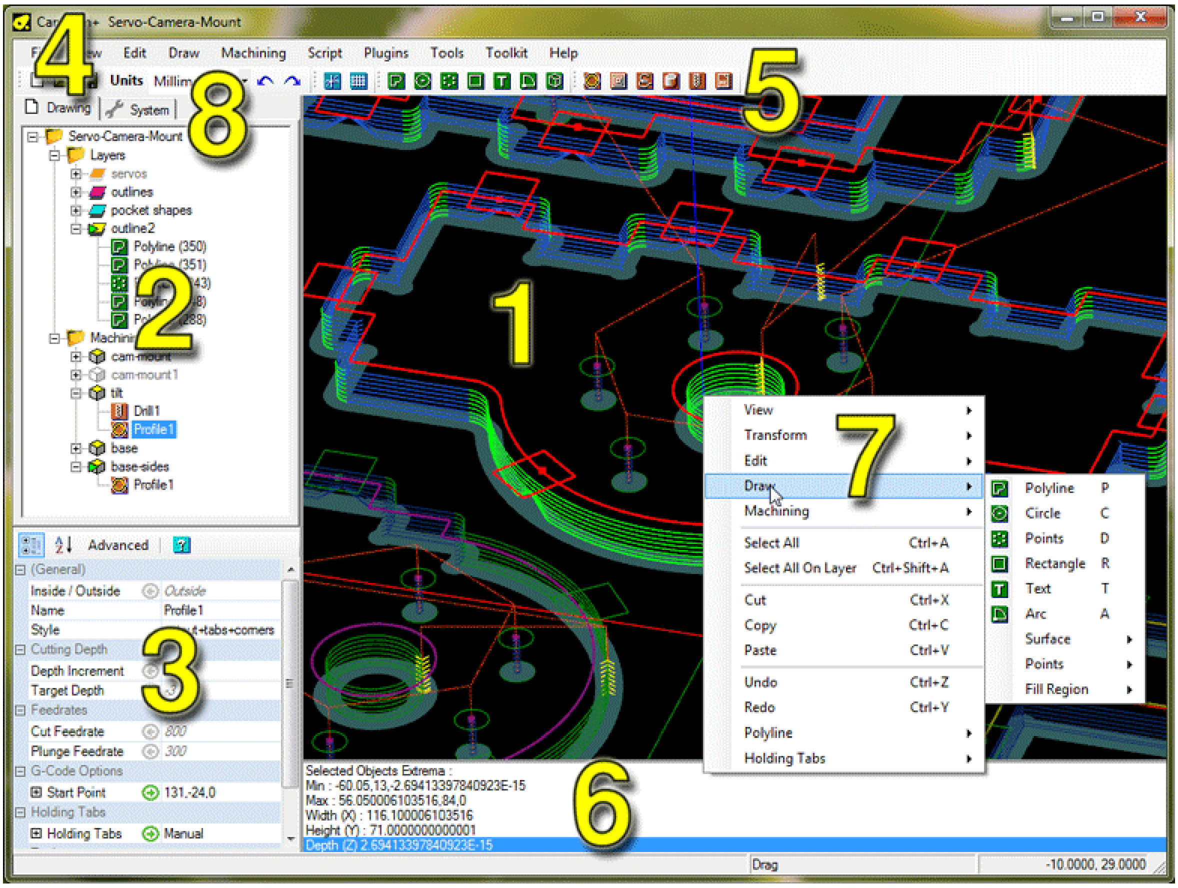
Contents

User Interface

This section introduces parts of the CamBam user interface and explains some terminology used.



1. Main Drawing Window

3D View of the current drawing and toolpaths.

2. Drawing Tree View

Shows all layers, drawing objects and machining operations (mops) in the current drawing.

3. Object Property Window

Display and edit properties of objects that are selected in the drawing window or drawing tree.

4. Main Menu Bar

Main menus for the application.

5. Tool Bar

Short cuts to commonly used tools and settings.

6. Message Window

Errors, warnings and informational messages are displayed here.

7. Drawing Context Menu

Menu for commonly used routines and operations applicable to selected objects.

8. System Tab

Provides access to settings common to all drawings such as general configuration settings, tool libraries, machining styles and post processors.

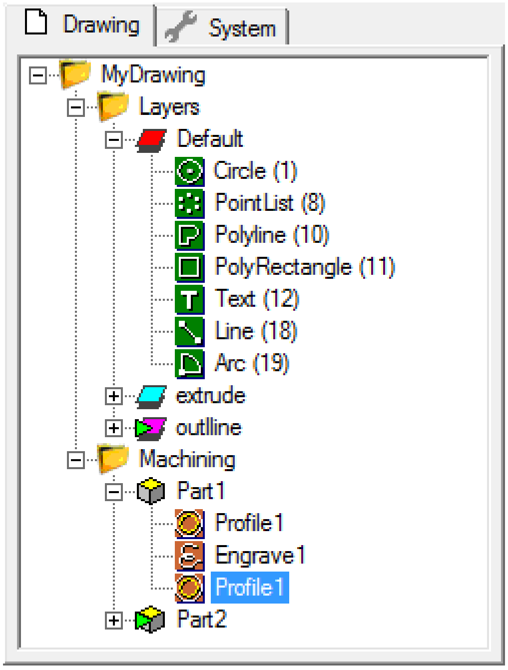
Drawing and System Tabs

Two tabs are available above the tree view, at the left side of the CamBam window: Drawing and System.

The ***Drawing Tab*** shows the contents of the current open CamBam drawing.

The ***System Tab*** contains libraries and settings common to all drawings.

Drawing Tab



The ***Drawing Tab*** displays the contents of the current drawing file, presented in a tree layout.

The first item of the drawing tree contains general settings specific to the drawing. This top object will be labeled using the name of the drawing file. In the example pictured, the file is titled : 'MyDrawing'.

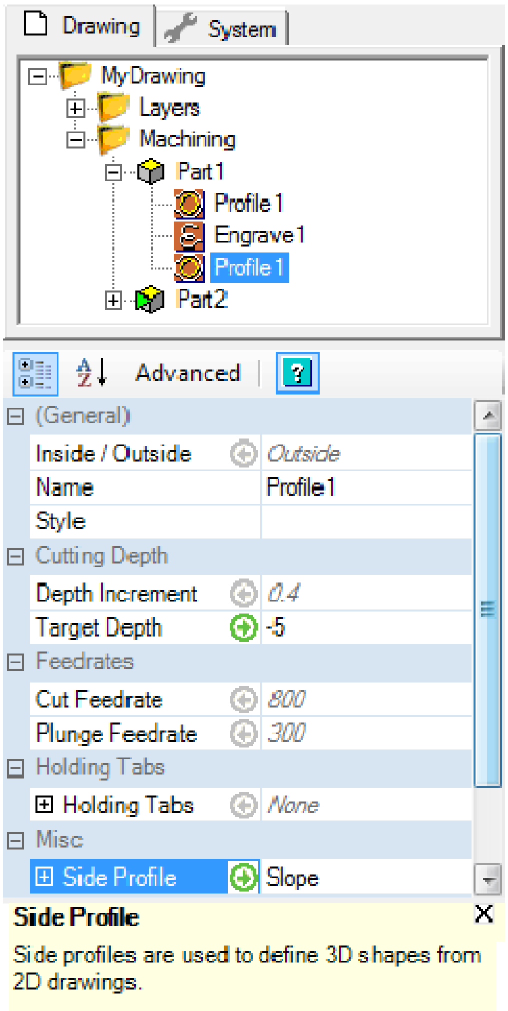
The drawing is then divided into two main sections: ***Layers*** and ***Machining***.

Layers are used to separate the drawings items into manageable sections which can be labeled, color coded, hidden and made visible to aid CAD design. The drawing tree shows the name of each layer and the color used to display drawing objects contained within the layer.

Expanding a layer within the tree shows the drawing objects in the layer. The icon and name of each item denote the drawing object's type. The object's ID is shown in brackets. All objects within the drawing have a unique identifier number.

The ***Machining*** folder is further divided into ***Parts*** which in turn contain all the machining operations used within the part.

Property Window



Selecting items in the tree view allows their properties to be modified in the property window below the tree.

In the image shown, the properties of the operation ‘Profile1’ in ‘Part1’ are displayed in the property window.

The size of the tree and property window can be adjusted by dragging the left mouse button on the dividing line between the two sections, when the mouse cursor changes to an icon. The property window’s column size can be adjusted by dragging with the left mouse button on the column divider, when the mouse cursor changes to: .

The tool bar at the top of the property window contains a number of buttons, used to customize the property display:


Switch between displaying properties alphabetically or by category.

Advanced / Basic

In ***Basic*** view mode, only a subset of the properties are shown; the most commonly used ones together with any values that have been changed from their default settings. Clicking ***Advanced*** will make all the selected item’s properties visible.

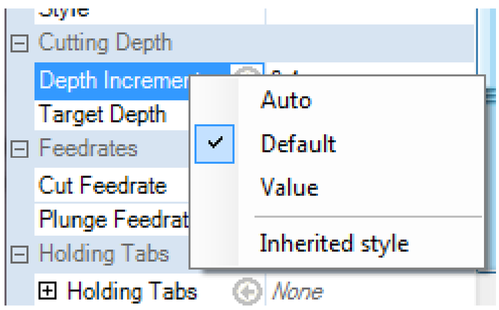

Displays a small window at the bottom of the property window, containing a brief description of the selected parameter.

For some objects, such as machining operations, a symbol may be shown to the right of the property name. These are:

Auto, indicating the value used will be automatically calculated.

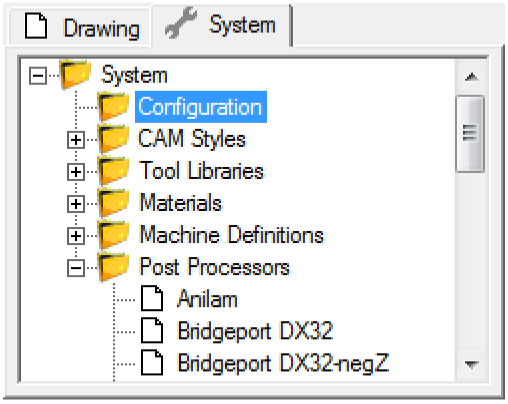
an explicit **Value** has been entered.

The current value is the **Default** (usually inherited from a machining operation’s style.



Clicking these icons will show a context menu where the type of value can be changed.

System Tab



The ***System Tab*** shows another tree view, this time displaying objects and settings available to all CamBam drawings, and contains the following sub folders:

Configuration: Equivalent to the ***Tools - Options*** menu and allows access to the system global ***configuration settings***.

CAM Styles : Folder containing ***machining style libraries***.

Tools : Folder containing ***libraries of cutting tools***.

Post Processor : ***Post Processor*** definitions, used to control how gcode is formatted from machining operations.

Materials/Machine Definitions : Both these sections are in an early stage of development and are intended for use in future releases.

Rotating and panning the drawing

Rotation

The 3D view is rotated by holding down the **ALT** key whilst dragging the left mouse button.

Other mouse and key combinations for rotations are available in the ***Rotation Mode*** option of the system configuration settings.

Panning

The drawing view is translated by dragging the center mouse button.

The cursor keys can also be used to translate the drawing view.

Zooming

Scrolling the mouse wheel will zoom in and out. Move the mouse cursor over the area you would like to zoom in on when scrolling.

The number pad + and - keys can also be used to zoom in and out.

Resetting

ALT + double click will reset the view orientation. If Left_Middle ***Rotation Mode*** is used, holding the middle mouse button whilst left double clicking will reset the view.

If the Left_Middle ***Rotation Mode*** is used, hold down the middle mouse button while double clicking the left button to reset the view.

The view can also be reset by selecting the ***View - Zoom To Fit*** menu option.

Selecting Objects

Objects can be selected by clicking them in the drawing view window, or by selecting them from the tree view on the left of the screen.
Clicking on empty space will clear any selections.

CTRL+**click** will select multiple objects. To deselect an object, **CTRL**+**click** it again.

CTRL+A will select all visible objects.

SHIFT+CTRL+A will select all objects in the active layer.

Multiple objects can be selected by dragging the left mouse button to form a selection rectangle. To be selected the entire object must be inside the rectangle.

Once selected, object properties can be viewed and modified in the property browser in the lower left.

Objects can be deleted by selecting them then pressing the **Delete** key.

Generating Toolpaths and GCode

CamBam uses CAM machining operations to generate toolpaths and machining instructions. CAM operations are sometimes referred to as ***MOPs*** (machining operations).

The following CAM operations are currently supported:

2.5D Profile - Creates toolpaths offset from selected geometry.

Pocketing - Fills a region bounded by geometry to create a pocket.

Engraving - Used to insert toolpaths that follow selected geometry.

Drilling - Creates drilling instructions from point list objects.

3D Surfacing - 3D Meshes can profiled using multi pass roughing or finishing profiles.
Front back and molds are also supported.

GCode - Gcode files can be imported as machining instructions.

Once the CAM operations are defined, GCode is generated by right clicking the ***Machining*** object in the tree view and selecting ***Create GCode File***.

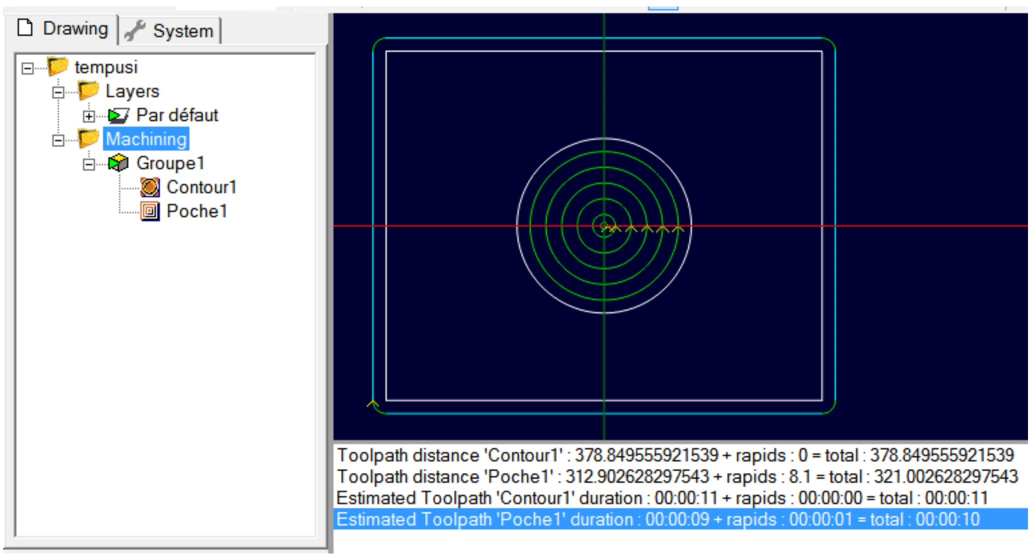
The ***Create GCode File*** option is also available by right clicking on each machining operation or each ***Part***. This will generate gcode for just the selected machining operation or part.

If the ***Diagnostic Level*** property in ***Tools/Options*** is set to >= 2, CamBam will display informations about toolpath length (in current drawing unit), as well as machining/rapids duration for each machining operation in the message window after the generation of toolpaths or Gcode.

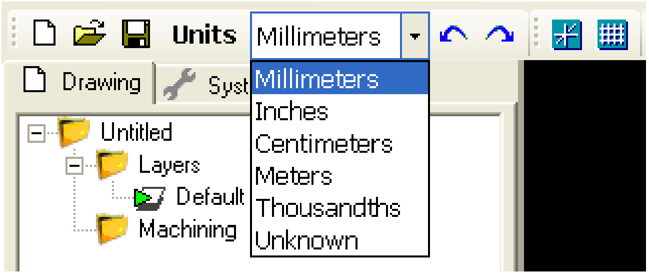
Some operations are not taken into account in the calculation ; this is the case of the drilling operation as well machining operations that has been added as plugins (Slotter, Thread milling, V-Engrave, etc …)

For machining operations using nesting, the displayed time is for only one instance of the nesting.

This is only a rough estimate ; for rapids the speed is fixed at about 600 mm/min, the G0 in Z are not taken into account.



Drawing Dimensions / Units



The current drawing's units can be changed from the drop down list on the toolbar.

After changing the drawing units, CamBam may prompt:

'Would you also like to change the default units for new drawings?'

If ***Yes*** is clicked, then the selected units will become the default drawing units.
If ***No*** is clicked, the current drawing's units will change but the default settings will remain unchanged.

Note: Changing the drawing units will not change the size of the drawing objects, only the units that the objects are measured in.
To scale objects, use the ***Transform - Resize*** command.

File Menu

File Open

CamBam can read the following drawing file types:

- CamBam native file format (*.cb)
- Autodesk DXF files - up to AutoCAD 2000 format (*.dxf)
- 3DStudio files (*.3ds)
- Stereo Lithographic 3D meshes (*.stl)
- STEP files (*.stp, *.step) New V1.0
- GCode files (.*tap*,.nc, etc)
- Gerber file (*.gbr)

Unrecognized file extensions are presumed to be GCode files.

Use the ***File - Open*** menu option to open the required file or drag and drop files from Windows Explorer onto the CamBam window.

When CamBam is installed, it will be associated with (*.cb) files, so these can be opened by double clicking them from Windows Explorer.

CamBam will also attempt to open any files passed to the application via the command line.

File New

Creates a new blank file.

The interface will be reset, the default settings stored in the general configuration will be used.

Hint: If a ***Drawing Template*** is defined in the system configuration settings, this file will be used as template for the new drawing. The drawing template can contain useful default settings such as ***Post Processor***, ***Fast Plunge Height*** and ***Stock***, as well as drawing objects and machining operations.

New from template

This will create a new drawing, based on an existing CamBam (.cb) file.

Template drawings are typically saved into the **templates** sub folder of the CamBam system folder. Use the ***Tools - Browse system folder*** menu to help find the templates location.

An example template drawing: **nameplate.cb**, is provided. This template allows the creation of a nameplate with raised lettering, commonly used for locomotive name plates. This template contains all the drawing objects and machining operations required. The default text can be quickly changed by double clicking the text object in the drawing view.

Changes made to a drawing based on a template will not affect the template file. To modify the template file, it will need to be opened from the template folder using ***File - Open***.

Save, Save As

Save your work using the menu ***File / Save*** or ***Save As.***

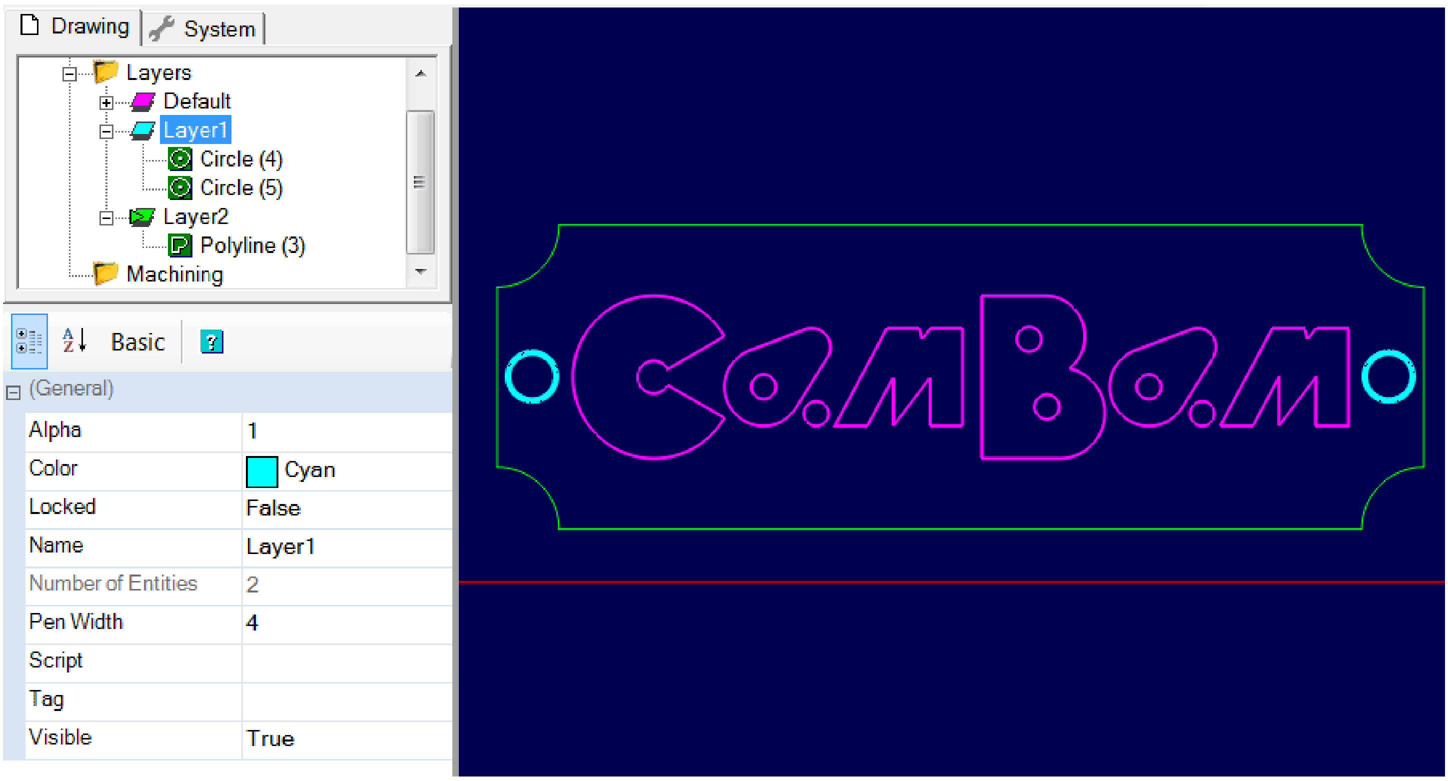
Depending on the value of the ***File Backups*** configuration setting, a number of backup files may be generated for each file. These backups are located in the same folder as the saved drawing and will have extensions such as .b1, .b2 etc. with .b1 being the most recent backup.

Print New V1.0

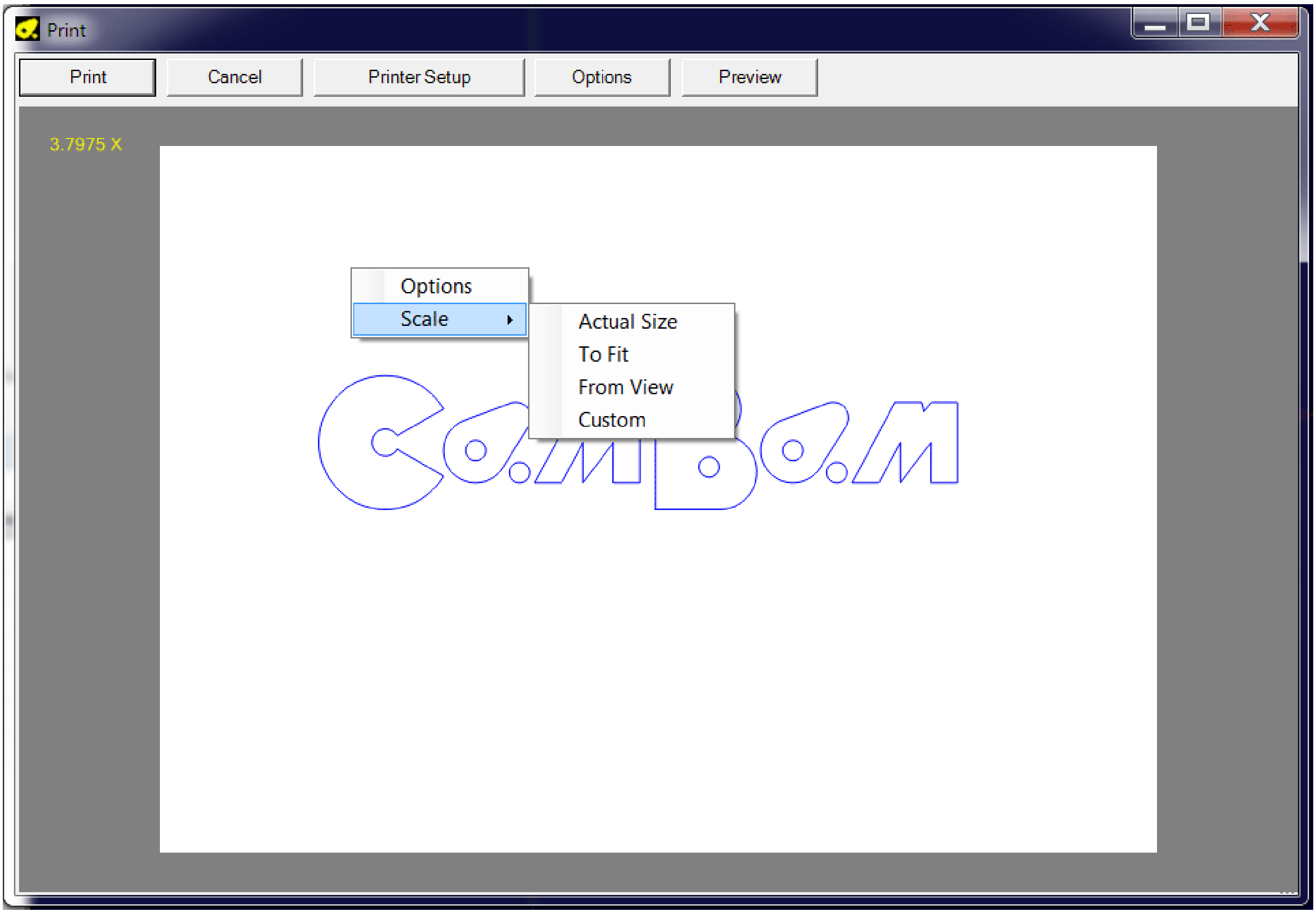
Use this menu to print the drawing area.

The colors and widths of the lines used for printing will be those defined in the ***Color*** and ***Pen Width*** properties of each layer. The background color will not be printed.

Be careful, with some printers, if the ***Pen Width*** of a layer is 0, the lines of this layer will not be printed.



The print dialog window allows you to position the drawing on the sheet, change its zoom factor with the mouse or with a pop-up menu, and access printer settings and other options.



Pan and zoom: The zoom and pan functions by mouse work in the preview window in the same way as in the drawing area so that the area of interest can be positioned in the page; **Alt** + double-click in this preview window centers the drawing on the sheet.

At the top left, the zoom factor is displayed in yellow.

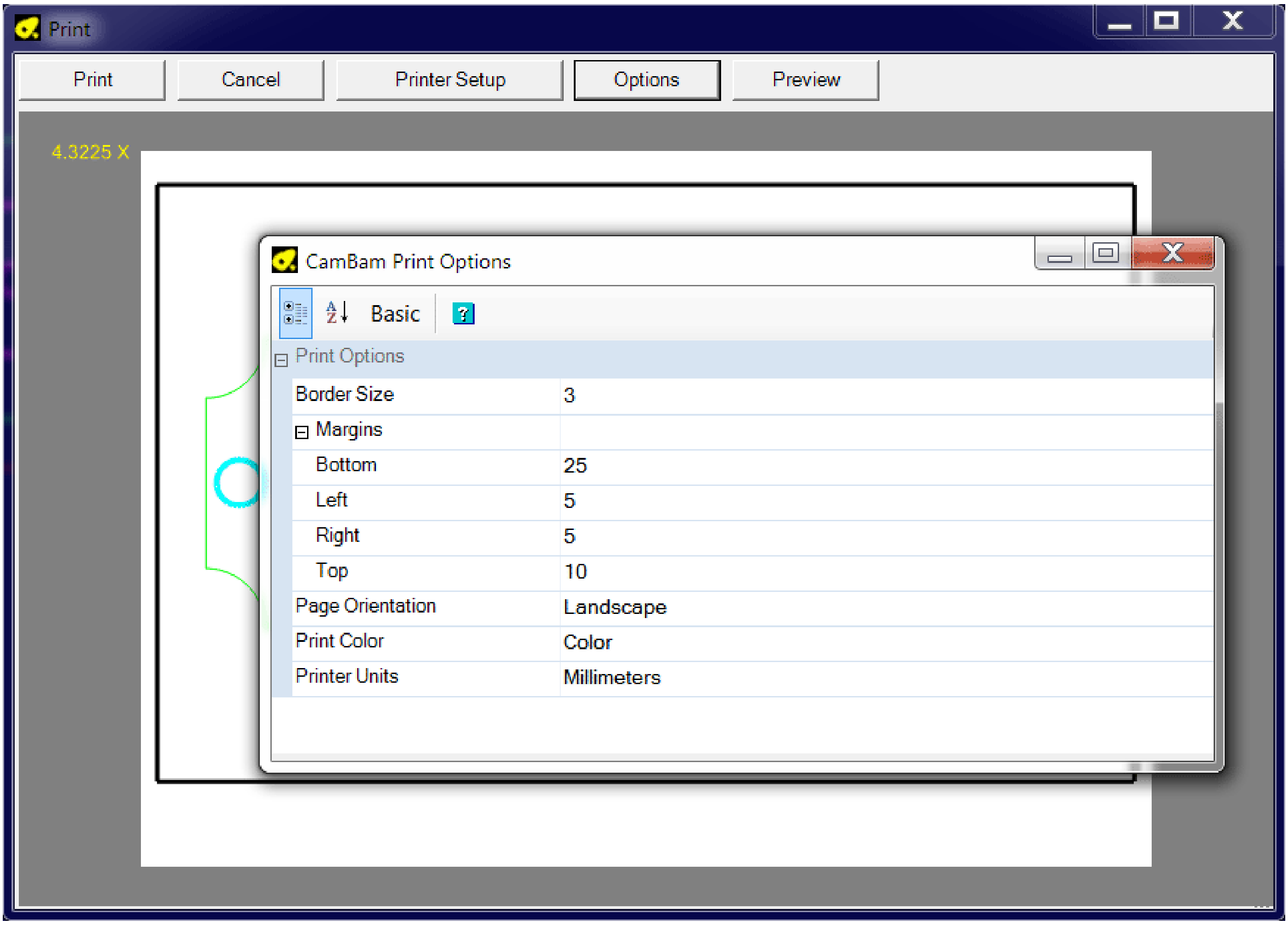
The zoom can be changed by turning the mouse wheel or using the ***Scale*** popup menu. You can use the ***Custom*** option to precisely set the desired zoom factor.

Printer setup button: Provides access to the selection panel and printer settings.

Print button: Provides access to the selection panel and printer settings, then starts printing via the OK button.

Preview button: Displays an exact preview of the print.

Options button: The context menu or the options button allows you to view the print options; these same options are also available in the general options of the program via the Tools / Options menu or via the Configuration folder of the system tab.



Print Color

: Color, Monochrome (gray scale), or Black & White.

Border Size

: If 0, the border display is disabled ; the values given in Margins define the position of the border.

Page Orientation

: portrait (vertical) or landscape (horizontal)

Export

To DXF: Exports the current file to DXF format (AC1009 = AutoCAD R11 / R12)

To STL: New V1.0 All surface objects on visible layers will be merged into one file and exported to STL format (mesh)

View Menu

Zoom

Three zoom options are available from the ***View*** menu:

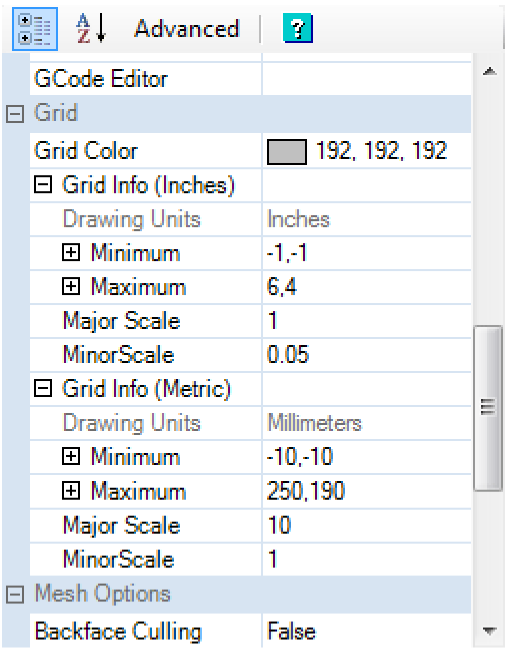
- ***Reset***
  reverts to a known position. (XY plane) and performs a ***Zoom To Fit***.
  Equivalent to **ALT** + double click.
  If ***Rotation Mode*** *Left_Middle* is active, the same operation can be done by a double left click while holding the middle button pressed.
  If ***Rotation Mode*** *Left_Right* is active, a double left click while holding the right button pressed.
- ***Zoom To Fit***
  Zoom so that all objects of all visible layers are visible, without changing the view orientation.
  Objects in hidden layers are not taken into account to calculate the zoom factor.
- ***Zoom Actual size***
  Zooms so that the drawing objects are shown approximately true sized (allowing for display size variations).

Displaying the grid and axis

The grid and axis display can be enabled and disabled using the following toolbar icons or by selecting ***Show grid*** and ***Show axis*** options from the ***View*** menu.

The appearance of the grid, including color, major and minor units, size and position, can be changed in the grid ***system configuration settings..***

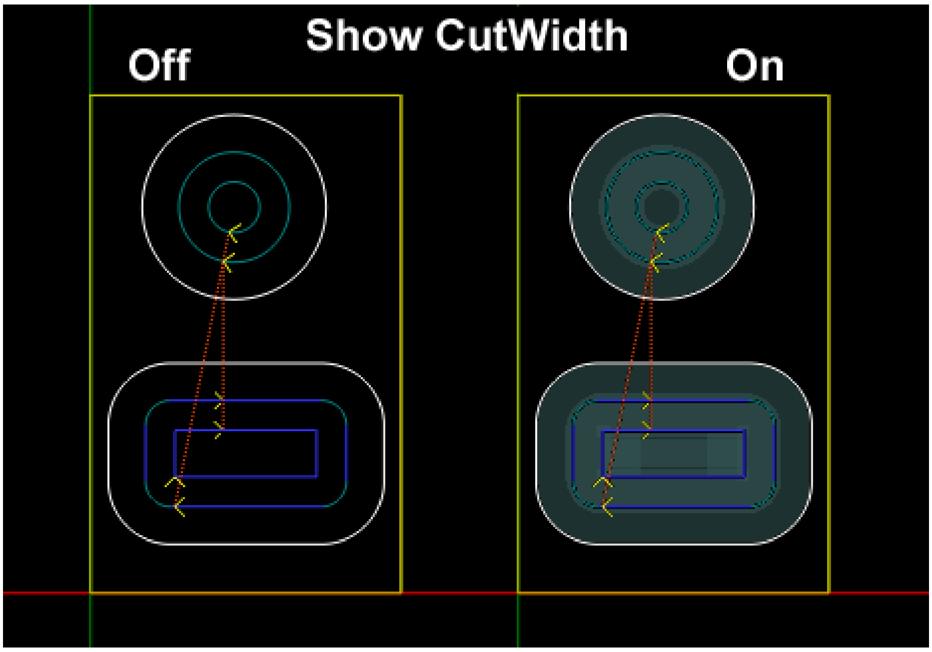
There are two sets of grid settings: One for inch drawing units and the other for metric.



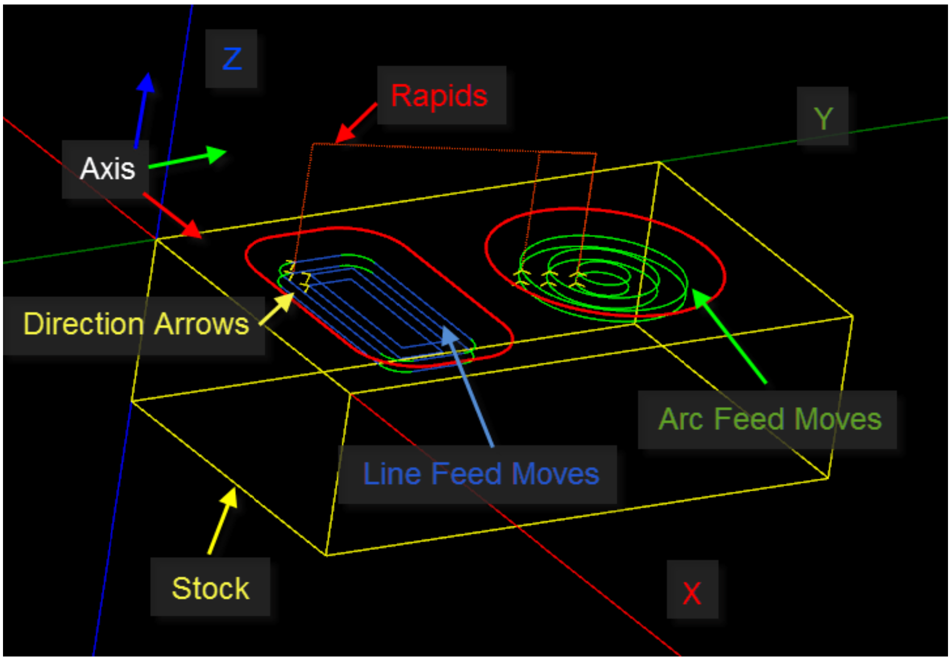
Display Setup

The following options enable or disable the display of graphical aids.

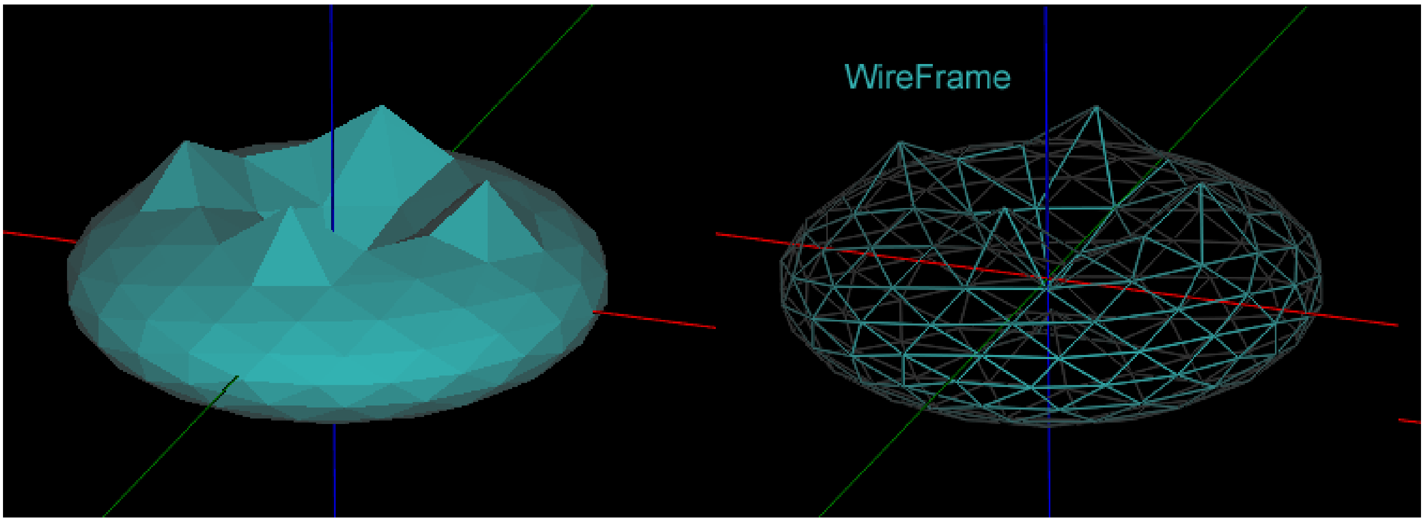
- ***Show toolpaths*** - Enable / disable the display of lines representing the toolpaths.
- ***Show cut widths*** - Enable / disable the display of a shaded area depicting the width of cuts along the toolpaths based on the specified tool diameters.



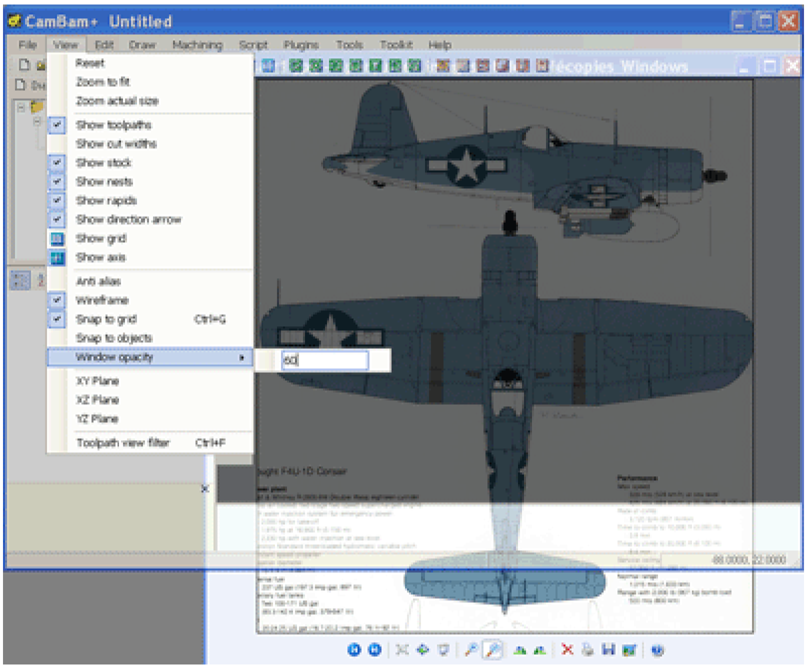
- ***Show stock*** - Enables / disables the display of the 3D representation of the block of material to be machined.
- ***Show nests*** - Enables / disables the display of arrays of machining operations, defined in the Part Nest properties.
- ***Show rapids*** - Enable / disable the display of dotted lines representing rapid moves (G0).
- ***Show direction arrows*** - Enable / disable the display of arrows indicating the direction of travel of the tool.
- ***Show grid*** - Enable / disable the display of the grid.
- ***Show axis*** - Enable / disable the display of XYZ axis lines of the 3D view.



- ***Anti-alias*** - Enable / disable anti-aliasing.
- ***Wireframe*** - Toggle the display of 3D objects between shaded surfaces or wireframe mode.

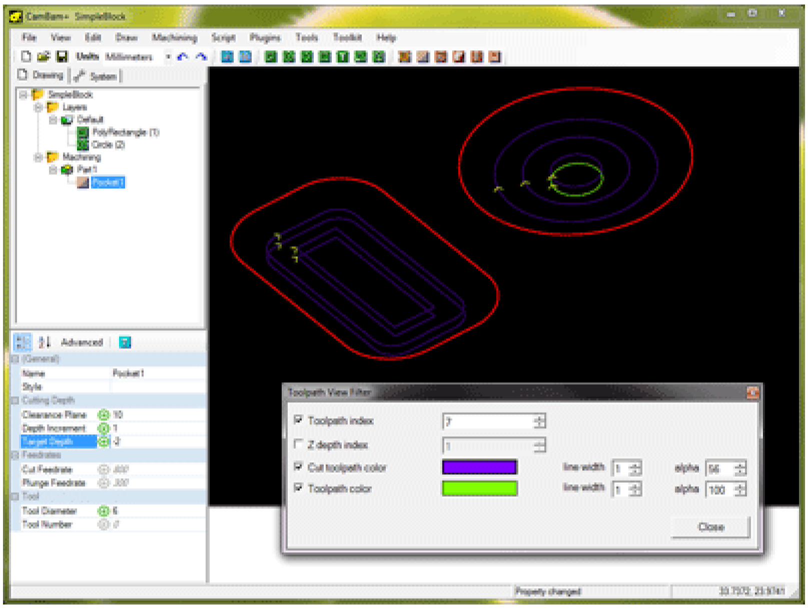


- ***Snap to grid*** - Enable / disable snap to grid.
- ***Snap to object*** - Enable / disable snapping to other drawing objects.
- ***Windows opacity*** - A value between 0 and 100% (opaque) which allows tracing over reference drawings in windows behind the CamBam drawing.



XY / XZ / YZ Plane - Switches the view seen from above (XY - default), Front (XZ) or side (YZ). For now only the XY plane can be used to draw with the mouse.

- ***Toolpath view filter*** - Used to view step by step the tool path according to their order of execution or level in Z.



Toolpath index: if checked, you can view the tool path in order of their execution by changing the numerical value on the right.

In this example, the 7th toolpath is highlighted in yellow, previously cut toolpaths are shown in purple and uncut toolpaths are not visible.

Z depth index: if checked, you can view the tool path in order of Z level by changing the numerical value on the right. All tool paths on the same Z level will be displayed simultaneously.

If both Toolpath index and Z depth index are ticked, the toolpath will be filtered by index down to the maximum Z depth specified.

Cut toolpath color display or hide toolpaths cut previous to the current toolpath.

Toolpath color when checked, the current toolpath will be highlighted in the selected color on the right; If unchecked, the current toolpath will be displayed using the standard arc and line move colors.

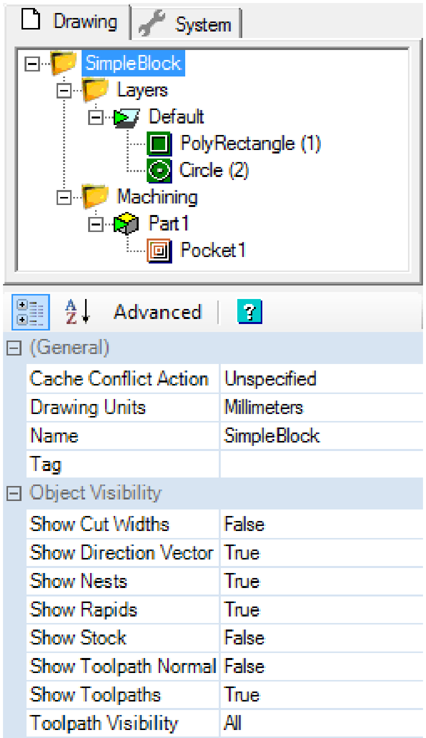
Click on the colored rectangles to change the display color of the toolpaths.

You can also choose the line width by changing the width of ***line width*** and transparency by the ***alpha*** value.

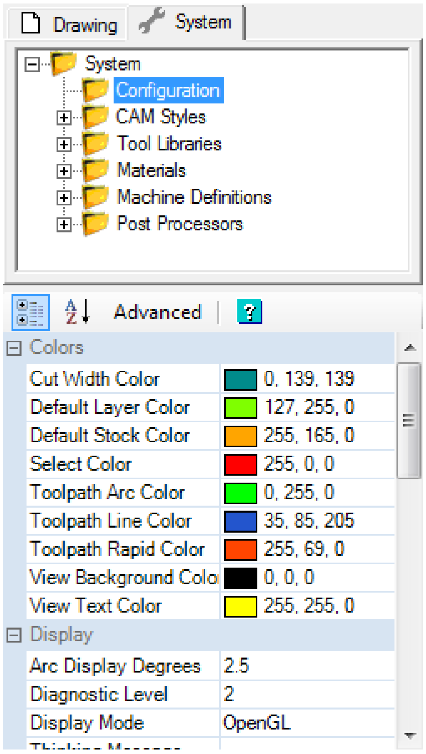
The ***Toolpath View Filter*** window can be kept open while manipulating the drawing, such zooming and panning the display.

Depending on the drawing's ***Toolpath Visibility*** setting (*All* or *Selected Only*), the filter will show the path of all machining operations or only those machining operations or Parts selected in the drawing tree.

Display settings are available in the property grid by selecting the top level (Drawing) object of the drawing tree.



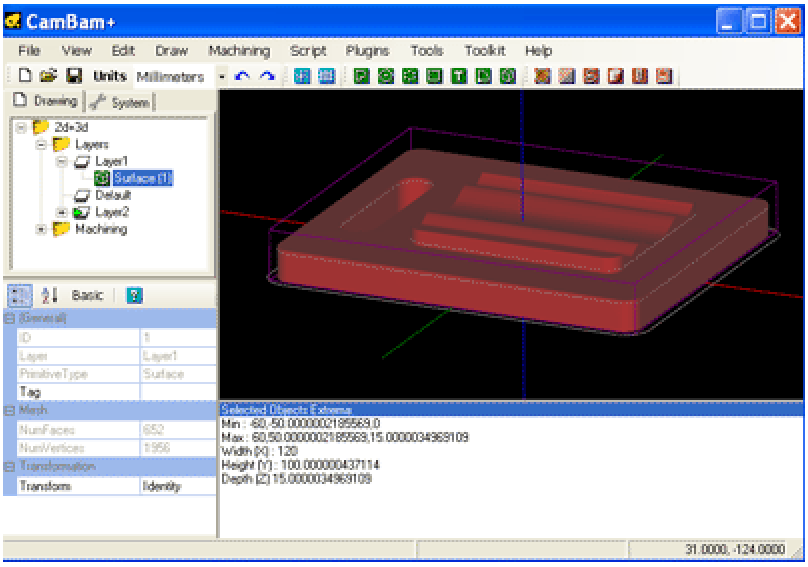
The display colors can be changed in the ***system configuration*** Colors group.



Tools Menu

CamBam has a number of utility functions grouped in the Tools menu.

- ***Save settings***
  Saves system configuration settings and any modified system libraries or post processors.
- ***Save settings on exit***
  If this menu item is checked, configuration and other system changes will be saved automatically when CamBam is closed.
- ***Browse system folder***
  Opens the folder containing CamBam system files (libraries, post processors, samples, scripts etc). The location of this folder can be specified in the ***System Path*** configuration setting.
- ***Options***
  Opens a window where system ***configuration settings*** can be maintained.
- ***Check for new version***
  Determines whether there are any newer CamBam updates available from the CamBam website.
- ***Clear messages***
  Clears messages from the information window below the drawing window.
- ***Get object extremas***
  Shows the extrema points and dimensions of the selected drawing objects.

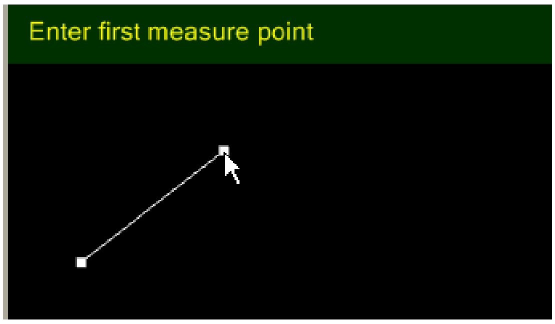


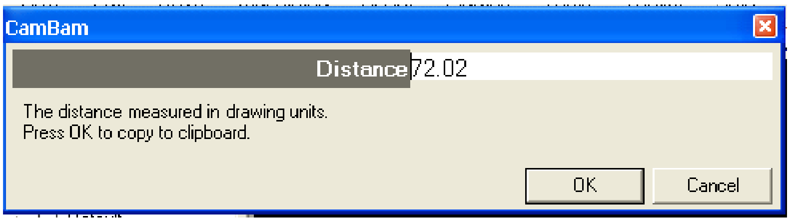
Min: minimum coordinates of the object in X, Y and Z are separated by a comma.
Example: X=-60, Y=-50.000…, Z=0

Max: maximum coordinates of the object in X, Y and Z are separated by a comma.
Example: X=60, Y=50.000…, Z=15.000…

Width, Height, Depth Maximum dimensions of the object in drawing units.

- ***Measure*** (**M** shortcut key) Allows you to draw a line to make a measurement between two points. The measurement result is displayed in a new window.

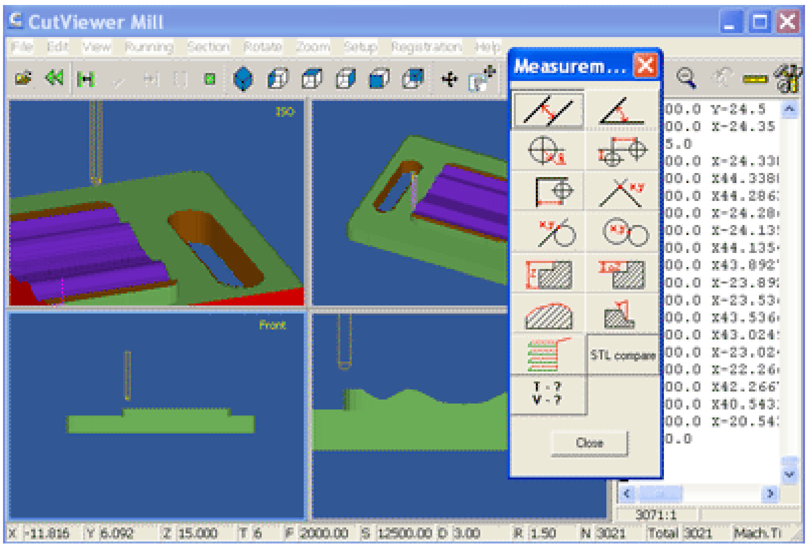




- ***Reload post processors***
  Reloads all the post processor definitions from disk. This may be needed if a post processor has been modified from another instance of CamBam.

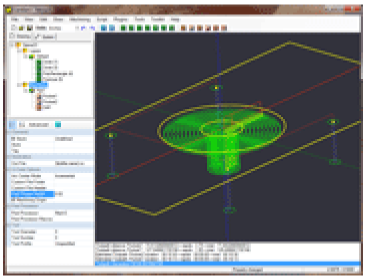
- ***Simulate with CutViewer***

  Starts the third-party software *CutViewer Mill*, to provide a 3D machining simulation from the Gcode file produced. To avoid having to provide *CutViewer* parameters manually, you must use a post processor designed to work with this software. (E.g. Mach3-CV for milling, Mach3-Turn-CV for turning). You must also define a ***stock object*** in the machining or Part objects.



Note: since V1.0, the plugin for cutviewer is not installed by default, you can download it on the plugins website.

Simple Example (Stepper Mount)



This sample project will demonstrate the general process involved in going from a new drawing to final gcode. The object is a mounting plate for a Nema 23 stepper motor and will include CAD, pockets and drilling machine operations.

The basic work flow for generating CAM files in CamBam is to first draw or load in drawing objects, then insert machining operations based on these geometric objects and finally generate gcode files.

Step 1 - Create and set up a new drawing.

Start with a empty drawing, use ***File - New*** or the new file icon from the toolbar.

In this example we are going to work in Inches, so the first step is to select the drawing units from the toolbar.

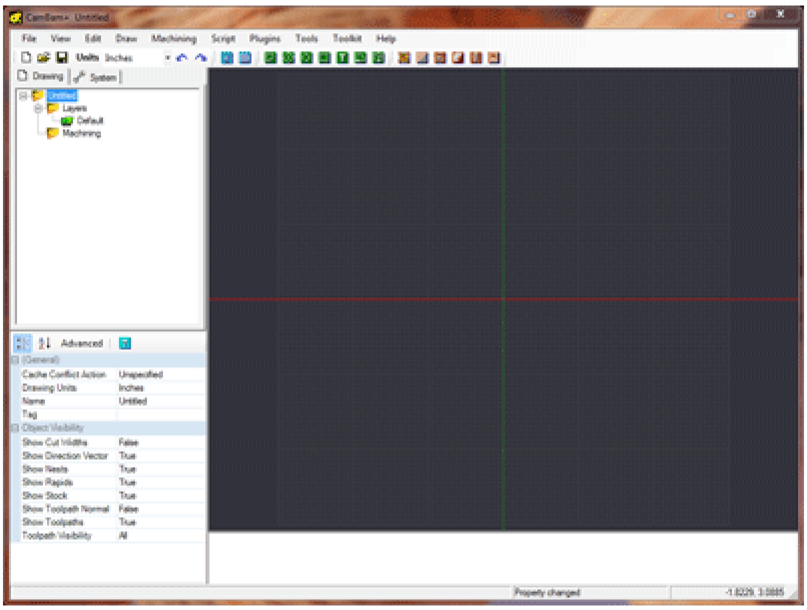
This will prompt :

"Would you like to change the default units to Inches?"

This question refers to the global drawing units property that is set in the ***system configuration settings***.
The global units option is used to set the drawing units for new drawings.
Select *Yes* to update the global setting as well as the current drawing.
Selecting *No* will change the current drawing to Inches but leave the current global units setting unchanged.

Show the drawing grid and axis by selecting the Show Axis and Show Grid buttons from the toolbar.

To zoom the image so that it fills the screen and makes it central, select ***View - Zoom Actual Size*** from the main menu



Step 2 - Drawing circles

We will draw a circle to define the raised circular area around the stepper shaft. This circle will later be used to form a circular pocket. For a Nema 23 stepper motor, this area is around 1.5” (38.1mm) diameter. We will also draw a circle to denote the shaft clearance hole with diameter 0.5” (12.7mm).

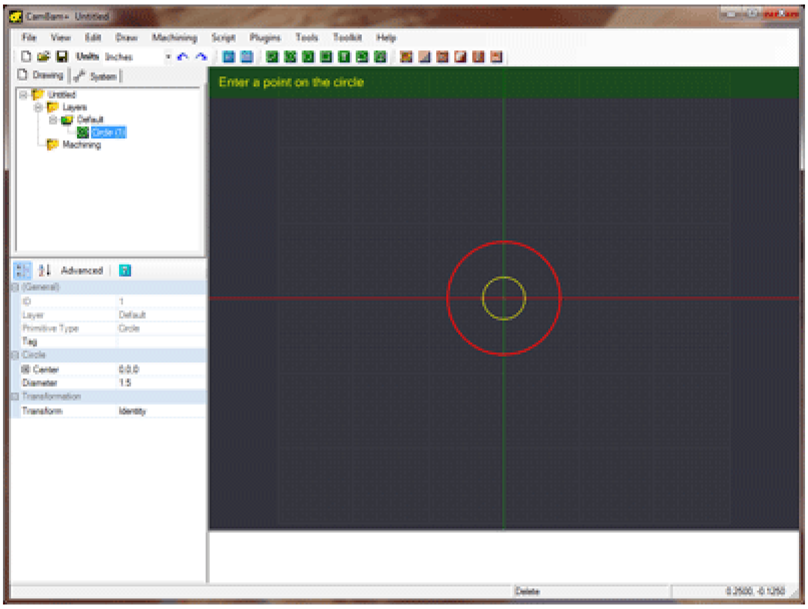
Select the circle drawing tool button from the tool bar. A prompt will be displayed at the top of the drawing window to guide the current drawing operation.

Select the center point for the circle on the drawing origin (0,0). If snap to grid is not turned on, right click the drawing to display the drawing context menu, then click ***View - Snap to Grid***.

Next, select another point on the circle. Chose the point (0.75,0). The point coordinates can be seen on the bottom right of the lower status bar. If the current grid settings will not allow selecting an exact point, chose a point nearby then the circle diameter can be modified later.

A circle drawing object will now appear in the drawing tree on the left. The properties for this circle will be displayed in the object properties window on the lower left. The ***Center*** property should read 0,0,0 and the ***Diameter*** should read 1.5.
These values can be modified in the object properties window if required.

Insert a second circle with center the origin and make the diameter 0.5.



Step 3 - Drawing a rectangle and making it central

The rectangular body of a Nema 23 stepper is about 2.36” (60mm). We will make our mounting plate 5” (127mm) wide and 2.375” (60.3mm) tall.

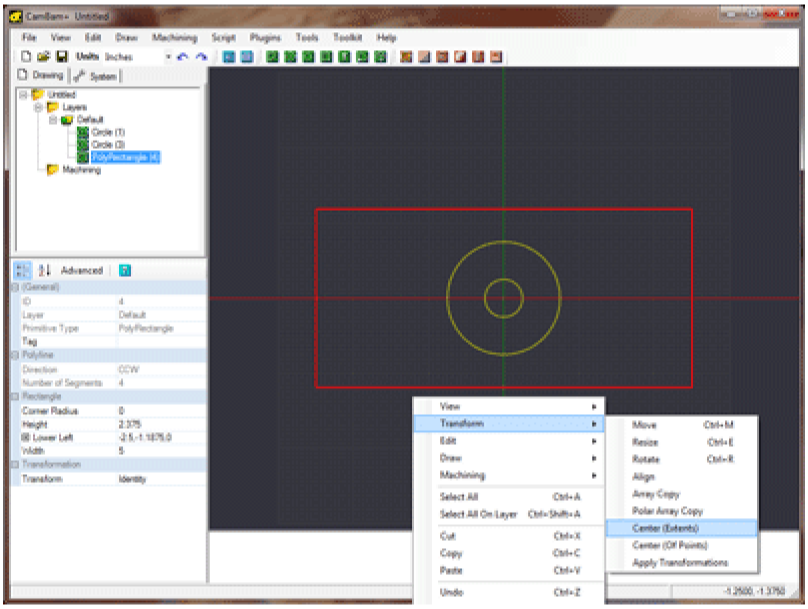
Select the rectangle drawing tool button from the tool bar. Once again, a prompt will be displayed at the top of the drawing window to guide the current drawing operation.

To simplify drawing, we will draw the rectangle with the lower left corner on the origin then center it. Click the origin for the lower left point then the point (5,2.375). Again, if the exact coordinates can not be selected then don't worry as these can be edited under the rectangle object's properties.

> ***Hint***: To pan the drawing view, click and drag the drawing with the center mouse button.
> The arrow keyboard keys can also be used to pan the display.
> To zoom the display, scroll with the mouse wheel.

A rectangle object should appear in the drawing tree and it's properties will be displayed in the object property window.
Change the ***Height***, ***Width*** and ***Lower Left*** point if required.

To center the rectangle, first make sure it is selected (it will be highlighted in bold red), then right click the drawing window and select ***Edit - Transform - Center (Extents)*** from the context window.



Step 4 - Inserting 4 points for mounting hole positions

The Nema 23 stepper motor has 4 bolt holes arranged in a square 1.856" (47.14mm) apart. We will be adding a drilling machining operation later to generate these holes so to prepare for that we need to insert 4 center points at the hole centers.

There are a number of ways to achieve this but here are a couple of options.

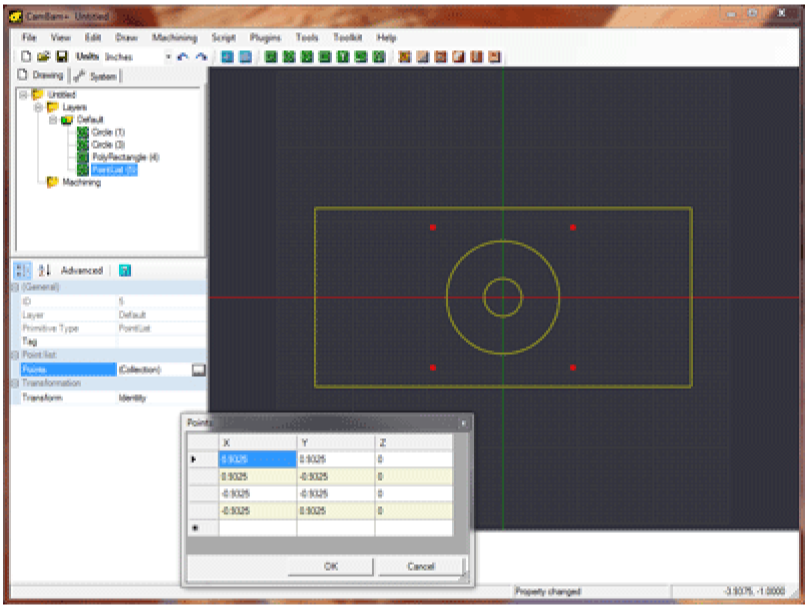
Select the point list tool button from the tool bar. Click 4 points around the origin with the following coordinates :

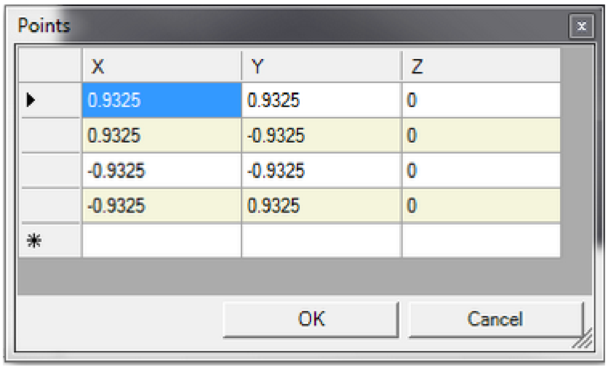
(0.928, 0.928), (0.928, -0.928), (-0.928, -0.928), (-0.928, 0.928)

Press the `Enter` key or click the middle mouse button to finish drawing the point list.

A PointList object will have been created in the drawing tree and it's properties will be visible in the object property window.
There is a property called ***Points*** which is followed by the word (Collection). The point coordinates can be

modified by clicking on the box that says (Collection), then clicking the ellipsis [...] button that appears after it. This will open the points editing dialog. The X and Y values can then be set to the values given in the list above.





An alternative way to achieve this is to first draw another rectangle with the lower left point on the origin then change the rectangles height and width properties to both be 1.856. Select the rectangle and center it (Right click, ***Edit - Transform - Center (Extents)***). Now draw a point list as before. This time the drawing points should snap to the rectangles corner points. It may be easier to turn off ***Snap to grid*** and make sure ***Snap to objects*** is turned on. Both these are set in the right click, ***View*** menu. Once the points are drawn, the guide rectangle can be selected then deleted.

The geometry for the stepper plate is now complete, so now would be a good time to make sure the drawing is saved.

Step 5 - Inserting a pocket and viewing the toolpath

Select the large circle drawing object then click the pocket machining operation button from the toolbar. A new pocket object will be created and displayed under the ***Machining*** folder in the drawing tree. The object property window will display the pocket's properties ready for editing.

CamBam will initially show only a limited number of common properties for the selected machining operation. Clicking the ***Advanced*** button at the top of the property grid will show the complete list of available properties.

For this example we are going to use a 0.125" (3.175mm) carbide cutter and cut at a feed rate of 7ipm (~180mm/min). The plunge feedrate will be 2ipm (~50mm/min) and a maximum of 0.02" (0.5mm) depth of material will be removed during any pass.

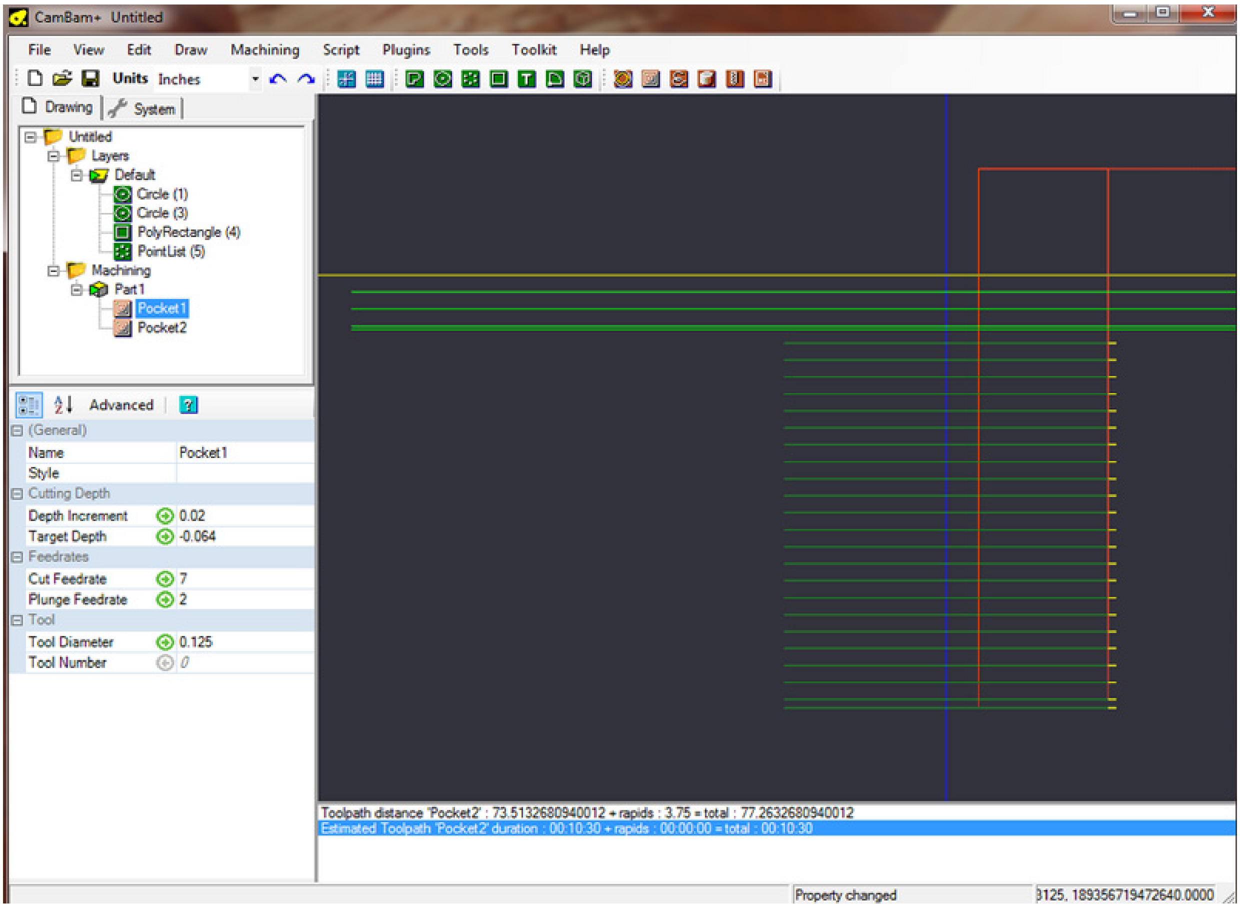
Change the pocket machine operation's properties to the following:

Tool Diameter	0.125
Stock Surface	0
Depth Increment	0.02
Target Depth	-0.064
Cut Feedrate	7
Plunge Feedrate	2
Clearance Plane	0.1

Note: The ***Target Depth*** value sets the final depth of the pocket and is the Z coordinate (relative to the origin) of the bottom of the finished pocket. CamBam assumes positive Z values are up, away from the stock and negative Z values are moving down into the stock or work table. If you try to enter a ***Target Depth*** above the stock surface the program will report a warning in the message window and set the target depth to the same as the stock surface.

To generate the resulting toolpath for the pocket, right click the drawing to bring up the drawing context menu, then select ***Machining - Generate Toolpaths***. This will display green circles indicating the path of the center point of the cutting tool.
Arc toolpaths are displayed in green, straight lines in blue.

To view the toolpath side on, select ***View - XZ Plane***. This shows 4 cutting levels. The X axis indicated by the red line is the level of the stock surface. The distance between each level is set in ***Depth Increment***. The



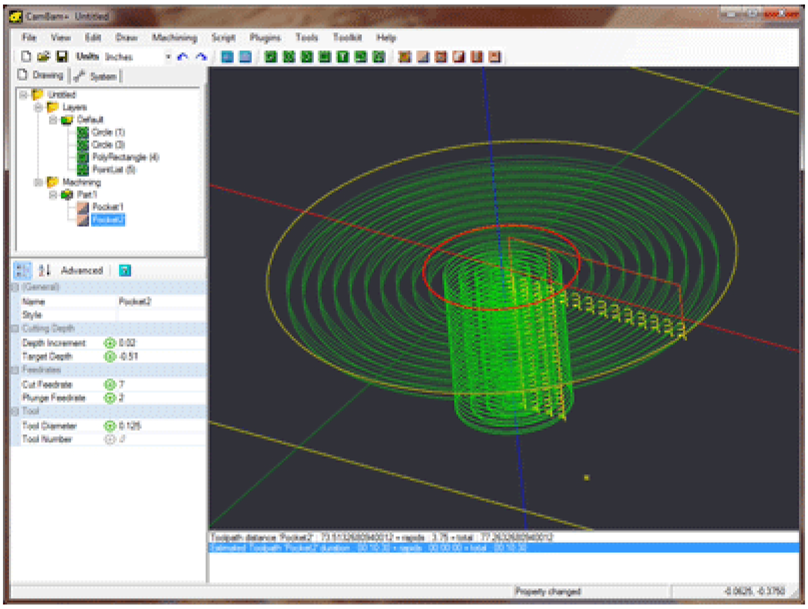
bottom toolpath will be a Z coordinate given in ***Target Depth***.

To rotate the 3D drawing view, hold the **ALT** key then click and drag on the drawing. To reset the view, hold the **ALT** key then double click the drawing. Other rotation modes can be set in the ***Rotation Mode*** setting of the system configuration.

Now we will insert a second pocket to cut the shaft clearance hole. Select the inner circle and insert a second profile machine operation.

This time use the following properties:

Tool Diameter	0.125
Stock Surface	-0.064
Depth Increment	0.02
Target Depth	-0.51
Cut Feedrate	7
Plunge Feedrate	2
Clearance Plane	0.1



Step 6 - Insert drilling machine operations

Select the point list object that defines the bolt holes, then click the drilling operation button from the toolbar. If you have trouble selecting the points from the drawing, you can also select them from the drawing tree view.

CamBam supports 3 different drilling methods:

Canned Cycles, which use gcode canned cycles G81, G82, G83 at each drilling point.

Spiral Milling, defines a spiral toolpath that cuts evenly through stock using a milling cutter and can cut a hole larger than the cutter diameter at arbitrary sizes.

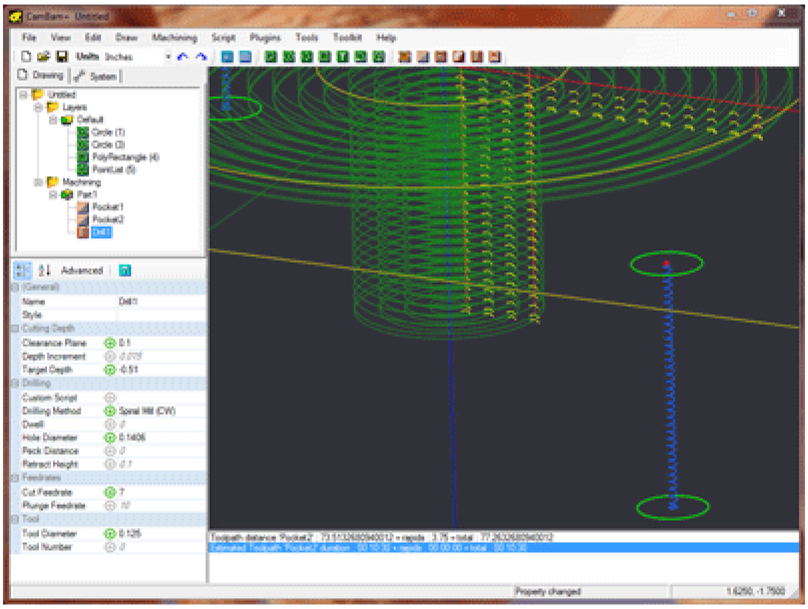
Custom Scripts, which allow snippets of gcode to be inserted at each drill point.

This example will drill 4 x 0.1406" (~3.6mm) that will then be tapped to accept a machine screw. The 0.125" cutter should still be in the CNC machine following the pocket so we will use spiral mill drill option to the correct hole diameter.

Change the drilling machine operation's properties to the following:

Tool Diameter	0.125
Stock Surface	0
Target Depth	-0.51
Cut Feedrate	7
Plunge Feedrate	4
Clearance Plane	0.1
Drillling Method	SpiralMill CW
Hole Diameter	0.1406

Generate the toolpaths again to view the resulting spiral paths.



Step 7 - Creating GCode

Before producing the gcode output, now would be a good time to save your drawing.

Then visually inspect the toolpaths and double check the parameters of each machining operations.

To create a gcode file (or post), right click to get the drawing menu then select ***Machining - Produce GCode***.

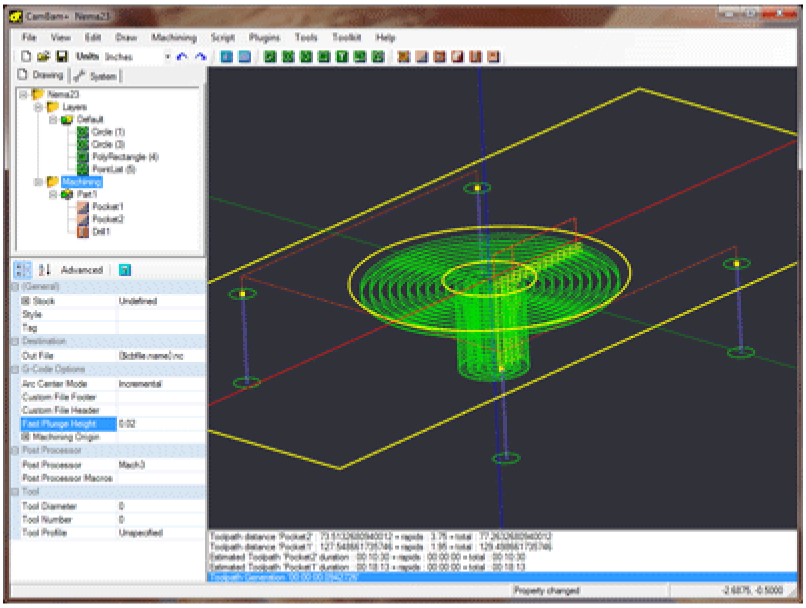
CamBam will then prompt for the location of the gcode file to produce. If the drawing file has been saved, the default file will be in the same folder as the drawing file with a .nc extension.

If the destination file already exists you will next be asked to confirm whether to overwrite it.

To control how the gcode file is produced, select the machining folder from the drawing tree. The machining properties for this drawing will then be displayed in the object properties window.

For NIST RS274 compatible interpreters such as LinuxCNC, Mach3 and USBCNC the default machining properties should be fine.

One setting to check is the ***Arc Center Mode*** property. This setting controls how the I and J (arc center) coordinates are defined for arc gcode (G02 and G03) and be Absolute or Incremental. This needs to be the same method as used by the interpreter and will result in crazy looking arcs or errors when opened in the interpreter.



Keyboard Shortcuts

Ctrl+A	Select all objects
Shift+Ctrl+A	Select all objects on the active layer
Ctrl+B	Edit - Break at intersections
Ctrl+C	Copy selected object to the clipboard
Ctrl+E	Resize selected drawing objects
Ctrl+F	Open the toolpath filter window
Ctrl+G	Toggle snap to grid mode
Ctrl+I	Convert to Region
Ctrl+J	Join selected drawing objects
Ctrl+M	Move selected drawing objects
Ctrl+O	Open a file
Ctrl+P	Convert selected objects to polylines
Ctrl+R	Rotate selected drawing objects
Ctrl+S	Save the current file
Ctrl+T	Regenerate all toolpaths
Ctrl+U	Union selected drawing objects
Ctrl+V	Paste from the clipboard
Shift+Ctrl+V	Copy the format from the clipboard object to the selected
Ctrl+W	Produce gcode file
Ctrl+X	Cut object and place on clipboard
Ctrl+Y	Redo the last undone operation
Ctrl+Z	Undo the last operation
Space	Hide/Unhide selected Layers, Parts or MOPs
Ctrl+Space (New V1.0)	Lock/Unlock selected Layers
A	Draw an arc
C	Draw a circle
D	Draw a point list (dots)
M	Measure
P	Draw a polyline
R	Draw a rectangle
T	Insert text
Cursor Up/Down/Left/Right	Pan the drawing view
Shift+Cursor Up/Down/Left/Right	Move the selected object of a minor unit of the grid

Shift+Ctrl+Cursor	Move the selected object of a major unit of the grid
Page Up or Num Pad -	Zoom out
Page Down or Num Pad +	Zoom in
Home	Reset view
Del	Delete selected Objects/Layers/Mops
F1	Help

Machining Operations

A machining operation is an object that will generate toolpaths and machining instructions used by a CNC machine. Typically these operations will be based on one or more drawing objects.

CamBam provides the following machining operation types:

Profile

This is a versatile 2D machining operation, typically used to cut around the inside or outside of a shape. Profiles support ***holding tabs*** (sometime called bridges), which will hold parts in place once the full depth of the stock is cut through.
Lead in and ***Lead out*** moves can be added to reduce the stresses on parts and tooling and the ***Side Profile*** property can be used to give 3D contours to the profile cut.

Pocket

Pockets are used to clear out stock within selected shape outlines. Pockets will detect selected *islands*, or closed shapes within other shapes to form more complex shapes. This can be used to create raised lettering effects such as on a name plate.

Drill

The drill operation is typically used to drill holes at selected point lists or circle centers using drill tooling. End mills can also be used to spiral mill holes larger than the tool diameter and complicated operations can be achieved using custom drilling scripts.

Engrave

Engraving operations are used to machine over selected lines. As well as 2D geometry in the XY plane, they can also be used to follow 3D lines with varying Z heights such as in bitmap heightmaps.

3D Profile

This operation is used to machine 3D shapes from surface mesh objects such as those imported from STL and 3DS files. V1.0 also support STEP files.
A number of different 3D methods are supported including waterline and scan-line methods with roughing and finishing options. Front and back face operations are provided as well as creating inverted 3D machining operations for molds.

Lathe

The turning operation is a new, experimental featured introduced with CamBam version 0.9.8. This can create roughing and finishing operations based on 2D profile lines drawn in the XY plane, but machined in the conventional lathe XZ plane.

NC File

The NC File operation is different to the other operations in that it is not based upon drawing objects, but can be used to include gcode from an external text file. This operation can also be used to display a toolpath or back plot gcode files. The contents of external gcode files will be included in the gcode output of the current CamBam drawing.

Inserting a machining operation

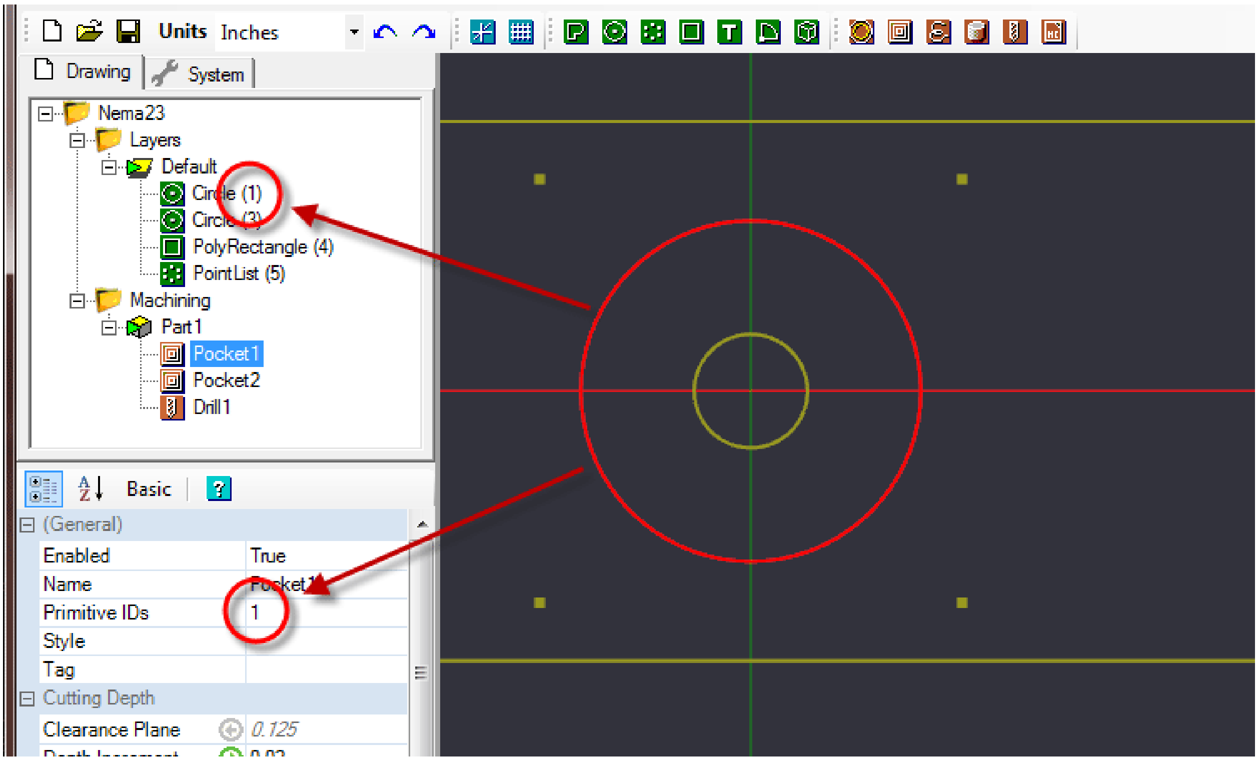
To add a machining operation, select one or more drawing objects (2D or 3D depending on the type of operation to be inserted), then click on the toolbar icon that corresponds to the desired operation, or choose from the ***Machining*** menu.

Machining operations can also be created by copying and pasting existing ones. Copies can be made from machining operations in the current file or from another file loaded in a second running instance of CamBam.

The new operation will appear in the drawing tree, within the currently active ***Part***, and its properties will be available to modify in the property window below the tree view.

When machining operations are selected in the drawing tree, all visible drawing objects associated with the operation will be highlighted in the drawing view.

The list of unique IDs identifying the drawing objects associated with the operation can be found in the ***Primitive IDs*** property.



Note: The ***Primitive IDs*** property is only displayed in the ***Advanced*** property view.

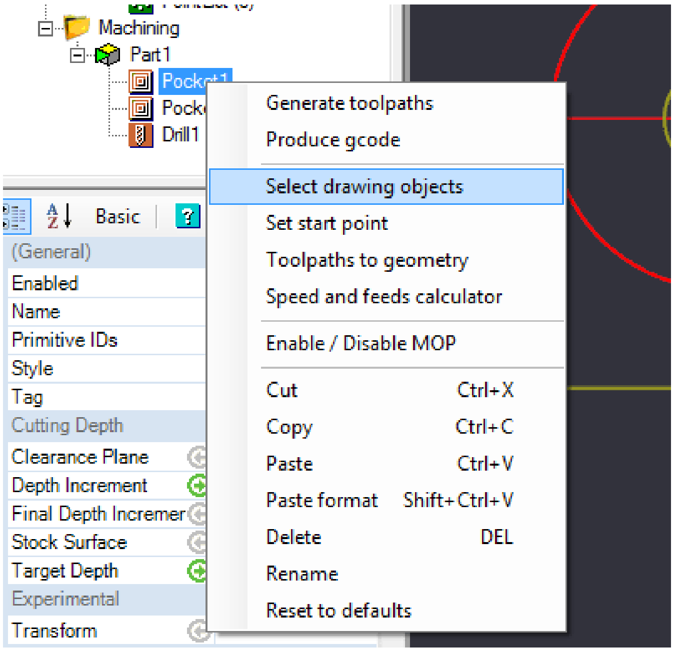
Changing machining operation source objects.

It may be necessary to change the drawing objects associated with a machining operation if:

- Additional objects need to be added to the operation.
- A drawing object has been modified and its ID no longer matches that currently linked to the machining operation (for example, after converting a rectangle to a polyline for editing, its ID number will be changed).
- A machining operation has been created by copying an existing operation and new drawing objects need to be assigned.

To change the assignment of source objects of a machining operation:

Click the right mouse button on the operation concerned to display the context menu for that operation, then use the ***Select Drawing Objects*** command.



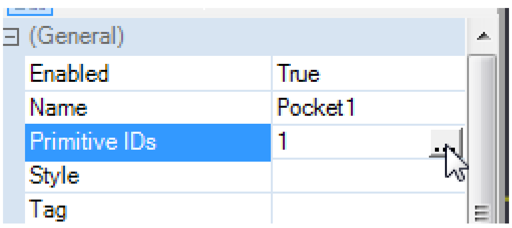
The drawing window displays the objects already assigned to the operation in red. All the ***object selection methods*** can be used to alter the current selection. Holding the **`Ctrl`** key and left clicking objects will add and remove objects from the selection.

Clicking an empty area of the drawing will deselect all.

When finished, click the *middle mouse button* or press the **`Enter`** key to apply the selection.

Press the **`Escape`** key to abort the selection and revert to the original.

Pressing the `[...]` button to the right of the ***Primitive IDs*** property will also invoke object selection function.



The ***Primitive IDs*** property can also be edited directly in the property grid, entering the ID values separated by commas.

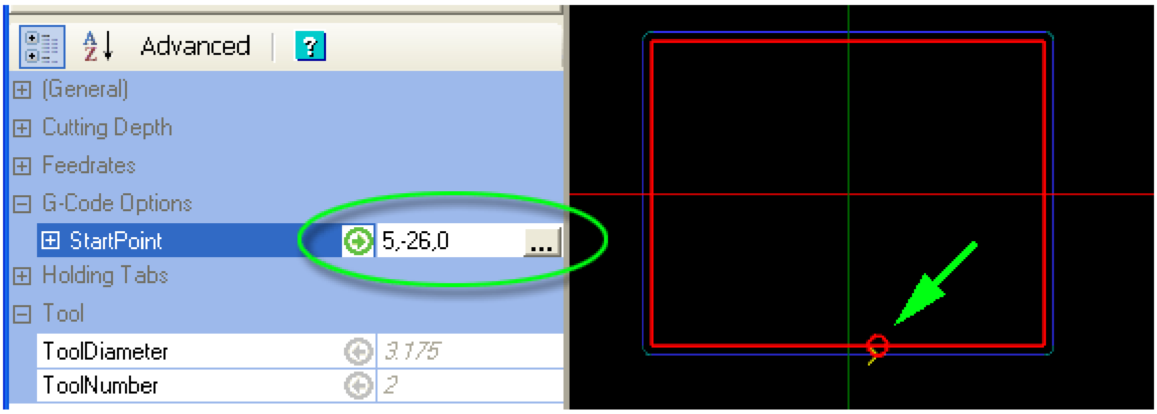
Managing machining operations

Right clicking a machining operation invokes a context menu with the following options.

Enable / Disable MOP: Activates or deactivates a machining operation. When disabled, the operation will appear greyed out, its toolpaths will be hidden and it will not be taken into account when creating the Gcode.

Set start point: Sets the starting point of a machining operation by clicking on the drawing at the desired start point. The operation will start at the closest point possible to the select start point.

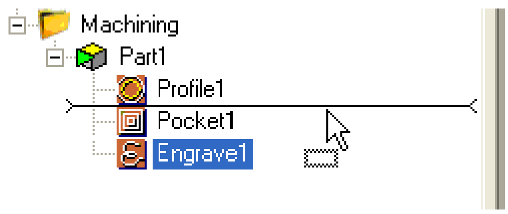
This starting point will be indicated by a red circle which can then be moved by dragging with the mouse. The coordinates of the chosen starting point will also be displayed, and can be edited directly, in the ***Start Point*** property of the machining operation.



Selecting the ... button to the right of the ***Start Point*** property will also invoke the interactive point selection function.

Cut / Copy / Paste: Uses standard clipboard routines to manage the machining operations. These functions allow copies of the selected machining operations to be made in the current drawing, or in a different drawing loaded in another running instance of CamBam.

Machining operations may be reordered or moved between Parts by dragging them within the drawing tree. A horizontal bar indicates where the operation will be inserted.



Paste format: This function copies most of the properties of a machining operation that has been copied to the clipboard using the Copy command, into the selected target machining operation. The target's name and source drawing objects are left intact.

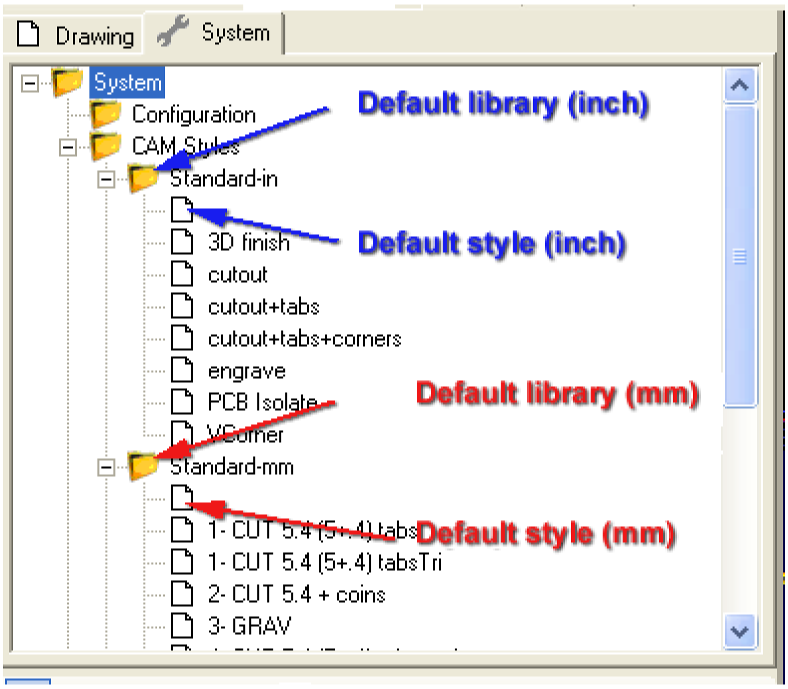
Paste format can also be used to copy the contents of a machining operation into a ***CAM Style*** object.

Delete: Removes the selected machining operation.

Rename: Renames the selected machining operation.

Reset to defaults: All the properties of the machining operation will be set to *Default* so that they will inherit their values from the parent ***CAM Style***.

If no style is specified for the machining operation, the Style set in the containing Part object will be used. If the Part does not have a defined Style, the Style set against the Machining object will be used. In the event that no style is defined at any of these levels, the default style will be used for the source of the *Default* values.



Note: The default style is the style with an empty name in the style library.

Warning: The default styles are very important for CamBam to function correctly and should not be renamed or removed.

Refer to the ***section on styles*** for more information.

Generate toolpaths: Calculate and display the tool paths for the selected machining operation only.

Produce Gcode: Creates the Gcode for this operation only, the suggested file name will be composed as follows.

```
Drawing name.part name.[machining operation].nc
```

See ***creating gcode section*** for more information.

Toolpaths to geometry: This feature allows you to create drawing objects from machining operation tool paths. These polylines can then be edited, used to create other toolpaths or exported as DXF.

Profile Machining Operation

A 2.5D Profile machining operation is typically used to cut out shapes.

Other uses include facing edges and with increased cut widths can be used to create pockets.

Cuts can be inside or outside a selected shape.

Lead in moves and holding tabs are supported.

Properties

Clearance Plane	The clearance plane (offset from the work plane). The clearance plane should be clear of the stock and any holding devices to allow free movement to any location.
Collision Detection	Makes sure adjacent toolpaths do not overlap. Multiple Toolpaths are unioned together.
Corner Overcut	 Set CornerOvercut to True to add an extra machining move, which will cut into inside corners that would not ordinarily be cut. This will result in some stock overcutting but is useful in cases where machined parts will be fitted together such as slot joints or inlays.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.
Custom MOP Header	A multi-line gcode script that will be inserted into the gcode post before the current machining operation.
Cut Feedrate	The feed rate to use when cutting.
Cut Ordering	Controls whether to cut to depth first or all cuts on this level first.
Cut Width	The total width of the cut. If this width is greater than the tool diameter, multiple parallel cuts are used.
Depth Increment	Depth increment of each machining pass. Determines the number of passes to reach the final target depth.
Enabled	*True*: The toolpaths associated with this machining operation are displayed and included in the gcode output *False*: The operation will be ignored and no gcode or tool paths will be produced

	for this operation.
Final Depth Increment	The depth increment of the final machining pass.
Holding Tabs	Defines holding tabs (bridges) to prevent cut parts moving while cutting. See the ***holding tab*** reference for more information.
Inside / Outside	Controls whether to cut Inside or Outside the selected shapes. For open shapes there is not inside or outside, so the point order controls which side of the line to cut.
Lead In Move	Defines the type of lead in move to use. ***Lead Move Type***: *None | Spiral | Tangent* ***Spiral Angle***: Used by spiral and tangents to control ramp angle. ***Tangent Radius*** : The radius of the tangent lead in ***Lead Move Feedrate*** : The feedrate to use for the lead move. If 0, Cut Feedrate is used. Refer to the ***lead move*** section for more information.
Lead Out Move	Defines the type of lead out move to use. Refer to the ***lead move*** section for more information.
Max Crossover Distance	Maximum distance as a fraction (0-1) of the tool diameter to cut in horizontal transitions. If the distance to the next toolpath exceeds MaxCrossoverDistance, a retract, rapid and plunge to the next position, via the clearance plane, is inserted.
Milling Direction	Controls the direction the cutter moves around the toolpath. *Conventional | Climb | Mixed*
Name	Each machine operation can be given a meaningful name or description. This is output in the gcode as a comment and is useful for keeping track of the function of each machining operation.
Optimisation Mode	An option that controls how the toolpaths are ordered in gcode output. *New (0.9.8)* - A new, improved optimiser currently in testing. *Legacy (0.9.7)* - Toolpaths are ordered using same logic as version 0.9.7. *None* - Toolpaths are not optimised and are written in the order they were generated.

Plunge Feedrate	The feed rate to use when plunging.
Primitive IDs	List of drawing objects from which this machine operation is defined.
Roughing / Finishing	Currently only supported by ***3D Profile*** and ***Lathe*** machining operations.
Roughing Clearance	This is the amount of stock to leave after the final cut. Remaining stock is typically removed later in a finishing pass. Negative values can be used to oversize cuts.
Side Profile	A composite property that enables the creation of pseudo 3D objects from 2D shapes by creating radii and slopes. See the ***side profiles*** reference for more information.
Spindle Direction	The direction of rotation of the spindle. *CW | CCW | Off*
Spindle Range	The pulley number or dial setting of the spindle for the target speed.
Spindle Speed	The speed in RPM of the spindle.
Start Point	Used to select a point, near to where the first toolpath should begin machining. If a start point is defined, a small circle will be displayed at this point when the machining operation is selected. The start point circle can be moved by clicking and dragging.
StepOver	The cut is increased by this amount each step, expressed as a fraction (0-1) of the cutter diameter.
Stepover Feedrate	The feed rate to use for crossover moves.
Stock Surface	This is the Z offset of the stock surface at which to start machining.
Style	Select a ***CAM Style*** for this machining operation. All default parameters will be inherited from this style.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Target Depth	The Z coordinate of the final machining depth.

Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill | BullNose | BallNose | Vcutter | Drill | Lathe*
Transform	Used to transform the toolpath. **Warning!** This property is experimental and may give unpredictable results.
Velocity Mode	Instructs the gcode interpreter whether or to use look ahead smoothing. *Constant Velocity* - (G64) Smoother but less accurate. *Exact Stop* - (G61) All control points are hit but movement may be slower and jerky. *Default* - Uses the global VelocityMode value under machining options.
Work Plane	Used to define the gcode workplane. Arc moves are defined within this plane. Options are *XY | XZ | YZ*

Pocket Machining Operation

Pockets are used to clear out stock within boundary shapes.

If selected shapes contain other shapes, CamBam will automatically detect these as 'Islands'. That is, the area around them will be cleared and the islands will remain prominent.

Properties

Clearance Plane	The clearance plane (offset from the work plane). The clearance plane should be clear of the stock and any holding devices to allow free movement to any location.
Collision Detection	Makes sure adjacent toolpaths do not overlap. Multiple Toolpaths are unioned together.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.
Custom MOP Header	A multi-line gcode script that will be inserted into the gcode post before the current machining operation.
Cut Feedrate	The feed rate to use when cutting.
Cut Ordering	Controls whether to cut to depth first or all cuts on this level first.
Depth Increment	Depth increment of each machining pass. Determines the number of passes to reach the final target depth.
Enabled	*True*: The toolpaths associated with this machining operation are displayed and included in the gcode output *False*: The operation will be ignored and no gcode or tool paths will be produced for this operation.
Final Depth Increment	The depth increment of the final machining pass.
Finish Stepover	The horizontal stepover distance used for the final cut of the pocket.
Finish Stepover At Target Depth	If *True*, the finish stepover move is only used once the final target depth is reached. If *False*, a finish stepover will be applied at each depth increment.

Lead In Move	Defines the type of lead in move to use. ***Lead Move Type***: *None | Spiral | Tangent* ***Spiral Angle***: Used by spiral and tangents to control ramp angle. ***Tangent Radius*** : The radius of the tangent lead in ***Lead Move Feedrate*** : The feedrate to use for the lead move. If 0, Cut Feedrate is used. Refer to the ***lead move*** section for more information.
Lead Out Move	Defines the type of lead out move to use. Refer to the ***lead move*** section for more information.
Max Crossover Distance	Maximum distance as a fraction (0-1) of the tool diameter to cut in horizontal transitions. If the distance to the next toolpath exceeds MaxCrossoverDistance, a retract, rapid and plunge to the next position, via the clearance plane, is inserted.
Milling Direction	Controls the direction the cutter moves around the toolpath. *Conventional | Climb | Mixed*
Name	Each machine operation can be given a meaningful name or description. This is output in the gcode as a comment and is useful for keeping track of the function of each machining operation.
Optimisation Mode	An option that controls how the toolpaths are ordered in gcode output. *New (0.9.8)* - A new, improved optimiser currently in testing. *Legacy (0.9.7)* - Toolpaths are ordered using same logic as version 0.9.7. *None* - Toolpaths are not optimised and are written in the order they were generated.
Plunge Feedrate	The feed rate to use when plunging.
Primitive IDs	List of drawing objects from which this machine operation is defined.
Region Fill Style	This option controls the pattern used to fill the pockets. The effects of each option can be seen when using the new Draw ***- Fill Region*** menu option.

	Options are: • *Horizontal Hatch* region filled with horizontal lines • *Vertical Hatch* region filled with vertical lines • *Inside+Outside Offsets* region filled with progressive offsets from outside in, unioned with offsets from islands radiating outward. • *Outside Offsets* region filled with progressive offsets from outside in (like current pocket method). • *Inside Offsets* region filled with offsets from islands radiating outward.
Roughing / Finishing	Currently only supported by ***3D Profile*** and ***Lathe*** machining operations.
Roughing Clearance	This is the amount of stock to leave after the final cut. Remaining stock is typically removed later in a finishing pass. Negative values can be used to oversize cuts.
Spindle Direction	The direction of rotation of the spindle. *CW | CCW | Off*
Spindle Range	The pulley number or dial setting of the spindle for the target speed.
Spindle Speed	The speed in RPM of the spindle.
Start Point	Used to select a point, near to where the first toolpath should begin machining. If a start point is defined, a small circle will be displayed at this point when the machining operation is selected. The start point circle can be moved by clicking and dragging.
StepOver	The cut is increased by this amount each step, expressed as a fraction (0-1) of the cutter diameter.
Stepover Feedrate	The feed rate to use for crossover moves.
Stock Surface	This is the Z offset of the stock surface at which to start machining.
Style	Select a ***CAM Style*** for this machining operation. All default parameters will be inherited from this style.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.

Target Depth	The Z coordinate of the final machining depth.
Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill | BullNose | BallNose | Vcutter | Drill | Lathe*
Transform	Used to transform the toolpath. **Warning!** This property is experimental and may give unpredictable results.
Velocity Mode	Instructs the gcode interpreter whether or to use look ahead smoothing. *Constant Velocity* - (G64) Smoother but less accurate. *Exact Stop* - (G61) All control points are hit but movement may be slower and jerky. *Default* - Uses the global VelocityMode value under machining options.
Work Plane	Used to define the gcode workplane. Arc moves are defined within this plane. Options are *XY | XZ | YZ*

Drilling Machining Operation

Used to create circular holes from selected point lists or circles.

Properties

Clearance Plane	The clearance plane (offset from the work plane). The clearance plane should be clear of the stock and any holding devices to allow free movement to any location.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.
Custom MOP Header	A multi-line gcode script that will be inserted into the gcode post before the current machining operation.
Custom Script	Custom GCode script used for drilling if DrillingMethod=CustomScript Various macros can be used in this script which will be expanded by the post processor. | - denotes a new line $c - Clearance Plane $d - Hole diameter $f - plunge feedrate $h - Z coordinate of each drill point $n - tool number $p - Dwell $q - Peck distance $r - Retract height $s - Stock Surface $t - tool diameter $x - X coordinate of each drill point $y - Y coordinate of each drill point $z - Target depth
Cut Feedrate	The feed rate to use when cutting.
Depth Increment	The depth increment controls the pitch of the spiral toolpath if ***Drilling Method*** = *Spiral Mill*. This is the depth of cut for each loop of the spiral.

Drill Lead Out	For spiral drilling only. If *True*, then move toward or away from the center of the hole before retracting.
Drilling Method	Method used to generate the drilling instruction. Options are: *Canned Cycle* - Uses G81,G82 or G83 *SpiralMill_CW* - Clockwise spiral toolpath *SpiralMill_CCW* - Counter clockwise spiral toolpath *CustomScript* - Uses the CustomScript property script
Dwell	The time to pause at the bottom of the drill cycle. The unit of time measurement depends on the machine interpreter configuration and may be seconds or milliseconds.
Enabled	*True*: The toolpaths associated with this machining operation are displayed and included in the gcode output *False*: The operation will be ignored and no gcode or tool paths will be produced for this operation.
Hole Diameter	Used for spiral mill drilling and is the diameter of the hole required. If this is set to Auto, then the sizes of the selected shapes are used to calculate the hole diameter.
Lead Out Length	For spiral drilling only. The distance to move in the lead out direction if ***DrillLeadOut*** = True. If length is positive, move toward the hole center. If length is negative, move away from the center.
Max Crossover Distance	Maximum distance as a fraction (0-1) of the tool diameter to cut in horizontal transitions. If the distance to the next toolpath exceeds MaxCrossoverDistance, a retract, rapid and plunge to the next position, via the clearance plane, is inserted.
Name	Each machine operation can be given a meaningful name or description. This is output in the gcode as a comment and is useful for keeping track of the function of each machining operation.
Optimisation Mode	An option that controls how the toolpaths are ordered in gcode output.*New (0.9.8)* - A new, improved optimiser currently in testing. *Legacy (0.9.7)* - Toolpaths are ordered using same logic as version 0.9.7. *None* - Toolpaths are not optimised and are written in the order they were generated.

Peck Distance	The incremental depth to drill before a retract. If 0, then doesn't peck drill.
Plunge Feedrate	The feed rate to use when plunging.
Primitive IDs	List of drawing objects from which this machine operation is defined.
Retract Height	For peck canned cycles, retract to this value after each peck.
Roughing / Finishing	Currently only supported by ***3D Profile*** and ***Lathe*** machining operations.
Roughing Clearance	This is the amount of stock to leave after the final cut. Remaining stock is typically removed later in a finishing pass. Negative values can be used to oversize cuts.
Spindle Direction	The direction of rotation of the spindle. *CW | CCW | Off*
Spindle Range	The pulley number or dial setting of the spindle for the target speed.
Spindle Speed	The speed in RPM of the spindle.
Spiral Flat Base	For spiral drilling only. If *True*, a full circle is added to the spiral base, to ensure a flat hole bottom. *False* will avoid the full circle cut, which may be useful for thread milling.
Start Point	Used to select a point, near to where the first toolpath should begin machining. If a start point is defined, a small circle will be displayed at this point when the machining operation is selected. The start point circle can be moved by clicking and dragging.
Stock Surface	This is the Z offset of the stock surface at which to start machining.
Style	Select a ***CAM Style*** for this machining operation. All default parameters will be inherited from this style.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Target Depth	The Z coordinate of the final machining depth.

Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill | BullNose | BallNose | Vcutter | Drill | Lathe*
Transform	Used to transform the toolpath. **Warning!** This property is experimental and may give unpredictable results.
Velocity Mode	Instructs the gcode interpreter whether or to use look ahead smoothing. *Constant Velocity* - (G64) Smoother but less accurate. *Exact Stop* - (G61) All control points are hit but movement may be slower and jerky. *Default* - Uses the global VelocityMode value under machining options.
Work Plane	Used to define the gcode workplane. Arc moves are defined within this plane. Options are *XY | XZ | YZ*

Engraving Machining Operation

Engraving machining operations ‘follow’ their selected shapes, including Z movements.

Properties

Clearance Plane	The clearance plane (offset from the work plane).The clearance plane should be clear of the stock and any holding devices to allow free movement to any location.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.
Custom MOP Header	A multi-line gcode script that will be inserted into the gcode post before the current machining operation.
Cut Feedrate	The feed rate to use when cutting.
Depth Increment	Depth increment of each machining pass. Determines the number of passes to reach the final target depth.
Enabled	*True*: The toolpaths associated with this machining operation are displayed and included in the gcode output *False*: The operation will be ignored and no gcode or tool paths will be produced for this operation.
Final Depth Increment	The depth increment of the final machining pass.
Max Crossover Distance	Maximum distance as a fraction (0-1) of the tool diameter to cut in horizontal transitions. If the distance to the next toolpath exceeds MaxCrossoverDistance, a retract, rapid and plunge to the next position, via the clearance plane, is inserted.
Name	Each machine operation can be given a meaningful name or description. This is output in the gcode as a comment and is useful for keeping track of the function of each machining operation.
Optimisation Mode	An option that controls how the toolpaths are ordered in gcode output.*New (0.9.8)* - A new, improved optimiser currently in testing. *Legacy (0.9.7)* - Toolpaths are ordered using same logic as version 0.9.7. *None* - Toolpaths are not optimised and are written in the order they were generated.

Plunge Feedrate	The feed rate to use when plunging.
Primitive IDs	List of drawing objects from which this machine operation is defined.
Roughing / Finishing	Currently only supported by ***3D Profile*** and ***Lathe*** machining operations.
Roughing Clearance	This is the amount of stock to leave after the final cut. Remaining stock is typically removed later in a finishing pass. Negative values can be used to oversize cuts.
Spindle Direction	The direction of rotation of the spindle. *CW | CCW | Off*
Spindle Range	The pulley number or dial setting of the spindle for the target speed.
Spindle Speed	The speed in RPM of the spindle.
Start Point	Used to select a point, near to where the first toolpath should begin machining. If a start point is defined, a small circle will be displayed at this point when the machining operation is selected. The start point circle can be moved by clicking and dragging.
Stock Surface	This is the Z offset of the stock surface at which to start machining.
Style	Select a ***CAM Style*** for this machining operation. All default parameters will be inherited from this style.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Target Depth	The Z coordinate of the final machining depth. For engraving operations, the Z coordinate of the source drawing object point will also be added to the toolpath so that the engraving toolpath can 'follow' the shape's Z contour.
Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.

Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill | BullNose | BallNose | Vcutter | Drill | Lathe*
Transform	Used to transform the toolpath. **Warning!** This property is experimental and may give unpredictable results.
Velocity Mode	Instructs the gcode interpreter whether or to use look ahead smoothing. *Constant Velocity* - (G64) Smoother but less accurate. *Exact Stop* - (G61) All control points are hit but movement may be slower and jerky. *Default* - Uses the global VelocityMode value under machining options.
Work Plane	Used to define the gcode workplane. Arc moves are defined within this plane. Options are *XY | XZ | YZ*

3D Profile Machining Operation

3D Profiles can be used to machine 3D objects from triangular mesh files. Currently 3DS, STEP **(New 1.0)** and STL files are supported.

3D Profiles support the following features.

- Waterline roughing and finishing methods.
- Z scanline roughing and finishing methods.
- Front face and back face machining.
- Generation of negative molds from positive shapes.
- Restriction of machining boundary to save machining time.
- Experimental additive support for extrusion heads.

This method replaces the Bas Relief method in older CamBam versions.

See also:
3D Profile Tutorial, 3D Profile Tutorial - Back face

Properties

Additive	If set to **True**, an additive toolpath will be generated, suitable for extrusion heads. Additive toolpaths are generated from lowest to highest Z levels with the lowest (starting) level at Z = ***Stock Surface***. For best results, this setting would be combined with a Waterline Rough 3D profile methods, and a small ***Depth Increment***. This method is very experimental at the moment and more work is needed to tie in with the post processor to control the extruder.
Arc Fit Tolerance	The tolerance used when automatic arc fitting is applied. Zero will use an automatically calculated value.
Auto Arc Fitting	Whether to apply arc fitting. Arc fitting will make toolpaths smoother to machine and faster to calculate, but may introduce some inaccuracy.
Back Face	When set to *True*, a toolpath for the back face of the model will be generated. If the back face option enabled, a valid ***Back Face Zero Z*** setting should also be supplied.

Back Face Culling	To improve code generation speed, model faces pointing away from the front are ignored. This can cause problems when the triangle winding order is inconsistent, so this behaviour can be disabled by setting ***Back Face Culling*** to *False*.
Back Face Zero Z	If the ***Back Face*** setting is enabled, this is the current Z coordinate that will be at Z=0 after the model is 'Flipped' about the ***Flip Axis***.
Boundary Margin	The outer boundary shape, as determined by the ***Boundary Method*** setting, is extended by the distance give in the ***Boundary Margin*** setting. It is recommended that a margin greater than 0 is used when using waterline profile methods in combination with *Shape Outline* boundary methods.
Boundary Method	This property controls the shape of the area around the model to machine. Boundary shapes options are: *Shape Outline* - the outline shape of the source 3d models. *Bounding Box* - a rectangle enclosing the source geometry. *Selected Shapes* - A list of 3D or 2D shapes specified in ***Boundary Shape Ids***.
Boundary Shape IDs	A list of drawing entity IDs that represent the shapes to use to determine the boundary shape.
Boundary Taper	Angle in degrees from vertical to taper the outer boundary edge.
Clearance Plane	The clearance plane (offset from the work plane). The clearance plane should be clear of the stock and any holding devices to allow free movement to any location.
Clip Area Max	A 2D Point, used with ***Clip Area Min*** to define a clipping area.
Clip Area Min	A 2D Point, used with ***Clip Area Max*** to define a clipping area. If ***Clip Area Max*** and ***Clip Area Min*** coordinates are both 0, the machining area will not be clipped.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.

Custom MOP Header	A multi-line gcode script that will be inserted into the gcode post before the current machining operation.
Cut Feedrate	The feed rate to use when cutting.
Cut Ordering	Controls whether to cut to depth first or all cuts on this level first.
Depth Increment	Depth increment of each machining pass. Determines the number of passes to reach the final target depth.
Enabled	*True*: The toolpaths associated with this machining operation are displayed and included in the gcode output *False*: The operation will be ignored and no gcode or tool paths will be produced for this operation.
Flip Axis	The axis around which you would flip the stock to machine the back face.
Lead In Move	Defines the type of lead in move to use. ***Lead Move Type***: *None | Spiral | Tangent* ***Spiral Angle***: Used by spiral and tangents to control ramp angle. ***Tangent Radius*** : The radius of the tangent lead in ***Lead Move Feedrate*** : The feedrate to use for the lead move. If 0, Cut Feedrate is used. Refer to the ***lead move*** section for more information.
Lead Out Move	Defines the type of lead out move to use. Refer to the ***lead move*** section for more information.
Max Crossover Distance	Maximum distance as a fraction (0-1) of the tool diameter to cut in horizontal transitions. If the distance to the next toolpath exceeds MaxCrossoverDistance, a retract, rapid and plunge to the next position, via the clearance plane, is inserted.
Milling Direction	Controls the direction the cutter moves around the toolpath. *Conventional | Climb | Mixed*
Mold	If set to *True*, a negative mold toolpath is generated from a positive shape.

Name	Each machine operation can be given a meaningful name or description. This is output in the gcode as a comment and is useful for keeping track of the function of each machining operation.
Optimisation Mode	An option that controls how the toolpaths are ordered in gcode output. *New (0.9.8)* - A new, improved optimiser currently in testing. *Legacy (0.9.7)* - Toolpaths are ordered using same logic as version 0.9.7. *None* - Toolpaths are not optimised and are written in the order they were generated.
Plane Slice Only	CamBam's waterline routines have been designed to work best with natural / curved shapes. Engineering shapes with perpendicular sides can potentially cause problems. If problems are encountered, setting ***Plane Slice Only*** to *True* can help but will only work with shapes that do not have any overhangs.
Plunge Feedrate	The feed rate to use when plunging.
Primitive IDs	List of drawing objects from which this machine operation is defined.
Profile 3D Method	The method used to generate the 3D toolpath. • *Horizontal* - Use a Z scanning method in horizontal direction. • *Vertical* - Use a Z scanning method in vertical direction. • *Waterline Rough* - Uses a series of waterline slices which are pocketed up to the boundary shape. • *Waterline Finish* - Creates a profile using the tool offset at each waterline slice.
Region Fill Style	This option controls the pattern used to fill the pockets. The effects of each option can be seen when using the new Draw ***- Fill Region*** menu option. Options are: • *Horizontal Hatch* region filled with horizontal lines • *Vertical Hatch* region filled with vertical lines • *Inside+Outside Offsets* region filled with progressive offsets from outside in, unioned with offsets from islands radiating outward. • *Outside Offsets* region filled with progressive offsets from outside in (like current pocket method). • *Inside Offsets* region filled with offsets from islands radiating outward.

Resolution	For *Horizontal* and *Vertical* 3D profile methods, this is the distance along each scan line (expressed as a fraction (0-1) of the cutter diameter), for each Z height test point. Larger resolution values are faster but could result in some features being over cut.
Roughing / Finishing	Used control whether this should be a roughing or finishing pass. For horizontal and vertical scanline operations: If *Roughing* is selected, toolpaths at each ***Depth Increment*** are calculated. If *Finishing* is selected, only the toolpath at the final ***Target Depth*** is calculated.
Roughing Clearance	This is the amount of stock to leave after the final cut. Remaining stock is typically removed later in a finishing pass. Negative values can be used to oversize cuts.
Scanline Gradient Threshold	For horizontal and vertical scanline methods, this property will suppress tool path segments steeper than a given gradient. The value is specified in degrees where 90 degrees is vertical (Z). A scanline finish with a reduced gradient threshold is useful when combined with a waterline finish operation. Waterline finish is best suited for steep areas but may result in uncut bands in shallow areas due to the limits its depth increment. Whereas scanlines work well on flat areas but can result in scalloped tool marks on steep model sides. Using a ***Boundary Margin*** of (minus) the tool radius is recommended when using this property to restrict tool paths to shallow areas.
Spindle Direction	The direction of rotation of the spindle. *CW | CCW | Off*
Spindle Range	The pulley number or dial setting of the spindle for the target speed.
Spindle Speed	The speed in RPM of the spindle.
Start Corner	Corner to start profiling. Used in *Horizontal* and *Vertical* 3D profiling methods only.

Start Point	Used to select a point, near to where the first toolpath should begin machining. If a start point is defined, a small circle will be displayed at this point when the machining operation is selected. The start point circle can be moved by clicking and dragging.
StepOver	The cut is increased by this amount each step, expressed as a fraction (0-1) of the cutter diameter.
Stepover Feedrate	The feed rate to use for crossover moves.
Stock Surface	This is the Z offset of the stock surface at which to start machining.
Style	Select a ***CAM Style*** for this machining operation. All default parameters will be inherited from this style.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Target Depth	The Z coordinate of the final machining depth.
Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill | BullNose | BallNose | Vcutter | Drill | Lathe*

Transform	Used to transform the toolpath. **Warning!** This property is experimental and may give unpredictable results.
Velocity Mode	Instructs the gcode interpreter whether or to use look ahead smoothing. *Constant Velocity* - (G64) Smoother but less accurate. *Exact Stop* - (G61) All control points are hit but movement may be slower and jerky. *Default* - Uses the global VelocityMode value under machining options.
Work Plane	Used to define the gcode workplane. Arc moves are defined within this plane. Options are *XY | XZ | YZ*

Lathe Machining Operation

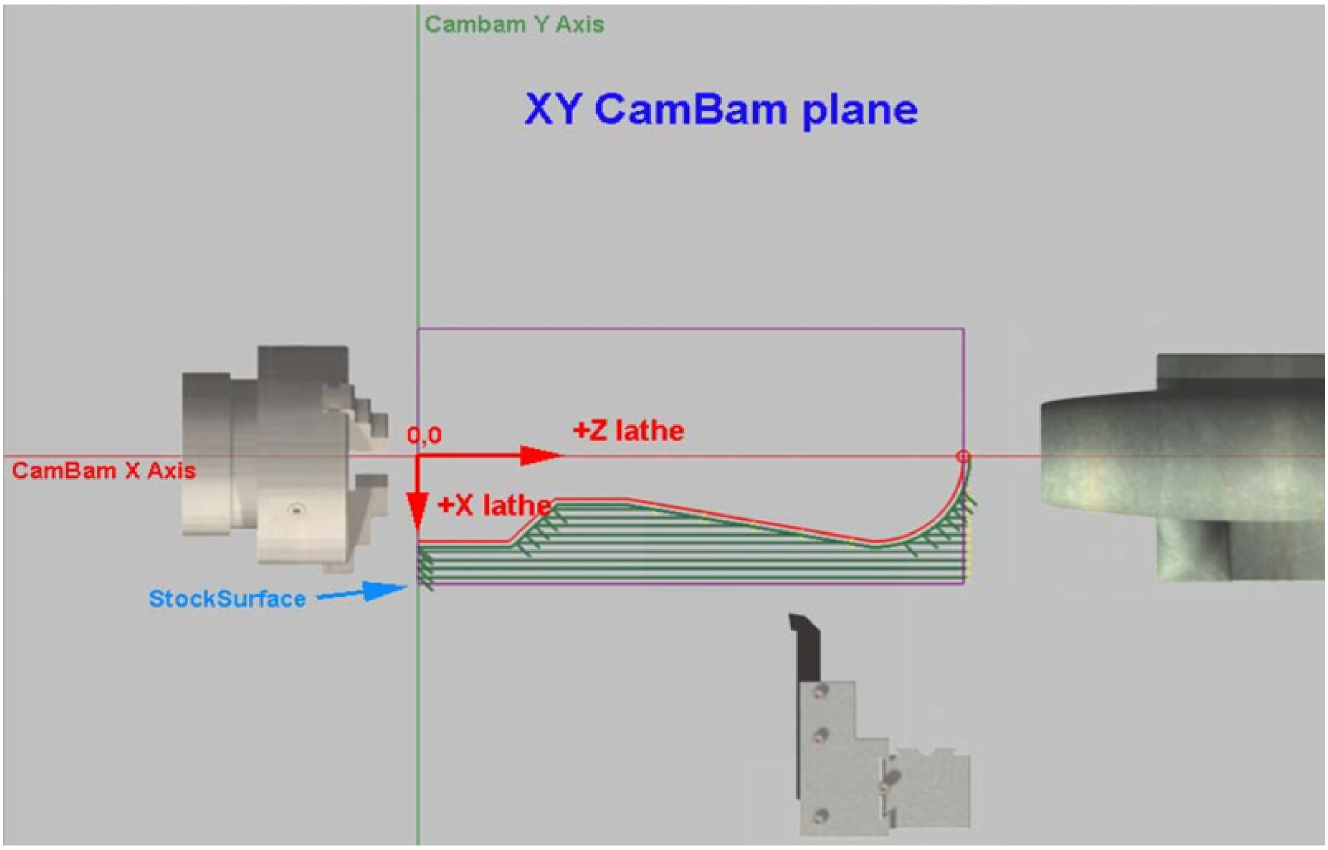
***Note*: The lathe code is new to version 0.9.8 and is still undergoing testing and development.
Treat any lathe gcode with caution and run simulations or air cuts before machining.

The Lathe machining operation has been provided as a plugin. In this way the plugin can be developed and updated independently of the main CamBam application. It is also a demonstration of the ability to extend CamBam's machining capability using user written plugins.

The file **lathe-test.cb** in the CamBam samples folder demonstrates the new lathe operation.

In this initial lathe release there are a number of limitations:

- Only profiling operations are currently supported. No facing, boring or threading support yet.
- Apart from the tool radius, there is no mechanism to define a lathe tool shape. The part should be drawn to allow for the cutter size and shape.



Drawing

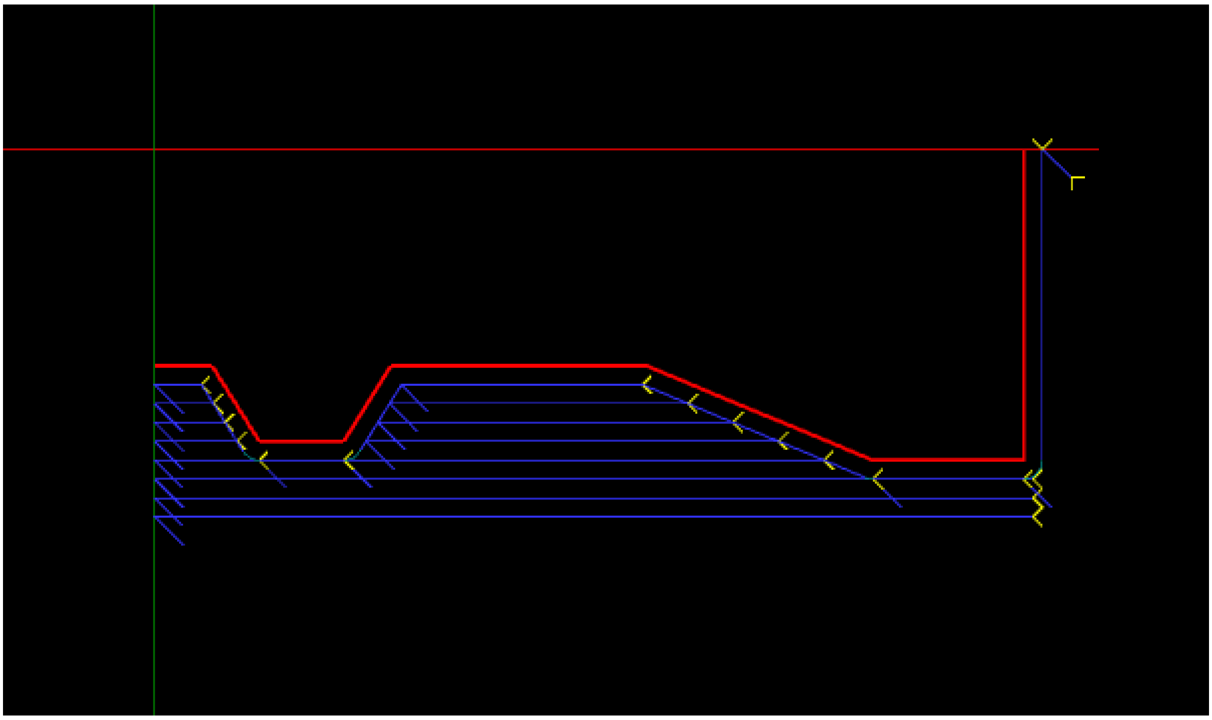
A lathe profile can be generated from a 2D line representing the shape to machine. The shape should be drawn so that: The lathe **+X** axis is drawn in the **-Y** direction and The lathe **+Z** axis is drawn in the **+X** direction.
This is so that the drawing will appear in the same orientation as when standing in front of a conventional

lathe.
The toolpaths will be converted to standard lathe X and Z coordinates when the gcode is produced.

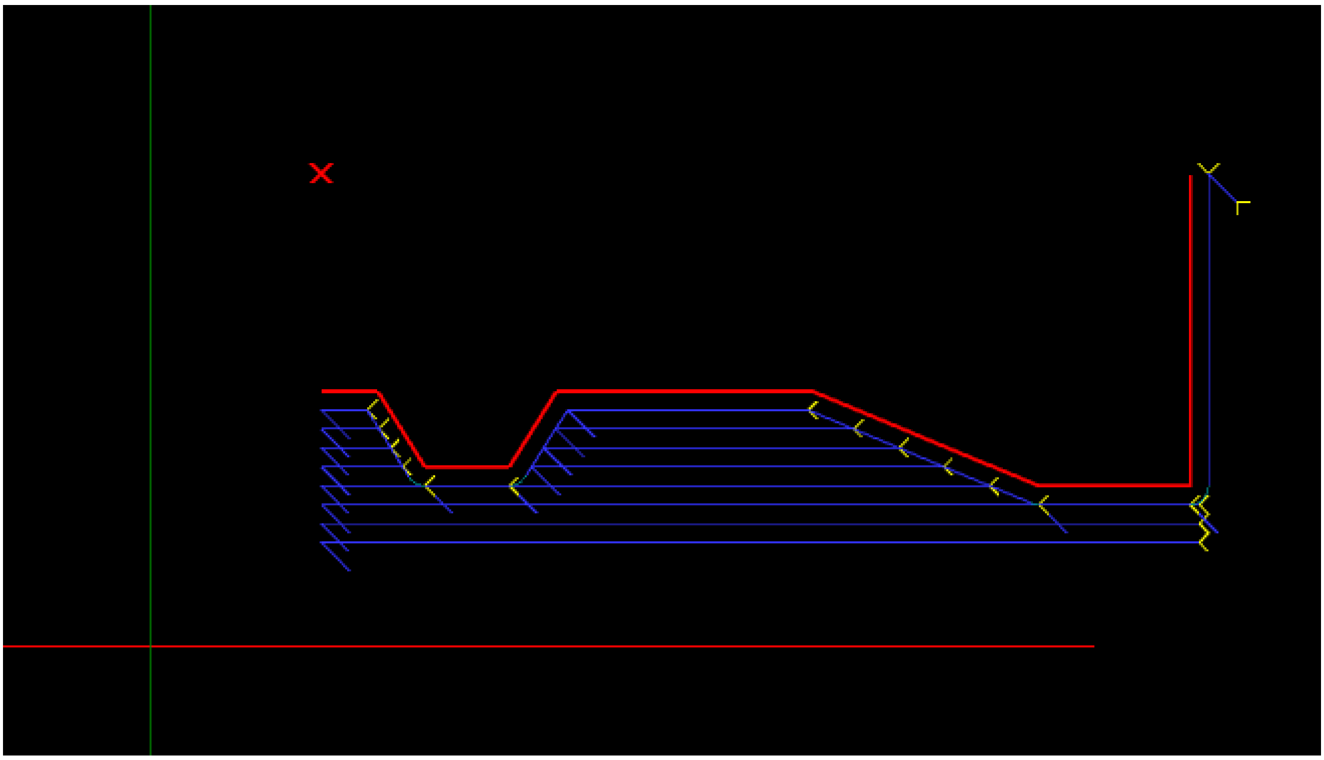
Only draw the profile line to be cut. Do not draw closed polylines, mirrored lines on the opposite side of the turning axis or lines along the turning axis as the lathe operation will try to cut these as well which will cause problems.

The profile line can be drawn anywhere in the drawing. If this line is away from the origin, the Machining Origin should be set so that it lies on the axis of rotation and at the Z=0 (lathe coordinate).

An example showing a profile where the machining 0,0 point is the same as the drawing origin.



The same pattern drawn away from the origin, where the machining origin (red X) has been moved to indicate the lathes X=0, Z=0 point.



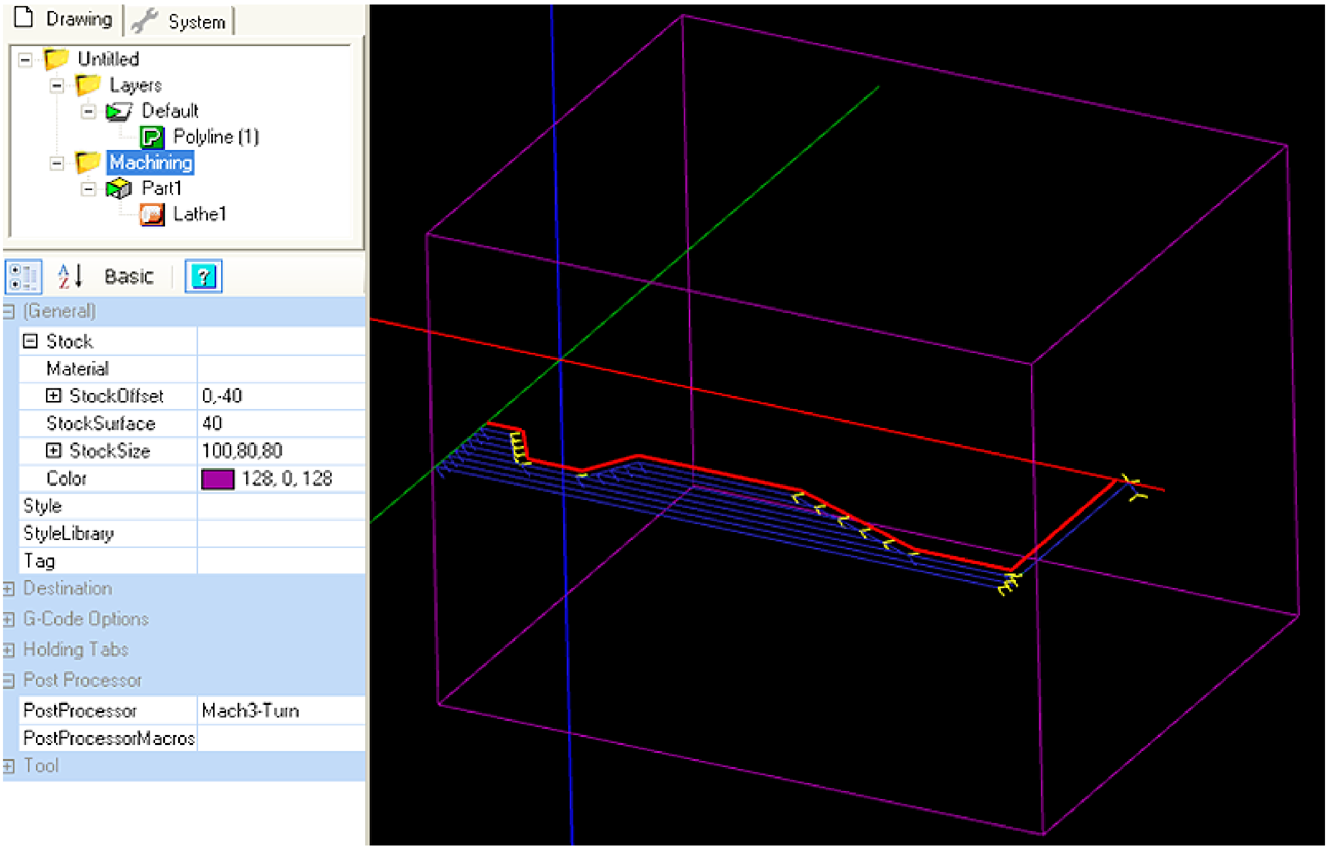
You can set the machine zero by setting MachiningOrigin property of the machining or part objects. Click the ... button to the right of the MachiningOrigin property to select the machine zero point on the drawing.

Stock Object

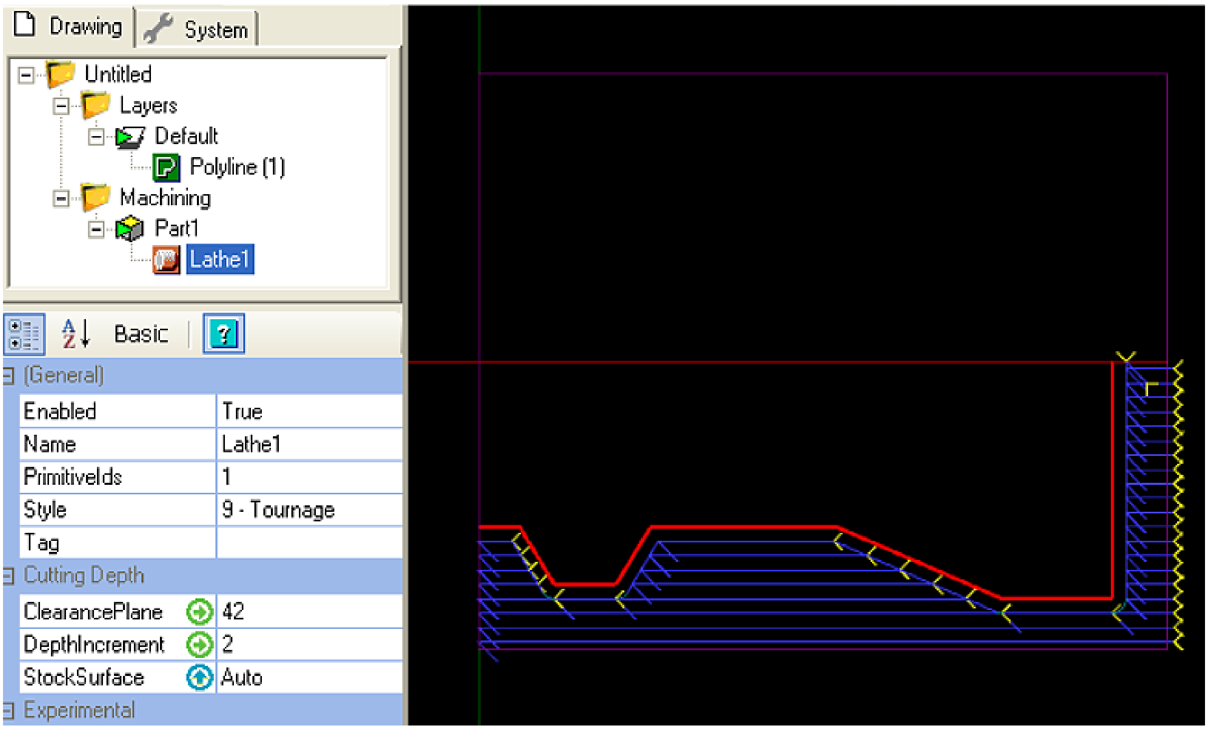
The lathe operation can use information from the stock object if one is defined, to determine properties such as stock surface and the machining envelope.

CamBam's stock definition does not currently support cylindrical stock so the stock will be shown as a rectangular block.

The following image shows a stock objects of 9-mm diameter and 100mm long (purple cube).



If the *Stock Surface* property is set to *Auto*, the stock object size is used to define it.



- The X size will be the length of the stock (along the lathe's Z axis).
- The Z and Y size should both be set to the stock diameter.
- StockSurface should be set to the stock radius.
- The Stock offset Y value should be set to **negative** the stock radius.

Using the Lathe Operation

Select a suitable profile line, then insert a lathe operation by selecting the top Machining menu, then select Lathe.

Note - The lathe plugin does not currently add an icon to the toolbar or drawing context menu.

Make sure the following are set:

- Workplane is set to XZ.
- Stock surface equals the radius of the stock.
- Clearance plane is greater than the radius of the stock.
- The machining origin is set along the axis of rotation.
- The tool diameter is set to twice the tool nose radius.
- The tool profile is set to Lathe.
- The correct RoughingFinishing option is set.
- If Roughing, a small RoughingClearance value is set.
- DepthIncrement and feedrates are appropriate for the material.
- Define the stock object if needed.
- A suitable post processor such as Mach3-Turn or EMC-Turn are selected in the Machining properties.

Properties

Clearance Plane	The safe **X** lathe coordinate to avoid any stock. The ***clearance plane*** value should always be expressed as a **radius**.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.
Custom MOP Header	A multi-line gcode script that will be inserted into the gcode post before the current machining operation.
Cut Feedrate	The feed rate to use when cutting.
Depth Increment	When roughing, this is the radial X distance between each parallel cut.
Enabled	*True*: The toolpaths associated with this machining operation are displayed and included in the gcode output *False*: The operation will be ignored and no gcode or tool paths will be produced for this operation.
Lathe Cut Direction	• *Right Hand* - Cuts will move from right (+Z) to left (-Z). • *Left Hand* - Cuts will move from left (-Z) to right (+Z).
Lathe Lead In Length *New [0.9.8N]*	Controls the length of the 45 degree lead in moves. A zero value will disable these moves.
Lathe Lead Out Length *New [0.9.8N]*	Controls the length of the 45 degree back away moves. A zero value will disable these moves.
Max Crossover Distance	Maximum distance as a fraction (0-1) of the tool diameter to cut in horizontal transitions. If the distance to the next toolpath exceeds MaxCrossoverDistance, a retract, rapid and plunge to the next position, via the clearance plane, is inserted.
Name	Each machine operation can be given a meaningful name or description. This is output in the gcode as a comment and is useful for keeping track of the function of each machining operation.

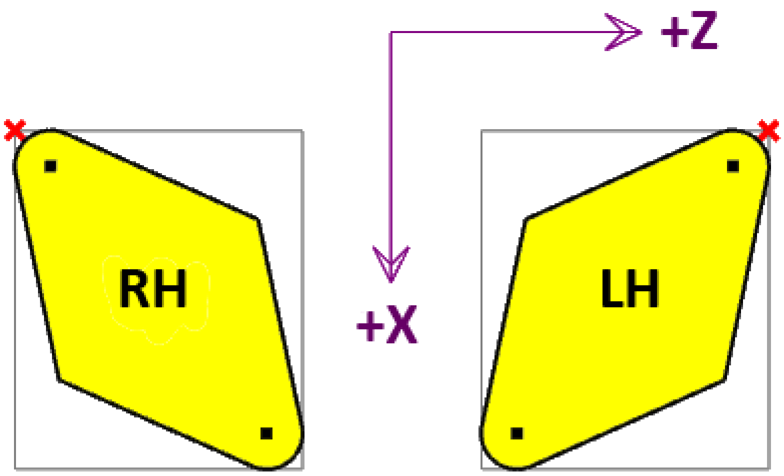
Optimisation Mode	An option that controls how the toolpaths are ordered in gcode output.*New (0.9.8)* - A new, improved optimiser currently in testing. *Legacy (0.9.7)* - Toolpaths are ordered using same logic as version 0.9.7. *None* - Toolpaths are not optimised and are written in the order they were generated.
Plunge Feedrate	The feed rate to use when plunging.
Primitive IDs	List of drawing objects from which this machine operation is defined.
Roughing / Finishing	The ***Roughing / Finishing*** property is used to select the machining method. If *Roughing* is selected, a number of straight passes are used at each depth increment, down to the source shape + roughing clearance, followed by a single cut at the roughing clearance distance that follows the shape. For *Finishing*, a single cut that follows the shape at the roughing clearance distance is used.
Roughing Clearance	This is the amount of stock to leave after the final cut. Remaining stock is typically removed later in a finishing pass. Negative values can be used to oversize cuts.
Spindle Direction	The direction of rotation of the spindle. *CW | CCW | Off*
Spindle Range	The pulley number or dial setting of the spindle for the target speed.
Spindle Speed	The speed in RPM of the spindle.
Start Point	Used to select a point, near to where the first toolpath should begin machining. If a start point is defined, a small circle will be displayed at this point when the machining operation is selected. The start point circle can be moved by clicking and dragging.
Stock Surface	This is the X offset of the stock surface at which to start machining. Can be set explicitly or determined from the stock object. ***Stock surface*** should always be expressed as a **radius**.
Style	Select a ***CAM Style*** for this machining operation. All default parameters will be inherited from this style.

Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter. The new *Lathe* tool profile should always be used. If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used.
Velocity Mode	Instructs the gcode interpreter whether or to use look ahead smoothing. *Constant Velocity* - (G64) Smoother but less accurate. *Exact Stop* - (G61) All control points are hit but movement may be slower and jerky. *Default* - Uses the global VelocityMode value under machining options.
Work Plane	**Should always be set to XZ for lathe code!**

Post Processor

Three sample lathe specific post processor definitions have been provided : Mach3-Turn, Mach3-Turn-CV (Mach3 with CutViewer definitions) and EMC2-Turn.
These definitions may need to be customised to suit the configuration of those controllers.

This section describes some post processor properties that are relevant to customising the lathe gcode output.

Clearance Plane Axis	Used to specify which direction clearance moves are made. Usually Z for normal milling, but must be set to X for lathe turning operations.
Lathe X Mode	Controls whether the X lathe coordinates will be written to gcode as *Radius*, or a *Diameter*. ***Depth Increment***, ***Stock Surface*** and ***Clearance Plane*** parameters should always be specified as a radius, regardless of the post processor ***Lathe X Mode*** setting.
Lathe Tool Radius Offset	If *False*, the toolpath at the center of the tool radius is output. If *True*, an appropriate tool radius offset is applied. The toolpath will be offset by a negative tool radius in the lathe X axis. The direction of the Z tool radius offset is determined by the cut direction. For right hand cuts the toolpath Z will be offset by a negative tool radius. For left hand cuts, a positive tool radius Z offset is used.  In the diagram above, the red cross represents the toolpath reference point when ***Lathe Tool Radius Offset*** is set *True*. If *False*, the dot at the tool radius center will be the reference point. The reference point is sometimes referred to as the 'Imaginary' or 'Virtual' tool point.
X Mode Diameter	Code to use to set X diameter mode (eg G7 for LinuxCNC)
X Mode Radius	Code to use to set X radius mode (eg G8 for LinuxCNC)
Invert Arcs	If set *True*, CW arcs will be output as CCW and vice versa. This may be useful for front face lathe operations.

Arc Output	*Normal* is the preferred setting and will use G2 and G3 codes to output arcs. *Convert To Lines* may be used as a last resort if CamBam can not generate arc codes in a format compatible with the destination controller. *Convert To Lines* is used with the ***Arc To Lines Tolerance*** property, where smaller tolerances will result in smoother curves but larger files.

Tool Definitions

A sample lathe tool library 'Lathe-mm' is provided. The tool library can be selected by changing the ***Tool Library*** property in the ***Machining*** or ***Part*** options.

Tool libraries are currently designed to support milling cutters, rather than lathe. However there are a couple of parameters than are useful to store in the tool library.

Tool Profile

should always be set to the new *Lathe* option. Among other things, this instructs the post processor to determine the tool radius from the tool diameter.

A new ***Comment*** property has been added. This is a text value that can be included by the post processor when using the {$tool.comment} macro from with the ToolChange post processor section.

For example. CutViewer Turn recognises a gcode comment that defines the geometry of the lathe tool in the following format:

```
TOOL/STANDARD,BA,A,R,IC,ITP
```

Refer to the CutViewer Turn documentation for details of this description. Here is a summary of the parameters:

- **BA** - Back angle.
- **A** - Angle.
- **R** - Radius.
- **IC** - Inner circle.
- **ITP** - Imaginary Tool Point. 0=Tool Center, 3 for right hand offset, 4 left hand offset.

This example ***Comment*** property defines a right hand cutter with a 2mm radius, 40 degree back angle and 40 degree taper.

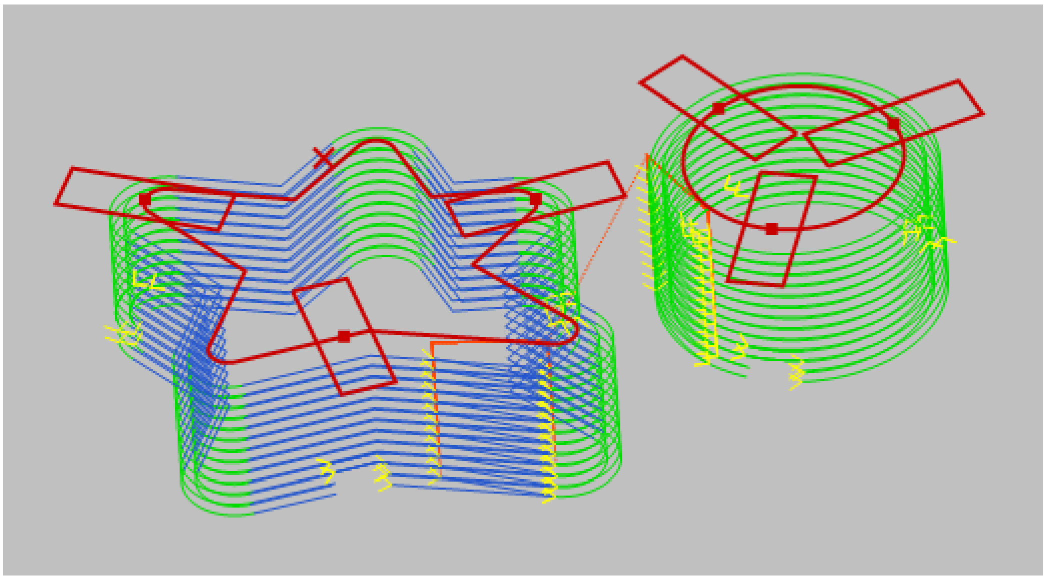
```
{$comment} TOOL/STANDARD,40,40,{$tool.radius},2,3 {$endcomment}
```

Creating GCode

1. The basic work flow for creating Gcode files is:
2. Create or Import drawing objects.
3. Select drawing objects and define Machining Operations.
4. Generate Toolpaths and visually inspect.
5. Create the destination Gcode file.

Generating and Inspecting Toolpaths

Toolpaths are generated by selecting the ***Machining - Generate Toolpaths*** menu item, Pressing **CTRL+T**, or by right clicking on individual machining operations in the drawing tree and selecting ***Generate Toolpath*** from the context menu.



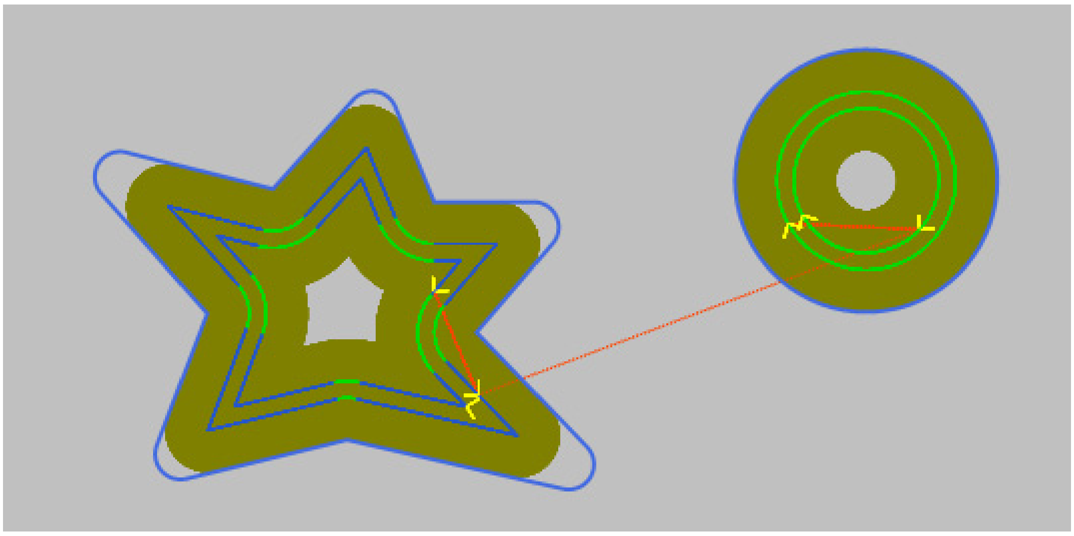
CamBam provides a 3D drawing view. Rotate the view (using **ALT**+*Drag*) to see more detail of the toolpaths including the different depth levels.

The toolpaths indicate the path that will be traveled by the central tip of the cutting tool. Different colors are used to differentiate straight line moves and arc moves. Small arrows indicate the cutting directions. Rapids are displayed using dotted red lines.

There are a number of settings which control the appearance of the toolpaths.

Toolpaths can be shown or hidden using the ***View - Show Toolpaths*** top menu and context menu options.

Other toolpath viewing options are available from the ***View*** menu.



Setting the drawing view option ***Show Cut Widths*** = *True* is a useful way of seeing the areas of stock that will cleared.

Another useful setting is the ***Toolpath Visibility*** property, which is available when selecting the top level (drawing) object in the tree view. This setting can be *All*, to show all enabled toolpaths, or *Selected Only*, which will only display toolpaths for machining operations that have been selected in the drawing tree, or whose containing ***Part*** object has been selected.

Select a post processor

A post processor will help covert machining operations and toolpaths into gcode, suitable for specific target machines.

If no post processor is defined, the default post processor is used.

Each drawing file can specify its own post processor, set in the machining options. To set a default post processor for all drawings, a default drawing template can be created which contains the required post processor.

See the ***post processor*** section for more details.

Creating Destination Gcode File

Once the machining operations have been correctly defined and inspected, a Gcode file can be produced to send to the CNC controller.
This is done by selecting the ***Machining - Produce Gcode*** menu option.

If a gcode file has not been previously created, a destination file location prompt is displayed.

The gcode filename is stored, and can be changed, by selecting the machining object in the drawing tree, then changing the ***Out File*** property under the Destination group of the machining properties. Selecting the ***Out File*** value will cause a […] button to appear which can be used to open a file browser.

A default filename is suggested by appending the default Gcode file extension to the current filename. In the system configuration settings there is a setting called ***Default GCode Extension*** which is used to set the file extension.

Often it is useful to be able to create Gcode from a single machining operation. This is particularly useful for new designs, where each machining step can be exported and tested separately. To do this, right click a specific machining operation in the drawing tree and select ***Create Gcode File*** from the context menu.

Machining Options

Parameters that control how machining operation toolpaths are generated, as well as how gcode is produced, can be set by selecting the ***Machining*** folder in the drawing tree and inspecting the property window.

> ***Note*****:** In earlier CamBam versions, settings that controlled how toolpaths were displayed were also found in the Machining options. In version 0.9.8, these have been moved to the top level *Drawing* object of the file tree and are also accessible from the *View* menu.

Properties

Arc Center Mode	This property controls whether the I and J parameters for arc moves (G2, G3) use absolute coordinates or incremental, relative to the arc end points. If this setting is different to the way the CNC controller interprets arc moves, the resulting toolpath may look a mess of random arcs in the controller. *Default* When default is set in the drawing's machining properties, the post processor Arc Center Mode will be used. A default value in the post processor will use *Incremental (C-P1).* *Absolute* I & J are absolute coordinates of the arc center point. *Incremental (C-P1)* I & J are coordinates of the arc center, offset from the first arc point. This is the typical incremental mode. In previous versions this option was just called *Incremental*. *Incremental (P1-C)* I & J are offsets of the first arc point from the arc center. *Incremental (C-P2)* I & J are arc center offsets from the second arc point. *Incremental (P2-C)* I & J are offsets of the second arc point from the arc center.
Custom File Footer	This text is inserted at the end of the gcode output. It can contain multiple text lines or pipe characters '|' to denote new lines. It can also contain $macros. Common available macros are described in the post processor section.
Custom File Header	This text is inserted at the beginning of the gcode output. It can contain multiple text lines or pipe characters '|' to denote new lines. It can also contain $macros. Common available macros are described in the post processor section.

Fast Plunge Height	This value is used when moving down to the stock surface or next cutting level. If set to 0, the current machining operation's ***Plunge Feedrate*** is used (which can result in slow machining times). If a non zero ***Fast Plunge Height*** is specified, a rapid move is used (G0) to the specified height above the stock. This can significantly improved cutting times in some files. A typical example might be 0.1 or Metric or 0.004 for Inches. The default value is (-1), which will use one minor grid unit as the fast plunge height.
Holding Tabs: ***Inner Tab Scale, Outer Tab Scale*** *New! [0.9.8i]*	Adjusts the length of the holding tabs by scaling the length by these amounts. ***Outer Tab Scale*** is the length toward the toolpath and ***Inner Tab Scale*** is the length away from the toolpath.
Machining Origin	A drawing point that will be used as the machining origin (X=0,Y=0) point when gcode is created. The ellipsis button to the right of this property can be used to select a point in the drawing. An 'X' icon will be displayed on the drawing at the machining origin point. This cross can be dragged to a new location using the mouse. ***Note***: MachiningOrigin replaces the GCodeOrigin and GCodeOriginOffset properties of earlier releases.
Number Format	Controls how decimal numbers are output to the gcode file. This property is overridden by the ***Number Format*** specified in the selected post processor. See the Post Processor section for more information.
Out File	This is the location of the destination gcode file. Clicking the button to the right of this property will open a file browser.
Post Processor	A selection from a drop down list which contains a list of all the post processors available. The post processor controls how the gcode files are formatted and are user configurable using XML based post processor files.
Post Processor Macros	This is a text field containing multiple macro definitions (one per line), of the format $macro=value. These macros can be used by the selected post processor and are a handy way of passing parameters from the drawing to the post processor.

Rebuild Toolpath Before Post	Controls whether to regenerate toolpaths before creating gcode post. • *Always* - Toolpaths will automatically be regenerated before posting the gcode. • *Prompt* - Prompts whether or not to regenerate toolpaths before posting. • *If Needed* - Toolpaths will be regenerated if machining properties or drawing objects change. *Prompt* or *If Needed* are useful when the toolpaths take a long time to generate such as with some 3D operations.
Show Cut Widths	*True* | *False*. Show cut widths will shade the areas that will be cut. This feature currently only works when the drawing view has not been rotated. It should be easy to spot any areas that are not shaded and will therefore have stock remaining.
Show Direction Vector	*True* | *False*. Controls the visibility of a small arrow at the start point of each toolpath that indications the direction of machining.
Show Rapids	*True* | *False*. Controls the visibility of a dashed line that indicates rapid moves from one toolpath to the next. ***Note***: Rapids are currently only displayed within each machining operation. Rapids from one machining operation to the next are not yet shown but should be in the next release.
Show Toolpaths	*True* | *False*. Shows or hides the toolpaths. This is the same as using the ***View - Show Toolpaths*** menu option.

Stock	The stock object is used to define the dimensions of a block of material from which the part will be cut. The properties of the stock object can be used to automatically determine some machining properties. • If a machining operation or style's ***Stock Surface*** property is set to *Auto*, the stock's stock surface value will be used. • If a machining operation or style's ***Target Depth*** property is set to *Auto*, the stock's stock surface and Z size will be used to determine the target depth, so a machining operation will by default machine all the way through the stock. Stock properties: ***Material***: Informational text that describes the stock material. ***Stock Offset***: X and Y offset of the lower left corner of the stock block. For example, a stock offset of -10,-20 would position the stock 10 units to the left of the Y axis (X=0) and 20 units below the X axis (Y=0). ***Stock Surface***: The Z location of the top of the stock block. ***Stock Size***: The X, Y and Z dimensions of the stock block. ***Color***: Color to use when displaying this stock object. Stock is undefined if the X,Y and Z sizes are all zero. Stock can be defined at the part or machining level. Stock defined at the part level will override and machining level stock definitions and will be used for all operations within the part. The stock object dimensions can also be passed to simulators such as CutViewer when post processors with appropriate stock macros are included, such as the *Mach3-CutViewer* post processor.
Style	Select a default ***CAM Style*** for this part. All machining operations in the part will use this style unless set otherwise in the machining operation's Style property.
Style Library	This property is used to locate the style definitions used in the Part or machining operations.
Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.

Tool Library	If left blank, the default tool library will be used (Default-{$Units}), otherwise the specified library will be used when looking up tool numbers.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill* | *BullNose* | *BallNose* | *Vcutter* | *Drill* | *Lathe*
Toolpath Visibility	*All* | *Selected Only* When there are a lot of machining operations, it can get visually confusing as to which toolpath belongs to which machining operation. By setting ***Toolpath Visibility*** to Selected Only, only the toolpaths for the machining operation selected in the drawing tree are visible.
Velocity Mode	*Constant Velocity* | *Default* | *Exact Stop* Controls the use of G61 and G64 commands in gcode output. This global velocity mode setting can be overridden by individual machine operations. For example it may be useful to have a global value of *Constant Velocity* set for the drawing and use *Exact Stop* for finishing machine operations. If *Default* is used, no velocity mode gcode is written (or the global velocity mode is used for machining operations). *Constant Velocity*, sometimes referred to as 'Look Ahead', is a useful feature implemented in some CNC controllers so that motion is smoothed between control points. This is particularly useful with geometry that involves a sequence of many small movements, often trying to approximate a natural shape. The downside is a potential loss of accuracy.

Viewing and editing gcode

CamBam can be used to view and edit the output gcode. It is also possible to specify an external gcode editor for this purpose.

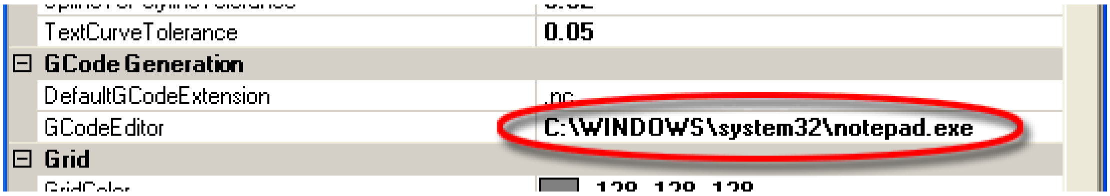
To invoke the gcode editor, use the ***Machining - Edit gcode*** menu option, or from the context menu presented when right clicking the machining folder in the drawing tree.

Edit gcode currently only edits the top level ***Machining*** gcode file. To edit gcode created from ***Parts*** or individual machining operations it is necessary to open these manually.

To find the location of the created gcode file, ***Browse gcode folder*** from the Machining context menu can be used. This will open Windows Explorer to the gcode folder location.

The tool used to open gcode files can be set in the ***GCode Editor*** property of the ***system configuration settings.***

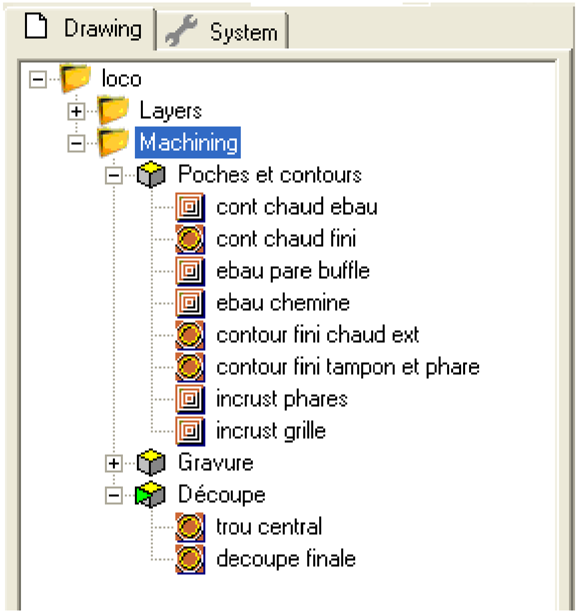
In the following example, Windows notepad has been defined as the gcode editor.



Gcode files can also be view and their toolpaths displayed (or back plotted) using the ***NC File***. Double clicking the ***NC File*** object in the drawing tree will invoke the gcode editor on the source gcode of the ***NC File*** operation.

See the ***Back Plotting*** section for more information.

Part Machining Object



A Part is a way of grouping multiple, related machining operations into a single object. A single drawing file can contain many different part objects.

Parts can be enabled or disabled individually. As with layers and machining operations, pressing the space bar when the item is selected in the drawing tree, will toggle a part's enabled state.

To generate the toolpaths for all the machining operations in a part, right click the part in the drawing tree, then select ***Generate toolpaths***. Right click an individual machining operation to generate toolpaths for just that mop, and right click the Machining folder (or press **CTRL+T**) to generate toolpaths for all enabled operations in the drawing.

By default, generating gcode will write the output from all the enabled parts in the drawing. To create gcode for just one part, right click the part in the drawing tree, then select ***Produce gcode***.

The file **heart-shaped-box.cb**, in the CamBam samples folder illustrates a good use of different parts. Here machining operations are separated into parts for front and back faces for the lid and base of a small wooden box.

Some of the Part properties such as ***Stock*** and ***Tools*** are repeated in the parent ***Machining*** folder. Usually it is best to define these properties at the Machining folder level, so they need only be defined once per drawing. If the Part properties are unspecified, the corresponding value will be used from the machining object. It may be useful to define the properties at the part level if they differ from the global Machining settings, for example if a part uses a different stock definition.

Properties

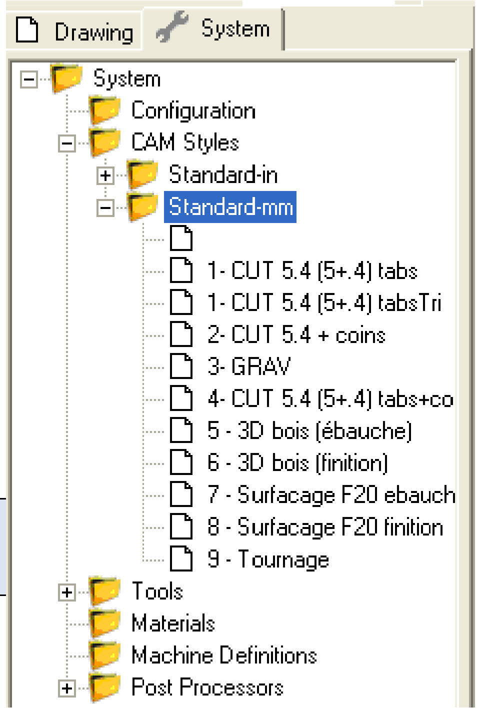
Enabled	If ***Enabled*** is *True*, the (enabled) machining operations in this part will have their toolpaths displayed and they will be included in the gcode output.
Machining Origin	A drawing point that will be used as the machining origin (X=0,Y=0) point when gcode is created. The ellipsis button ... to the right of this property can be used to select a point in the drawing. An 'X' icon will be displayed on the drawing at the machining origin point. This cross can be dragged to a new location using the mouse. ***Note***: MachiningOrigin replaces the GCodeOrigin and GCodeOriginOffset properties of earlier releases.

Name	A descriptive name for the part. This name will be used to generate a filename when creating gcode output from the part.
Nesting	This composite property provides a method of generating an array or nest of parts. ***Nest Method***: Change this to *Grid* or *Iso Grid*, then set the ***Rows*** and ***Columns*** values to determine the number of copies of each part. The ***Spacing*** value will control the distance between each copy. When the toolpaths are generated, an outline should be displayed to indicate the location of each copy. The centre of each outline contains a triangular icon. Clicking and dragging this icon will change the nesting pattern and will also change the nesting method to *Manual*. ***Grid Order*** Controls the direction of the grid layout. For example *Right Up* will make copies to the right of the original, then move up to the next row. ***Grid Alternate*** If set to *True*, the grid will alternate the direction of each row or column (depending on ***Grid Order***). If *False* then each row or column will proceed in the same order with a rapid back to the start of each. ***Nest Method*** = *Point List* The location of each nest copy is taken from a point list drawing object which is set in the ***Point List ID*** property. A new ***Nest to point list*** Part context menu function has been added, in this way a list of nest points can effectively be copied from one part to another by sharing a common point list. ***GCode Order*** Controls how the nested machining operations are ordered in the gcode output. • *Auto* - All consecutive MOPs within the part with the same toolnumber will be posted then repeated for each nest copy, before moving to the next MOP (which would require a tool change). • *Nest Each MOP* - Each MOP is output at each nest location before moving to the next MOP. • *All MOPs Per Copy* - All the MOPs in the part are posted before moving to the next nest location. Multiple copies of the part's toolpaths will be written to the gcode output. This will increase the gcode file size, but does avoid some of the issues encountered when using subroutines. More detail is available on the Nesting documentation page.

Out File	This is the location of the destination gcode file. Clicking the button to the right of this property will open a file browser.
Stock	The stock object is used to define the dimensions of a block of material from which the part will be cut. The properties of the stock object can be used to automatically determine some machining properties. • If a machining operation or style's ***Stock Surface*** property is set to Auto, the stock's stock surface value will be used. • If a machining operation or style's ***Target Depth*** property is set to Auto, the stock's stock surface and Z size will be used to determine the target depth, so a machining operation will by default machine all the way through the stock. Stock properties: ***Material***: Informational text that describes the stock material. ***Stock Offset***: X and Y offset of the lower left corner of the stock block. For example, a stock offset of -10,-20 would position the stock 10 units to the left of the Y axis (X=0) and 20 units below the X axis (Y=0). ***Stock Surface***: The Z location of the top of the stock block. ***Stock Size***: The X, Y and Z dimensions of the stock block. ***Color***: Color to use when displaying this stock object. Stock is undefined if the X,Y and Z sizes are all zero. Stock can be defined at the part or machining level. Stock defined at the part level will override and machining level stock definitions and will be used for all operations within the part. The stock object dimensions can also be passed to simulators such as CutViewer when post processors with appropriate stock macros are included, such as the *Mach3-CutViewer* post processor.
Style	Select a default ***CAM Style*** for this part. All machining operations in the part will use this style unless set otherwise in the machining operation's Style property.
Style Library	This property is used to locate the style definitions used in the Part or machining operations.

Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.
Tool Library	If left blank, the default tool library will be used (Default-{$Units}), otherwise the specified library will be used when looking up tool numbers.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill* | *BullNose* | *BallNose* | *Vcutter* | *Drill* | *Lathe*

CAM Styles



CAM Styles are a way of grouping machining operation parameters into reusable objects to help simplify common machining tasks.

Each machining operation has a ***Style*** property. This refers to a style definition, stored in a system library, which is available to all drawings. The value of machining operation properties that are marked as *Default*, will be taken from the style associated with the operation. In this way, any changes to a CAM style will immediately affect all operations that refer to it.

If no style is selected, a default style will be selected automatically.

> ***Note***: Styles replace a system of Templates that were used in previous versions of CamBam and provided a similar purpose.

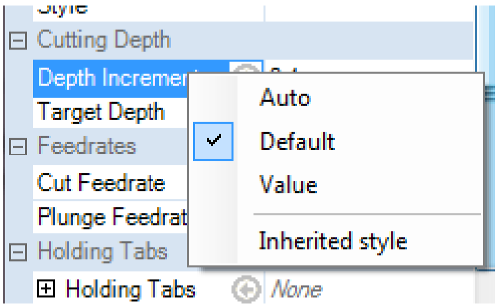
Default, Value and Auto properties

Machining operation and Style properties can have multiple states, indicated by the icon to the right of the property name.

Default - The value of the property will be taken from the CAM style associated with the machining operation. Default property values will be displayed in grey italics and will show the default value that will be used.

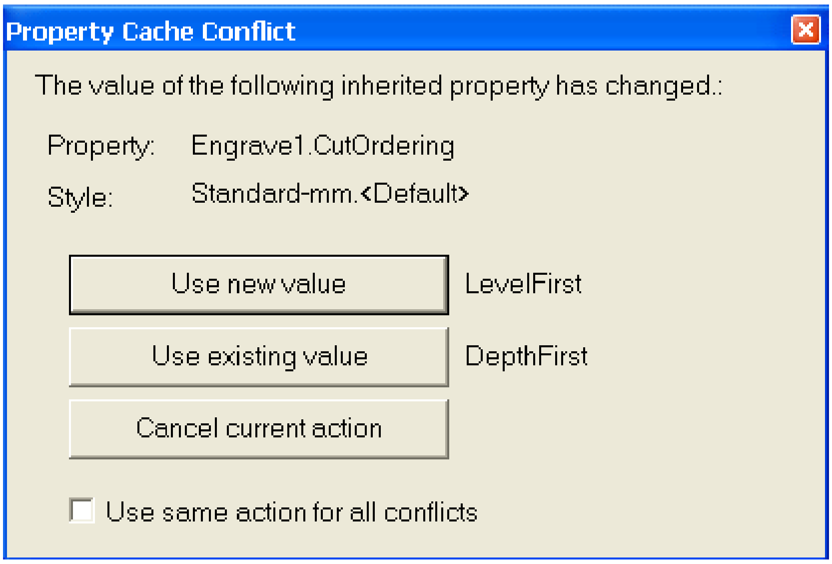
Auto - Indicates the property value is to be calculated internally by CamBam, often based on other settings.
For example if the ***Target Depth*** property is set to *Auto*, the depth will be calculated to cut the full thickness of the stock object.

Value - The property value has been entered explicitly. In this way, the property is overridden from the value stored in its parent style.



Clicking the value icon to the right of the property name, or right clicking the property, will display a menu where the property value state can be selected. This context menu also contains an ***Inherited style*** command. ***Inherited style*** invokes a message box showing where the value of the selected property will be taken from.

Property cache conflict alert



If the result of a *Default* property has changed from the previous value used, a ***Property Cache Conflict*** message may be shown. This may occur if the value stored in the parent style has changed, or if the style uses an *Auto* value and parameters that affect the result of the automatic calculation have changed.

The warning message provides the following options:

Use new value - the newly calculated default value will be used.

Use existing value - the old value will continue to be used. This will change the property from a *Default* to an explicit Value.

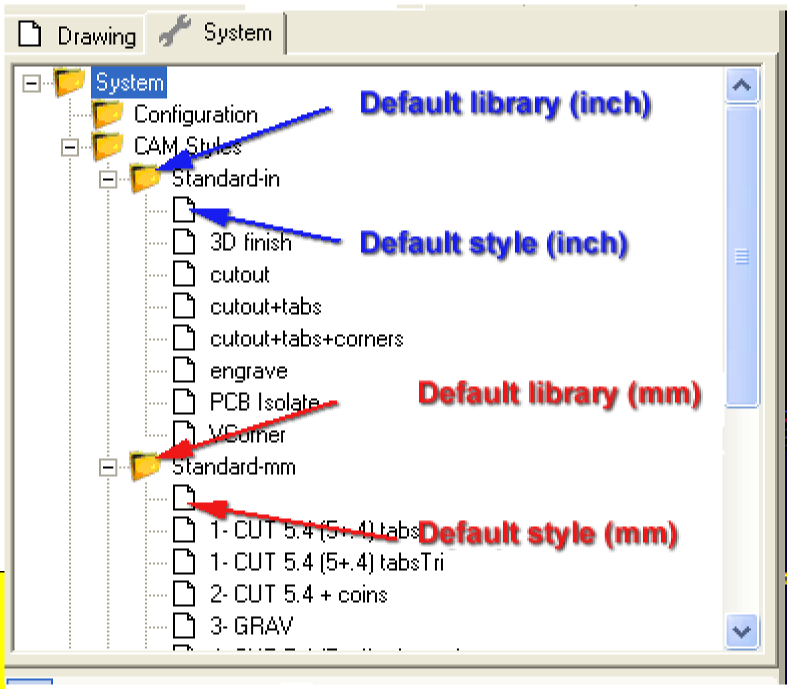
Cancel current action - the old value will continue to be used and left as a *Default*, but the current action will be canceled.

If **Use same action for all conflicts** is checked, the same response will be used whenever a new conflict is detected. This remains in effect until the file is closed. The next time the file is opened, changed default properties will once again be reported.

The cache conflict alert was added to prevent inadvertent changes to a drawing resulting from changing a style or other dependent system library. In this way, if a drawing is transferred to another computer or sent to another person, it is not necessary to also provide the dependent style definitions, as all the required information will be preserved within the file.

Machine operation, Part and Machining CAM styles

CAM Styles can be specified at the ***Part*** level and in the top level ***Machining*** folder.



If the ***Style*** property is left blank for a machining operation, the Style specified in the containing ***Part*** object will be used. If the ***Part*** does not have a defined Style, the Style set against the ***Machining*** object will be used. In the event that no style is defined at any of these levels, the default style will be used for the source of the *Default* values.

The default style is the style with an empty name in the style library.

Warning: The default styles are very important for CamBam to function correctly and should not be renamed or removed.

Style definitions and style libraries

Style definitions can be maintained from the ***CAM Styles*** section of the ***System*** tab.

CAM Styles contain a ***Parent Style*** property, so that styles can be based upon other styles.
If the parent style parameter is not set, the default (blank name) top level style will be used to resolve default properties.

Hint: If the properties in the default CAM style are set as close as possible to the values used by the majority of machining operations encountered, then in many cases, extra CAM styles may not need to be defined at all.

Styles are grouped into style libraries. A style library may be used to group parameters for machining particular materials or different drawing units.

The ***Machining*** and ***Part*** objects contain a ***Style Library*** property. This can be used to determine the correct style to use when the same Style name is present in multiple libraries.

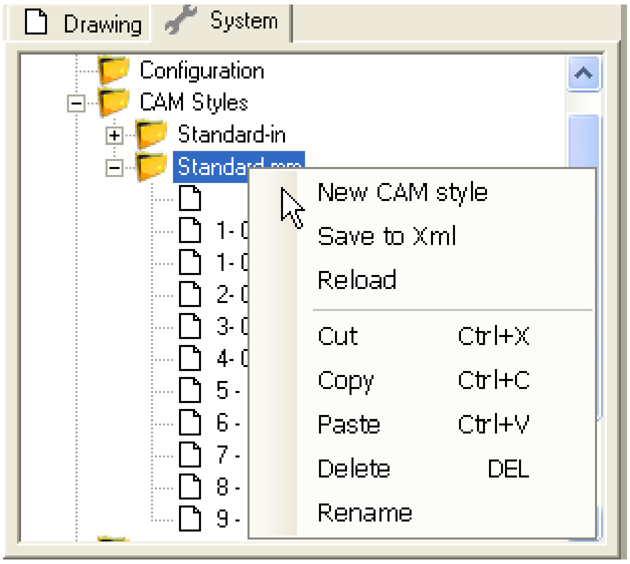
The style library property can contain the following macros:

{$Material} This will be expanded to the name of the material used in the stock object.

{$Units} This will be expanded to the drawing units abbreviation (e.g. `'mm'` for Millimeters and `'in'` for Inches).

If no style library is specified, libraries will be searched in the following order:

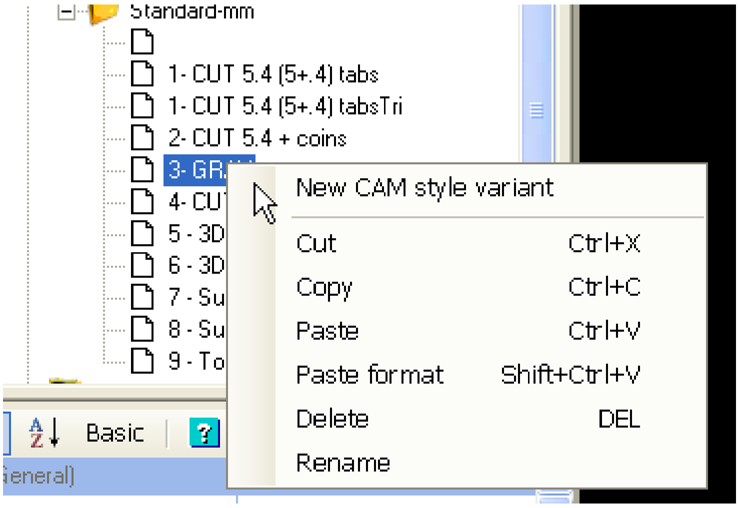
1. {$Material}-{$Units} (if a stock material is defined)
2. Standard-{$Units}



Styles and style libraries can be cut, copied, pasted, deleted and renamed within the System tree. A right click context menu gives access to many of these commands.

Styles can also be moved from one style library to another by clicking and dragging them within the system tree.

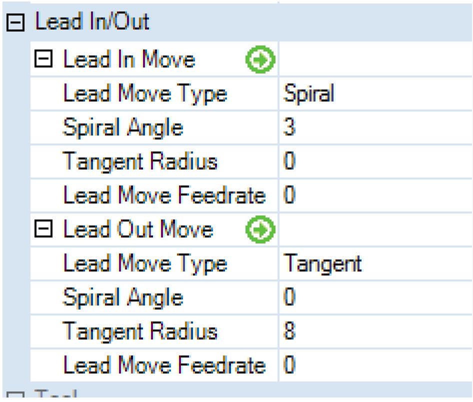
If a style library has been modified externally or by another instance of CamBam, the ***Reload*** operation will load the latest changes into the current program instance.



The context menu shown when right clicking a style also contains a ***New CAM style*** variant option. This will create a new style that inherits its default parameters from the selected 'parent' style.

It is also possible to copy settings from machining operations into a style by copying the machining operation to the clipboard, selecting the system tab, right clicking a destination CAM style then selecting ***Paste format*** from the context menu. This provides a similar functionality to the ***Copy MOP to Template*** operation of previous CamBam versions.

Lead Moves (Lead In and Lead Out)



Many machining operations support lead in and out moves, which control the movement used when entering and exiting a cut.

Lead in moves can be ramped so the cutter will gradually reduce the cutter Z height while simultaneously feeding in X and or Y. This can be crucial when using certain cutters that do not support directly plunging into stock.

There are two main types of lead moves; *Spiral* and *Tangent*. Setting the ***Lead Move Type*** to *None* will prevent the use of a lead move and a direct plunge at the ***Plunge Feedrate*** will be used instead.

The lead move properties also support a ***Lead Move Feedrate*** parameter. If this is 0 then the machining operation's ***Cut Feedrate*** is used, otherwise the feedrate entered into the ***Lead Move Feedrate*** parameter is used.

Spiral Lead

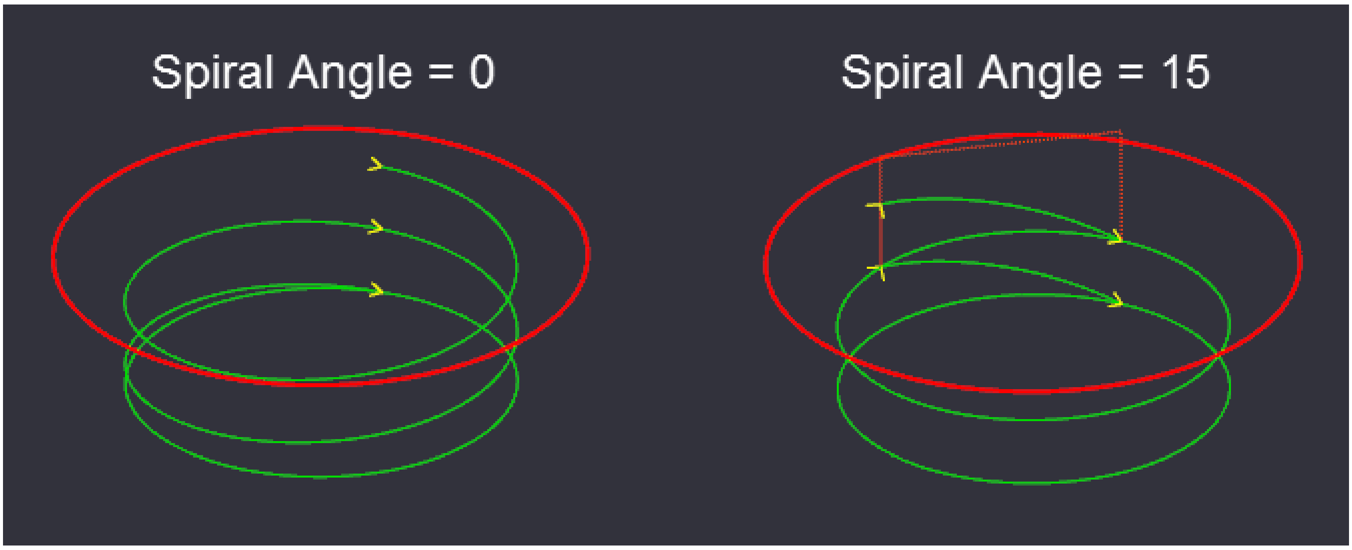
The entry move will follow the path of the underlying toolpath in X and Y, while decreasing the cutter Z from the previous stock level, down to the next target depth.

The angle of the spiral ramp is defined in the ***Spiral Angle*** property. If an angle is specified, once the target depth is reached, a complete pass of the toolpath is then made at a constant Z depth. A lead in move will be used at each depth increment. This may make it necessary for the cutter to lift up to the clearance plane, then plunge to the start of the next lead move.

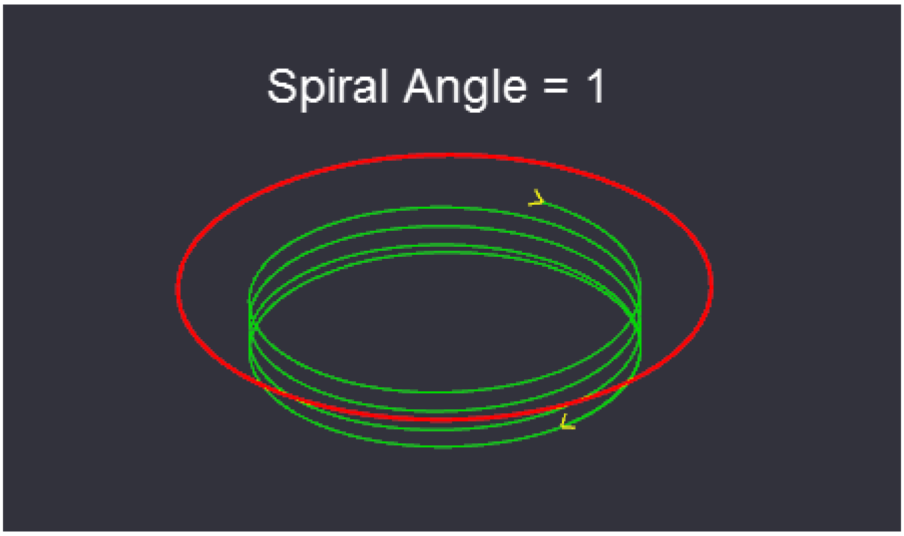
Hint: The plunge down to each depth increment can slow machining times considerably. To reduce this, a ***Fast Plunge Height*** value can be set in the ***Machining*** options. This allows a rapid move to be used down to the next cut level.

If ***Spiral Angle*** is set to 0, an angle is calculated so that the ramp will complete one depth increment along one pass of the target toolpath, in a continuous feed move.
For closed shapes, the lead move will then replace the toolpath at each depth level, with just a single constant Z toolpath inserted at the final target depth to ensure a level base to the cut.
For example; if the source shape used was a circle, the resulting toolpath would be a continuous spiral, with each loop cutting one depth increment, followed by a circular cut at the target depth.

The following images compare a spiral lead in move with an explicit 15 degree ramp angle and a spiral with a 0 degree angle where the ramp angle is then automatically calculated.



If a very shallow spiral angle is used, it may be necessary for the lead move to complete a number of circuits of the toolpath before reaching the target depth, as shown in the following image, where a 1 degree angle is used.

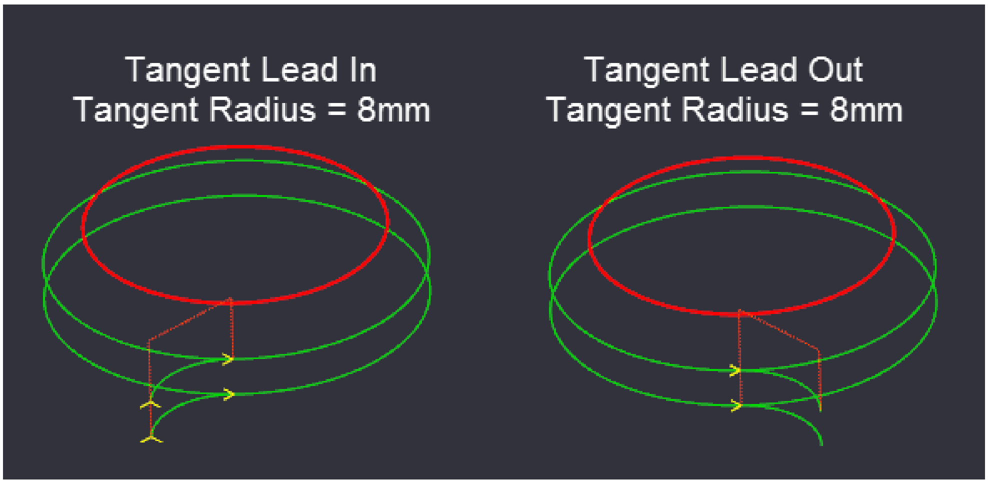


Tangent

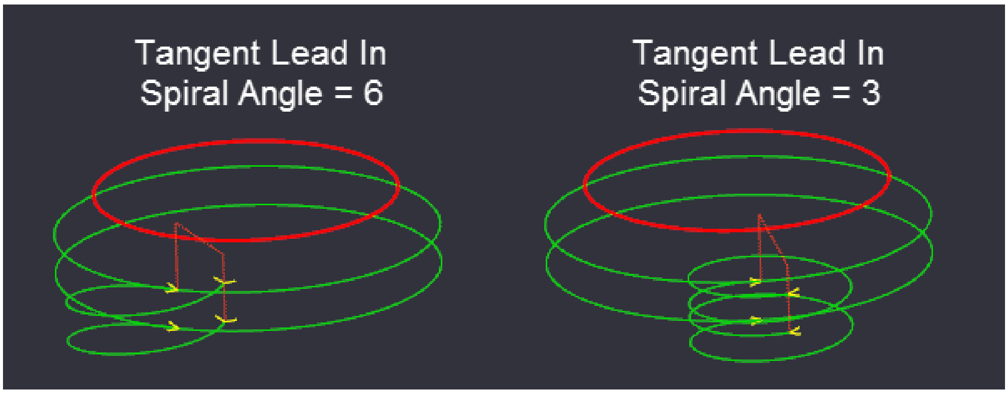
The tangent lead move will use a circular arc move to enter or exit the stock, meeting the target toolpath start point at a tangent.

As well as setting the ***Lead Move Type*** to Tangent, the ***Tangent Radius*** property must also be defined.

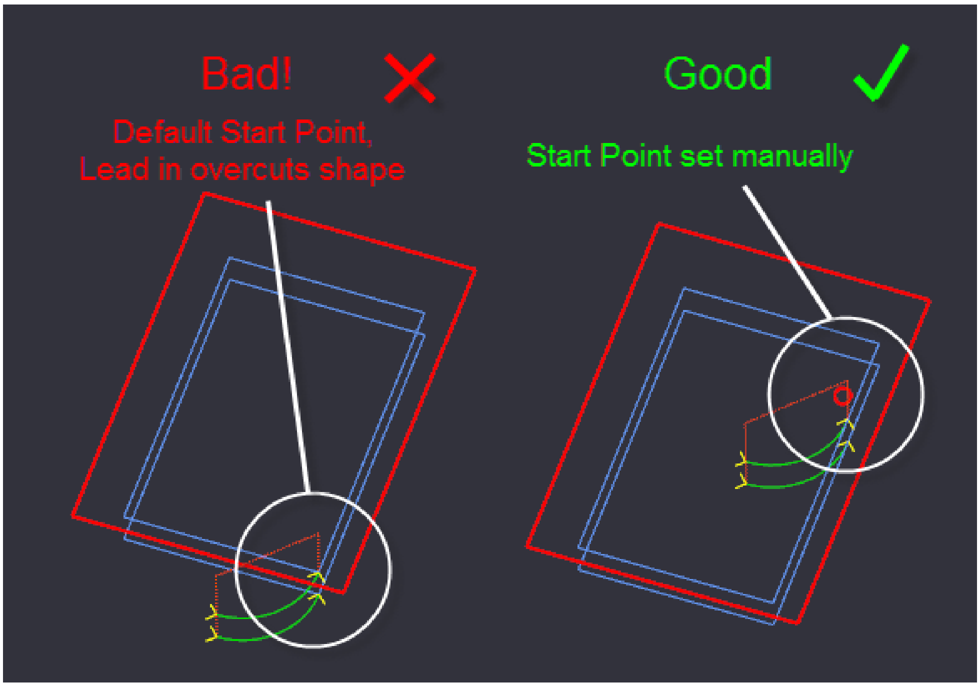
Tangent moves are particularly useful for lead out moves, to avoid tool marks as the cutter moves away from the stock.



Tangent moves will also make use of the ***Spiral Angle*** parameter, where the arc move will also plunge in the Z direction to form a circular spiral or spiral segment. As with spiral lead moves, if the spiral angle is shallow, multiple loops may be needed for the lead move to reach the target depth.



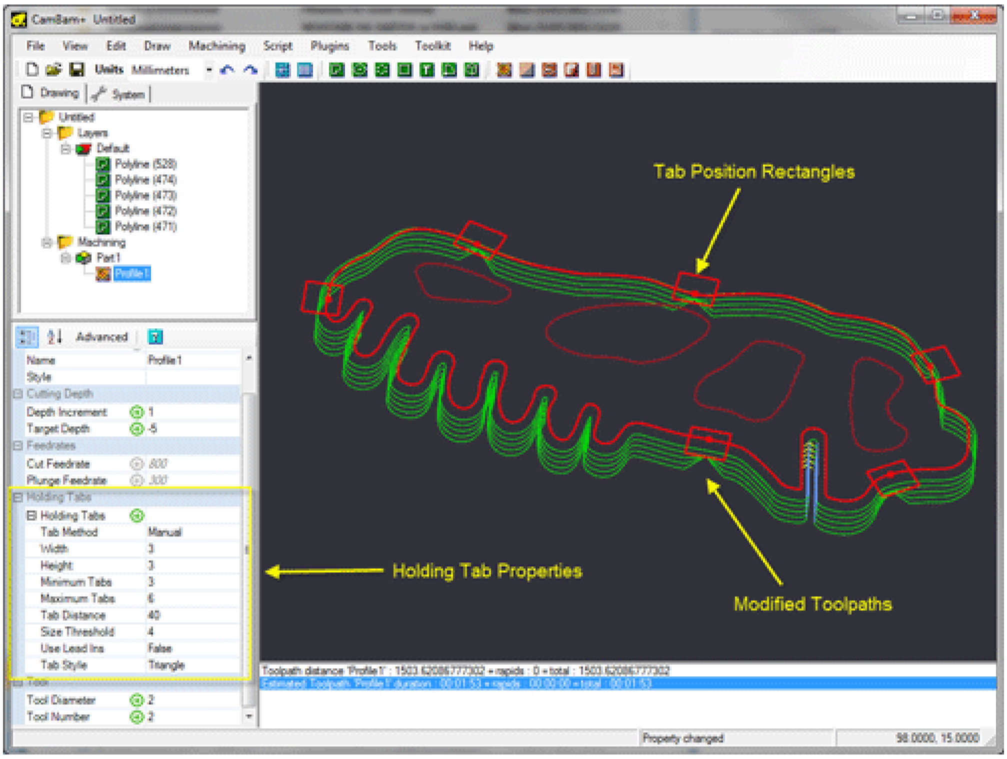
In some cases, such as an inside profile cut with internal corners, the default toolpath start point may lead to problems when using tangent lead moves. In these cases, the machining start point should be modified to move it to a more sensible location, away from inside corners.



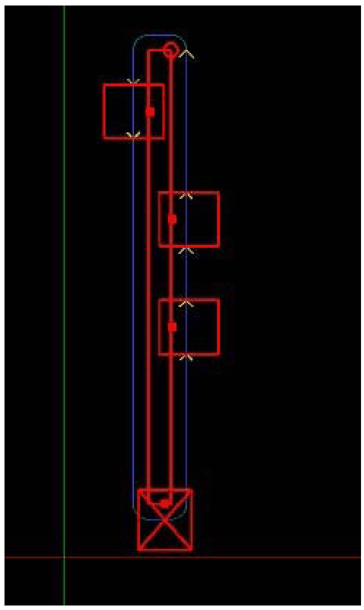
Holding Tabs

Holding tabs or bridges are used to hold material in place when cutting through the entire thickness of stock. They are formed by breaking or bending the toolpaths at the lower depths of the cut, to leave areas of stock intact.

The ***Profile machining operation*** contains a ***Holding Tabs*** composite property. Click on the + sign to the left of this to expand the property and modify the sub properties.



The quickest way to enable holding tabs is to select the profile machining operation in the drawing tree. Then right click the drawing window to open the drawing context menu. At the bottom of the context menu, a ***Holding Tabs*** sub menu is displayed. From here, select ***Autocalc***. This is similar to setting the holding tab ***Tab Method*** property to *Automatic* and rebuilding the toolpaths.



The holding tabs will be displayed as a series of rectangles spaced around the source drawing shapes. If the automatically generated holding tabs are in inconvenient positions, they can be quickly moved by clicking and dragging them to an alternate position. This will also change the ***Tab Method*** to *Manual*.

If a tab is displayed with a red cross marker through it, this indicates a holding tab that cannot be applied to any toolpaths. This is often caused when holding tabs are positioned on the corners of shapes. In these cases, manually adjusting the tab position will resolve the problem. The X marker will not be cleared until the toolpaths are regenerated.

When a profile machining operation is selected, the drawing context menu can also be used to ***Add*** and ***Remove*** holding tabs. When removing tabs, right click the mouse within the rectangle of the holding tab to remove. Similarly, when adding tabs, first right click on the source shape where the new shape should be located, then select ***Add Tab*** from the resulting Holding Tabs context menu.

The number and spacing of the automatically generated holding tabs is controlled by the ***Tab Distance*** parameter as well as the ***Minimum*** and ***Maximum Tabs*** properties. For example, if the perimeter of an object is 160mm and a tab distance of 30mm is used, the nearest whole number to 160/30 ie 5 holding tabs will be considered. If however this number is greater than the ***Maximum Tabs*** property, the maximum tabs number will be used instead. Similarly, if the automatic number of tabs is less than the ***Minimum Tabs*** property, the minimum number will be inserted.

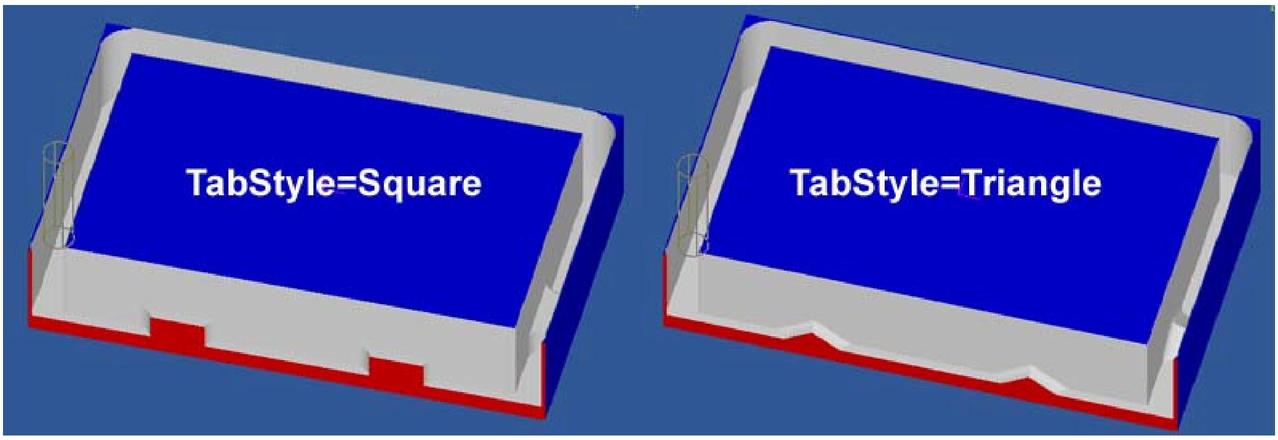
If ***Tab Distance*** is set to 0, the minimum number of tabs is always presumed.

The ***Size Threshold*** is used such that if a source shape's perimeter is smaller that this value, no holding tabs will be inserted for that shape.

The size of the holding tabs is controlled by the ***Height*** and ***Width*** holding tab properties. Height is taken to be as measured from the target depth of the profile, to the top of the desired holding tab. The width will be the width as measured at the thinnest part of the holding tab. The rectangles used to display tabs and the resulting gaps in the toolpaths will appear wider than this width setting. This is to compensate for the tool diameter.

Experience will dictate the optimum tab height and widths. Too large tabs will hold parts securely but require extra manual cleanup to remove the tab stock. Too small tabs run the risk of the parts breaking free, which can damage both parts and cutting tools. The type of material will also affect this choice. Metals typically can use smaller holding tabs while woods and plastics will need wider or thicker tabs to compensate for the brittle nature of the material.

There are two types of holding tab cross-section shapes available which is defined in the ***Tab Style*** property: *Square* and *Triangle*. Triangles are a good all-round tab shape, easy to clean up and provide a degree of ramping back down into the stock. Squares can be stronger and can also be used with lead in moves when ***Use Lead Ins*** is set true. This is useful for holding tabs in harder materials.

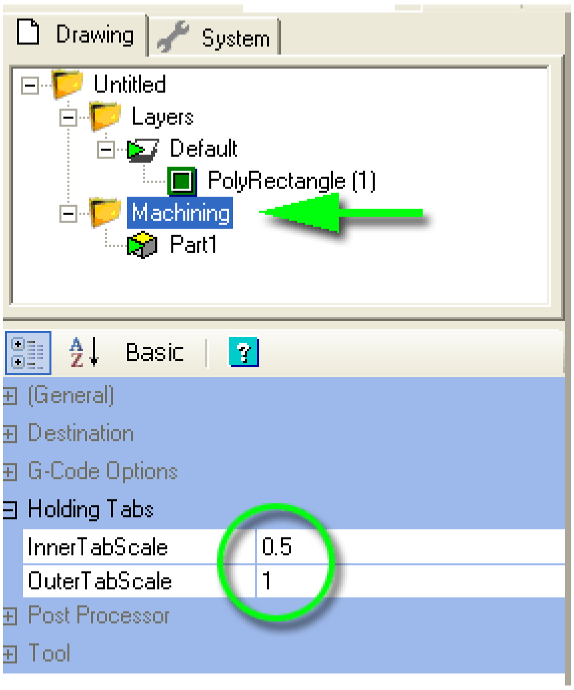


Properties

Height	The height of the holding tab measured from the stock base or target depth.
Maximum Tabs	The maximum number of auto tabs to insert around each shape.
Minimum Tabs	The minimum number of auto tabs to insert around each shape.
Size Threshold	Shapes with perimeters less than this value will not have any automatically calculated tabs.
Tab Distance	The approximate distance between each automatically generated holding tab.
Tab Method	*None* - No holding tabs will be inserted. *Automatic* - Tab positions will be automatically determined. *Manual* - Tab positions have been modified or set manually. *Automatic (Outer | Inner)* - Similar to Automatic, except tabs will only be added to the outside (or inside) shapes of regions.
Tab Style	*Square* - Square cross section tabs. *Triangle* - Triangle cross section tabs. *Skip* - Similar to the Square style except a rapid move will be used across the top of the tab. This method is commonly used in plasma cutting to insert a holding tab or bridge without having to turn off the plasma.
Use Lead Ins	Square holding tabs will result in a vertical plunge on the trailing edge. This can be hard on cutters, especially in harder materials. If ***Use Lead Ins*** is set *True*, an extra lead in move (as defined in the profile's ***Lead In Move*** property) is inserted at the trailing edge.
Width	The final width of the holding tab, measured at the thinnest part of the tab.

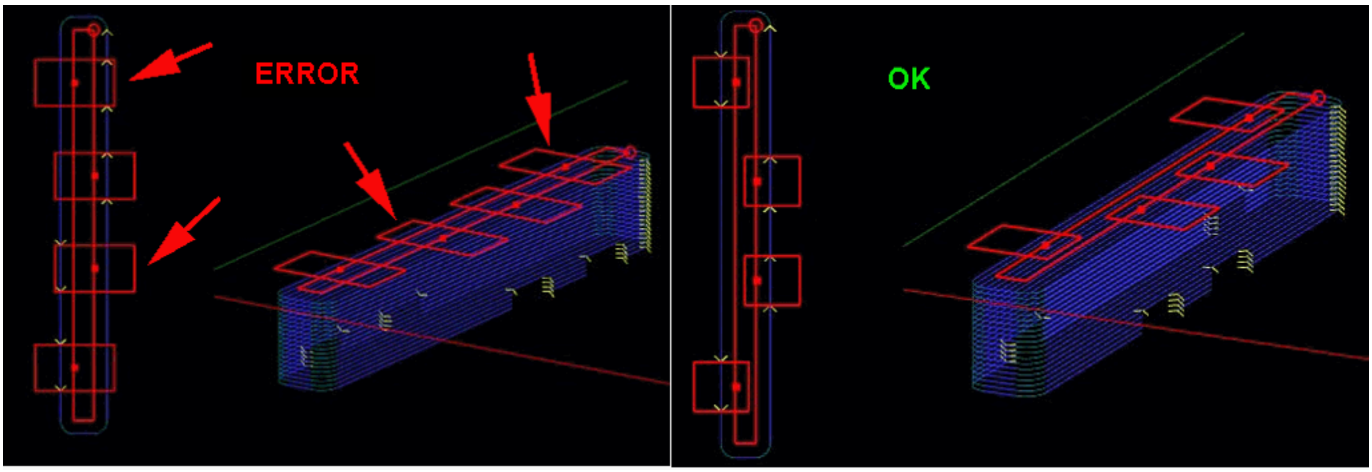
Advanced Settings

In some cases, such as very narrow source shapes, a problem can occur where the shape of the holding tab may extend to the toolpath on the other side of the part and holding tabs incorrectly assigned to the wrong side.



To help resolve this problem, two parameters are available in the ***Machining*** properties of the drawing: ***Inner Tab Scale*** and ***Outer Tab Scale***. These are used to extend or contract the size of the tab 'rectangle'. The inner tab scale will alter the size of the tab rectangle that extends towards the source shape. The outer tab scale will affect the tab size away from the stock shape.

The following image show a narrow source shape that has caused an incorrect holding tab. Reducing the inner tab scale resolves this problem.



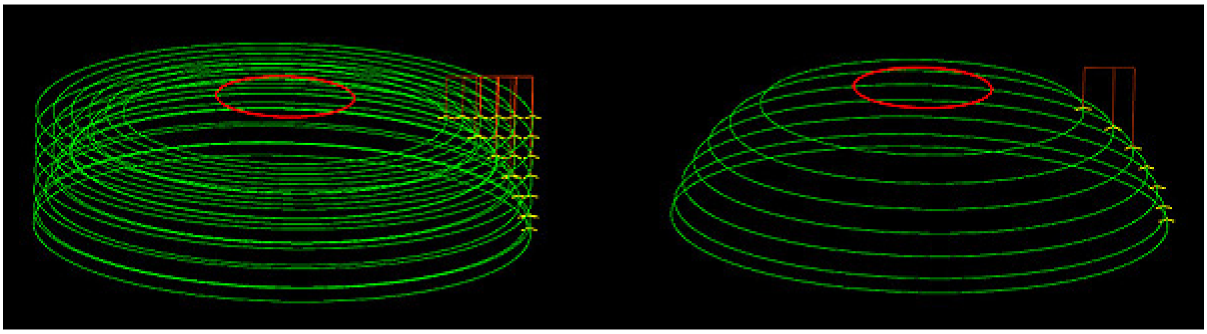
Side Profiles

Side profiles are a method of producing 3D contours from 2D shapes, by creating radii and slopes. Side profiles are created by manipulating the ***Side Profile*** composite property of the ***2D Profile*** machining operation.

The files side **profiles.cb** and **heart-shaped-box.cb**, in the CamBam samples folder illustrate various uses of side profile operations.

Properties

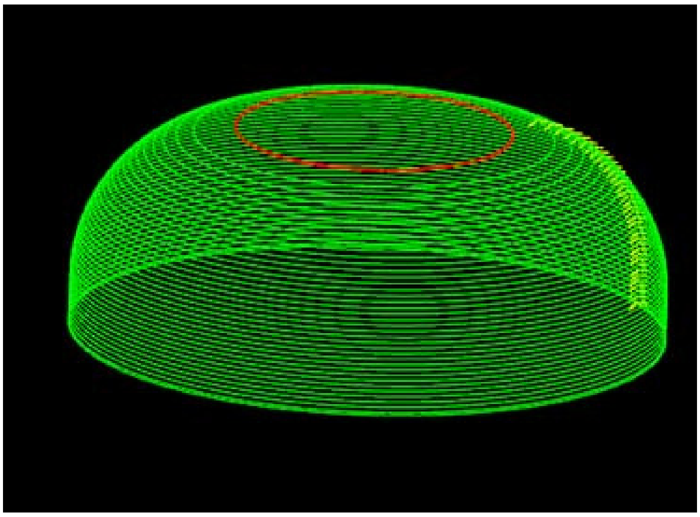
Method	*None* - Normal perpendicular sides. *Slope* - ***Value*** contains the angle in degrees from vertical of the slope (or bevel). *Convex Radius* - ***Value*** contains the radius of the convex contour. *Concave Radius* - ***Value*** contains the radius of the concave contour.
Value	A value that controls the selected side profile method.
Adjust Cut Width	When *False*, the toolpaths will just follow the calculated profile. This is fine for a finishing pass, but is not suitable for clearing stock. Set ***Adjust Cut Width*** = *True* to machine all the stock layers above as well as on the profile. This is useful for roughing operations.

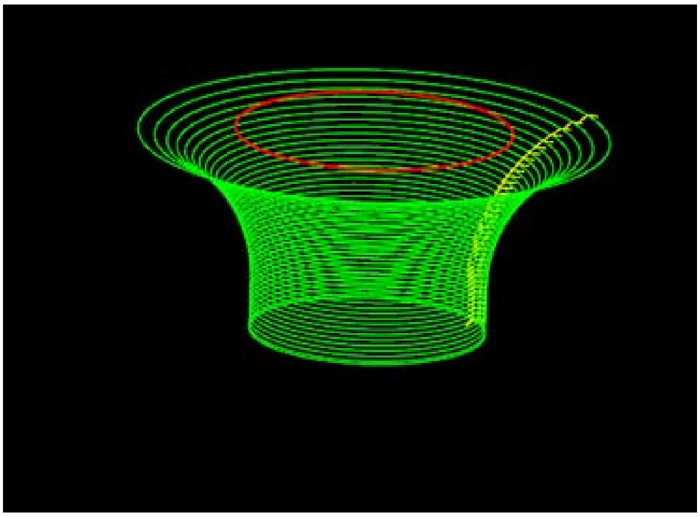


Adjust Cut Width = *True* ***Adjust Cut Width*** = *False*

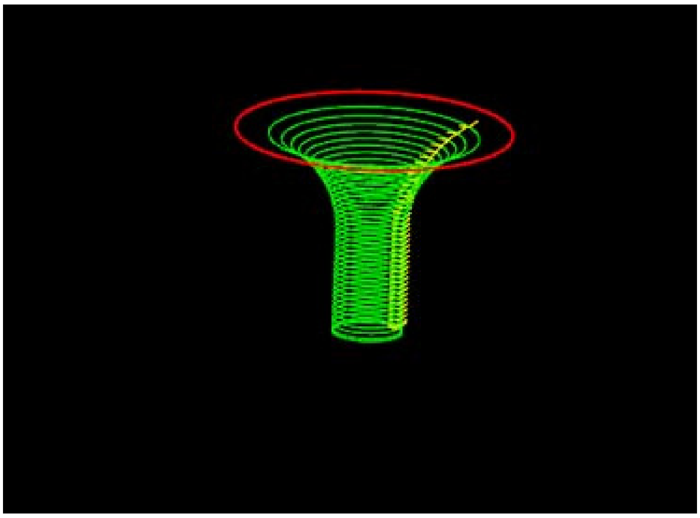
The sign of the ***Value*** parameter is significant and reversing the sign will result in different effects.

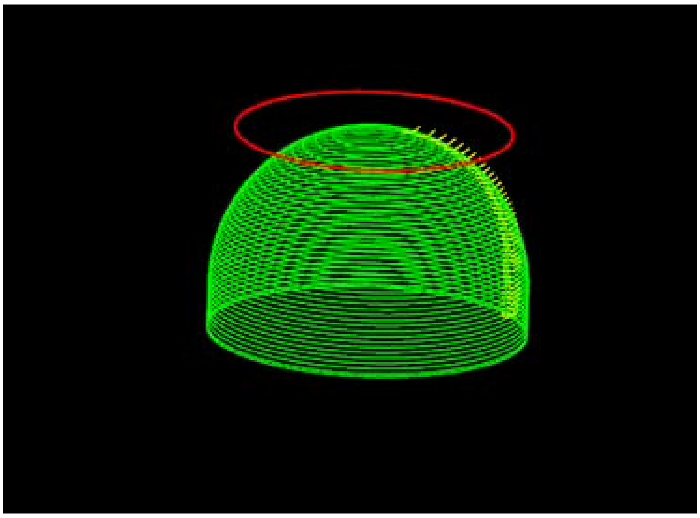
Below are some examples of various combinations of side profile methods, value signs and profile ***inside/outside*** settings.
These images were created from the side **profiles.cb** sample file.



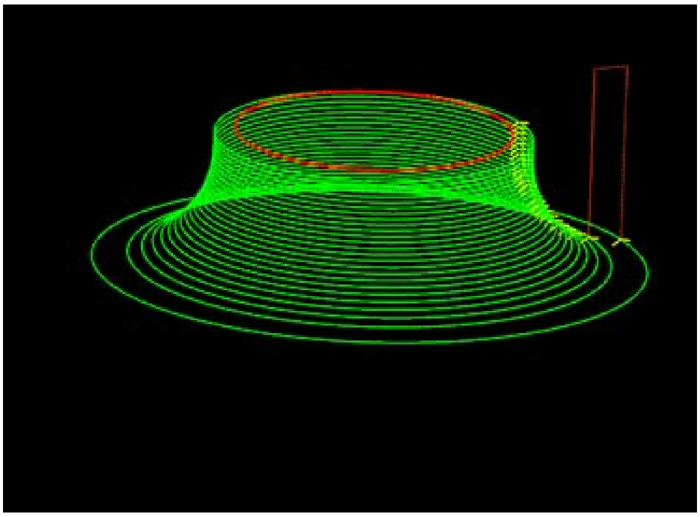


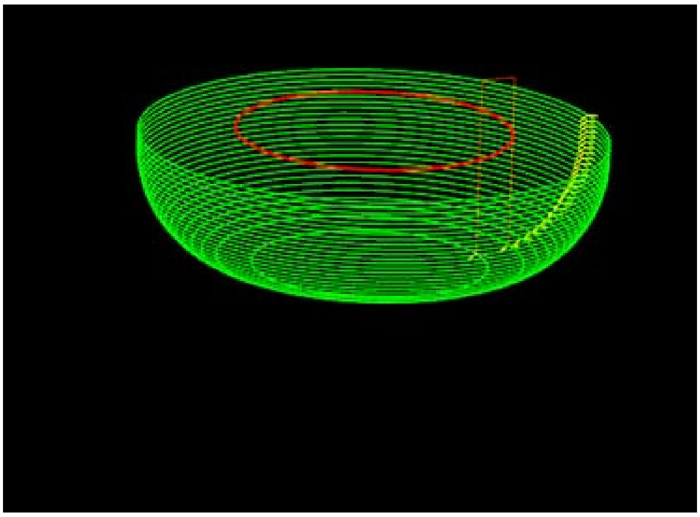
Method=ConvexRadius, Value=+Ve, Profile=Outside Method=ConvexRadius, Value=-Ve, Profile=Outside



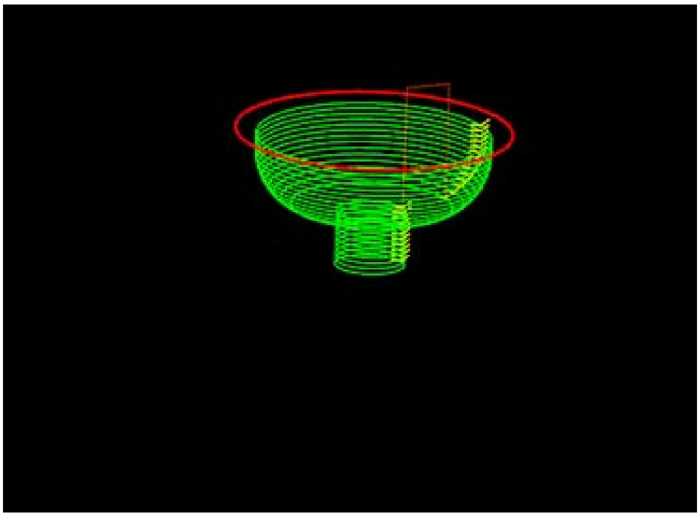


Method=ConvexRadius, Value=+Ve, Profile=Inside Method=ConvexRadius, Value=-Ve, Profile=Inside

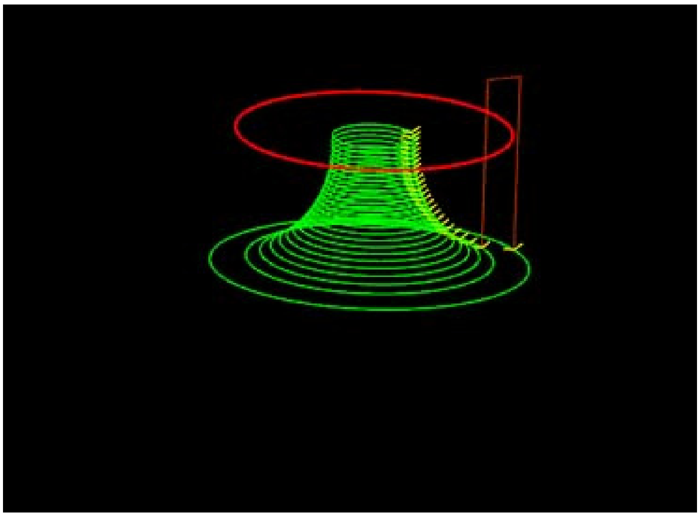




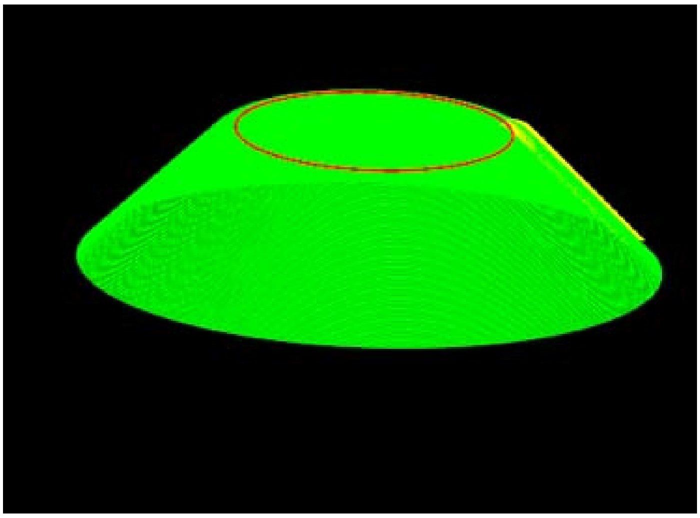
Method=ConcaveRadius, Value=+Ve, Profile=Outside Method=ConcaveRadius, Value=-Ve, Profile=Outside



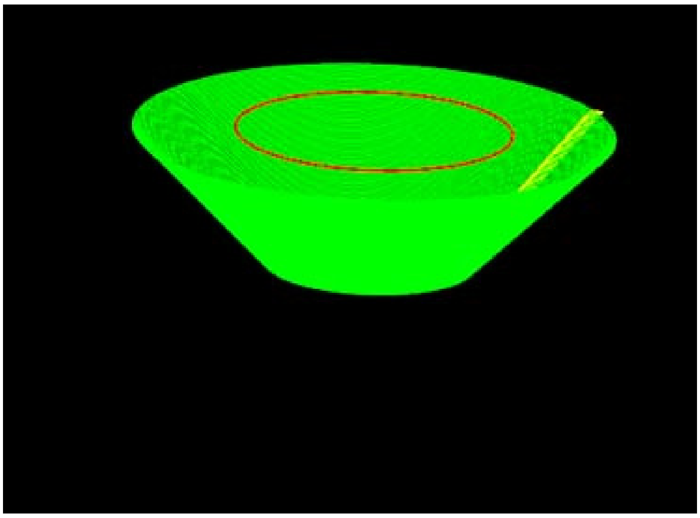
Method=ConcaveRadius, Value=+Ve, Profile=Inside



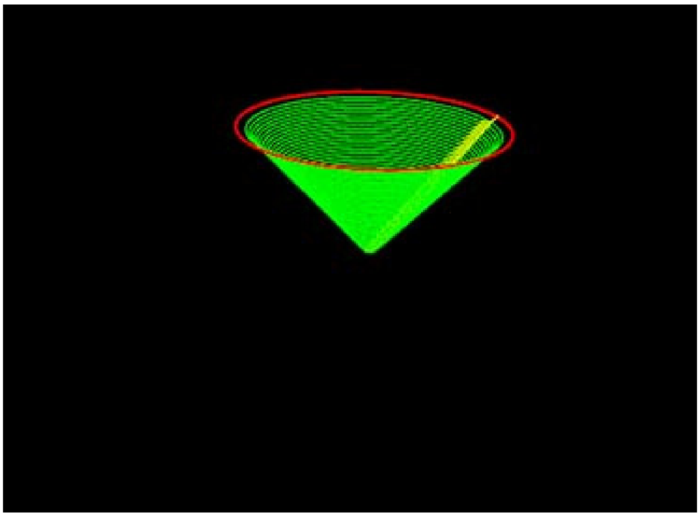
Method=ConcaveRadius, Value=-Ve, Profile=Inside



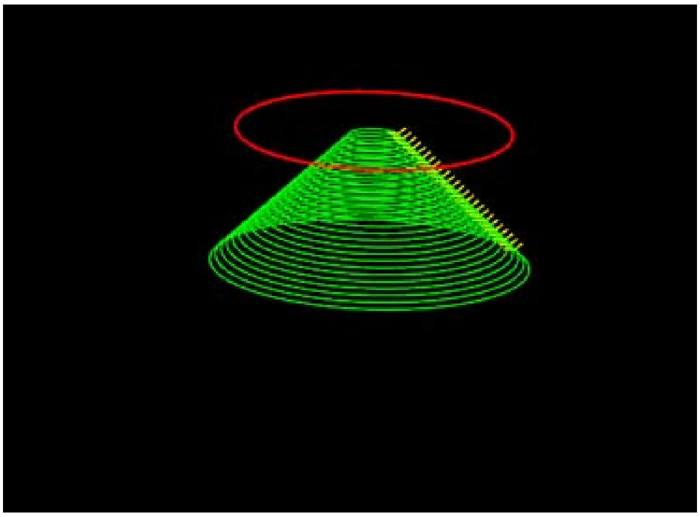
Method=Slope, Value=+Ve, Profile=Outside



Method=Slope, Value=-Ve, Profile=Outside



Method=Slope, Value=+Ve, Profile=Inside



Method=Slope, Value=-Ve, Profile=Inside

Post Processor System

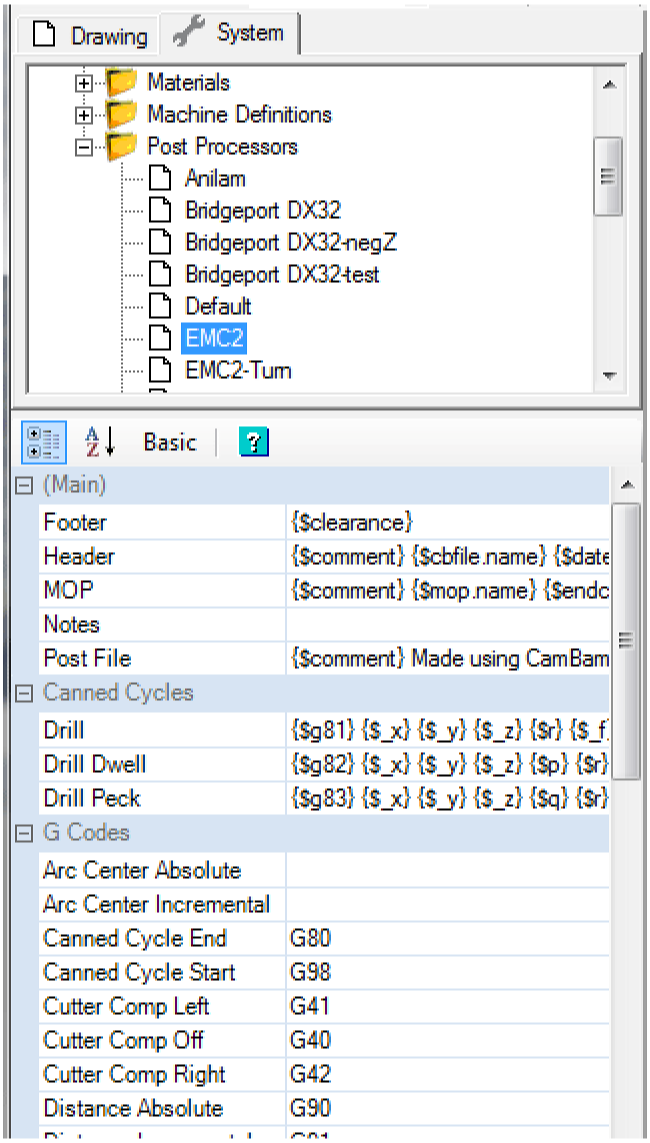
The format of generated gcode files can be controlled using post processor definitions. These definitions can be created, copied and modified within the Post Processors section of the System tab.

The post processor used for a specific drawing is set under the machining options. Select the machining folder in the drawing tree and look in the Post Processor group of the machining properties. If no post processor is specified, the default post processor will be used.
To set the default post processor, right click the definition in the Post Processors section of the System tab, then select Set As Default. The default definition will be marked with a green arrow.

Machining Properties

Post Processor	This option is a drop down list that contains all the custom post processors defined in the system folder. Leave this blank to use the default post processor.
Post Processor Macros	This option is used to pass user defined macros from the drawing to the post processor. This is a multi-line text field containing multiple macro definitions in the format $macro=value. Example: $o=1234 $stock_height=0.4

Post Processor Management



The list of available post processors is accessed from the ***Post Processor*** folder of the system tab. Here, post processor definitions can be created, modified, copied, renamed and deleted.

New post processors can be created via the context menu visible when right clicking the post processor folder. Alternatively, existing definitions can be copied, pasted then modified. This is a good way of creating variations of a working post processor.

If post processor files are modified or new ones created outside of CamBam or in another CamBam instance, the post processor list should be refreshed using the ***Tools - Reload Post Processors*** menu option.

Post processors are XML files with a .cbpp file extension, stored in the \post sub folder of the system folder.

Install a post processor downloaded from the Internet

1. If necessary, unzip the file to obtain a ***xxx.cbpp*** file
2. In CamBam, menu ***Tools/Browse System Folder*** to open the location of the “system” of CamBam.
3. Copy your .cbpp file into the ***post*** folder of the “system” of camBam
4. Use the ***Tools/Reload Post Processors*** menu to add it to the list of available post processors (or restart CamBam)

Post Processor sections

The post processor definition contains a number of sections. Each section can contain a mixture of literal text, which is output to the destination gcode file directly, and text macros of the format {$macro}. The macro definitions are defined within other sections of the post processor, or by defining user macros in the ***Post Processor Macros*** property of drawing’s machining options. The macros are evaluated and the resulting text values are written to the gcode output.

Note: If any of the following sections are not visible in the property editor, make sure the Advanced property view button is selected.

(Main) - Post File

This section defines the general structure of the gcode file. It typically includes three macros that are evaluated internally based on rules defined in further sections of the post processor.

{$header} - This macro is evaluated using the ***Header*** section described below.
{$mops} - This macro is evaluated to a list of blocks of text, one block per each machining operation. Each block is formatted using the rules in the ***MOP*** section.
{$footer} - This macro is evaluated using the ***Footer*** section described below.

Example:

```
%
O{$o}
( MY FANUC POST )
{$header}
G0 X10Y10Z0
{$mops}
{$footer}
%
```

Note: The value of {$o} macro is passed to the post processor using the drawing's ***Post Processor Macros*** property which may contain a value such as '$o=1234'.

The % characters are output literally and would be omitted if not using an RS232 file transfer program.

(Main) - Header

Defines the text rules used by the {$header} macro.

Example:

```
{$comment} {$cbfile.name} {$date} {$endcomment}
{$tooltable}
{$comment} CUTVIEWER {$endcomment}
{$comment} FROM/0,0,5 {$endcomment}
{$comment} TOOL/MILL,1,0,20.0,0 {$endcomment}
{$comment}STOCK/BLOCK,{$stock_width},{$stock_length},
    {$stock_height},{$stock_x},{$stock_y},{$stock_z} {$endcomment}
{$cbfile.header}
{$units} {$distancemode} {$velocitymode} {$cuttercomp(off)}
{$toolchange(first)}
G54 ( Use fixture 1 )
{$clearance}
```

Once again, the ***Post Processor Macros*** property is used to pass the {$stock_...} macros to the post processor, which in this example may contain text such as:

```
$stock_length=150
$stock_width=150
$stock_height=12.7
$stock_x=75
$stock_y=75
$stock_z=12.7
```

(Main) - Footer

Defines the text rules used by the {$footer} macro.

Example:

```
{$clearance}
G28 G91 Z0
G90 G53 X-15.0 Y0.0
M09
{$spindle(off)}
{$endrewind}
```

(Main) - MOP

Defines how each item of the {$mops} macro is formatted. This information will be repeated in the gcode output for each active machining operation.

Example:

```
{$comment} {$mop.name} {$endcomment}
{$toolchange}
{$velocitymode} {$workplane}
{$mop.header}
{$spindle} {$s}
{$blocks}
{$mop.footer}
```

(Main) Start Cut

Macro to use when about to feed cut. This may be used for plasma or laser cutters to power on the cutting tool.

The start of cutting is determined when a feed move is detected where Z is below the stock surface.

(Main) End Cut

Macro to use when finished a feed cut. This may be used for plasma or laser cutters to power off the cutting tool.

The end of cutting is determined when a rapid is detected or a feed move where Z is at or above the stock surface.

For example, to power off a laser to avoid holding tabs, use *square* holding tabs and set the holding tab height so that the top part of the tab move is above the stock surface. The ***Start Cut*** macro will then be invoked when the feed move resumes below the stock surface.

(Main) Post Processor Macros

This property can be used to set default values for any custom macros used in the post processor.

Custom macro values will be overridden by the values set in the Post Processor Macros property of the machining options.

Tools - Tool Table Item

Defines how each item of the {$tooltable} macro is produced. Tool tables are typically inserted in the header of a file and contain commented text describing the list of tools that will be used in the gcode file.

Example:

```
{$comment} T{$tool.index} : {$tool.diameter} {$endcomment}
```

Tools - Tool Change

Defines how the {$toolchange} macro is formatted.

Example:

```
{$clearance}
{$comment} T{$tool.index} : {$tool.diameter} {$endcomment}
{$comment} Tool Radius and Taper coming soon {$endcomment}
{$comment} TOOL/MILL, {$tool.diameter}, {$tool.radius},
    {$tool.length}, 0 {$endcomment}
T{$tool.index} M6
```

G Codes - G0, G1, G2, G3, G81, G82, G83

These sections define how the commonly used gcode operators are output.

G Codes - Arc Center Absolute

Used in the {$mop.header} macro to specify that ArcCenterMode is set to Absolute. Mach3 recognizes G90.1

G Codes - Arc Center Incremental

Used in the {$mop.header} macro to specify that ArcCenterMode is set to Absolute. Mach3 recognizes G91.1

G Codes - Canned Cycle Start

Code sequence used at the start of a group of canned cycle blocks. Typically G98 for initial level return after canned cycles.

G Codes - Canned Cycle End

Code sequence used at the end of a group of canned cycle blocks. Typically G80.

G Codes - Cutter Comp Off, Cutter Comp Left, Cutter Comp Right

Used in the {$cuttercomp(off|L|R)} macros. Typically Off=G40, Left=G41, Right=G42.

G Codes - Distance Absolute, Distance Incremental

Typically absolute=G90, incremental=G91. **NOTE!** Incremental distance mode is not currently supported.

G Codes - Units (Inches), Units (Metric)

Typically inches=G20, millimeters=G21.

G Codes - Velocity Mode - ExactStop, Velocity Mode - Constant Velocity

Typically exact stop=G61, constant velocity=G64.

G Codes - Workplane XY, Workplane XZ, Workplane YZ

Typically XY=G17, XZ=G18, YZ=G19.

G Codes - X Mode Diameter

Used in the {$lathexmode} macro to specify that X values are in diameter mode. For example; EMC2 recognizes G7.

G Codes - X Mode Radius

Used in the {$lathexmode} macro to specify that X values are in radius mode. For example; EMC2 recognizes G8.

M Codes - End Rewind

Typically M30.

M Codes - Repeat

Typically M47.

M Codes - Spindle CW, Spindle CCW, Spindle Off

Typically CW=M3, CCW=M4, Off=M5.

M Codes - Stop

Typically M0.

Moves - Rapid, Feed Move, Arc CW, Arc CCW

These sections define how the commonly used gcode move instructions are formatted.

Example:

Rapid

```
{$g0} {$_f} {$_x} {$_y} {$_z} {$_a} {$_b} {$_c}
```

Feed Move

```
{$_g1} {$_f} {$_x} {$_y} {$_z} {$_a} {$_b} {$_c}
```

Arc CW

```
{$g2} {$_f} {$_x} {$_y} {$_z} {$i} {$j}
```

Arc CCW

```
{$g3} {$_f} {$_x} {$_y} {$_z} {$i} {$j}
```

Note: The gcode operators {$g...} and their parameters can specified using an underscore (_) prefix. This is to show values that are modal (or sticky). That is, they will only be output if the current value has changed. Omitting the underscore will force the parameter to be always output.

Canned Cycles - Drill, Drill Dwell, Drill Peck

These sections define how the commonly used canned cycle instructions are formatted.

Drill

```
{$g81} {$_x} {$_y} {$_z} {$_r} {$_f}
```

Drill Dwell

```
{$g82} {$_x} {$_y} {$_z} {$p} {$_r} {$_f}
```

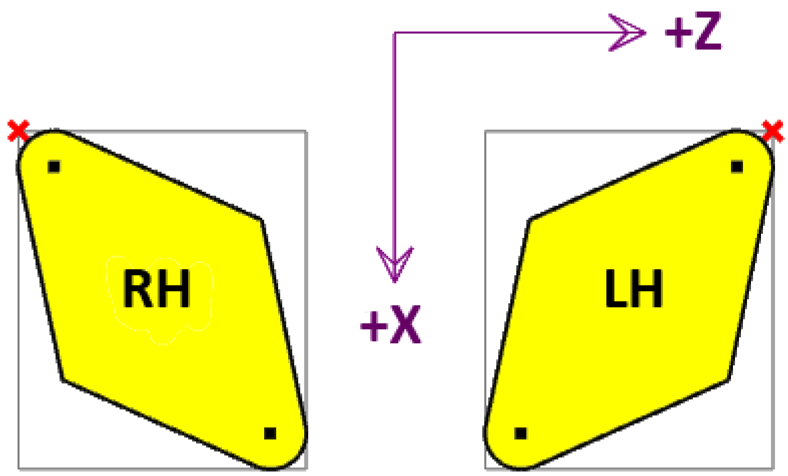
Drill Peck

```
{$g83} {$_x} {$_y} {$_z} {$p} {$_q} {$_r} {$_f}
```

Lathe - Lathe Tool Radius Offset

If *False*, the toolpath at the center of the tool radius is output.

If *True*, an appropriate tool radius offset is applied. The toolpath will be offset by a negative tool radius in the lathe X axis.
The direction of the Z tool radius offset is determined by the cut direction. For right hand cuts the toolpath Z will be offset by a negative tool radius. For left hand cuts, a positive tool radius Z offset is used.



In the diagram above, the red cross represents the toolpath reference point when ***Lathe Tool Radius Offset*** is set *True*. If *False*, the dot at the tool radius center will be the reference point. The reference point is sometimes referred to as the 'Imaginary' or 'Virtual' tool point.

Lathe - Lathe X Mode

For lathe operations, specifies whether X values are radius or diameter mode.

Line Numbering

Line Numbering - Add Line Numbers

If *True* then line numbers will be inserted at the beginning of g-code lines.

Line Numbering - Line Number Format

Controls how the line number values are presented. '0' characters denote a place holder that will contain either a significant digit or a 0. A '#' character will output a significant digit or space character where there is no significant digit at that position.

Line Numbering - Line Number Increment

Line numbers will be incremented by this amount each time a line number is added.

Line Numbering - Line Number Prefix

This text (typically an 'N' character) will be written before the line number value.

Line Numbering - Line Number Skip

Lines where the first non space character is in this list will not receive a line number.

Line Numbering - Line Number Space After

If True then a space character will be inserted after the line number value.

Line Numbering - Line Number Start

The initial for the first line number used.

Options - Arc Output

Controls how arcs are output to gcode.

If *Convert To Lines*, small line moves are used rather than arc commands.
Helix Convert To Lines is similar to *Convert To Lines*, but only for helical arcs (i.e. arcs with varying Z).

Options - Arc Center Mode

This property controls whether the I and J parameters for arc moves (G2, G3) use absolute coordinates or incremental, relative to the arc end points.
If this setting is different to the way the CNC controller interprets arc moves, the resulting toolpath may look a mess of random arcs in the controller.

Default When default is set in the drawing's machining properties, the post processor Arc Center Mode will be used. A default value in the post processor will use Incremental (C-P1).

Absolute I & J are absolute coordinates of the arc center point.

Incremental (C-P1) I & J are coordinates of the arc center, offset from the first arc point. This is the typical incremental mode.
In previous versions this option was just called Incremental.

Incremental (P1-C) I & J are offsets of the first arc point from the arc center.

Incremental (C-P2) I & J are arc center offsets from the second arc point.

Incremental (P2-C) I & J are offsets of the second arc point from the arc center.

Options - Arc To Lines Tolerance

If Arc Output = *Convert To Lines* is used, this value controls the maximum allowed error when converting arcs to lines. Smaller tolerances will result in smoother curves but larger files.

Options - Clearance Plane Axis

Used to specify which direction clearance moves are made. Usually Z, but may be set to X or Z for lathe operations.

Options - Comment, End Comment

Defines the text to be used at the beginning and end of a comment.

Example 1:

Comment: (
End Comment:)

Example 2:

Comment: ;
End Comment:

Options - End Of Line

Character sequence used at the end of a line. Escape code \r and \n can be used.

Options - Invert Arcs

Controls the behaviour of XZ (G18) arcs only.

For milling operations, this should be set *False*. The direction of the g-code arcs will then be relative to the positive Y axis (using a right hand coordinate system).

For lathe operations, this should usually be set *True*. Lathe arc directions are typically relative to the 'Up' direction. This would imply a positive Y axis using a left hand coordinate system. CamBam's drawing view is a right hand coordinate system so XZ arcs would need to be inverted when written to g-code.

Note: Coordinate handedness can be determined by pointing your thumb in the direction of positive X, second finger in the positive Y axis and third (middle) finger in the positive Z axis direction.

Options - Minimum Arc Length

A numerical value that controls the precision limits used for outputting arc moves (G2, G3). If the length of an arc is less than the ***Minimum Arc Length*** value then a straight move (G1) is used instead.
This is useful for TurboCNC users where very small arcs can cause glitches that may appear as dimples in the toolpath.

Example:
Minimum Arc Length: 1e-4

Options - Maximum Arc Radius

A numerical value that controls the maximum radius allowed for arc moves. If an arcs radius exceeds this value, a straight line move (G1) is used.

Example:
Maximum Arc Radius: 1e6

Options - Number Format

This is a string formatting pattern that controls how floating point numbers are displayed.

A hash character (#) denotes an optional digit place holder and a 0 character denotes a digit that is always displayed, adding padding zeros if required.

It can also change the gcode instructions that are required. For example, if a toolpath contains a move from X=1.234 to X=1.233 and a number format of #.#0 is used, no move instruction will be written to the gcode as the coordinates are identical when formatted to 2 decimal places.

Options - Rapid Down To Clearance

If set *True*, and Z is above the clearance plane a rapid down to the clearance plane is used.
If *False* the current Z is maintained.

Options - Suppress Parser Errors

The post processor will parse gcode as it is created to update internal values such as registers. This can produce error messages for post processors that produce non-standard gcode. In many cases the gcode will still be correctly generated and the error messages can be ignored.

Setting ***Suppress Parser Errors*** to True will prevent the gcode parsing errors being displayed, which may otherwise hide genuine error messages.

Options - Upper Case

If set to *True*, the post processor converts all text written to the gcode file to upper case. This is particularly useful for Fanuc posts that do not support lower case characters.

Post Build - Post-Build Command and Post-Build Command Args

Post-Build Command can be used to specify an external application to modify the gcode produced from the post processor.

Post-Build Command Args contains any arguments to pass to the application.
The following macros are recognised:

{$outfile} is the filename of the raw gcode output.
{$cbfile.name} is the short name of the current CamBam document.

Note: Double quotes should be used in command arguments to avoid problems with spaces in filenames.

Example:

Post-Build Command: `C:\bin\gcodelinenums.exe`

Post-Build Command Args: `"{$outfile}" "{$outfile}.out"`

Rotary [New! 1.0]

Rotary - Axis Of Spin

The axis about which the stock is rotated.

Rotary - Rotary Axis

The rotary (4th) axis letter.

Rotary - Rotary Wrap

If *True*, allow Gcode to be 'wrapped' about a rotary axis.

If ***Rotary Wrap*** is set to *True*, the post processor will convert all toolpaths to lines only, then wrap all the toolpaths selected around the rotary axis.

The radius of rotation is taken from each machining operations ***Stock Surface*** property. It is up to the user to make sure the toolpath is within a width of **2 x PI x stock surface**.

Note: The post-processor should also be modified so as not to output the axis registers for the non rotational axis. For example, if rotating about the Y axis, the X{$x} move parameters should not be output.

see post processors *RotaryX.cbpp* and *RotaryY.cbpp* in the CamBam post folder

Post Processor Macros

$arc.i ***$arc.j*** ***$arc.k***	Outputs the I, J or K register value of the current arc move. The register 'I', 'J' or 'K' prefix is not output.
$arc.radius	Outputs the radius of the current arc move. Arcs that sweep 0 to 180 degrees will have a positive radius and arc sweeps > 180 to 360 degrees will output a negative radius.

$arc.start ***$arc.sweep*** ***$arc.end***	Outputs the start, end or sweep angle of the current arc move. Angles are measured in degrees, with 0 degrees along +X axis. CCW arcs will have a positive sweep angle. CW arcs will have a negative sweep angle.
$arccentermode	Outputs *Arc Center Absolute* (typically **G90.1**) or *Arc Center Incremental* (typically **G91.1**) depending of the *Arc Center Mode* selected
$blocks	This macro is generated internally and contains all the move instructions required by the current machine operation (MOP)'s.
$comment	Inserts the text defined in the ***Comment*** section of the post processor.
$cbfile.footer	Inserts the drawing's ***Custom File Footer*** Machining option.
$cbfile.header	Inserts the drawing's ***Custom File Header*** Machining option.
$cbfile.name	Inserts the drawing's ***Name*** property.
$check(x,y,z)	Generated internally, this macro checks the x,y,z coordinate parameters against the current tool location. If different, a sequence of moves will be inserted to move to the new position, using the clearance plane and plunge feed rates where necessary.
$clearance	Rapids to the clearance plane.
$cuttercomp(off|L|R)	Cutter radius compensation. **Note:** CamBam does not currently calculate radius compensation codes for toolpaths. Inserts the text defined in the Cutter Comp Off, Cutter Comp Left or ***Cutter Comp Right*** sections of the post processor. Typically Off=G40, L=G41, R=G42
$date	Inserts the current date time stamp
$distancemode	Inserts the distance mode in use. The values are defined in the ***Distance Absolute*** and ***Distance Incremental*** sections of the post processor. Currently this always equates to ***Distance Absolute*** (typically G90).
$endcomment	Inserts the text defined in the ***End Comment*** section of the post processor.

$endrewind	Inserts the text defined in the ***End Rewind*** section of the post processor. Typically M30.
$footer	Evaluates the text in the ***Footer*** section of the post processor.
$g0, $g1, $g2, $g3 ***$g81, $g82, $g83*** ***$_g0, $_g1, $_g2, $_g3*** ***$_g81, $_g82, $_g83***	These Gcode macros control how the gcodes are output. The format of each code is taken from the G... definitions in the post processor. This may be useful to control zero padding (eg G1 vs G01), or to use alternative G codes. If the underscore (_) prefix is used, these instructions are assumed to be modal (or sticky). That is; the first occurrence of the code will be written but omitted if following blocks use the same instruction.
$header	Evaluates the text in the ***Header*** section of the post processor.
$mop.clearanceplane	Outputs the ***Clearance Plane*** value of the current machining operation.
$mop.cutfeedrate	Outputs the ***Cut Feedrate*** value of the current machining operation. The 'F' register code prefix is not output.
$mop.depthincrement	Outputs the ***Depth Increment*** value of the current machining operation.
$mop.dwell	Outputs the ***Dwell*** value of the current drilling operation.
$mop.first.x ***$mop.first.y*** ***$mop.first.z***	Insert the X, Y or Z coordinate of the first toolpath point of the current machining operation. This macro may be useful after a tool change command, to move to the next machining X, Y coordinate at the tool change height, before plunging to the clearance plane.
$mop.footer	Inserts the current machining operation's ***Custom MOP Footer*** property.
$mop.header	Inserts the current machining operation's ***Custom MOP Header*** property.
$mop.holediameter	Outputs the ***Hole Diameter*** value of the current drilling operation.
$mop.name	Inserts the current machining operation's ***Name*** property.
$mop.peckdistance	Outputs the ***Peck Distance*** value of the current drilling operation.
$mop.plungefeedrate	Outputs the ***Plunge Feedrate*** value of the current machining operation.

	The 'F' register code prefix is not output.
$mop.retractheight	Outputs the ***Retract Height*** value of the current drilling operation.
$mop.stocksurface	Outputs the ***Stock Surface*** value of the current machining operation.
$mop.tag	Outputs the ***Tag*** value of the current machining operation.
$mop.targetdepth	Outputs the ***Target Depth*** value of the current machining operation.
$move.x ***$move.y*** ***$move.z***	Outputs the X, Y or Z register value of the current move. The register code is not output.
$mops	Inserts a list of objects, one item for each enabled machining operation. Each list item is defined using the ***MOP*** section definition of the post processor.
$part.name	Inserts the name of the current part.
$post.toolchange	Inserts the post processor tool change macro. This may be useful to include in the tool definition ***Tool Change*** property.
$repeat	Inserts the text defined in the ***Repeat*** section of the post processor. Typically M47.
$s	Inserts the current machining operation's ***Spindle Speed*** property.
$set(x|y|z|a|b|c|f|p|q|r,<value>)	Sets the current value of the specified X, Y or Z register. No gcode will be output. Example: $set(z,5.5) This may be useful after a custom, controller based tool change macro, to inform the post processor of the controller's new coordinates. The value NaN can also be used to set the register to an undefined state.
$spindle	Inserts a macro depending on the current machine operation's ***Spindle Direction*** property. Nothing will be written to the gcode if the spindle is already in this state.

$spindle(off|cw|ccw)	Inserts the text defined in the ***Spindle Off***, ***Spindle CW*** or ***Spindle CCW*** sections of the post processor. Typical values are cw=M3, ccw=M4, off=M5
$stock.xsize ***$stock.width*** ***$stock_width***	The X size of the stock block defined in the Machining or Part object. Example: (For CutViewer STOCK definition) {$comment} STOCK/BLOCK,{$stock_width},{$stock_length},{$stock_height},{$stock_x},{$stock_y},{$stock_z} {$endcomment}
$stock.ysize ***$stock.length*** ***$stock_length***	The Y size of the stock block defined in the Machining or Part object.
$stock.zsize ***$stock.height*** ***$stock_height***	The Z size of the stock block defined in the Machining or Part object.
$stock.xoffset	The X coordinate of the lower left corner of the stock block (relative to the machine's XY(0,0)), defined in the Machining or Part object.
$-stock.xoffset ***$stock_x***	The *minus* X coordinate of the lower left corner of the stock block (relative to the machine's XY(0,0)), defined in the Machining or Part object.
$stock.yoffset	The Y coordinate of the lower left corner of the stock block (relative to the machine's XY(0,0)), defined in the Machining or Part object.
$-stock.yoffset ***$stock_y***	The *minus* Y coordinate of the lower left corner of the stock block (relative to the machine's XY(0,0)), defined in the Machining or Part object.
$stock_z	The *minus* Z coordinate of the lower left corner of the stock block (relative to the machine's XY(0,0)), defined in the Machining or Part object.
$stop	Inserts the text defined in the ***Repeat*** section of the post processor. Typically M0.
$tool.comment	Inserts the ***Comment*** property from the tool library for the current tool.

$tool.diameter	Inserts the current machining operation's ***Tool Diameter*** property. **Note:** The $tool.diameter macro will not be defined until there has been a tool change command. If used in the header section, use a tool change such as ***$toolchange(first)*** before referring to ***$tool.diameter***.
$tool.index	Inserts the current machining operation's ***Tool Number*** property.
$tool.length	Inserts the tool length property from the tool definition in the tool library.
$tool.name	Inserts the current tool's ***Name*** property (from tool library) or T(tool number) if there is no tool library entry.
$tool.profile	Inserts the ***Tool Profile*** property of the current tool.
$tool.radius	Uses the current machining operation's ***Tool Profile*** property to determine a radius. 0 for end mills and Diameter / 2 for bullnose.
$tool.veeangle	Inserts the current tool's ***Vee Angle*** property (from tool library) or 0 if there is no tool library entry.
$toolchange	Inserts a tool change instruction, based on the ***Tool Change*** definition in the post processor. If the tool number has not changed, no tool change code is inserted.
$toolchange(first)	Inserts a tool change instruction using the first tool in the current drawing's tool table.
$tooltable	Inserts a description for each tool that is referenced in the current drawing. Each item in the list is formatted using the ***Tool Table Item*** definition in the post processor.
$units	Outputs the drawing's ***Units*** property. The codes used are taken from the ***Units (Inches)*** or ***Units (Metric)*** sections of the post processor. Typically Inches = G20, Millimeters = G21.
$velocitymode	Inserts the current machining operation's ***Velocity Mode*** property. The codes used are taken from the ***Velocity Mode - Constant Velocity*** or ***Velocity Mode - Exact Stop*** sections of the post processor. For example: Mach3 uses Exact Stop=G61, Constant Velocity=G64.

$workplane	Inserts the current machining operation's ***Work Plane*** property. The codes used are taken from the ***Workplane XY | XZ | YZ*** sections of the post processor. Typically XY=G17, XZ=G18, YZ=G19.
$x, $y, $z, $a, $b, $c $i, $j, $f, $r, $p, $q $_x, $_y, $_z, $_a, $_b $_c, $_i, $_j, $_f, $_r $_p, $_q	These macros insert the parameters used in common Gcode move operations. If an underscore (_) prefix is used, these parameters are treated as modal. That is they will only be output if the current value has changed. Omitting the underscore will force the parameter to be always output. These macros will include the register code as well as the value, for example $x = X1.23
$xneg, $yneg, $zneg, ...	The same as the other register macros ($x, $_y etc), but with the value sign reversed.
$xabs, $yabs, $zabs, ...	The same as the other register macros ($x, $_y etc), but with the value always positive.

Nesting

CamBam's nesting provides a method to repeat a series of machining operations at different positions on the machine table. In this way, it is possible to produce multiple copies of a workpiece from a single drawing and one set of machining operations.

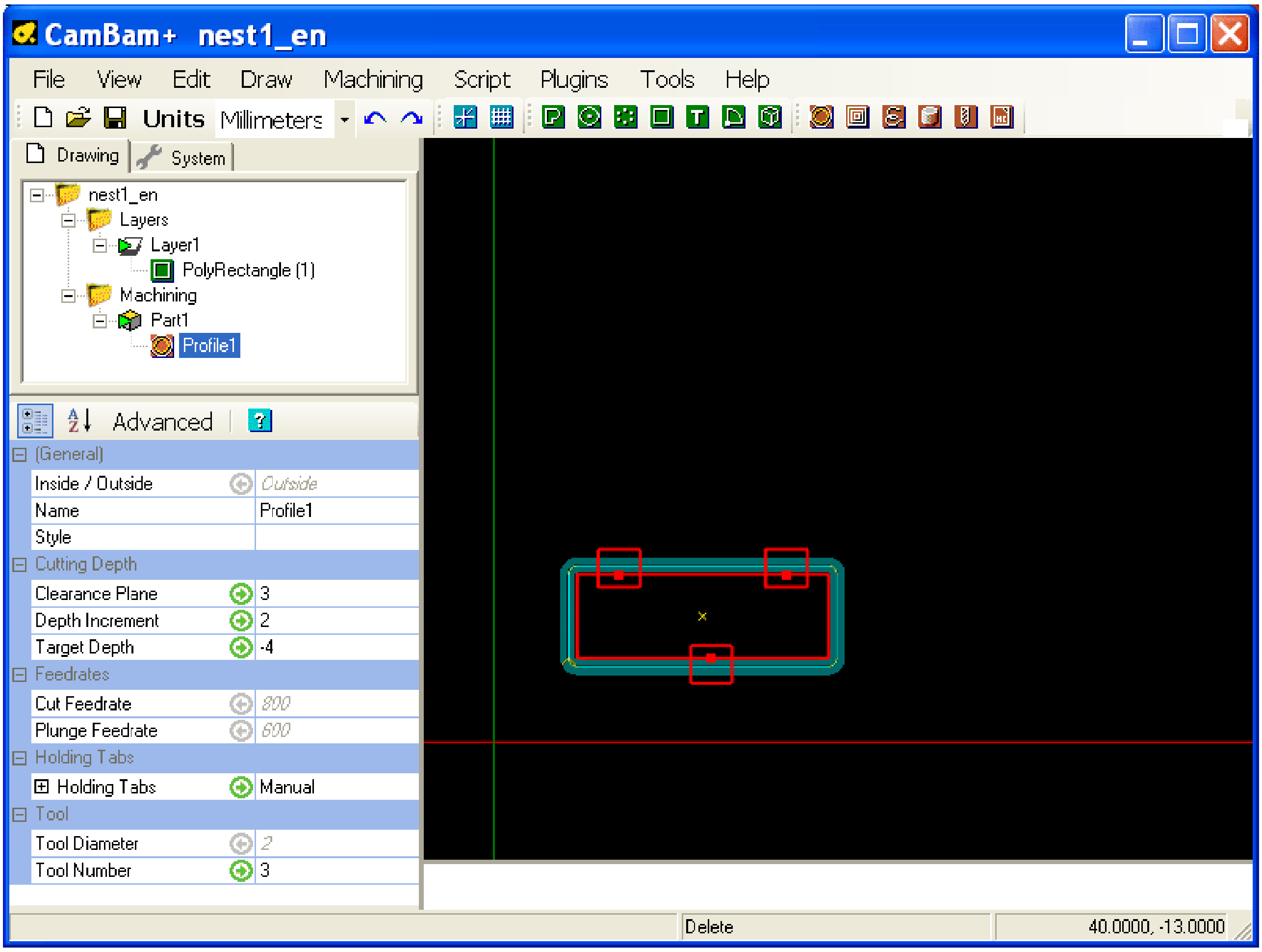
Nesting operates on the contents of an entire ***Part***. All machining operations contained in a part will be machined at the positions and order defined in the ***Nesting*** property of that part.

Implementation of a nesting

First create one or more machining operations.

For this example, we will use a single operation; an external cut of a rectangle of 10 x 30 mm, with a tool of 2 mm diameter.

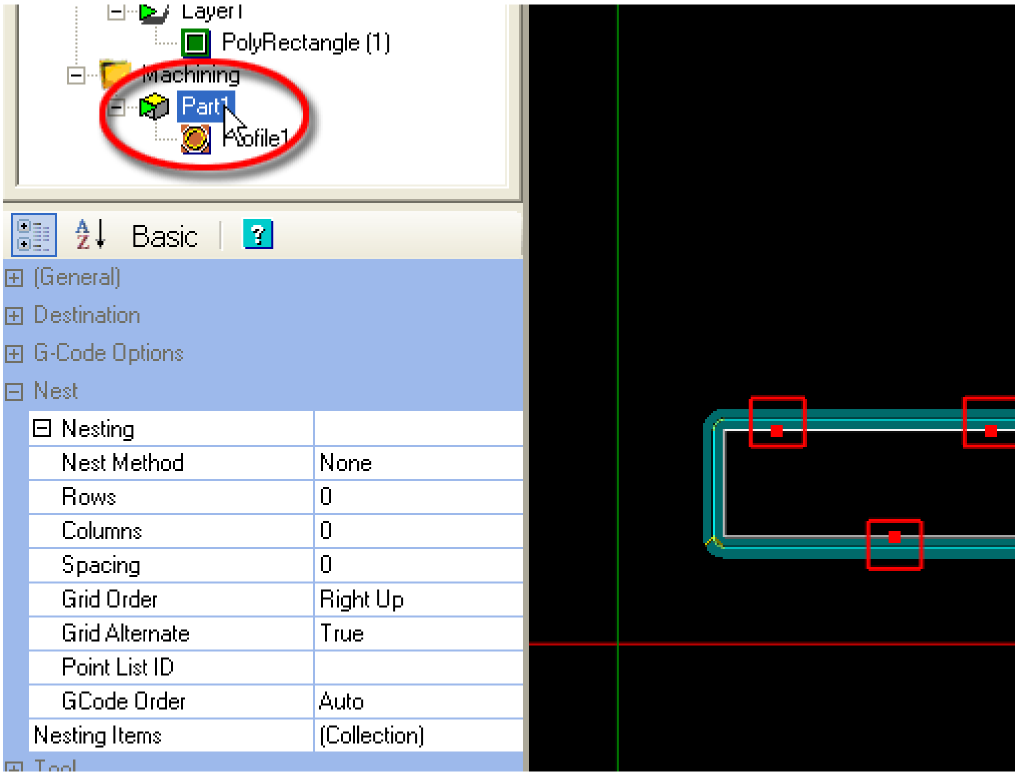
The following image shows the ***Profile*** operation, with tabs and cut widths of the tool visible.



Setting the nesting

All the properties defining the nesting are located at the Part object.

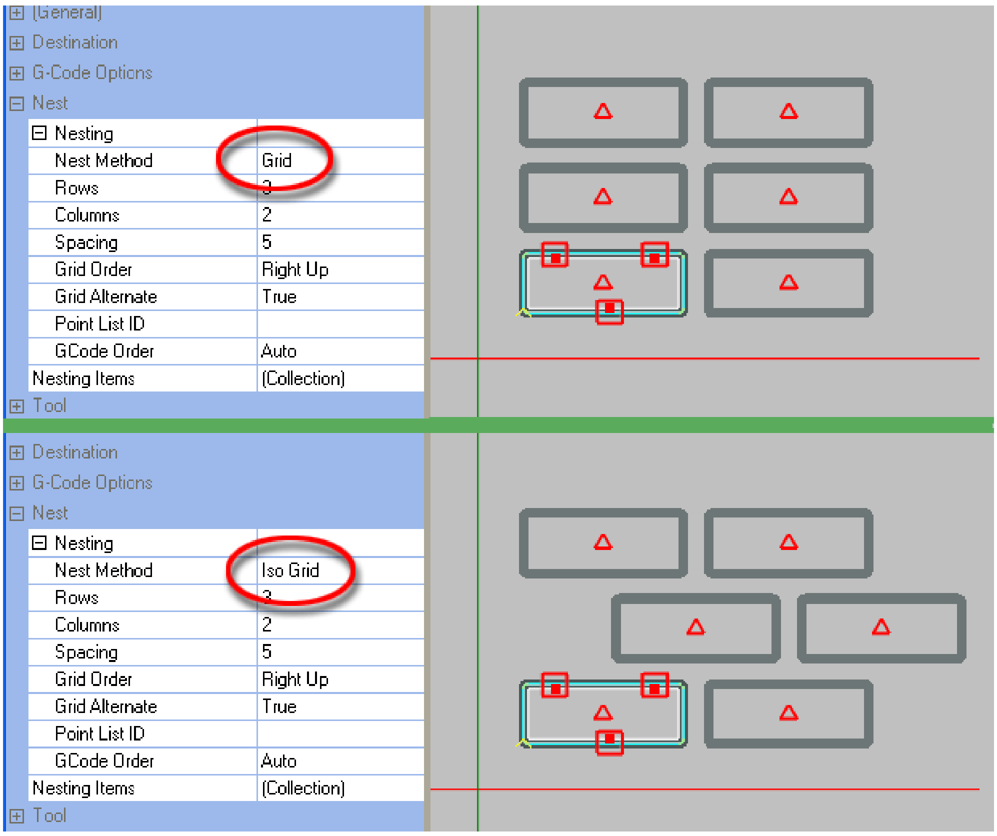
Select the Part containing the Machining Operations (MOP) that are to be nested.



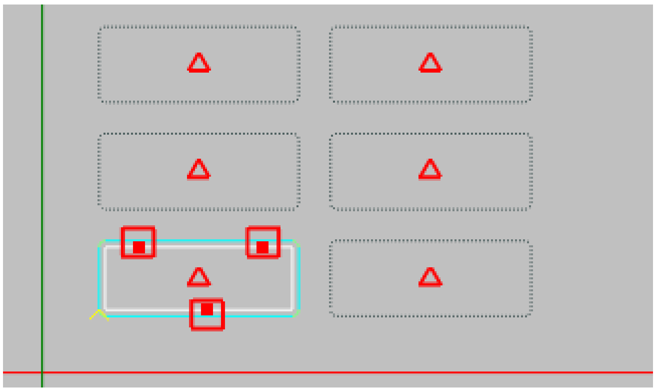
In the part properties, click the + sign to the left of the ***Nesting*** property to display all options.

Nest Method

The ***Grid*** and ***ISO Grid*** methods are used to distribute copies of machining operations in **rows** and **columns** automatically. The ISO Grid method additionally adds a horizontal / vertical shift of half the pitch at each row / column. The shift is performed on the lines or the columns depending on the value of ***Grid Order***.



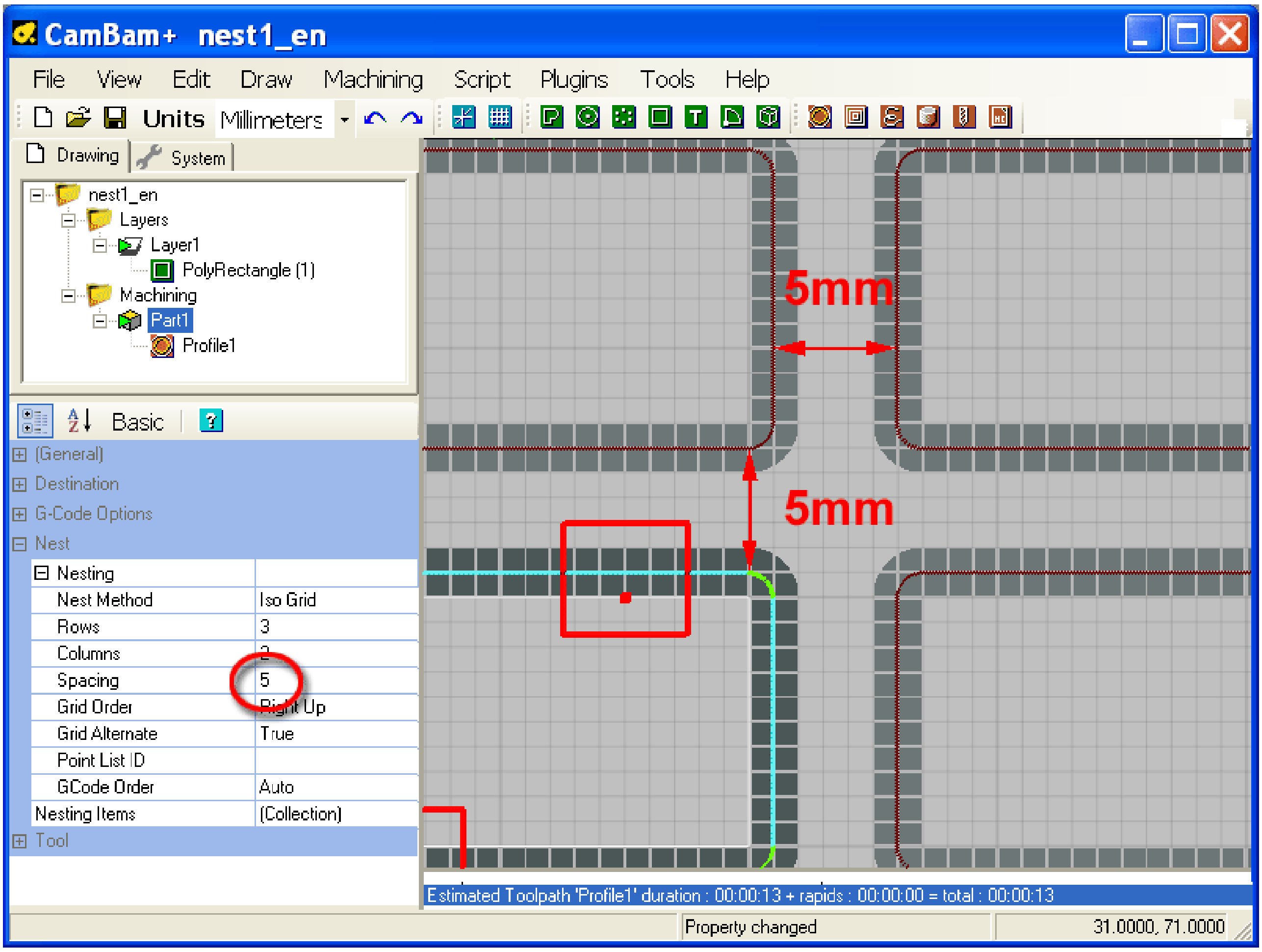
Note: Copies are shown as a dotted outline of the same color as used for the display of the cut widths (from revsion N), with a red triangle in each center as shown in the picture below. You can also activate the menu ***View/ Show cut widths***, allowing a better view of the copies as on the images above.



Note: The Part must be selected in the drawing tree, for the copies to be visible.

To create the nesting of this example, select a type of grid, enter the number of ***Rows*** and ***Columns*** required, a ***Spacing*** value, and then generate the toolpaths of the ***Part*** via its context menu.

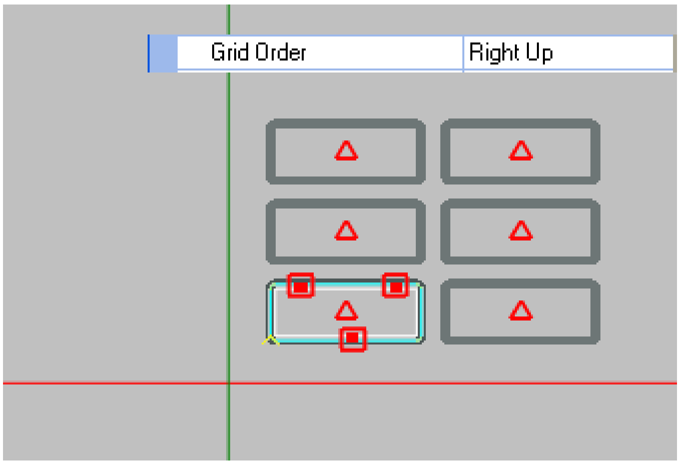
Important: The *Spacing* value defines the distance between the outermost toolpaths of the active machining operation contained in the Part.

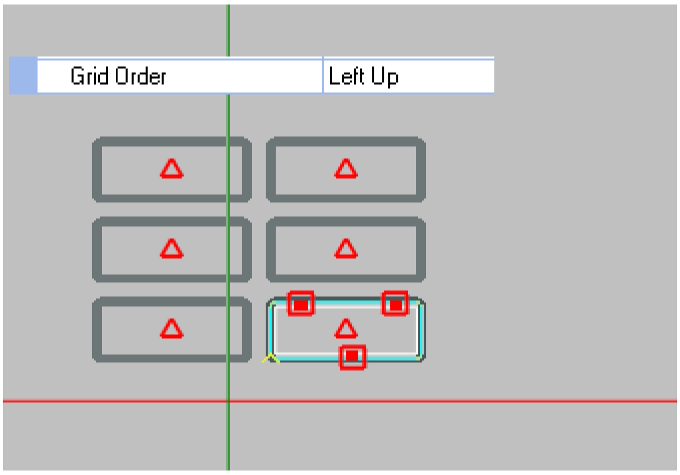


In the picture above, with ***Show Cut Widths*** enabled, a distance of 5 mm is visible, between the toolpaths, leaving 3mm of stock between the widths of the 2mm tool diameter cut.

Grid order

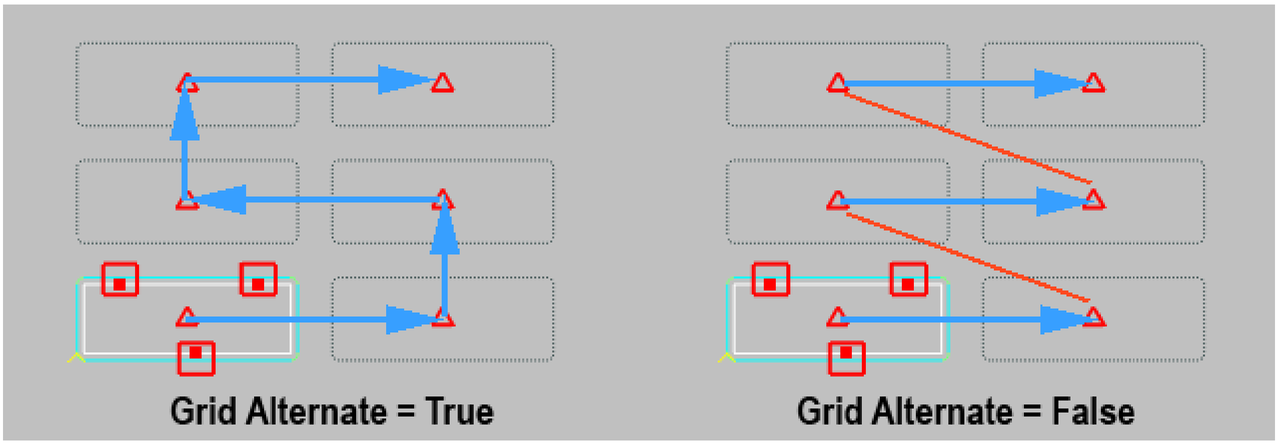
Grid Order controls the direction of the grid layout. For example *Right Up* will make copies to the right of the original, then move up to the next row.





Grid Alternate

If ***Grid Alternate*** is set to *True*, the grid will alternate the direction of each row or column (depending on ***Grid Order***). If *False* then each row or column will proceed in the same direction, with a rapid back to the start of each.

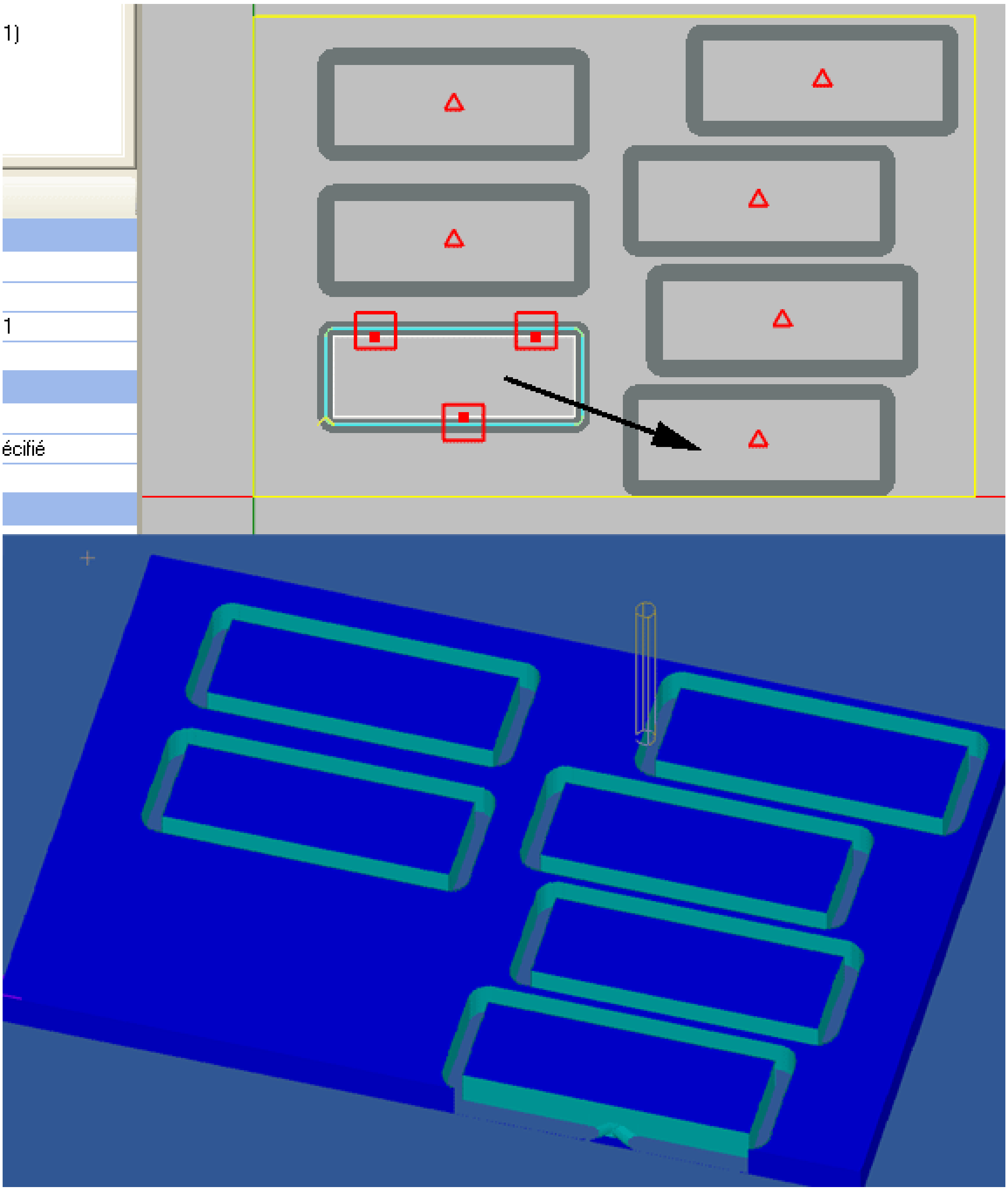


Moving the copies

Nested copies can also be positioned manually.

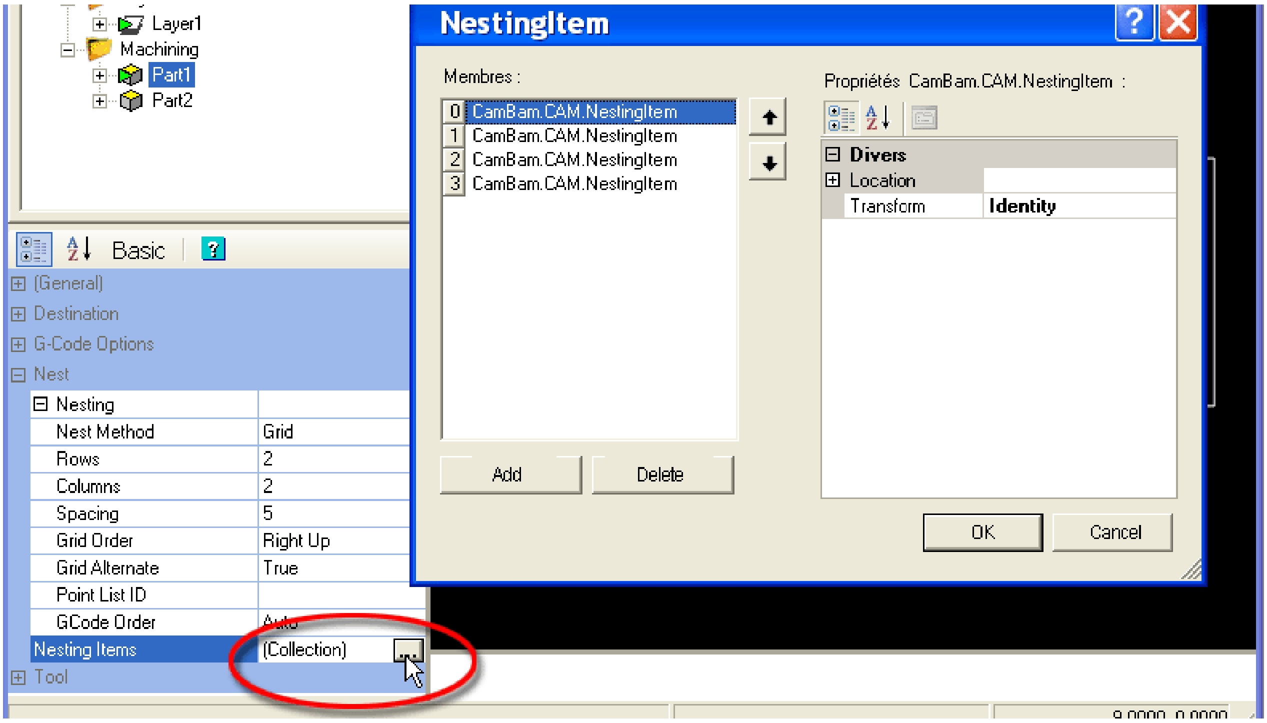
Click and hold the left button of the mouse over the red triangle of a copy, then drag it to the desired position. The ***Nest Method*** property will automatically change to *Manual*.

Note that if you move the original copy, the original will always appear at the same position on the display (but without the red triangle), but **will not be machined** as the image of the simulation below shows (the black arrow indicates the displacement of the original). Only the nested copies will be output to the gcode.



You can also edit the location of copies, delete or add, modify the order in which they will be machined using the Nesting Item collection editor.

Set the ***Nest Method*** to *Manual*, and then edit the *(collection)* object. The ***Advanced*** properties view needs to be enabled to access this.



The left side of the window allows you to add or remove copies using the ***Add*** and ***Delete*** buttons.

With the two vertical arrows to the right of the list you can change the order of the list, and thus the machining order of the copies, by selecting a copy then clicking on an arrow to move up or down.

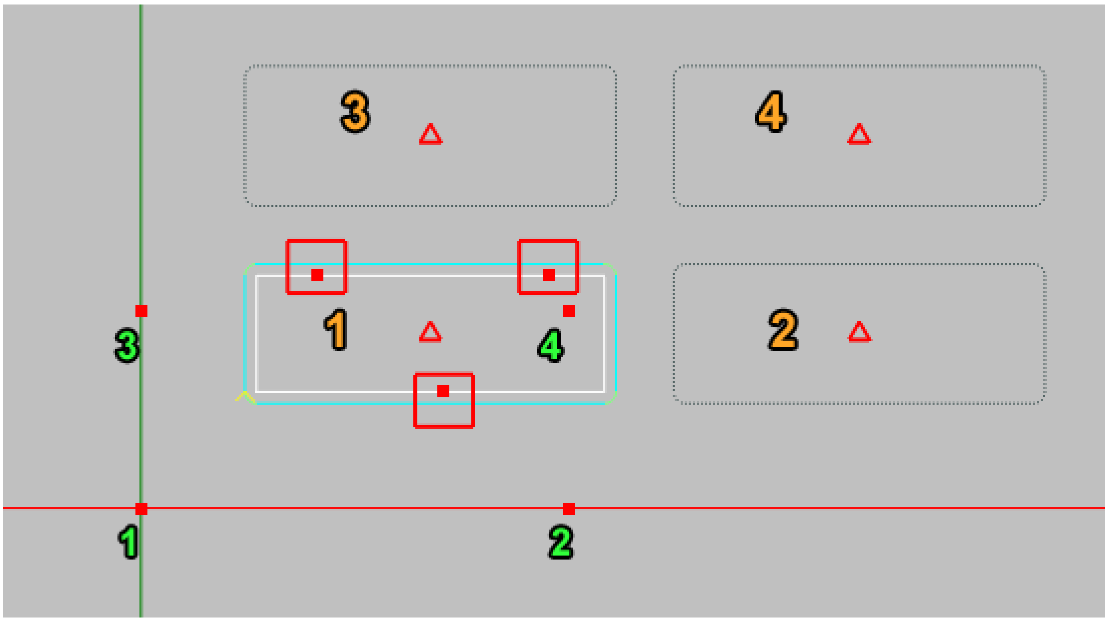
The right part of the nesting collection editor allows you to edit the coordinates of each copy.

Using a points list to define a nesting

It is possible to use a points list drawing object to define the positions of nesting copies, by using the ***Nesting Method*** set to *Point List*, then assigning the ***ID*** of the point list in the ***Point List ID*** property.

The default coordinates of the nesting system are at the drawing origin, (at the center of the X and Y axes), unless an alternative *Machining Origin* has been set (shown by a small red cross), either on the ***Part*** or ***Machining*** options.

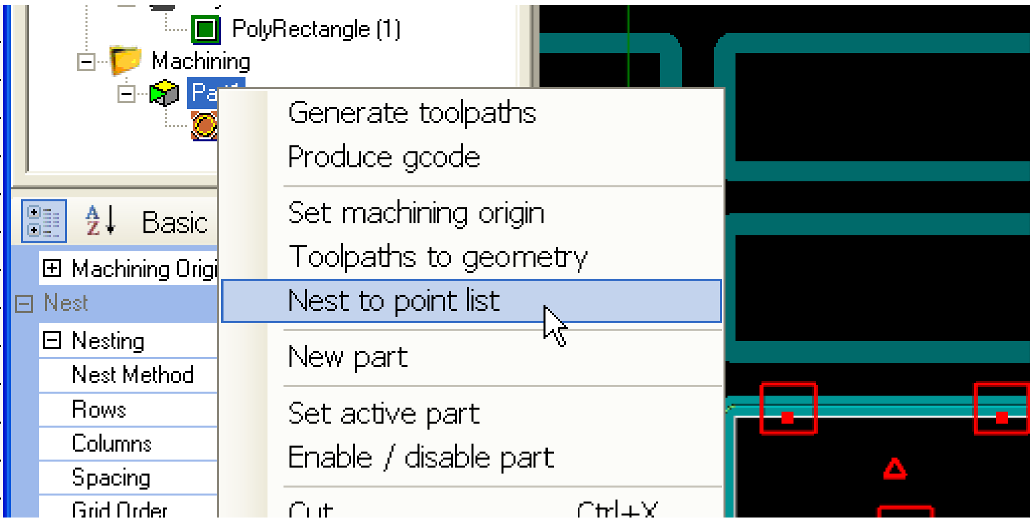
The image below shows the points corresponding to different copies.



It is also possible to generate a list of corresponding points to an existing nesting. This is what I did to get the point list corresponding to the different copies of the image above.

Copies having been previously created with the automatic placement obtained with the Grid method.

For create this point list, open the context menu of the relevant ***Part***, and use the menu item: ***Nest to Point list***



Why use a point list ?

There are some cases where the use of a ***Point list*** is required.

- If you want to assign a value from center to center distance between the copies rather than spacing between them, (or more precisely, between the outermost toolpaths).
- If a specific placement is needed, such as the result of a mathematical calculation. The point list can be created in a spreadsheet and then imported into CamBam.
- If several different Parts need the same nesting placement.

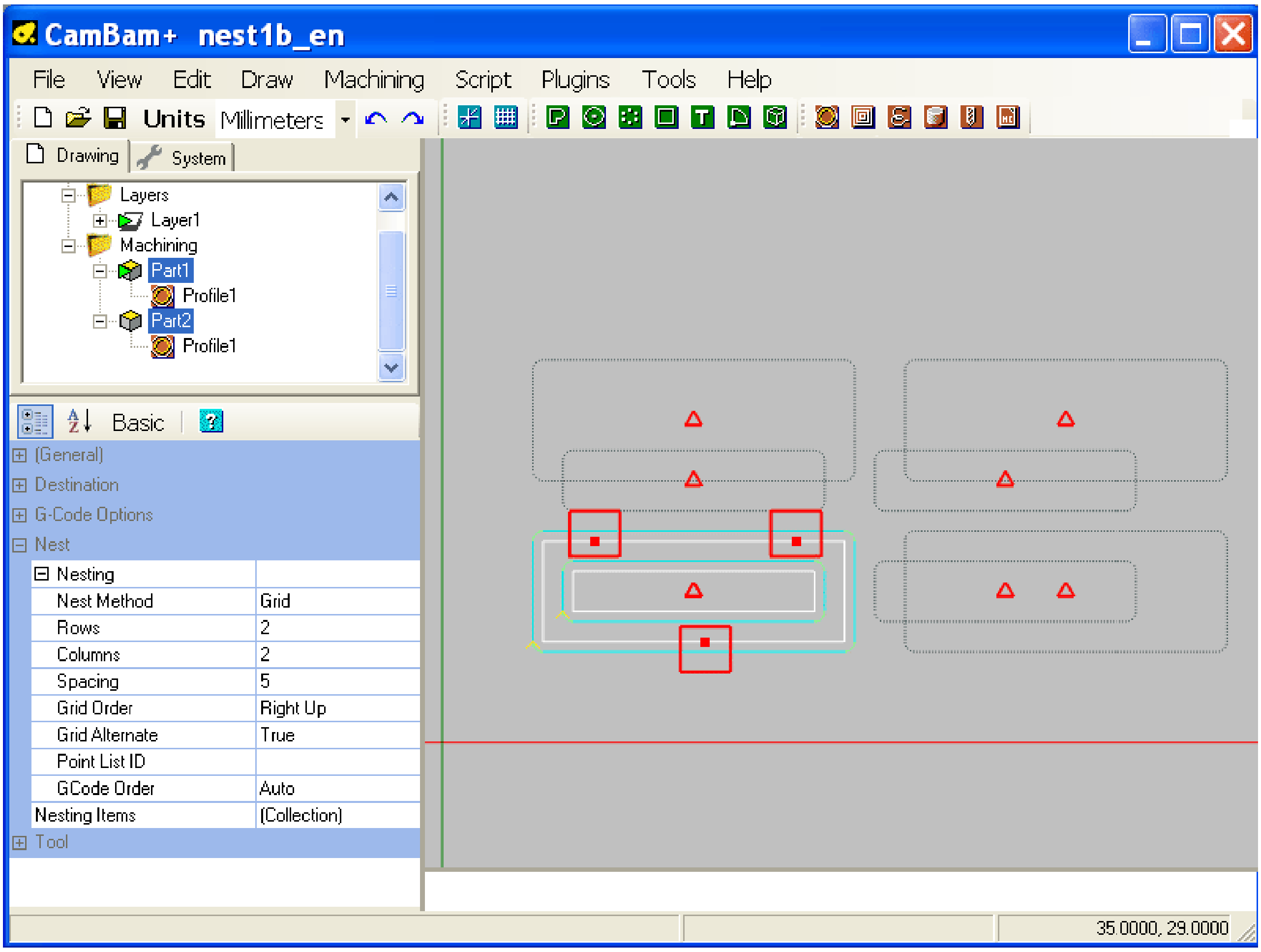
In these cases the spacing between copies is not usable as it is based on the distance between the outermost toolpath of the Part, which can be different from one Part to another, depending on the machining operations that compose it.

If each ***Part*** uses the same, common point list object for their nesting, their nested copies will have identical positioning.

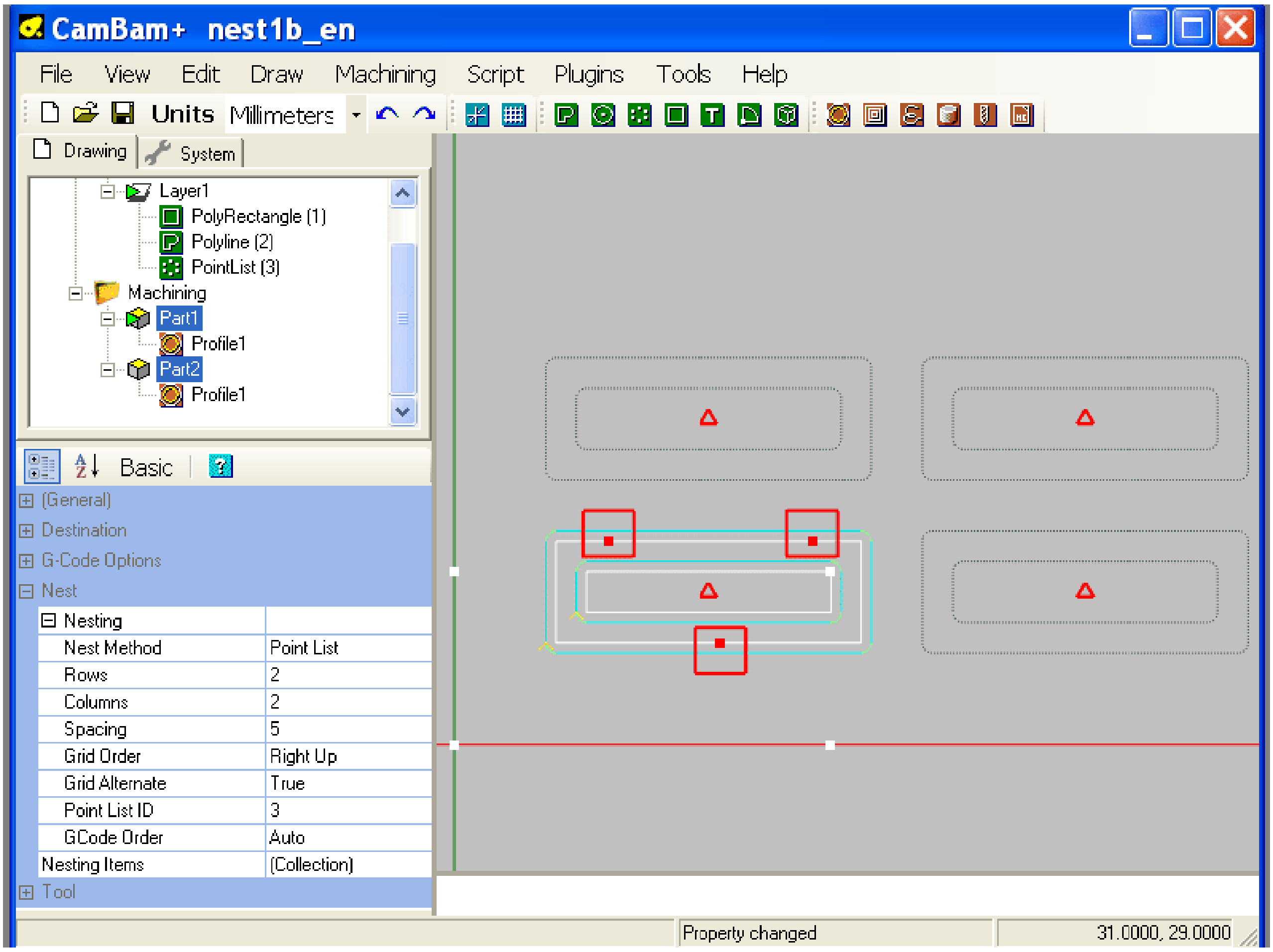
Note: If some machining operations are disabled in the Part, they will not be taken into account when calculating the spacing between copies. Enabling and Disabling machining operations can alter the nest spacing when *Auto* nesting methods are used, whenever the toolpaths are regenerated.

The ***Nest to Point list*** function on the ***Part*** context menu can be used to generate a suitable placement point list. The resulting point list can then be assigned to all other Parts that must use the same nesting pattern

In this image, the *Grid* nesting method is used with the same parameters for both Part's nesting properties. As can be seen, the operations of the second Part (the small rectangle) are not executed in the right location.



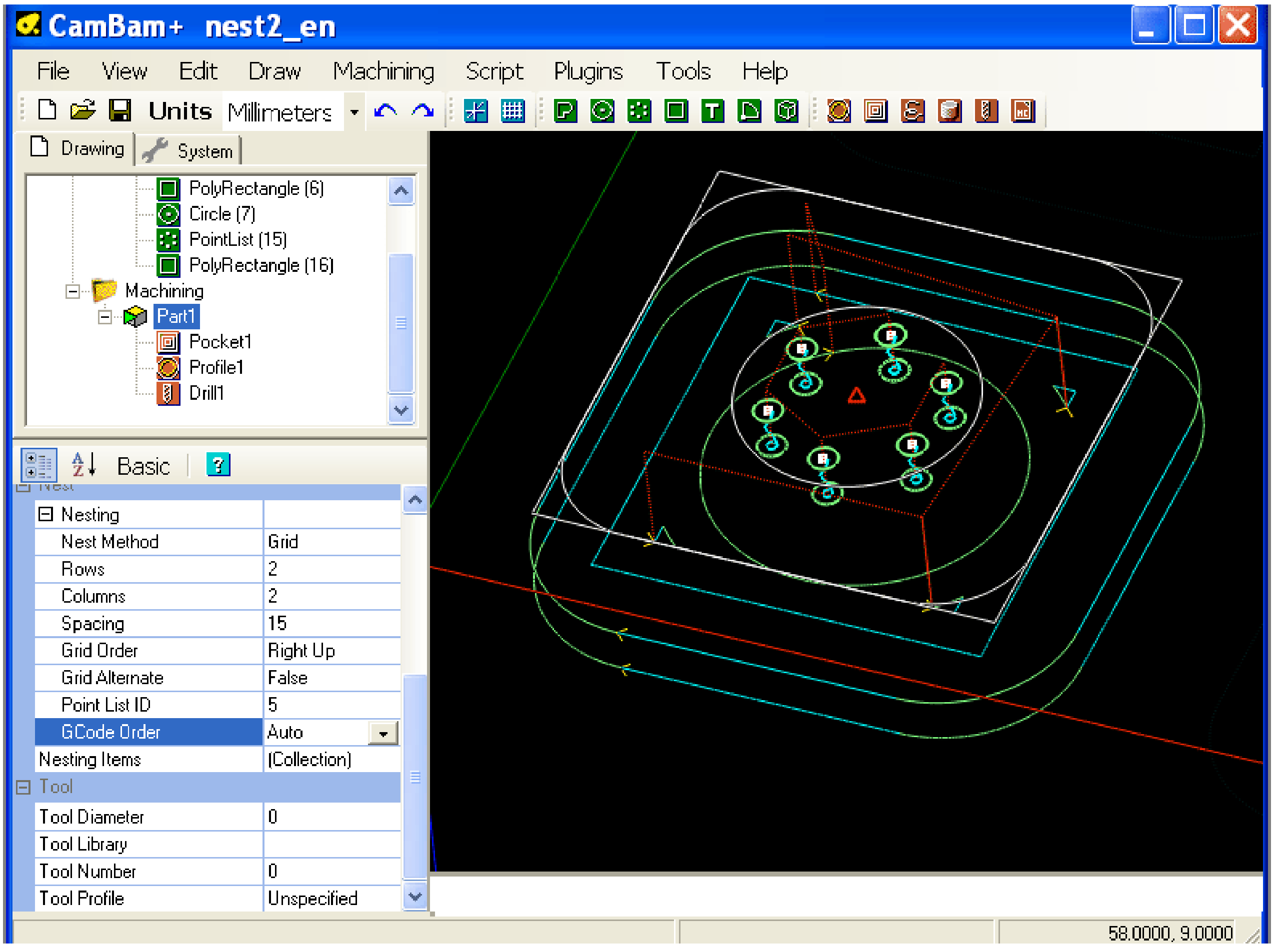
After generating a point list for *"Part1"*, and assigning this point list to the both Parts, and setting ***Nest Method*** to *Point list* everything is in order.


CamBam+ nest1b_en
File View Edit Draw Machining Script Plugins Tools Help
Units Millimeters
Drawing System
Layer1
PolyRectangle (1)
Polyline (2)
PointList (3)
Machining
Part1
Profile1
Part2
Profile1
Basic
(General)
Destination
G-Code Options
Nest
Nesting
Nest Method Point List
Rows 2
Columns 2
Spacing 5
Grid Order Right Up
Grid Alternate True
Point List ID 3
GCode Order Auto
Nesting Items (Collection)
Tool
Property changed
31.0000, 29.0000

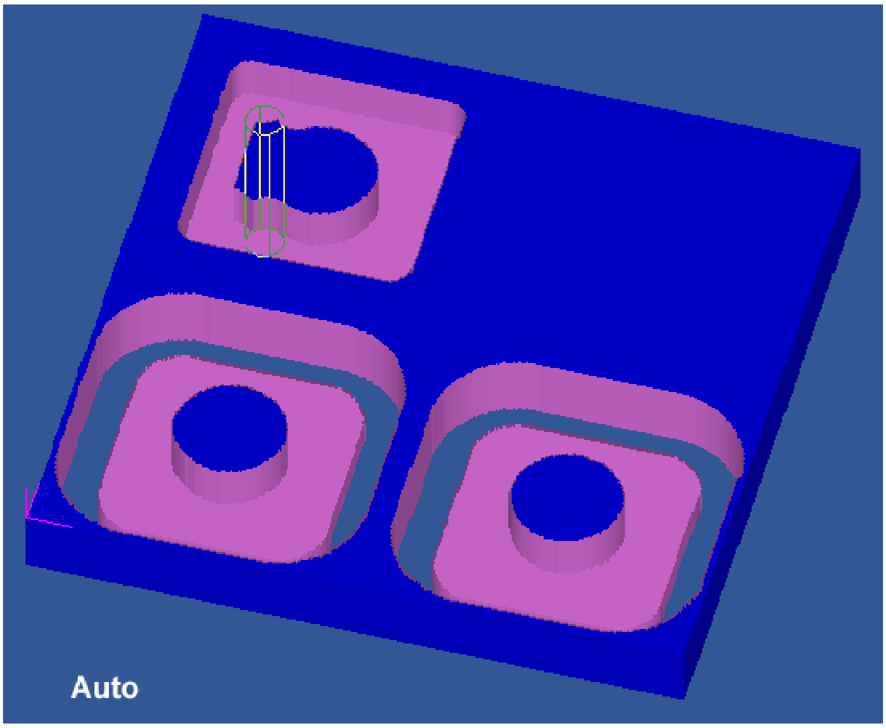
Output order of machining operations

The ***GCode Order*** property is used to define the order in which multiple machining operations contained in a Part are executed.

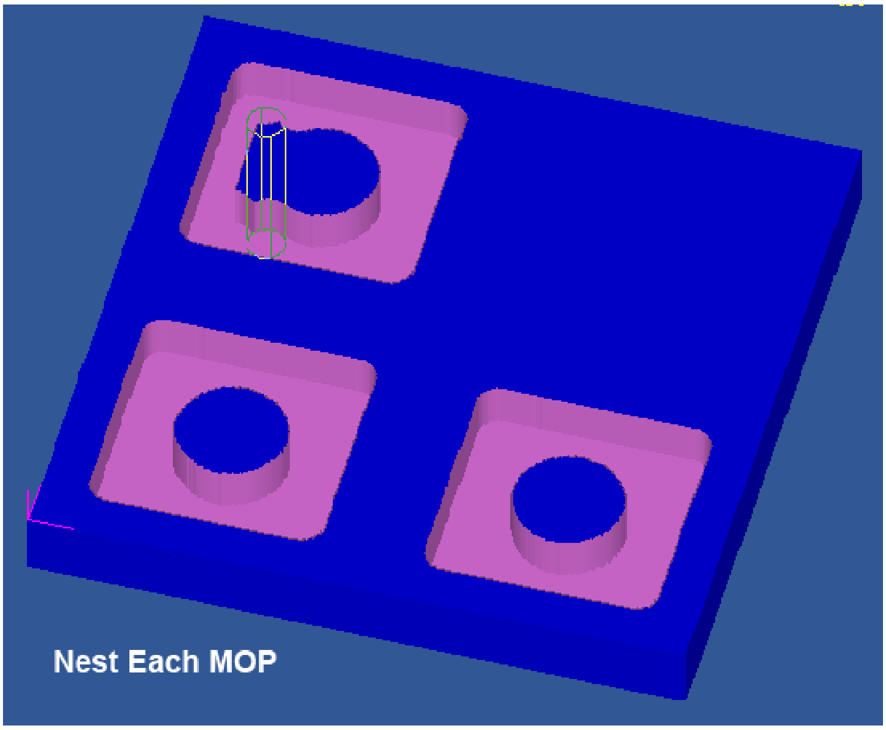
In the following example, the machining *"Pocket1"* (an island) and the cutting *"Profile1"* are machined with the same tool of 8mm diameter, the operation *"Drill1"* is performed with a tool of 2mm diameter, so there is a tool change at the beginning of the *"Drill1"* operation.



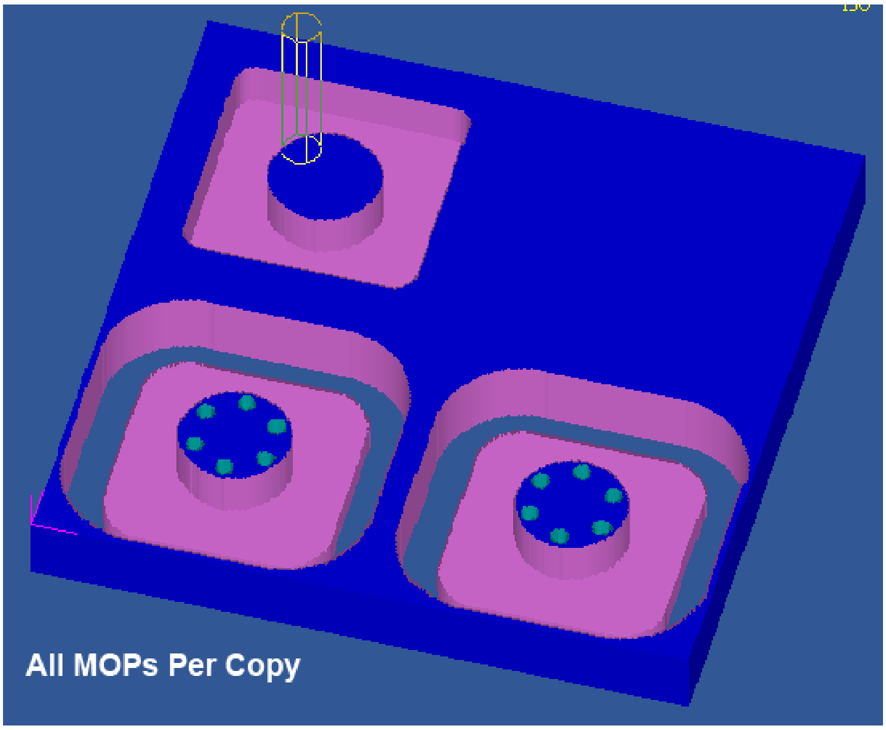
GCode Order = *Auto* - All consecutive MOPs within the part ***with the same tool number*** will be posted, then repeated for each nest copy, before moving to the next MOP (which requires a tool change).



GCode Order = *Nest Each MOP* - Each MOP is output at each nest location before moving to the next MOP.



GCode Order = *All MOPs Per Copy* - All the MOPs in the part are posted before moving to the next nest location.



Use of the nesting with a 4th rotary axis

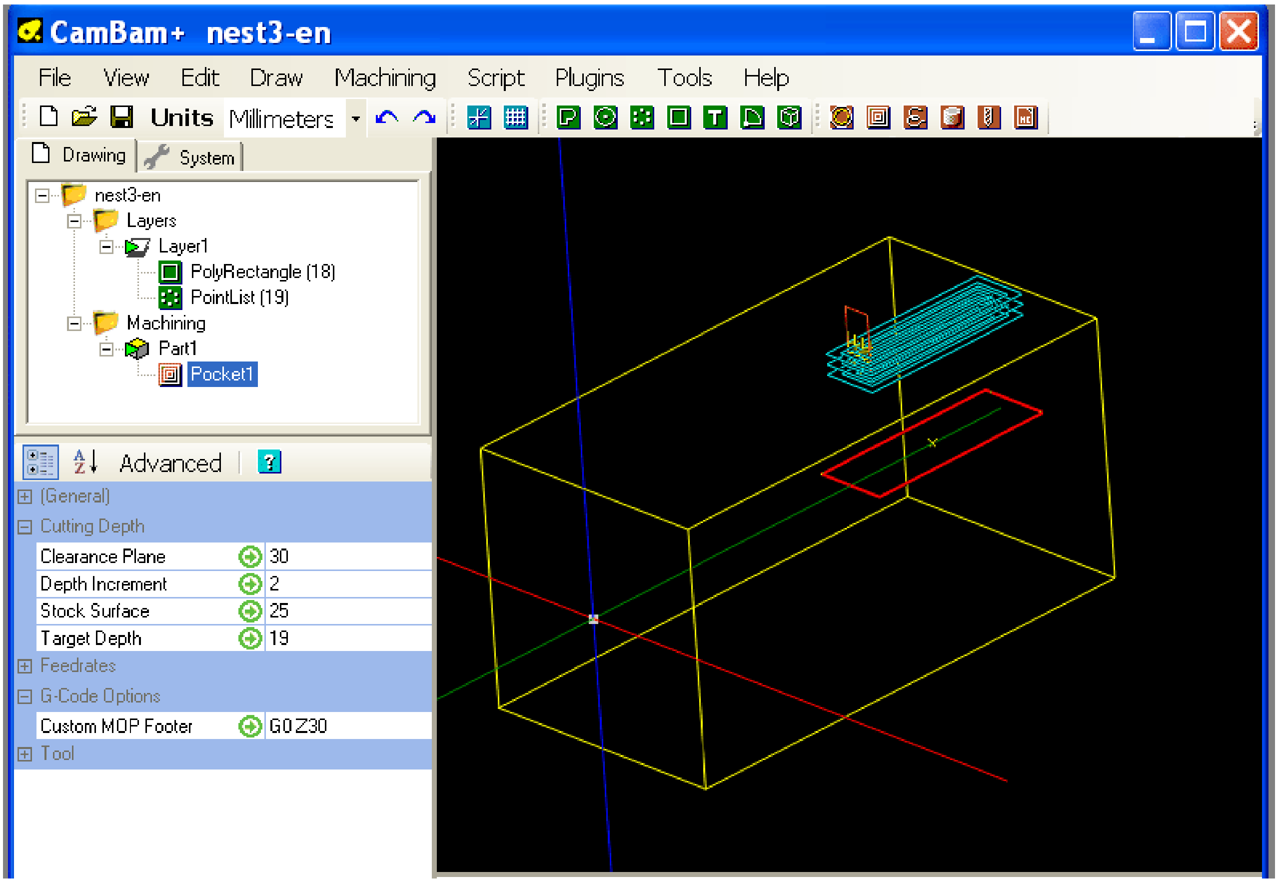
To use the nesting system with a rotary axis to reproduce the same machinings at different angular positions, use a point list containing the same number of points as the number of copies to be made, **with each point's X and Y coordinates set to 0**.

The machining operations in the Part will then be repeated at the same place. Add a Gcode command to the ***Custom MOP Footer*** of the last machining operation of the Part to rotate the 4th axis the desired angle before performing the following sequence of machining operations.

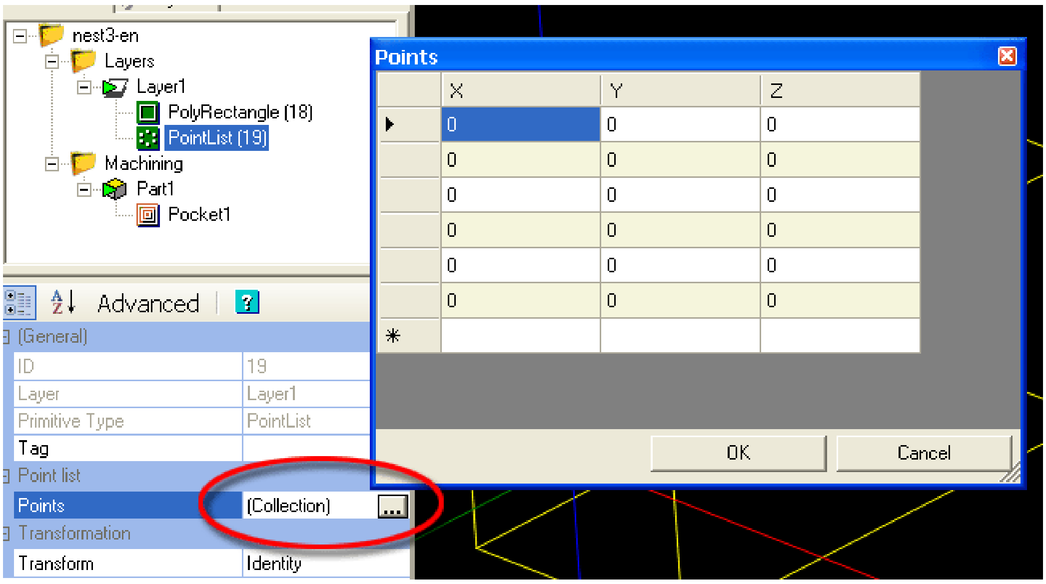
Example:

To machine 6 grooves distributed around a shaft of 50mm diameter.

Create a ***Pocket*** operation with a ***Stock surface*** value equal to the radius of the workpiece to be machined, in this example ***Stock surface*** = 25 (Z=0 is set at the axis of the chuck on the CNC rotary axis).



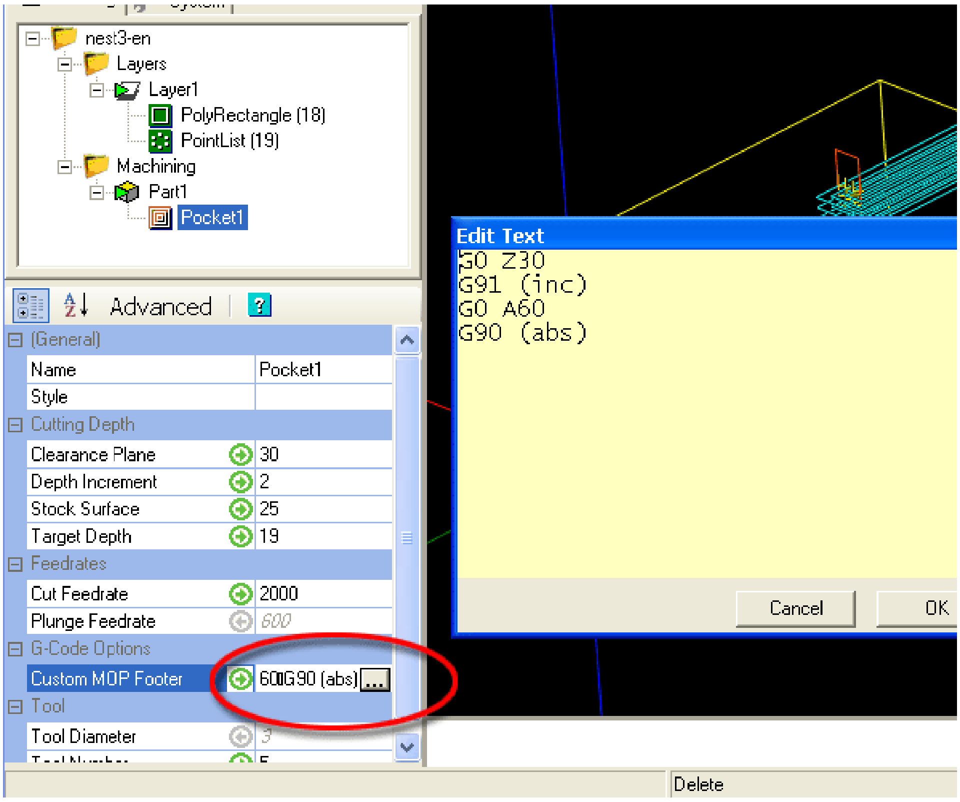
Create a Point list with 6 points, all at (X,Y,Z) = 0,0,0.



Define a nesting with the ***Nest method*** = *Point list* for this Part, then assign to it the point list created above.

The pocketing operation will be repeated six times in the same place.

To rotate the chuck at the end of the part's operations, the following Gcode sequence is added in the ***Custom MOP Footer*** property of the last machining operation in the Part.



Go back to the clearance plane:

```
G0 Z30
```

Use relative coordinates so as to give a rotation angle rather than an absolute position:

```
G91
```

Rotate the A axis of 60° i.e. 360/6:

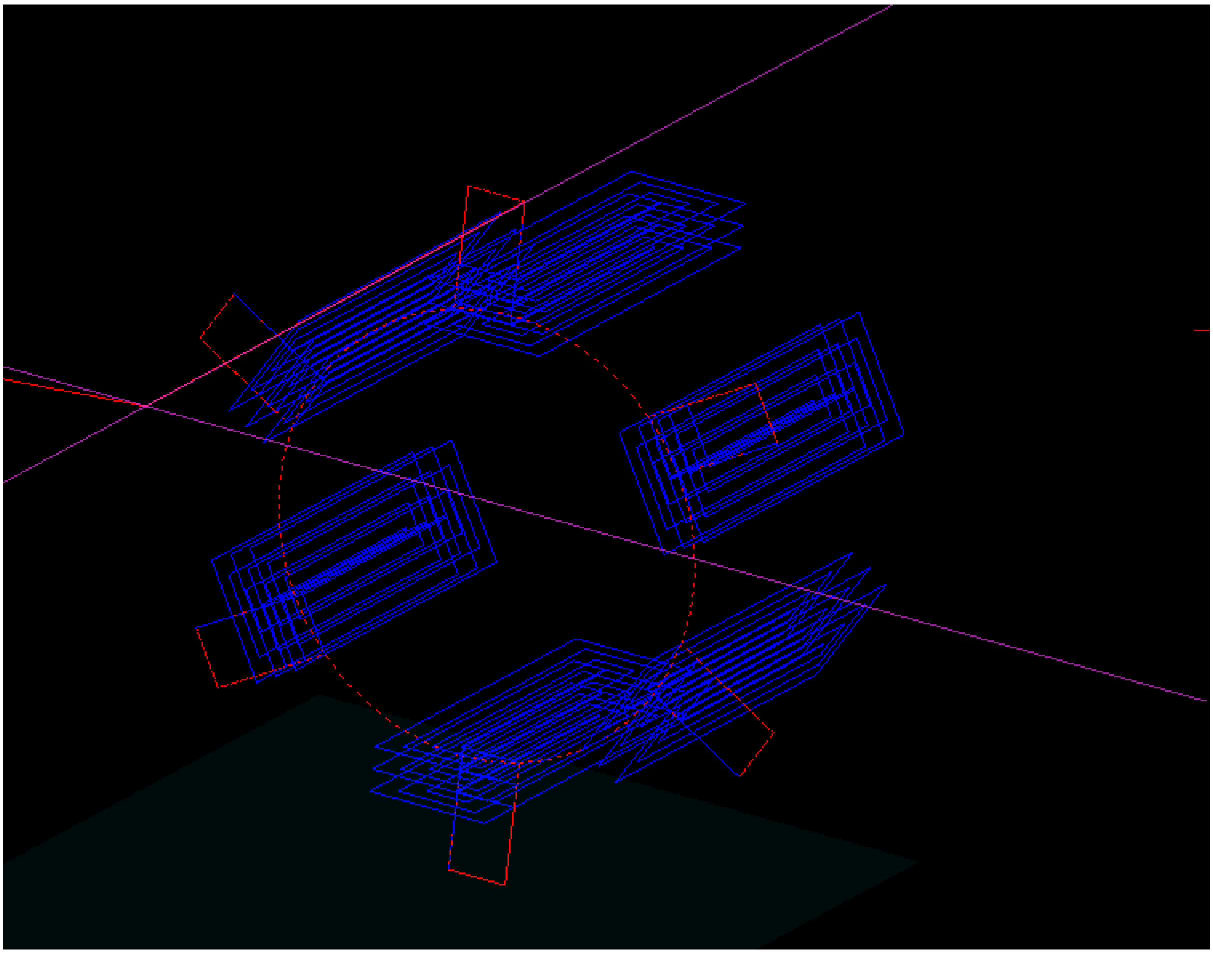
```
G0 A60
```

Change back to absolute coordinates:

```
G90
```

You can also use a ***NCFile*** object to put the Gcode that produces the rotation instead of putting it into the ***Custom MOP Footer*** property. In this case the GCode must be created in a separate text file. You can create it with the CamBam text editor. (***Scripts / New / VBscript*** ; delete the existing text, enter your Gcode and save the file with a .txt, .nc, or .ngc extension ….)

The result cannot be seen in *CamBam* or *CutViewer*, but is visible in the Mach3 simulation window .



Properties

Nesting	This composite property provides a method of generating an array or nest of parts. ***Nest Method***: Change this to *Grid* or *Iso Grid*, then set the ***Rows*** and ***Columns*** values to determine the number of copies of each part. The ***Spacing*** value will control the distance between each copy. When the toolpaths are generated, an outline should be displayed to indicate the location of each copy. The centre of each outline contains a triangular icon. Clicking and dragging this icon will change the nesting pattern and will also change the nesting method to *Manual*. ***Grid Order*** Controls the direction of the grid layout. For example *Right Up* will make copies to the right of the original, then move up to the next row. ***Grid Alternate*** If set to *True*, the grid will alternate the direction of each row or column (depending on ***Grid Order***). If *False* then each row or column will proceed in the same order with a rapid back to the start of each. ***Nest Method*** = *Point List* The location of each nest copy is taken from a point list drawing object which is set in the ***Point List ID*** property. A new ***Nest to point list*** Part context menu function has been added, in this way a list of nest points can effectively be copied from one part to another by sharing a common point list.

	GCode Order Controls how the nested machining operations are ordered in the gcode output. • *Auto* - All consecutive MOPs within the part with the same toolnumber will be posted then repeated for each nest copy, before moving to the next MOP (which would require a tool change). • *Nest Each MOP* - Each MOP is output at each nest location before moving to the next MOP. • *All MOPs Per Copy* - All the MOPs in the part are posted before moving to the next nest location. Multiple copies of the part's toolpaths will be written to the gcode output. This will increase the gcode file size, but does avoid some of the issues encountered when using subroutines.

Backplotting + NCFile object

CamBam can be used to view toolpaths contained within many gcode files.

GCode files can be opened using ***File - Open***, or dragged onto the main drawing view from Windows Explorer.

The gcode file is associated with a special ***NCFile*** machining operation that will appear in the machining tree view. This operation contains properties that can change the way the gcode is interpreted and displayed. If any options are changed, the toolpaths should then be regenerated.

CamBam currently only supports basic gcode and does not recognise more complex gcode syntax such as subroutines.

The contents of the gcode file referenced in the ***NCFile*** object, will be written to the gcode output of the parent drawing. Also, by double clicking the ***NCFile*** machining operation in the drawing tree, the gcode source file will be opened in the configured gcode editor.

Another useful feature of backplotting is the ability to convert the gcode toolpaths to drawing objects. Right click the NCFile object under the machining tree and select ***Toolpath To Geometry*** from the context menu.

Properties

Arc Center Mode	GCode distance mode (*Absolute* or *Relative*), used to determine I and J coordinates in G02 and G03 (arc) commands.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.
Custom MOP Header	A multi-line gcode script that will be inserted into the gcode post before the current machining operation.
Cut Feedrate	The feed rate to use when cutting.
Distance Mode	GCode distance mode (*Absolute* or *Relative*), used to determine X, Y and Z coordinates.
Enabled	*True*: The toolpaths associated with this machining operation are displayed and included in the gcode output *False*: The operation will be ignored and no gcode or tool paths will be produced for this operation.

Max Crossover Distance	Maximum distance as a fraction (0-1) of the tool diameter to cut in horizontal transitions. If the distance to the next toolpath exceeds MaxCrossoverDistance, a retract, rapid and plunge to the next position, via the clearance plane, is inserted.
Name	Each machine operation can be given a meaningful name or description. This is output in the gcode as a comment and is useful for keeping track of the function of each machining operation.
Optimisation Mode	An option that controls how the toolpaths are ordered in gcode output. *New (0.9.8)* - A new, improved optimiser currently in testing. *Legacy (0.9.7)* - Toolpaths are ordered using same logic as version 0.9.7. *None* - Toolpaths are not optimised and are written in the order they were generated.
Plunge Feedrate	The feed rate to use when plunging.
Source File	The filename of the gcode file which will be read, back plotted and inserted into output gcode.
Start Point	Used to select a point, near to where the first toolpath should begin machining. If a start point is defined, a small circle will be displayed at this point when the machining operation is selected. The start point circle can be moved by clicking and dragging.
Style	Select a ***CAM Style*** for this machining operation. All default parameters will be inherited from this style.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Tool Diameter	This is the diameter of the current tool in drawing units. If the tool diameter is 0, the diameter from the tool information stored in the tool library for the given tool number will be used.
Tool Number	The ToolNumber is used to identify the current tool. If ToolNumber changes between successive machine ops a toolchange instruction is created in gcode. ToolNumber=0 is a special case which will not issue a toolchange. The tool number is also used to look up tool information in the current tool

	library. The tool library is specified in the containing Part, or if this is not present in the Machining folder level. If no tool library is defined the Default-(units) tool library is assumed.
Tool Profile	The shape of the cutter If the tool profile is Unspecified, the profile from the tool information stored in the tool library for the given tool number will be used. *EndMill | BullNose | BallNose | Vcutter | Drill | Lathe*
Work Plane	Used to define the gcode workplane. Arc moves are defined within this plane. Options are *XY | XZ | YZ*

Tool Libraries

Libraries of tools can be maintained in the system tab's ***Tools*** folder.

Multiple libraries can be defined. This may be useful to group tools for specific purposes, materials or drawing units. It may also be convenient to create a master library of all tools, then smaller libraries or 'palettes', customised to specific jobs, into which tools from the master library can be copied.

Tool libraries can be specified in CamBam drawings in the ***Machining*** options or ***Part*** objects. Libraries specified at the ***Part*** level will take precedence over any set at the ***Machining*** level.

Each machining operation can specify a ***Tool Number***. This number is used to look up information about that particular tool in the relevant tool library.

If no tool library is specified in the drawing, the default libraries will be searched for entries of this tool number. The default libraries are labeled 'Default-in' and 'Default-mm', where the units of the current drawing will be used to choose the correct library according to the '-in' or '-mm' suffix.

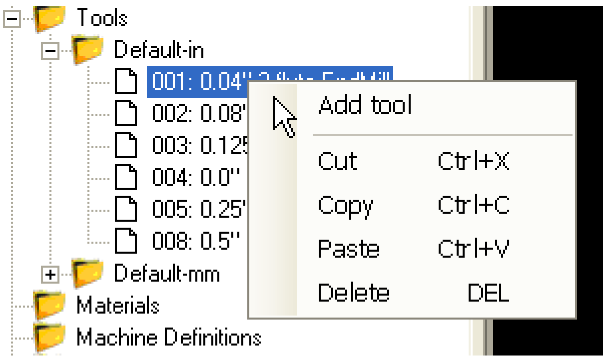
Tool numbers can also be set at the ***Machining*** and ***Part*** levels. If a tool number is set at the machining level, this will be the default tool, used by all parts and machining operations, unless explicitly set in the part or machining operation. The tool selected for the part will override any default machining tools and will be used for all operations within the part, unless they contain non zero tool numbers.

The tool definitions in the tool library contain information such as tool diameters and profiles, which can be used in the referring machining operation. If the tool diameter or profile is set explicitly in the machining operation then this will take precedence over the information from the tool library.

It is possible to use tool numbers without having any matching entries in the tool libraries. In these cases the tool diameter and profile must be defined in the machining operation.

Managing tools

Like the other system libraries, new tools and tool libraries can be create from the context menus presented when right clicking the system tab's ***Tools*** folder and tool library sub folders.



Tool Properties

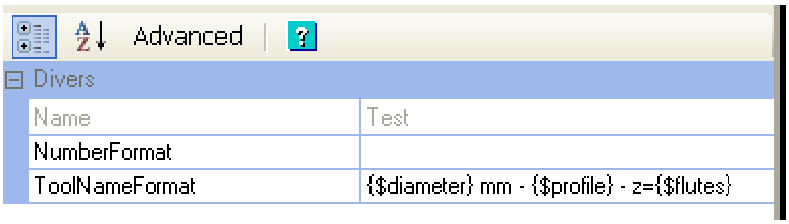
Tool libraries and definitions are a relatively new addition to CamBam. Some of the properties available in the tool definitions are intended for future functionality, but for the current release can be considered for informational use only.

Axial Depth Of Cut *Informational*	The maximum (Z) depth of cut for this tool.
Coating *Informational*	
Comment	This is a text value that can be included by the post processor when using the {$tool.comment} macro from with the ToolChange post processor section.
Diameter	The diameter of the cutting part of the tool. This will be used to calculate toolpath offsets. For V cutters, this should be set to the diameter of the cut at typical depths of cut.
Flute Length *Informational*	The length of the cutting part of the tool.
Flutes *Informational*	The number of cutting teeth or flutes.
Helix Angle *Informational*	The helix angle for spiral type cutters.
Index	The tool number that uniquely identifies the tool within the library. The tool index will be used when looking up tool numbers referenced with the CamBam drawings. The tool index will also be used within gcode when signaling tool changes etc. This should be set to match any corresponding tool tables used by the controller which may contain tool height offsets etc.
Length *Informational*	The total length of the tool that typically extends from the collet.
Material *Informational*	Material from which the cutter is made.
Max Ramp Angle *Informational*	The maximum ramp angle. To be used for lead move calculations in future releases.
Name	The descriptive name of the tool which will be used in drop down tool selection lists within the drawing. The name can be automatically calculated from tool diameter, profile and other parameters by using ***the Tool Name Format*** property of the parent tool library.
Notes *Informational*	Free format text notes relating to the tool.
Part Code *Informational*	A general identifier that may be useful to relate the tool to an external library or tooling catalogue.

Radial Depth Of Cut *Informational*	The maximum 'stepover' to be used by this cutter for crossover cuts.
Shank Diameter *Informational*	The diameter of the non cutting shank of the tool.
Tool Change	The tool change property may contain text that will be used in the post processor when a tool change condition occurs. The code in this property will be output to the gcode file and will be used in preference to the default tool change definitions specified in the post processor (for this tool only).
Tool Profile	The shape of the tool profile: *End Mill* *Bull Nose* *Ball Nose* *V-Cutter* *Drill* *Lathe*
Tooth Load *Informational*	Feed per tooth. Intended for use in automatic speed and feed calculations in future releases.
Vee Angle *Informational*	The angle of the V cutter.

Tool numbering and naming

Tools can be re-numbered by simply changing their index number in the property grid. If the number entered already exists, the numbers of the following tools will be staggered.



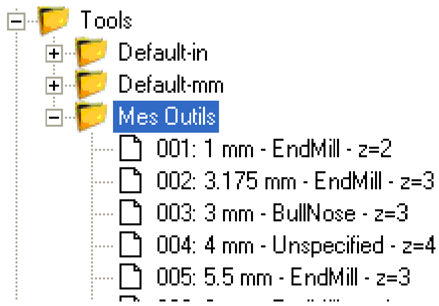
It is also possible to automatically rename using an expression entered into the tool libraries ***Tool Name Format*** property. The expression can contain the following macros:

{$diameter} = Tool diameter
{$flutes} = Number of teeth / flutes
{$index} = Tool number
{$length} = Tool length
{$partcode} = Part code
{$profile} = Tool shape / profile
{$veeangle} = Tool Vee angle

The tool name will be recalculated whenever the tool properties have changed. It is also possible to rename all the tools if the format expression has changed by using the ***Rename all tools*** from the tool library context menu.

The following image shows the tools renamed using the following format expression.

```
{$diameter} mm - {$profile} - z={$flutes}
```

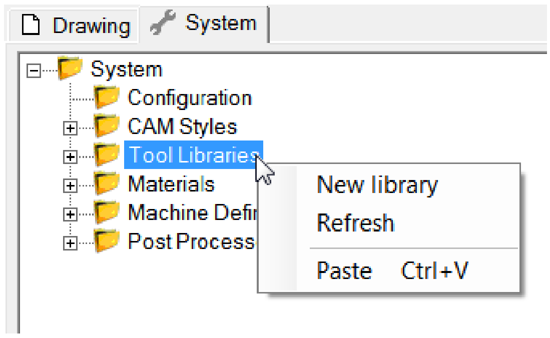


Managing Libraries

Like for tools, libraries can be copied, pasted, deleted or moved with all their content.

These operations works the same as for tools, but using the context menu of a library folder.

Create a new library: Two possibilities ; either through the tool libraries main menu with ***New Library*** or by copy / paste or cut / paste from an existing library.

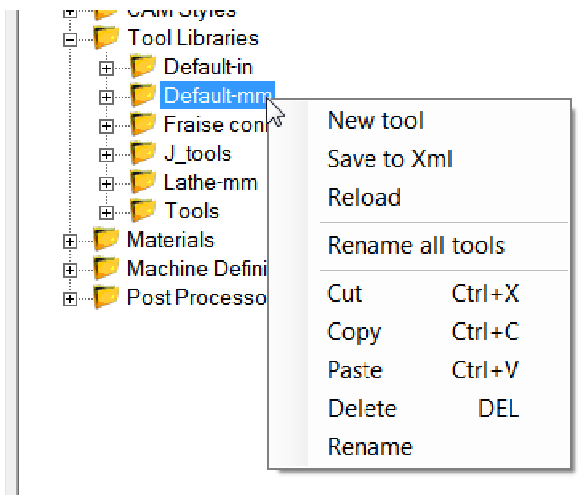


As long as you have not saved the libraries, you can restore the values saved on the hard disk by the ***Refresh*** command.

A ***Paste*** command is also available in this menu. It is identical to the ***Paste*** command in a library folder.

Copy and paste a library: You can cut, copy and paste a library from the context menu of this library with the ***Cut/Copy/Paste*** commands.

Delete Library: You can delete a library with the ***Delete*** command of the Library context menu, or by pressing the **`Del`** key on the keyboard when this library is selected.



Reload: Reloads the version of the library that is on the hard disk.

Save to XML: Saves the library to the hard disk. Libraries will also be saved by using the CamBam ***Tools/Save settings*** menu command. This backup will be automatic if the menu option ***Tools/Save settings on exit*** is checked

All tool libraries are saved as an .xml file (text files) in the tools folder of the CamBam system folder. You can quickly access this folder through the menu command ***Tools/Browse System Folder***.

Speeds and Feeds Calculator

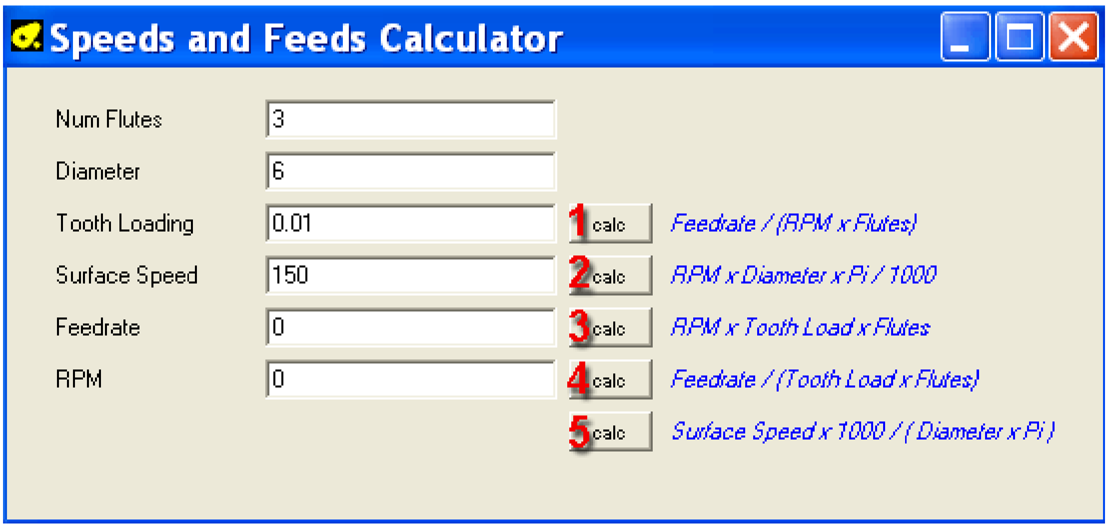
The speeds and feeds calculator is accessed from the context menu shown when right clicking a machining operation. This can be used to calculate feed rates, spindle rpms and other machining parameters.

The speeds and feeds calculator is rather basic at the moment. It requires an understanding of the theory of the calculations involved. It will also require information from external references, such as tooling data sheets from cutter manufacturers, and machinist's lookup tables.

Speed and feed formulas should also be considered as a rough guideline and are no substitute for practical experience gained, working with specific machines, cutters and materials. The formulas are often based on reference data which assumes optimal cutting conditions, coolant, rigid machines, and are often targeted towards industrial applications to optimise productivity and not necessarily tool life.

Many other factors will also need to be taken into account when judging appropriate speeds and feeds, such as: Machine rigidity and backlash, Spindle power, Sharpness of tooling, Depth of cut, Finishing or Roughing operations etc.

Some information (such as ***Tool Diameter*** and ***Cut Feedrate***) may be taken from the machining operation selected, or from the tool libraries (***Num Flutes***). No information is currently fed back into the machining operation, so the results of any calculations will need to be manually copy and pasted into the appropriate parameters.



Num flutes Number of teeth

Diameter The diameter of the tool.

Tooth Loading Feed per tooth in (inches or mm). This information will need to be looked up from cutter manufacturer data or machinist reference tables.

Surface Speed Cutting speed in m/min or inch/min. Also will need to be looked up from cutter manufacturer data or machinist reference tables.

Feedrate Feed rate in mm/min or inch/min.

RPM Rotation speed of the spindle in revs per min.

Usage

The general working method is to start with the ***Number of flutes*** and ***Diameter*** properties, which should remain fixed. Then enter the recommended ***Tooth Loading*** and ***Surface Speed*** values suggested for the cutter / stock material combination, taken from reference or manufacturer data.

The aim is to find suitable ***Feedrate*** and ***RPM*** values, which can then be fed back into the machining operation.

The values of ***Feedrate*** and ***RPM***, suggested by the formulas, may not be possible given the limitations of the CNC machine. In these cases, the machine's limits will be fed back into the calculation to determine the effect this will have on the tooth loads and surface speeds.

In the example image, the values of ***RPM*** and ***Feedrate*** have been set to 0 to highlight that these values will be calculated from the other parameters. In this case, a 6mm diameter cutter with 3 teeth, a tooth load of 0.01mm per tooth and a surface speed of 150m/min.

The buttons numbered 1 through 5 on the image are used to calculate a parameter based on other variables. The formula, and dependent variables used, are shown to the right of the calculate buttons.

1) Clicking button 5, will calculate the spindle speed (RPM) from the surface speed and tool diameter.
In this example, we get 7958 rev/min.

2) Clicking button 3 will then calculate the feed rate from the spindle rpms calculated in step 1, the tooth load and number of flutes. In this example the result is 238.74 (m / min).

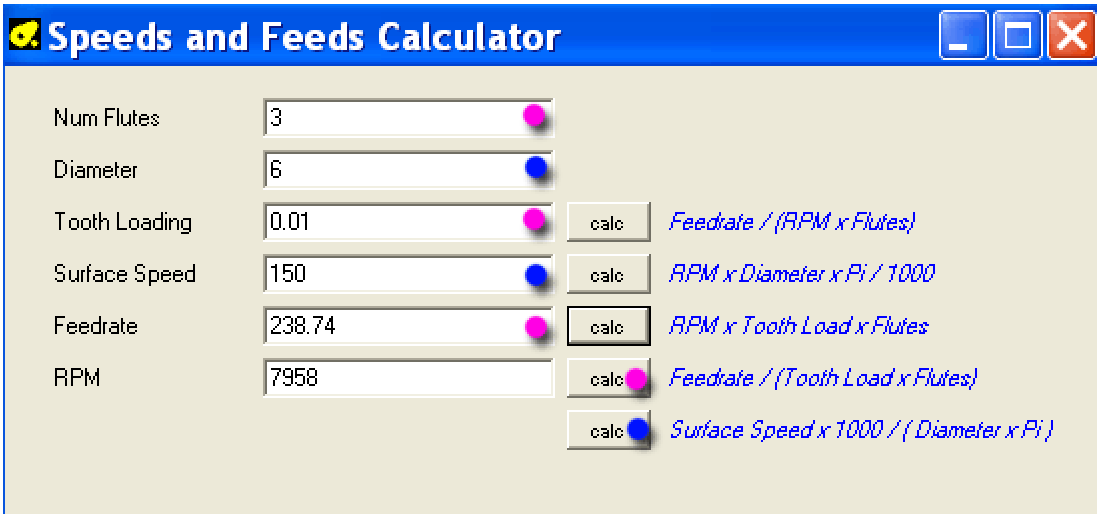
Warning: When the speed and feeds calculator is opened, it will contain feedrate and spindle speed information from the selected machining operation. These values may need to be recalculated to attain accurate values given the current tooth load and surface speeds.

Adjust the calculations based on the limitations of hardware

It is not always possible to use the ideal values calculated. The spindle may not turn fast enough, or, conversely slow enough. The machine may also not be able to achieve the required feed rate. In these situations, it will be necessary to compromise and change the values to suitable limiting values.

Buttons 1, 2 and 4 are used to calculate the value of their associated parameters if the feedrate or RPMs need to be manually modified to limiting values. The rotational speed (RPM) has 2 buttons because it can be calculated either taking into account the ***Feedrate*** and ***Tooth Load***, or the ***Surface Speed***.

These adjustments should be made only after completing steps 1) and 2) above.



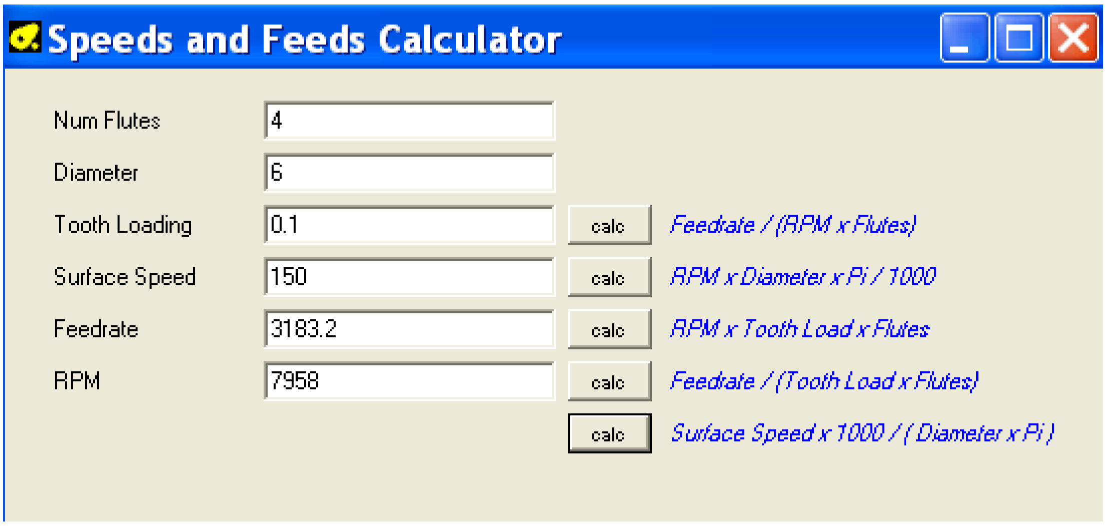
Example 1

Suppose our spindle does not drop below 10,000 rev / min, we can calculate the other parameters according to this speed. Enter 10000 for ***RPM***.

Next, calculate the other values to reflect the new spindle speed. In this case, the ***Feedrate*** and ***Surface Speed*** values. Click button 3 to calculate the new ***Feedrate*** based on the revised ***RPM*** value. The result in this example is 300 mm / min. Clicking button 2 will recalculate the ***Surface Speed***, also based on the revised ***RPM*** value. In this example the revised ***Surface Speed*** is 188.5 m / min. If this is outside the range of recommended cutting speeds, extra care should be taken and the machining strategy may need to be revised.

Example 2

The cutting parameters selected for this second example are: ***Tool Diameter*** 6mm, 4 teeth, feed 0.1 mm / tooth, cutting speed 150 m / min

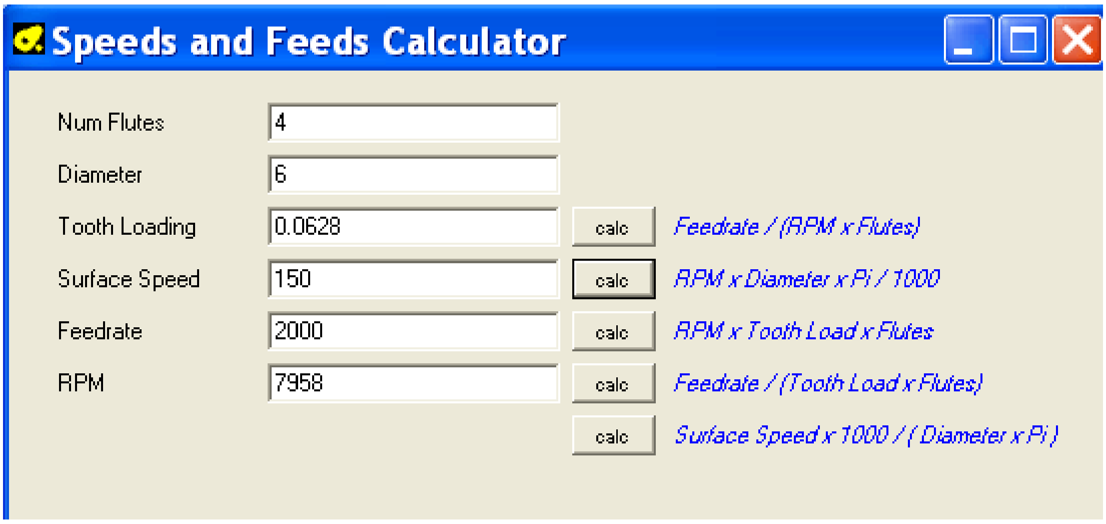


The button 5) calculation provides a spindle speed of **7958 rev/min** and button 3), a feedrate of **3183.2 mm/min**.

Suppose our machine is limited to a maximum speed of 2000 mm / min, we will enter that value as the feed rate (instead of 3183.2). We can then try different possibilities for other suitable values. In this case, we can recalculate the spindle speed, for example (depending on the feed rate) by clicking on button 4. This will give us a spindle speed of 5000 rev / min.

As with the previous example, we recalculate the ***Surface Speed*** (button 2) to verify that we are still within an acceptable range. In this case we get 94.25 m / min.

If we had wanted to keep the same ***RPM*** speed (7958) for this feed rate of 2000 mm / min (and thus keep the recommended surface speed), we could use button 1. to calculate a new ***Tooth Loading***. This would give a value of 0.0628 mm/tooth.



CAD Entities

Polyline

Polylines consist of multiple straight line and circular arc segments.

Polylines are used internally to represent toolpath shapes as they correspond well to gcode G1 (line) and G2,G3 (arc) moves.

The polylines are drawn with the mouse, making a left click to place the points. A double click on a polyline enter in edit mode and allow to see and to move the points which constitute it.

Click the middle mouse button (wheel) or hit **Enter** to confirm, or **Esc** to cancel.

Properties

Closed	*True* | *False* Open polylines have two ends and no defined inside or outside. Closed polylines are where the first and last points are the same and have a well defined inside and outside. **Note** Polylines with first and last points having the same coordinates are not necessarily closed. The closed marker should be set to True for these shapes otherwise unexpected results may occur.
Points	This property contains a collection of polyline points. Clicking the [...] button to the right of the property will open up a window where the points can be edited directly. Each point contains an X,Y and Z coordinate and a bulge parameter. ***Bulge*** is defined as tan(sweep angle/4) for arc segments, where bulge=0 is a straight line.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects. The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

Region

A region consists of a closed outer shape and a number of internal holes.

To create a region, select inner and outer shapes then use the ***Edit - Convert - To Region*** menu option, or press **CTRL+i**

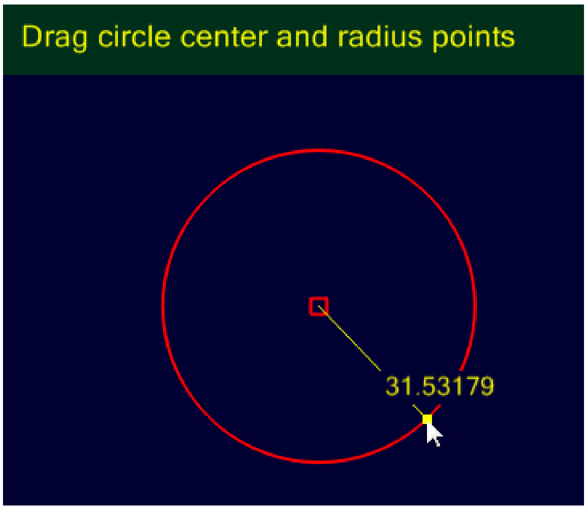
Circle

The circles are drawn with the mouse, making a left click to place the center, then after releasing the left button, moving the mouse to define the radius followed by a new click of the left button to finish.

New 1.0

A double click on a circle enter in edit mode and thus to see and move the center; click / move the center point to move it ; click / move around the circle to change its diameter.

Click the middle mouse button (wheel) or hit **Enter** to confirm, or **Esc** to cancel.



Properties

Center	The coordinates of the center of the circle.
Diameter	The diameter of the circle.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects.The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

PointList

Point lists are useful for defining points to be used for drilling operations.

A double click on a pointList allows you to enter in edit mode and to move or add points; click / move a point to move it ; click on a the drawing area to add a new one.

Click the middle mouse button (wheel) or hit **Enter** to confirm, or **Esc** to cancel.

As well as drawing directly, they can be created from the ***Draw-Point List*** menu operations

Divide Geometry	Evenly divides a selected shape into a given number and inserts a point at each division. This is useful for generating a bolt hole pattern.
Step Around Geometry	Inserts a point at given distances around a selected shape.
Fill Geometry	Fills a closed shape with points.
Offset Fill Geometry	Fills a closed shape with points where alternating rows are offset by half the stepping distance.
Centers	Inserts a point at the center of each selected object.
Extents	Inserts a point at the extremities and center of a boundary rectangle enclosing each selected object.

Properties

Points	This property contains a collection of points. Clicking the [...] button to the right of the property will open up a window where the points can be edited directly.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects. The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

Rectangle

The rectangles are drawn with the mouse, making a left click to place an angle, then after releasing the left button, moving the mouse to define the diagonal followed by a new click of the left button to finish.

A double click on a rectangle enter in edit mode and alow to move the points of angles (the object remains rectangular)

Click the middle mouse button (wheel) or hit **Enter** to confirm, or **Esc** to cancel.

Properties

Corner Radius	This will round the corners of the rectangle to a given radius.
Height	The height of the rectangle.
Lower Left	The coordinates of the lower left corner of the rectangle.
Width	The width of the rectangle.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects. The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

Text

Properties

Bold	Bold Font Style.
Char Space	This option scales the width used for each character. The default is 1. A setting of 2 would double the space used for each character (but not the character itself).
Font	This is the name of the font to use for the text.

Height	This is the text height in drawing units. The height is based on the **em** square, which is a property of the font that describes the largest dimensions possible of the font. To obtain an accurate height, given the text and font entered, the ***Edit - Resize*** command should be used.
Italic	Italic Font Style.
Line Space	This scales the distance between each text line. The default is 1.
Location	This is the first and at the moment, only alignment point. The TextAlignmentH and TextAlignmentV options are all relative to this point.
Regular	Regular Font Style.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Text	The text to enter. To enter multi line text, click the [...] button after this property.
Text Alignment Horizontal	Left, Center or Right (relative to ***Location***).
Text Alignment Vertical	Top, Center or Bottom (relative to ***Location***). NOTE: Bottom is actually the baseline of the text.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects. The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

Arc

The arcs are drawn with the mouse by clicking the two extreme points, then by positioning the center with a third click.

New 1.0

A double click on an arc allows you to enter in edit mode and to move its center start and end points.

Click the middle mouse button (wheel) or hit `Enter` to confirm, or `Esc` to cancel.

Note: While drawing or editing an arc, pressing the `Shift` key will invert the direction of the arc when moving the center.

Properties

Center	The center of the arc.
Radius	The radius of the arc.
Start	The start angle in degrees of the first arc point. Angle = 0 is along the positive X axis.
Sweep	The sweep angle in degrees from the first to second arc point. Positive angles are counter clockwise and negative angles clockwise.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects. The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

Line

Lines have multiple segments, similar to polylines, but can only contain straight sections.

New 1.0

Lines can be edited by double-clicking on it, like polylines

Surface

These are triangular faceted 3D meshes imported from an STL or 3DS file

New 1.0

They also can be created in CamBam with ***Draw/Surface/extrude solids*** and ***Draw/Surface/Fill***

Surfaces are triangle face meshes imported from STL, STEP and 3DS files.

Spline

Splines (or NURBS) are curves that are defined by a series of control points and weights.

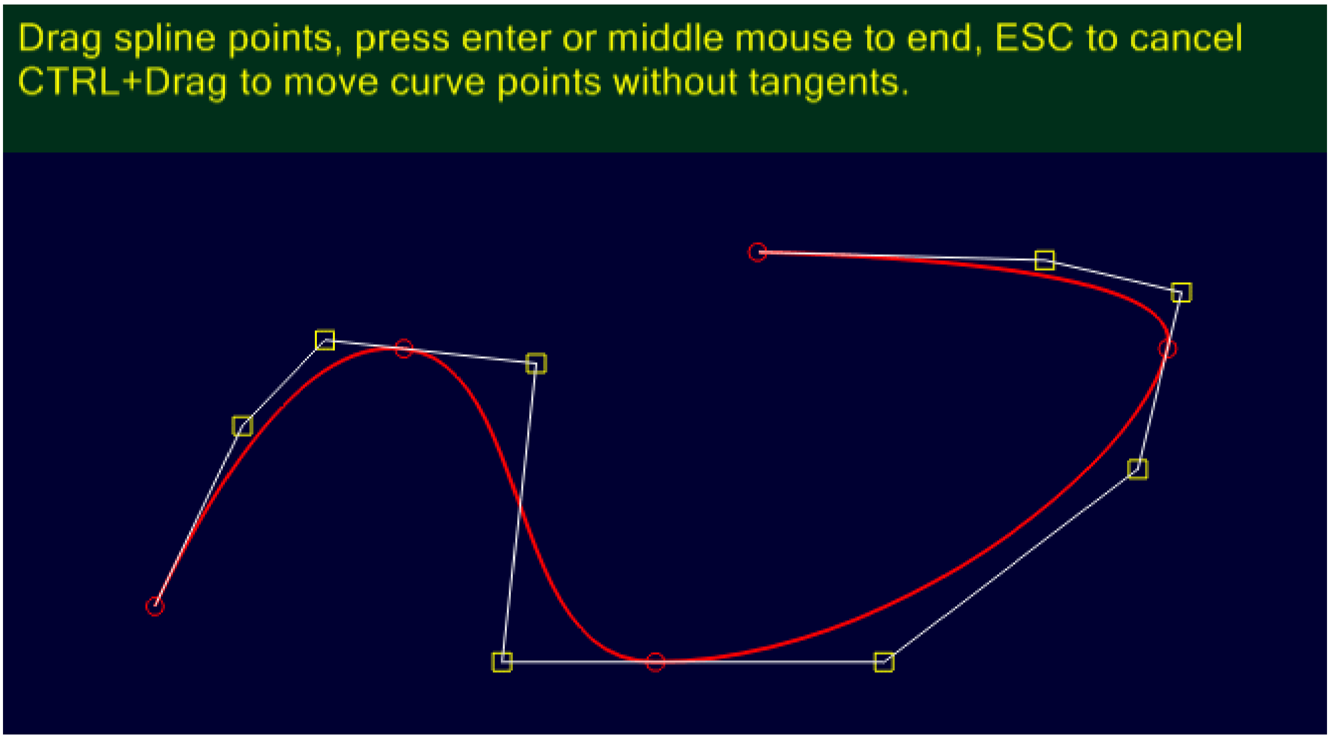
New 1.0

The splines are drawn with the mouse, making a left click to place the points of inflection.

A double click on a spline enter in edit mode and thus to see and move the control points that constitute it, as well as the points that make it possible to modify the curvature locally.

Pressing the **Ctrl** key while moving a control point (red circle) temporarily breaks the link with its curvature control points (yellow squares)

Click the middle mouse button (wheel) or hit **Enter** to confirm, or **Esc** to cancel.



Script object

New 1.0

A **script object** works like other drawing objects, but the geometry is created from a Python script.

See Script Object section

Bitmap object

New 1.0

A Bitmap Object is a picture that can be used as a "blue print" to draw over, that can be "numerized" to extract contours or just used as a picture to enhance the design.

See Bitmap Object section

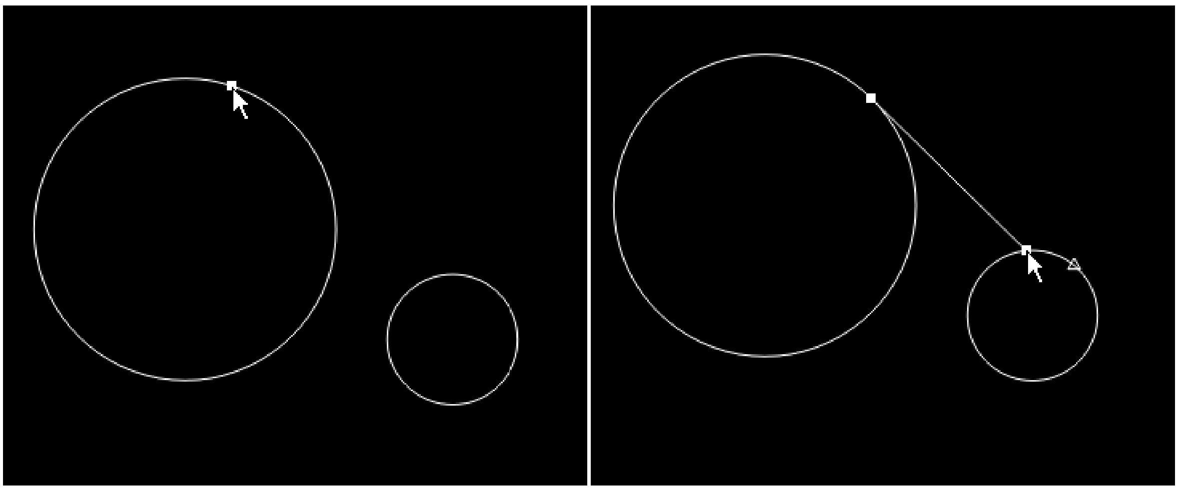
Draw/More

Tangent

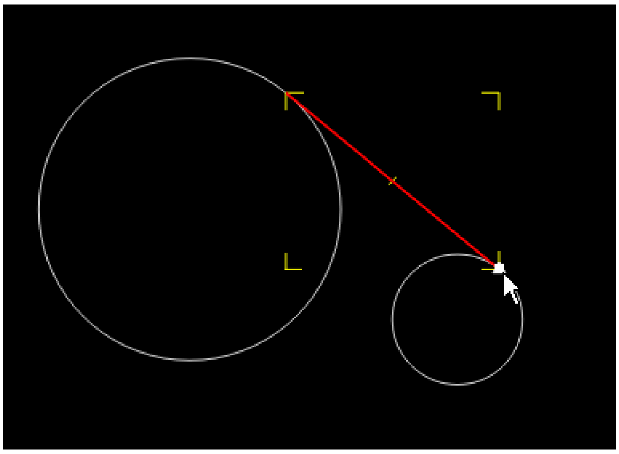
Draws a line between points of tangency of circles or arcs (or arcs contained in a polyline)

After choosing the ***Tangent*** function, move the mouse over the first circle / arc; the end of the line will follow and stick to the curve ; click the left mouse button to fix that first point (which will “slide” along the curve depending on the orientation you give to the line)

Then move the mouse cursor on the second circle / arc; a triangle will appear at the nearest point of tangency. This triangle will change places depending on the position of the mouse.



Hook the end of the line on the triangle, then click on the left button to finish.



Note: It may be easier to "feel" the magnetization to the curve/triangle if you disable grid snapping

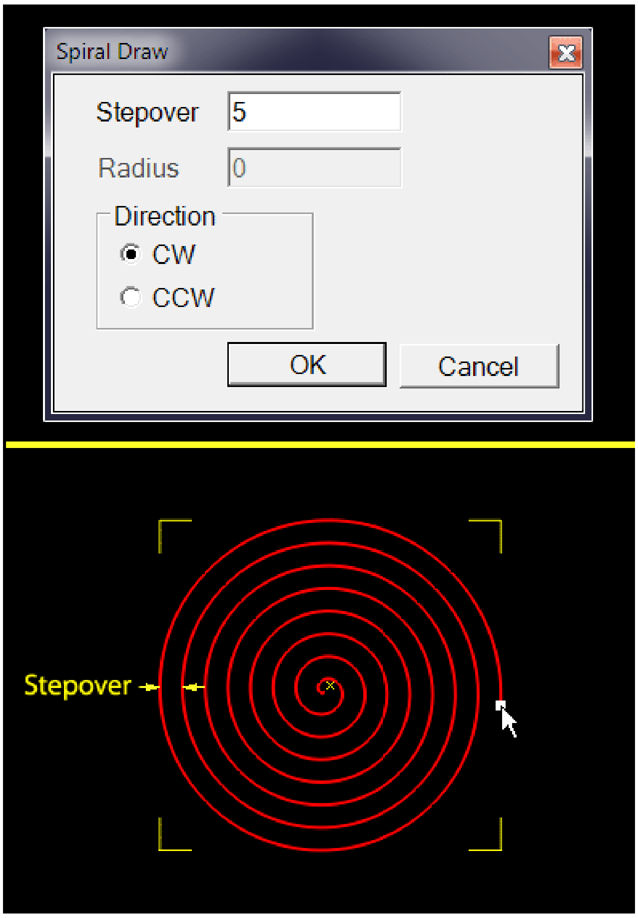
Flat Spiral

Draws a flat spiral.

The ***Stepover*** is the spacing between the turns in current drawing unit.

The ***Direction*** is the winding direction of the spiral, and can be clockwise (CW) or counter-clockwise (CCW)

After filling in these two informations, draw the spiral starting at the center and then defining the radius (as if to draw a circle)

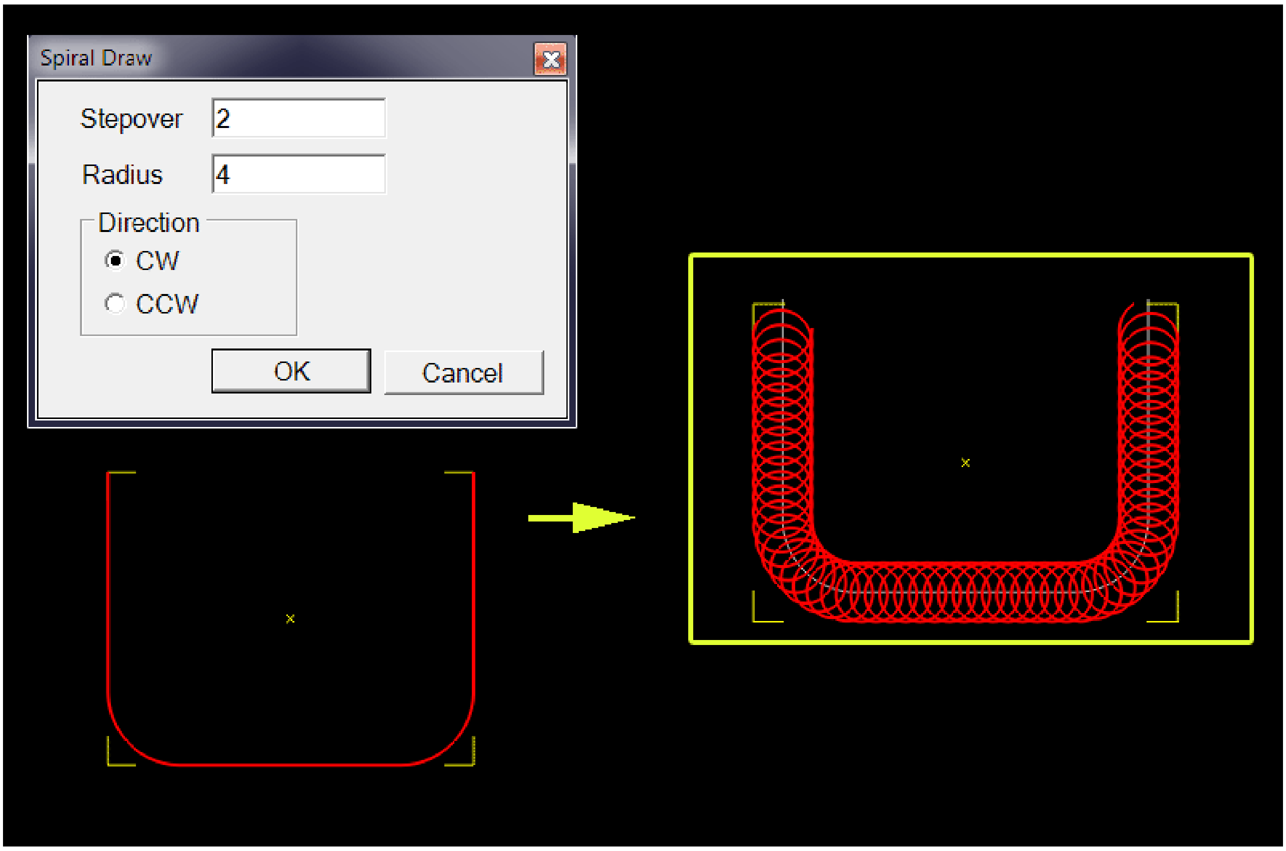


Spiral Path

Draws a spiral along a path.

First, select a shape that will serve as path, then choose the ***Spiral Path*** command from the menu.

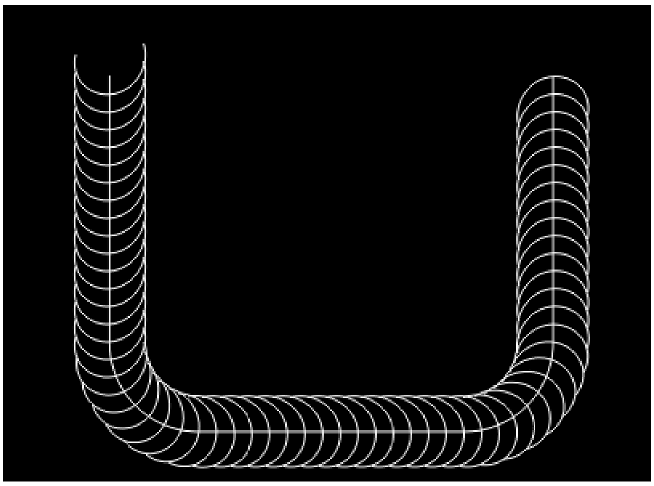
The same window as for a flat spiral will open. It allows to choose the ***Stepover*** and ***Direction***, as before, but also to specify the ***radius*** of the spiral.



Trochoidal Path

Same principle as above, but this time it is a succession of arcs that follow the path, and not a spiral.

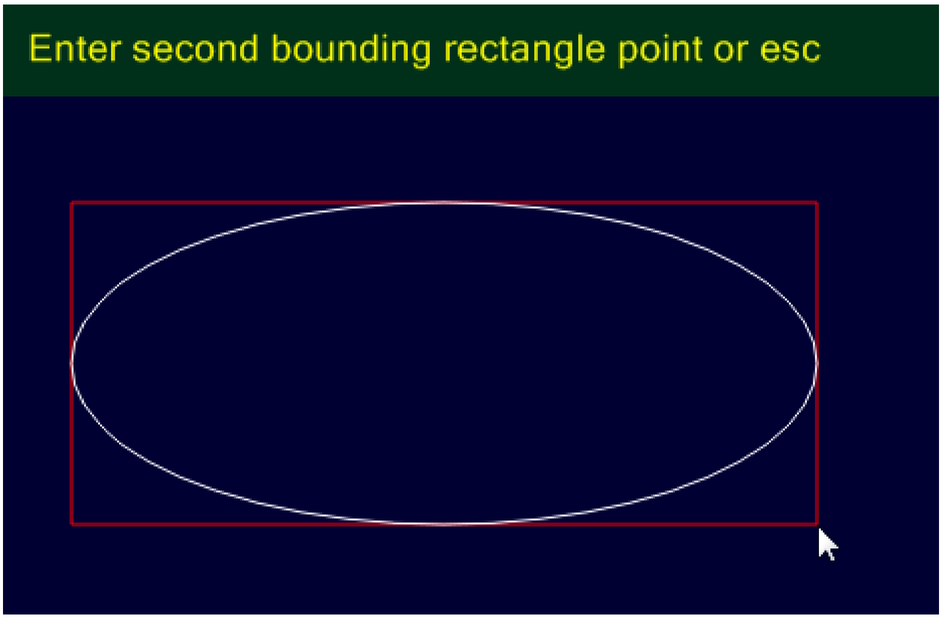
Combined with an engraving operation, this path allows you to quickly remove material with less stress for the tools.



Ellipse

Allows drawing ellipses by drawing a bounding rectangle with the mouse.

The ellipse once drawn becomes a polyline, consisting of arcs and lines.



Script Objects

A ***script object*** works like other drawing objects, but the geometry is created from a Python script.

Machining operations can be based on script objects in the same way as static drawing objects. They can also be moved, rotated and copied. Any changes made by the script will be automatically picked up by associated machining operations.

Scripts can be useful for generating parametric shapes (such as a gear, or tabbed box), creating copies of, or manipulating other drawing objects. Script objects can also be used to provide general functionality such as an animation.

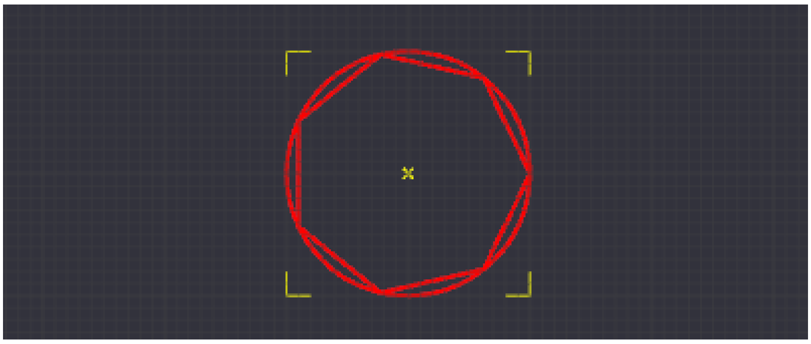
Drawing scripts will typically create primitive shapes and add them to the ***Entities*** collection property using:

```
this.Entities.Add(...)
```

The following example draws an N-sided shape of a given radius.

```
# variable that define the shape...
n = 7
radius = 10
# create a polyline object...
poly = Polyline()
poly.Closed = True
for i in range(0,n):
    th = i*2*Math.PI/n
    poly.Add( radius * Math.Cos(th), radius * Math.Sin(th), 0)
# add the polyline to the list of drawing objects...
this.Entities.Add(poly)
# multiple drawing objects can be added...
this.Entities.Add(Circle(0,0,radius))
```

Resulting in this…



Script objects will usually run when required to update geometry, but can be forced to run by selecting them in the drawing tree and pressing **F5**.

Use ***Edit - Explode*** to turn script objects into their component static drawing objects, which can be used with previous CamBam versions or further manipulated using CamBam CAD operations.

When a script object is inserted using the ***Draw - Script object*** command, or by clicking the script toolbar icon , a default script is inserted. The contents of this script can be set in the ***Default Script Entity*** system configuration property.

Properties

Script	The script text to be executed. The text can be edited by clicking the [...] button to the right of the property, or by double clicking any drawing objects that are created by the script.
Execute On Update	If *True*, the script will be executed automatically when required, such as when the drawing loads or if the script is modified. If *False* the script must be run manually by selecting the object and pressing **F5**. This can be useful for non-drawing scripts such as animations that only need to be run on demand.
Entities	The ***Entities*** property contains a list of drawing objects that are generated by the script.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects. The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

Copying drawing objects

Making copies of other drawing objects is a simple but powerful technique to allow easily managed drawings. Entities can be copied by looking for their ID, adding objects from a specific layer or including them from external drawings.

The following example makes 4 copies of a source object with each copy offset 18mm along the X axis.

This is much like using the ***Transform - Array Copy*** function with the benefit that any changes to the source object will be automatically reflected in the copies the next time the script is run by pressing **F5**.

```
# find drawing object with ID=7
ent = doc.FindPrimitive(7)

for i in range(1,5):
    clone = ent.Clone()
    clone.Transform.Translate(i*18,0,0)
    this.Entities.Add(clone)
```

Using Layers

Hard coding source object IDs in scripts can be error prone. If the source object IDs change, any referring scripts will fail.

An alternative is to copy objects from a source ***Layer*** specified by name.

```
for ent in doc.Layers["Border"].Entities:
    this.Entities.Add(ent.Clone())
```

Copying from external files

```
cbfile = CADFile()
filename = "samples\\skull-big-foam.cb"
cbfile.Open(FileUtils.GetFullPath(CamBamConfig.Defaults.SystemPath,filename))

for ent in cbfile.Layers["Default"].Entities:
    this.Entities.Add(ent.Clone())
```

Including scripts

The script object includes scripts from the ***Script*** property of the parent layer, as well as the ***Script*** property at the top level of the drawing file. This allows functions, classes or variables to be defined once in the drawing, then referenced in any number of drawing objects.

CamBam inserts an internally generated header block before each script is run, to import commonly used name spaces. This helps keep the user scripts succinct. The internal header contains the following code.

```
import clr
clr.AddReference('System')
from System import *
clr.AddReference('System.Drawing')
from System.Drawing import *
from CamBam import *
from CamBam.CAD import *
from CamBam.CAM import *
from CamBam.UI import CamBamUI
from CamBam.Geom import *
```

When the script is run, the script components are appended together, so the actual script executed is given by:

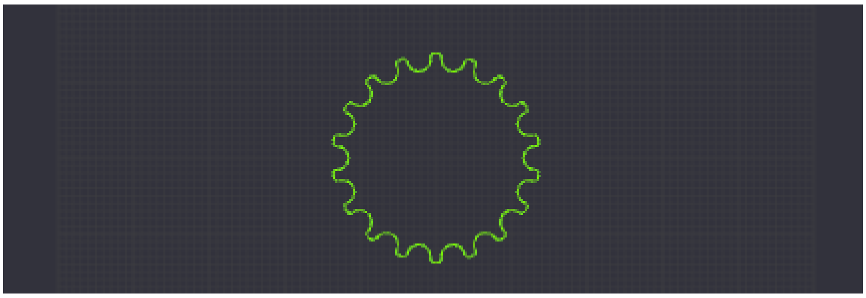
```
Header + Drawing.Script + Layer.Script + ScriptEntity.Script
```

Importing external scripts and modules

Scripts can also be imported from external sources using the standard Python ***import*** instruction. This provides a way of managing a common library of drawing objects and other functions which can be shared throughout multiple drawings. Any changes to modules will be automatically picked up whenever scripts that refer to them are run.

This example will import the ***Pulleys.py*** module from the common scripts folder.

```
# import the module
import Pulleys
# declare a timing pulley object with pitch=5mm and 18 teeth
HTD5_18 = Pulleys.HTDPulley(5.0,18)
# add the pulley outline to the script object's drawing entities
this.Entities.Add(HTD5_18.getFrontView())
```



The `import Pulleys` line instructs Python to search for a module called ‘Pulleys’. Python will look in a search path including the current drawing’s folder and the CamBam system \ scripts folder.

There is a script in the CamBam system\script folder called Pulleys.py so that module is found and imported. CamBam also provides the following modules: Pulleys.py, StepperMotors.py and Bearings.py. The ***C-BEAM-Belt-Drive.cb*** sample file makes use of these modules to generate drawing objects.

Modules are imported in their own name-space, any names used from the module need to be prefixed with the module name, such as ‘Pulleys.HTDPully’. It is also possible to import module names into the current scripts name-space using a line like:

```
import Pulleys
# import just the HTDPulley name
from Pulleys import HTDPulley
# or import all names from the Pulleys module
from Pulleys import *
```

Then names like ‘HTDPulley’ can be used without the ‘Pulleys’ prefix.

Script modules are pretty much like any other script, except they need to be ‘self contained’. The internally generated header block described above will not be automatically added, so all required modules and name-spaces need to be imported into the module.

The module search path can be defined within the script to allow importing standard Python modules.

```
# append the module search path
import sys
sys.path.append("C:/Python27/Lib")
# import the Python CSV handling module
import csv
reader = csv.reader(open("C:/devt/test.csv", 'rb'))
for row in reader:
    print str(row)
```

Animations

Scripts can be used to animate drawing objects by applying transformations such as rotations and translations then updating each frame within a loop. It is possible to rotate, scale and pan the drawing during the animation.

Animation scripts should be marked with ***Execute On Update*** set *False*, otherwise they will be run as soon as the file is loaded.

The script should also set the ***Animating*** property of the current view when the animation starts, then clear it at the end of the script. This allows the view to provide some optimizations as well as preventing non-selected objects to be displayed faded during the animation.

The script should also preserve the original transformation state of any objects being animated, then restore the transforms at the end of the script, otherwise the drawing will be left in a state with the source objects transformed as at the last animation frame.

The following example will rotate and translate an object with ID=1 along the X axis.

```
# find drawing object with ID=1
ent = doc.FindPrimitive(1)

# save the current transformation
tsave = ent.Transform

# start animating
view.Animating = 1

cycles = 4

for f in range(0,cycles*360):
    # apply transformations to the source object
    trans_translate = Matrix4x4F.Translation(f*0.1,0,0)
    trans_rotate = Matrix4x4F.RotationZ(-f/180.0*Math.PI)
    ent.Transform = tsave * trans_rotate * trans_translate

    # update the display
    view.RefreshView()
    app.Sleep(5)

# restore the original transformation
ent.Transform = tsave

view.Animating = 0
```

Bitmap Objects

Use ***Draw - Bitmap***, then select a drawing file such as a jpeg or bitmap. CamBam will detect edges in the bitmap, allowing you to scan a drawing, or draw a shape in an external application like Photoshop, then load it into CamBam and insert a machining operation based from it. Changes to the drawing will be picked up automatically by the machining operation.

The bitmap file is linked to the CamBam drawing, not saved within it, so transferring CamBam drawings containing bitmaps will also require the bitmap file to be copied.

To break the link to a bitmap file and include the vectors within the CamBam drawing, select the drawing object then use ***Convert To Polylines*** or `CTRL+P`.

Showing Bitmaps or Vectors

The ***Display Bitmaps*** and ***Display Vectors*** properties can be used to control what is displayed in the drawing.


showing bitmap only


showing bitmap and vectors


showing vectors only (***invert***=*false*)


showing vectors only (***invert***=*true*)

Smoothing

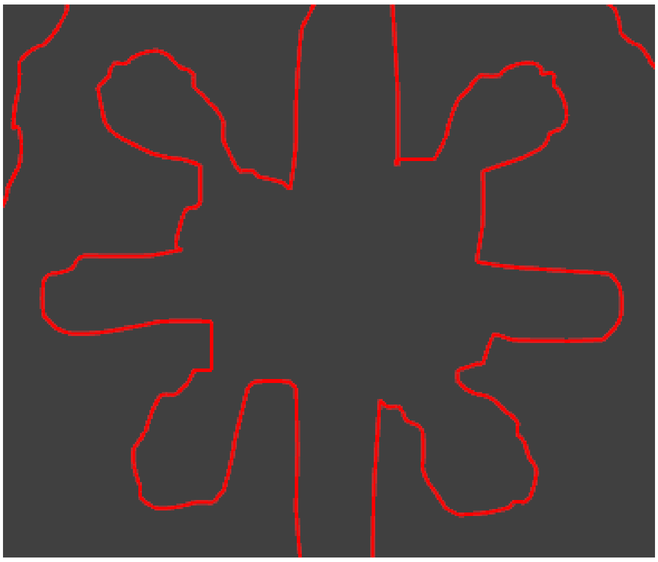
CamBam will automatically vectorize the bitmap using a quick and simple edge detect. You can change the vectorization method in the bitmap properties. Selecting ***Smoothing*** = *True* will generate a spline smoothed outline. This can take a few seconds for the spline to calculate.

Smoothing Pixels can be used to control the spline fit. ***Smoothing Pixels*** = *2* usually provides good results. Going lower than 1 can take a long time and will end up with a spline trying to follow each pixel.

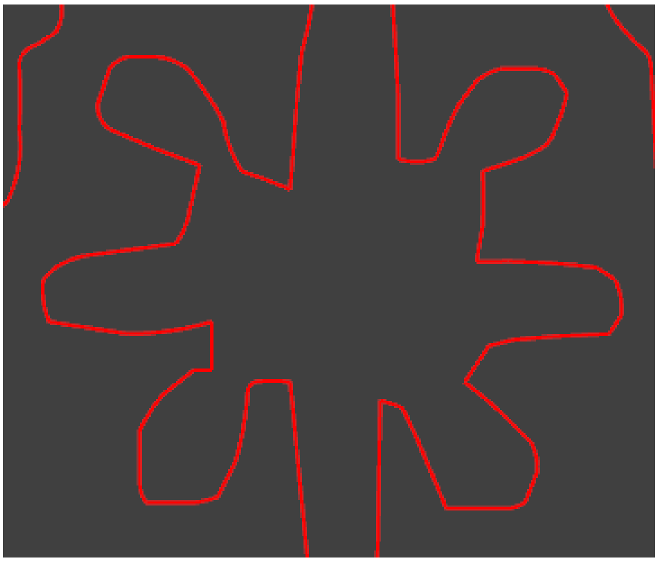
Small smoothing pixel values (around 1), can take a very long time to calculate (minutes!).

We are working to make this faster and run in a background process so as not to lock up the UI, but for now smoothing pixel settings of 2 or more are recommended.

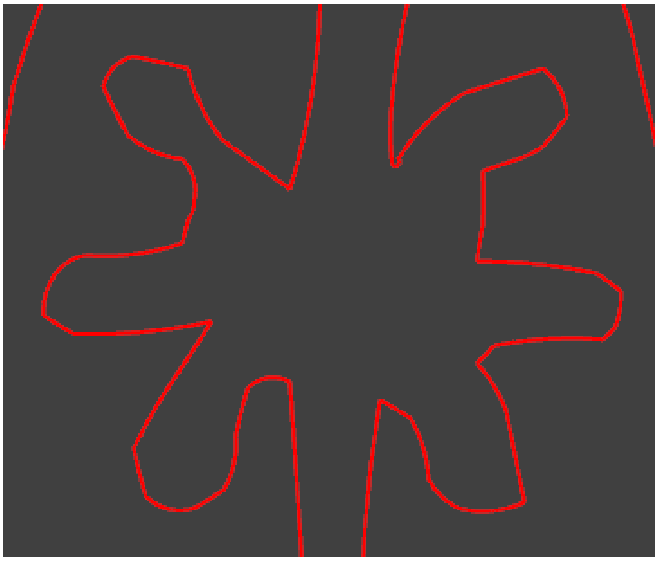
no smoothing



smooth pixels = 1



smooth pixels = 2



smooth pixels = 5

Properties

Display Bitmap	*True* | *False*. Controls whether the bitmap image is displayed.
Display Vectors	*True* | *False*. Controls whether the detected edge vectors are displayed.
Height	The height of the image in drawing units. The pixel height and vertical resolution stored in the bitmap are used to calculate the default height value.
Width	The width of the image in drawing units. The pixel width and horizontal resolution stored in the bitmap are used to calculate the default width value.
Invert	If True the edges of lighter pixels are detected. If the image contains dark lines on a light background, the edge detect will include the bitmap outline. If False the edges of darker pixels are detected. If the image contains dark lines on a light background, only the edges of the lines are included.
Position	This is the X,Y,Z drawing location of the lower left corner of the bitmap.
Smooth Pixels	If ***Smoothing*** = *True*, ***Smooth Pixels*** controls the maximum allowed error between a spline curve and the raw outline. The error is measured in number of pixels. Smaller values will produce splines that closely match the outline, but may take longer to calculate, produces larger splines and possibly result in unwanted detail such as jagged pixel outlines. Larger values will produce faster, simpler, smoother curves but with less accuracy.
Smoothing	*True* | *False*. If True, a spline will be used to fit the raw pixel edges. If False, only the raw pixel edges are used. Raw pixel edges will be faster to calculate, but result in 'jagged' pixel edges.

Source File	The path to the source bitmap file. The following bitmap formats are supported: BMP, GIF, EXIF, JPG, PNG and TIFF The filename will be relative to the current CamBam drawing path. Saving the CamBam drawing first, before inserting bitmaps is recommend so the relative rather than full paths will be detected. Relative paths make is easier to copy the .cb file and bitmap files together without needing to change paths within the drawing.
Threshold	The brightness threshold used to determine if a pixel is set when vectorizing. A value between 0 and 1, where 0 is black and 1 is white. Note that color images use monochrome edge detection so edges between different colors with similar brightnesses may not be detected.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Transform	A 4 x 4 matrix of numbers used for general transformations of the drawing objects. The transform matrix can be used for rotations, translations and scaling about all 3 axis. *Identity* indicates no transformations will be applied to the object.

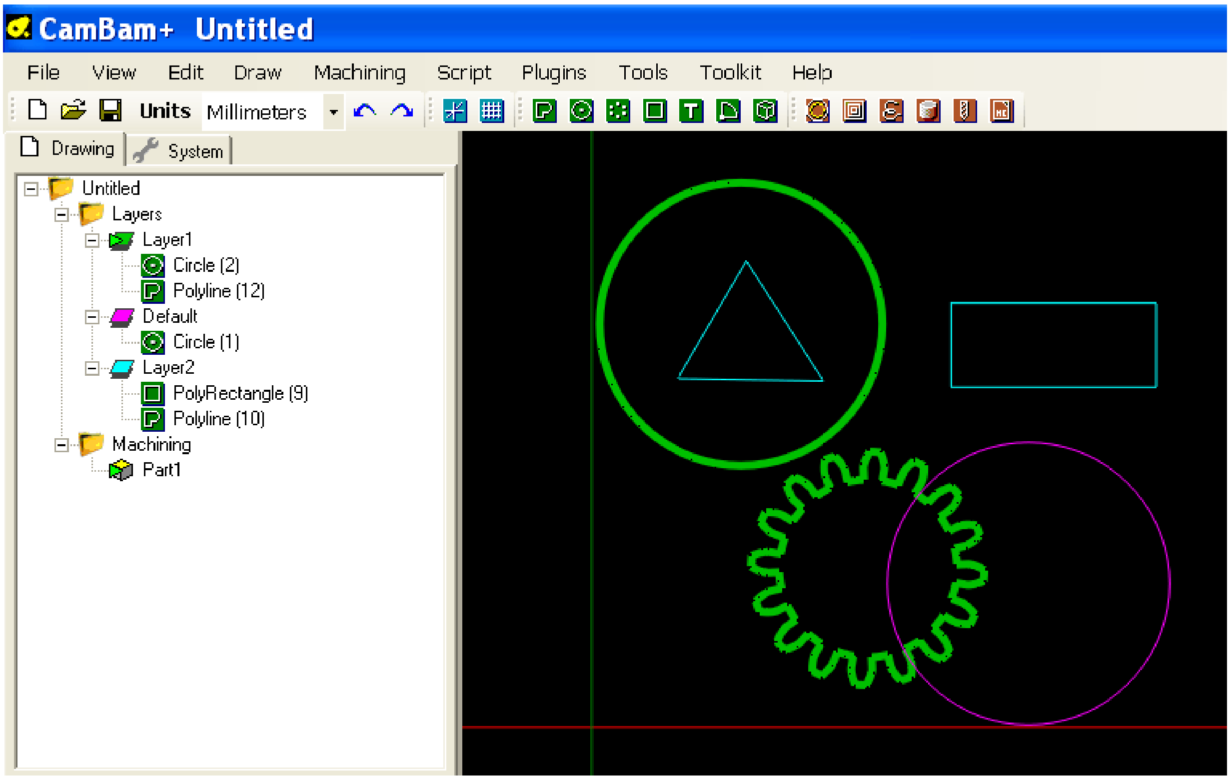
Layers

Drawing objects can be organised within multiple, color coded, layers.

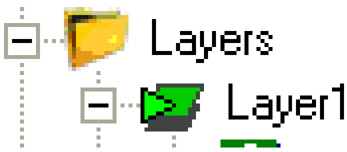
Layers (and the drawing objects contained within them), can be made hidden or visible, which can greatly simplify working on complicated drawings.

Drawing objects can be moved between layers using cut, copy and paste, or by simply dragging and dropping them within the drawing tree view.

Selecting a layer in the drawing tree allows its properties to be altered in the property grid. The appearance of the layers, such as colour and line widths, can be set in this way.

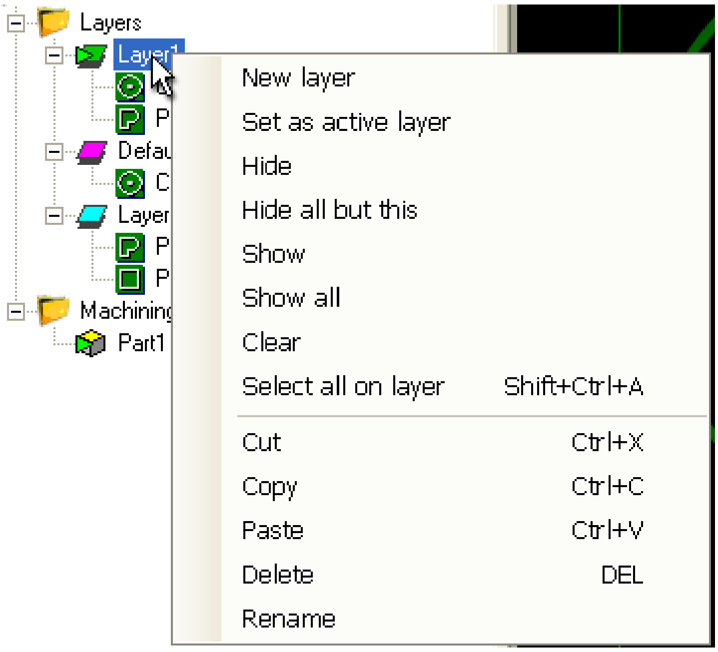


When drawing new shapes, they will be inserted into the layer marked as the ***Active*** layer, which is indicated in the drawing tree by a small green arrow icon. The active layer is set by right clicking a layer in the drawing tree, then selecting ***Set as active layer***.



Note: It is possible for the active layer to also be hidden, so new drawing objects will be inserted into the layer, but not displayed until the layer is marked as visible again.

Other operations for manipulating layers are available from the context menu, visible when right clicking on a layer.

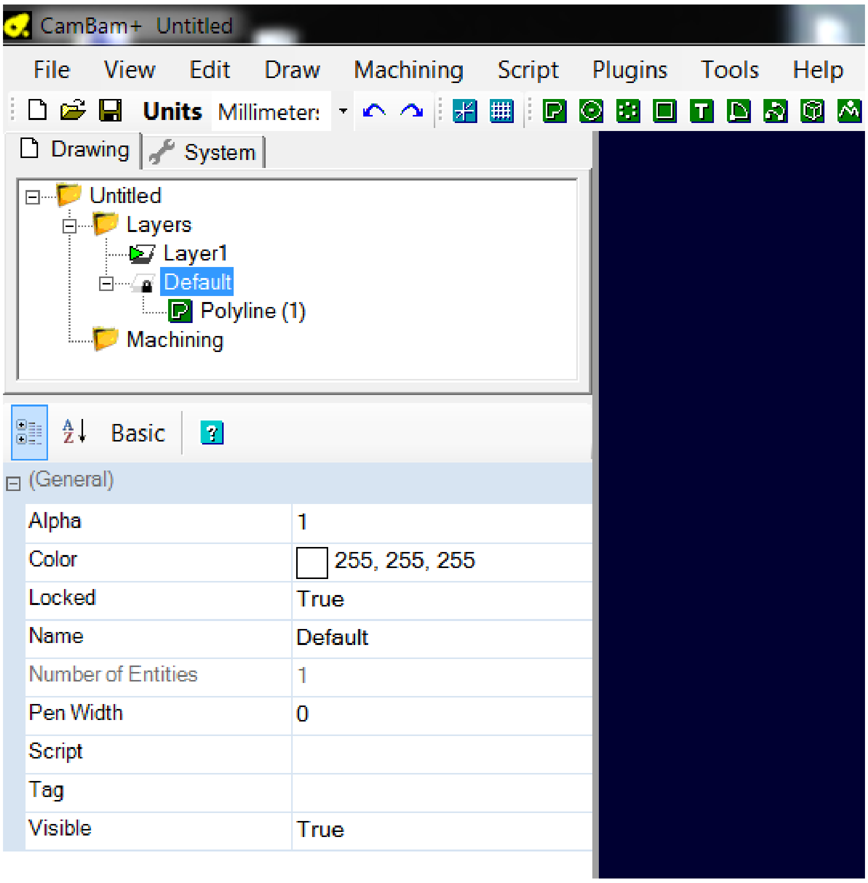


[New! 1.0]

A layer can now be locked. In this case, the objects it contains cannot be selected from the drawing area (but remain selectable in the list)

When a layer is locked, a small padlock appears on its icon. 

To lock a layer, once selected in the list, use **CTRL + Spacebar**, or toggle its ***Locked*** property to *True* in the property grid.



Layer Operations

New layer

Creates a new layer and also makes this the *active layer*. The default color of new layers can be changed in the ***Default Layer Color*** property of the system configuration settings.

Set as active layer

New drawing objects will be inserted into the current active layer.

Hide

The selected layer is marked as hidden and the drawing objects will not be displayed in the drawing view. These objects will also be prevented from being selected using operations such as ***Select All*** (**CTRL+A**). Hidden layers will be displayed greyed in the drawing tree.

Layers can be quickly toggled between visible and hidden by selecting them in the drawing tree view then pressing the **SPACE** key.

Hide all but this

Will hide all layers in the drawing, apart from the selected one.

Show

Makes the selected layer and drawing objects visible.

Show all

Makes sure all the layers in the drawing are marked visible.

Clear

This operation will delete all the drawing objects contained in the selected layer.

Select all on layer

Selects all the drawing objects on the selected layer.

Cut / Copy / Paste

Cut / Copy and Paste selected layers and all their drawing objects.

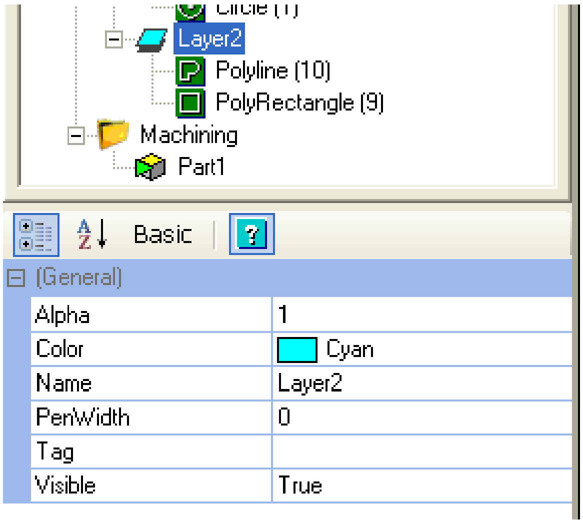
Delete

Removes a selected layer and contained drawing objects from the drawing.

Rename

Rename the selected layer. Layers can also be renamed by selecting them in the drawing tree and pressing **F2**, or by a slow double click on the layer name.

Properties

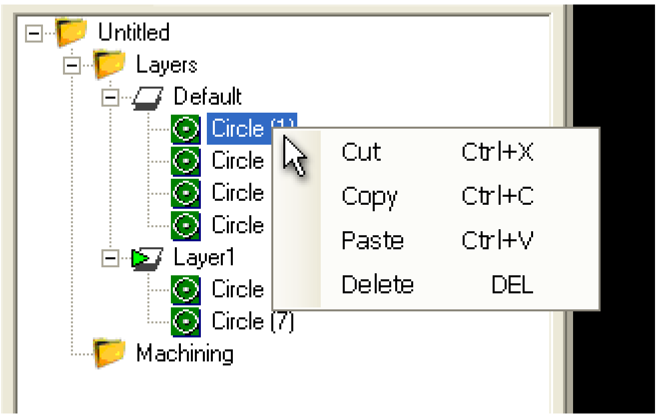


Alpha	The level of transparency of the drawing objects in the layer. 0 to 1, 1=opaque, 0 = completely transparent.
Color	Color used to display the drawing objects.
Locked *[New! 1.0]*	Locked layers prevent visible objects from being selected.
Name	The name of the layer.
Number of Entities *[New! 1.0]*	The number of entities in this layer.
Pen Width	Thickness of the drawing lines.
Script *[New! 1.0]*	This script is included into all child script entities.
Tag	A general purpose, multi-line text field that can be used to store notes or parameter data.
Visible	The state of the layer visibility: *True* = Visible, *False* = Hidden.

Moving drawing objects between layers.

You can move the drawing objects from one layer to another by simply dragging and dropping with the left mouse button.

You can cut, copy and paste the drawing objects between layers using the context menu for each drawing object in the tree or the context menu of the drawing area, or through the main Edit menu.



NOTE: The paste function will behave differently depending on which context menu it was called from:

From the context menu of the layer:
The object will be pasted into the layer that the context menu was opened.

From the context menu of the drawing view of the main Edit menu:
The object will be pasted into the selected layer, if none is selected, it will be pasted into the current active layer.

CAD Transformations

Moving

Objects can be moved by selecting them, then holding down the **SHIFT** key whilst dragging the objects with the mouse.

Using the keyboard only, selected objects can be moved holding the **SHIFT** key and using the arrow keys. This will move the object one minor grid unit in the arrow key direction (If using millimeters, this will be 1mm, if using inches then this will be 1/16"). If **CTRL+SHIFT** keys are held down, objects will be moved one major grid unit (If using millimeters, this will be 10mm, if using inches then this will be 1")

NOTE: The grid major and minor units can be defined in the system configuration, grid section.

Alternatively, the ***Transform - Move*** menu option can be used to position an object by first selecting a source point, then a destination point. This can be useful to accurately position one object relative to another, as the point selection will 'snap' to object points.

Three point Edit - Move New! [V1.0]

The ***Edit - Move*** command now supports up to 3 source and destination points.

Hold down the **CTRL** key and click to select up to 3 source points.

If only one point is selected, the move operation works as before by selecting the target destination point.

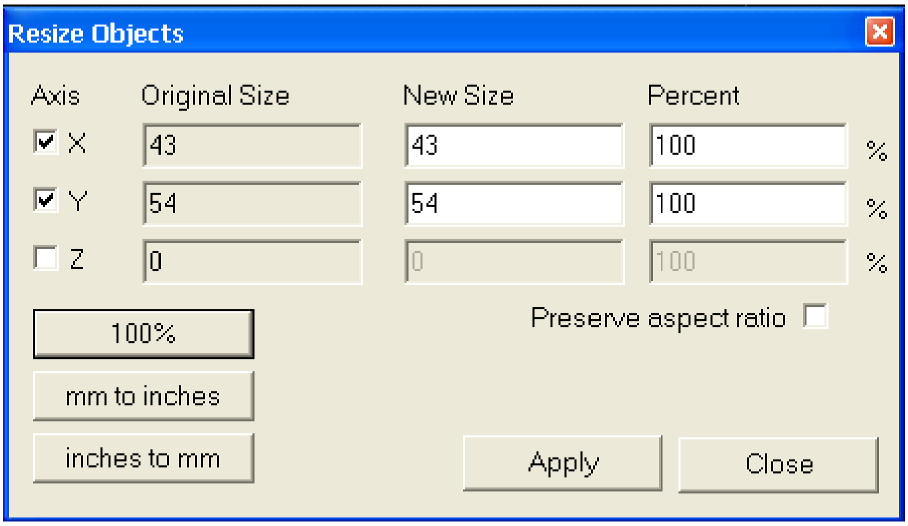
If two source points are selected, the first destination point will be the new location for the first point. The second destination point will form a line on which the moved second point lies. This in effect allows a move and a rotation in the XY plane in one operation.

If three source points are selected, the first destination point will be the new location for the first point. The second destination point will form a line on which the moved second point lies. The third destination point will form a plane on which the moved third point lies. This allows a shape to be moved in three dimensions. For example a 3D mesh could be aligned to a mounting plate by selecting three matching reference points at the center of three bolt holes.

The selected points are color coded: Red, Green, Blue to identify the current selection point.

Resizing

The ***Transform - Resize*** menu option (or **`Ctrl+E`**) is used to resize selected drawing objects.



Each axis can be scaled separately by using the check box to the left of the axis label. Unticked axis will retain their original size.

The ***Original Size*** column displays the current dimensions of the selected objects.

A specific size can be entered in the ***New Size*** column or a scaling factor entered in the ***Percent*** column.

If the ***Preserve aspect ratio*** box is checked, changing one axis will cause the other (enabled) axis to be scaled by the same amount

Short cut buttons are provided for common scaling factors

- ***100%*** will restore the objects their original size (100%).
- ***mm to inches*** will scale mm measurements to inches.
- ***inches to mm*** will scale inch measurements to millimeters.

Press ***Apply*** for the resize to take effect.

Rotating

The ***Transform - Rotate*** menu option (or **`Ctrl+R`**) is used to rotate selected objects.

This will first prompt for the center point of rotation.

Next, a reference (or start) angle is prompted. This can be useful when rotating a shape a set angle from a given edge. For example, to draw a perpendicular to an edge, draw a line along the edge, select the rotation center at one end of the line and the reference angle at the other line end point; then rotate 90 degrees using the angle snaps.

Pressing the middle mouse button will skip the reference angle selection and will use a 0 degrees reference, where 0 degrees is along the positive X axis.

Move the mouse about the rotation center point to control the rotation angle.

If the ***View - Snap to grid*** menu option is enabled, the rotation angle will snap to common angles (multiples of 30 and 45 degrees).

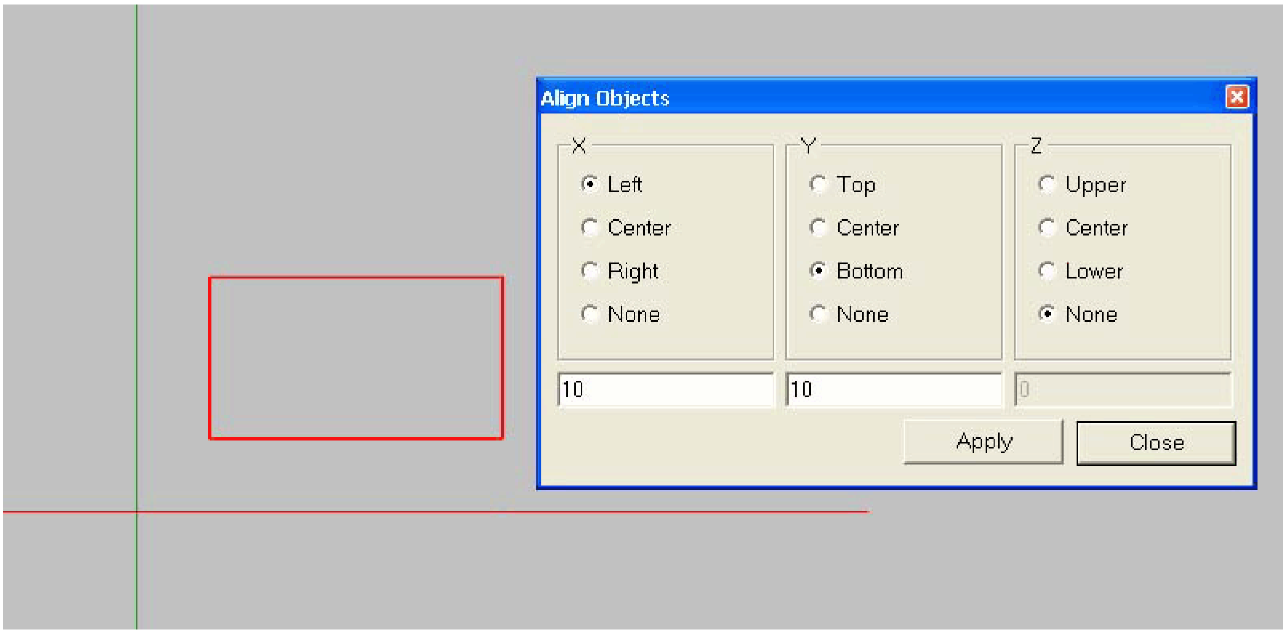
The rotation command can also be used to rotate about other axis. Pressing the **X**, **Y** or **Z** keys while rotating will select the axis of rotation. The angle of rotation is always set by moving the mouse around the center point in the plane of the drawing view, regardless of the axis setting.

Rotate can also be used to mirror an object, by selecting the Y axis of rotation, and rotating 180 degrees.

Selected objects can also be rotated 'freehand', by selecting them, holding down the **SHIFT** key, then using the view rotation mouse+keyboard combinations. For example, **ALT+SHIFT** and mouse drag. This method currently only rotates about the origin and does not snap to angles, so is only really useful for positioning 3D objects for artistic effects.

Align

Transform - Align can be used to position selected objects. This will display a form with 3 columns, one for each axis. Select the point of the selected axis to align, or none to leave the current axis position intact. Enter the drawing coordinate underneath which will be the new location of the alignment point, then press ***Apply***.

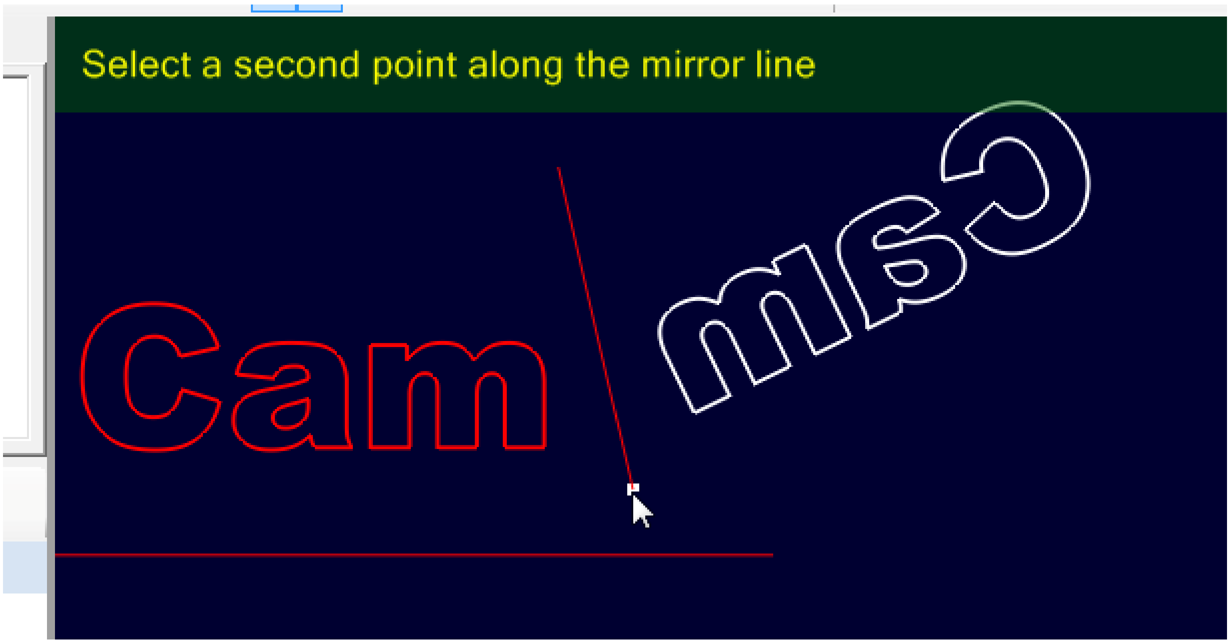


In this above example, the left most side of the rectangle is aligned to X=10 and the bottom part of the rectangle is aligned to Y=10. The Z location of the rectangle will remain unchanged.

Mirror

Creates a mirror copy of all selected drawing objects about a mirror line. The mirror line is specified by selecting two points along the line.

The **Shift** key can be used to constrain the reflection plane according to the current angles (0, 30, 45, 60 and 90 ° for each quadrant)



Array Copy

Array copy is used to create multiple copies of a drawing object, with each copy offset a specified distance.

First select the objects to copy, then select the ***Transform - Array Copy*** menu option. The routine first prompts for the number of copies to make, not including the original selected objects.

The routine then prompts for an offset distance for each copy, in the format X,Y,Z. The Z coordinate can be omitted and 0 will be assumed.

There is also an optional 4th parameter 'scale', which can be used to increase (scale > 1) or decrease (scale < 1) the size of each copy. Each copy is scaled using the following formula 1+(scale-1)*n, where n is the copy number.
For example 0,1,0,0.9 would offset each copy 1 unit in the Y direction and scale the copies 90%, 80%, 70%,etc of the original size.

Polar Array Copy

Polar array copy is used to create multiple copies of a drawing object around a point, with each copy offset by specified angle.

First select the objects to copy, then select the ***Transform - Polar Array Copy*** menu option. The routine first prompts for the center point of rotation, followed by the number of copies to make, not including the original selected objects.

The routine then prompts for an offset angle for each copy, about each axis, in the format X,Y,Z. The Z rotation coordinate can be omitted and 0 will be assumed. The angles are measured in degrees.

Rotation follows the right hand rule. To visualise this, with your right thumb pointing in the direction of the positive axis, the direction of rotation for a positive angle is the direction that the other fingers curl.

For example, to make 12 objects, evenly spaced around a point, set the number of copies to 11 (note the original copy is not counted), and use the following rotation value : 0,0,30 (that is 30 degrees around the positive Z axis).

Centring

The ***Transform - Center*** menu options can be used to center objects about the drawing origin.

There are two variations:

Center (Extents) will use the center point of the bounding rectangle to align the selected shapes.

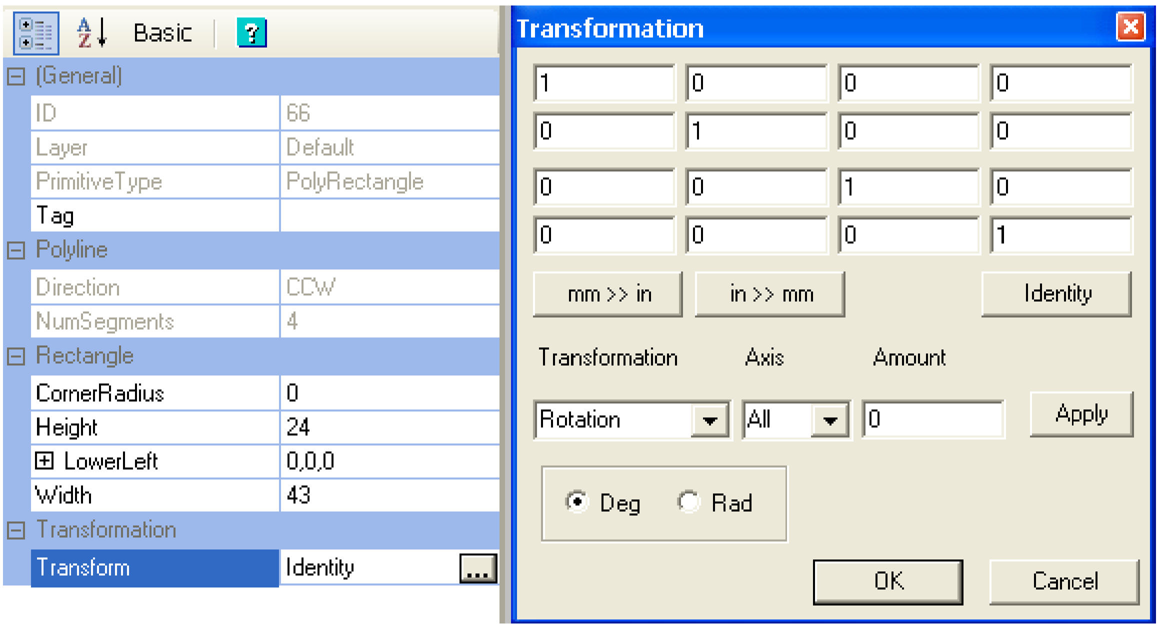
Center (Of Points) will align the 'average' point of all the control points contained in the selected shapes.

Transform matrix

More advanced transformations can be defined by changing the selected object's ***Transform*** property. This is a 4 x 4 matrix which is used to position, rotate and scale the object.

The transform property is located in the object property window for the selected objects(s).

Click the [...] button to the right of the ***Transform*** property to open up the transformation editor dialog.



Values can be entered into the matrix directly or a number of helper buttons can be used.

To rotate, scale or translate, select the required operation from the ***Transformation*** drop down list, select an ***Axis*** that the transformation applies to and an ***Amount***, then press the ***Apply*** button.

For rotations, the positive Z axis is coming out of the screen towards you.
If you place your right thumb in the positive Z direction, your fingers will curl in the direction of a positive rotation about the Z axis.
This right hand rule applies to rotations about all the axis.

Multiple transformations can be applied as along as ***Apply*** is clicked between each.

To reset the transformation matrix, click ***Identity***.

Apply Transformations

Changing the Transform property does not initially change any other object properties. For example, a circle center point and diameter, or polyline control points will remain unchanged. The transformed values will be automatically calculated when needed (during toolpath generation for example). To change these properties immediately, select an object then use the ***View - Apply Transformations*** menu option. This will transform all the shape's properties where appropriate and then reset the transform matrix back to Identity.

Note: As of version 0.9.8 many operations will now automatically apply transformations. This behavior can be controlled by changing the ***Auto Apply Transformations*** in the system configuration.

CAD Operations

Explode

Replaces a drawing object with its constituent parts.

For polylines, this will create individual line and arc objects.

For point lists, this will create individual point objects.

For text objects, each letter will be converted to a region.

For regions, the outer and inner shapes will be converted to polylines.

Join

This operation will attempt to join individual selected objects into single objects.

The join routine will first prompt for a join tolerance. This distance (measured in the current drawing's units) is used to determine how close the end points of shapes need to be before they can be joined.

Offset

Creates a polyline, offset from the selected shape by a given distance.

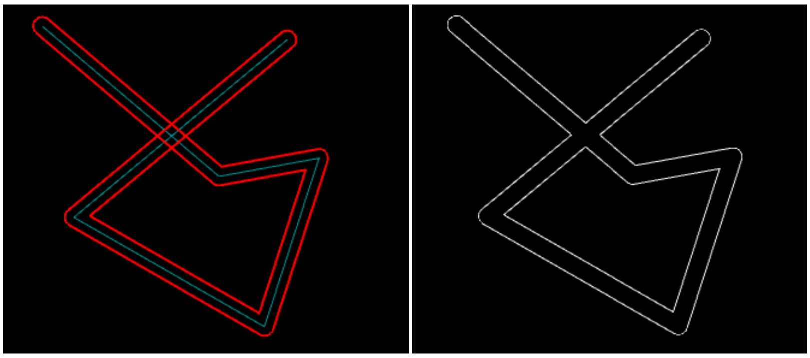
If a positive offset distance is supplied, the resulting offset polyline will be outside the selected shape. If a negative offset is given, the offset polyline should be inside the shape.

Open Offset

Creating an offset from an 'open' polyline will generate another open shape, offset from the original by the specified distance. ***Open Offset*** on the other hand will produce a closed shape that completely encloses the source shape.

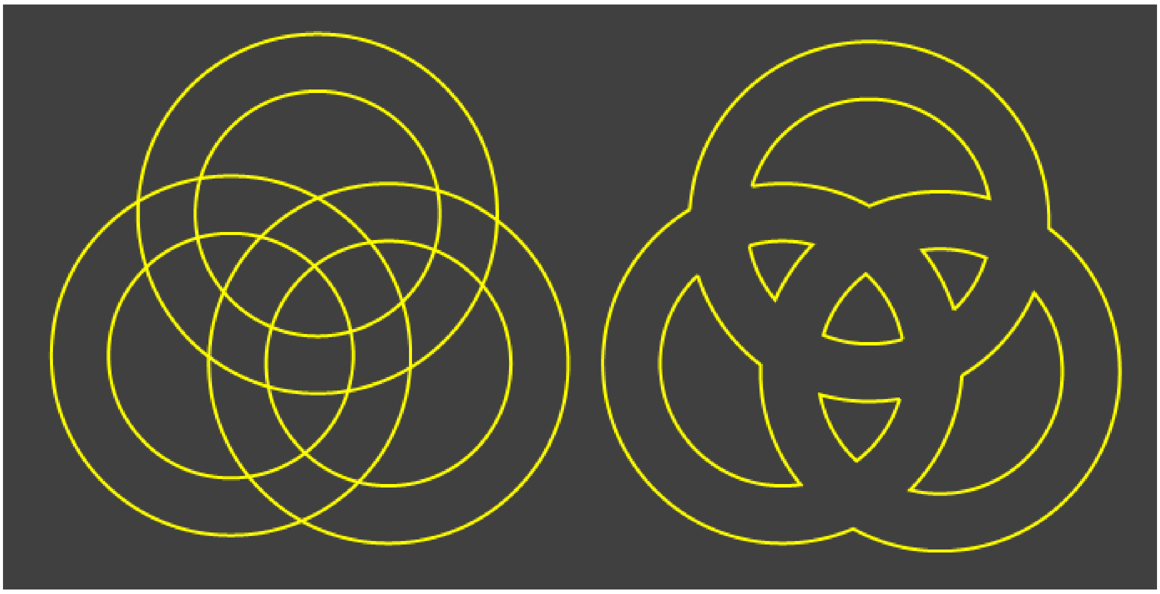
Open offsets are particularly useful when drawing slot shapes, by drawing the center line of the slot then using the ***Open Offset*** operation. Another typical use is for drawing tracks to be used in PCB milling.

The image on the left shows the result of the Open Offset command used on an open polyline. Note that the closed polyline can loop back upon itself. On the right hand image, this open offset has been modified by using the ***Break at intersections*** operation, then the unwanted inner segments deleted.



Union

Replaces shapes with the outer boundaries of all the selected shapes.



Subtract

Subtracts closed shapes from other closed shapes.

Intersection

Currently, this operation will only work on the first 2 selected objects.

Trim

Trims (deletes) parts of objects contained inside or outside selected trimming objects.

Fillet

Insert arc fillets of a specified radius into selected shapes.

Currently, fillets will only be inserted between adjacent straight (line) segments.

To create a fillet between distinct line segments, they must first be joined to form a single polyline.

Intersection Points

This operation inserts points at the intersections of selected shapes. This is useful when drawing, so that other drawing operations can 'snap' to these points.

Break At Intersections

Breaks selected shapes at the intersection points with other selected shapes.

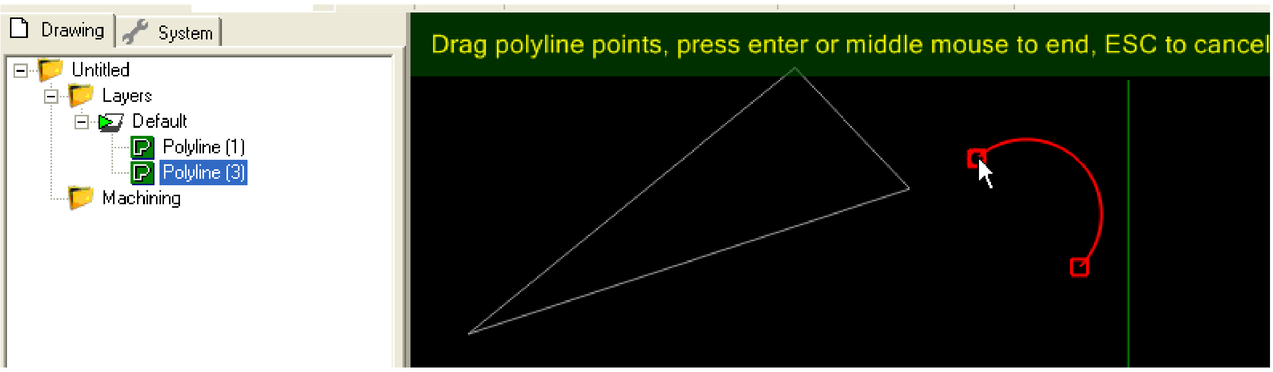
Edit Polyline

This section describes the operations available from the ***Edit - Polyline*** menu.

Edit

Modifies selected polylines by allowing the control points to be dragged interactively in the drawing view.

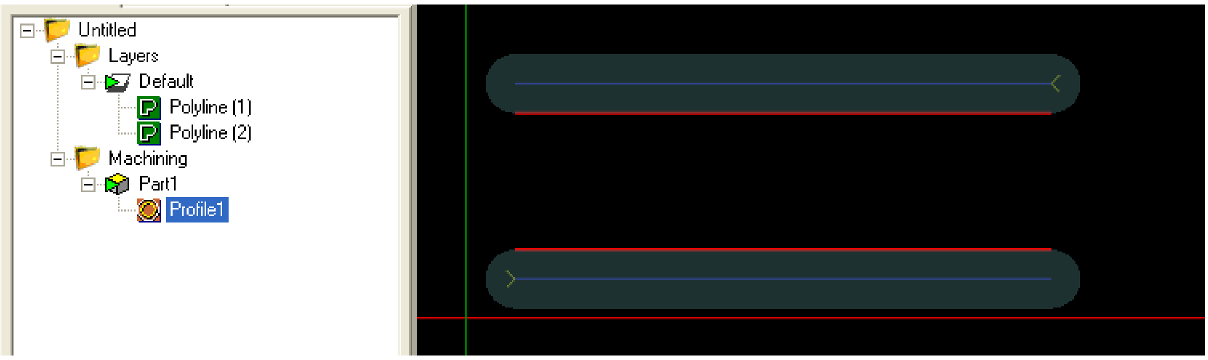
The polyline edit mode can also be entered by double clicking a polyline in the drawing view.



Reverse

Reverses the order of the points within a polyline.

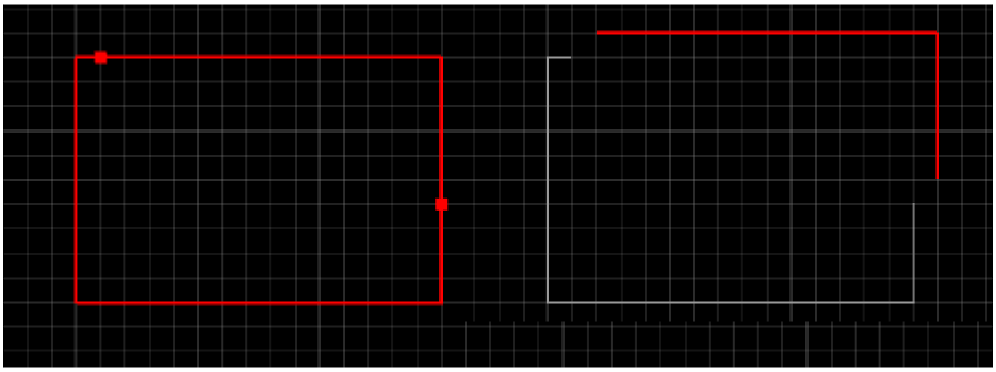
Reverse is useful in situations where there is a toolpath such as a profile, offset from an open polyline. For open polylines, the order of the points within the polyline will dictate which side of the polyline the toolpath will be offset from. As an alternative to changing the profile machining operations ***Inside / Outside*** property, the source polyline can be reversed. This will change the side of the polyline where the toolpath is offset.



Clean

Removes duplicate points from a polyline.

Break at points



Break a polyline at a set of points.

Select the polylines to cut, and also select the point lists that will define the cutting points and then select the ***Edit - Polyline - Break at Points*** menu option.

Set start point

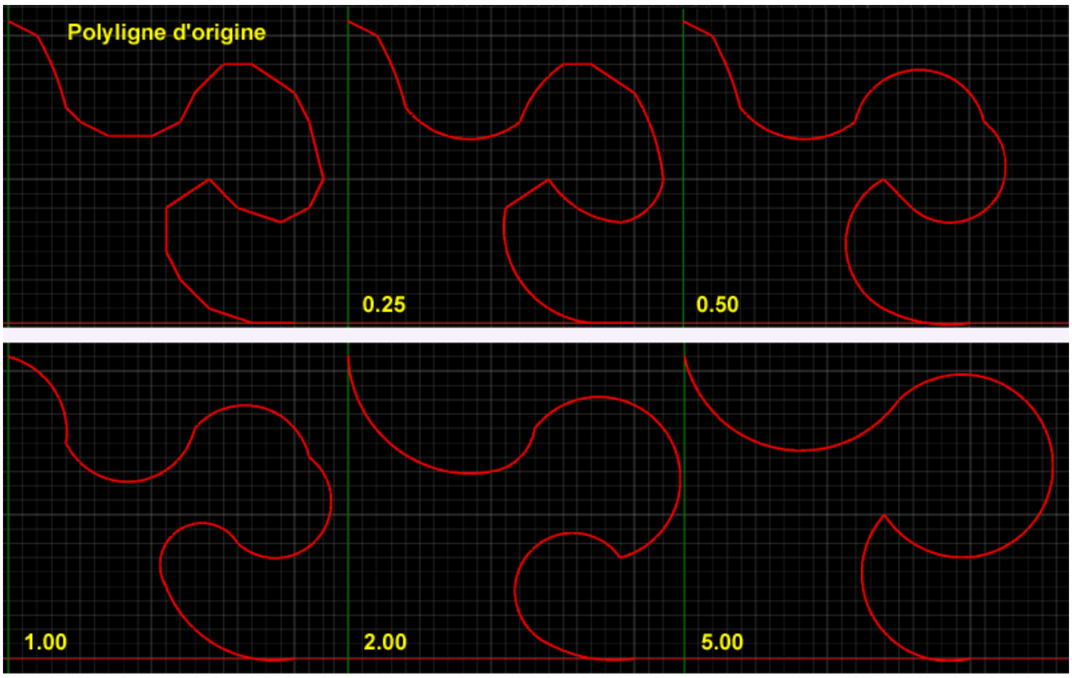
Changes the first polyline control point, (for closed polylines only).

Arc Fit

Arc fit will attempt to simplify a polyline by replacing a number of small segments with a single circular arc segment, that fits the polyline control points according to a specified tolerance. In some cases this can dramatically reduce the number of segments in the polyline, resulting in faster toolpath calculations and more compact gcode. The use of large arc segments rather than many small segments can also make the machining operations much smoother when cutting.

Arc fit will prompt for an ***Arc Fit Tolerance***. This is the maximum allowed deviation (in drawing units) from the fitting arc to the original segments. A larger tolerance can result in polylines with fewer segments but the deviation from the original shape is potentially greater.

The following images show the effect of different tolerances on a polyline arc fit.



Remove overlaps

Overlaps are polyline segments that back track along the polyline and then backtrack again, much like a compressed Z shape. These can cause problems for some of the routines in CamBam, such as polyline joining and toolpath generation.

These problems are commonly found in drawings that have been converted from bitmaps using vectorization software. These overlaps may be very small and hard to spot.

The ***Remove overlaps*** operation will create a copy of the source polyline in the active layer, with any detected overlaps removed. If the original polyline is used by a machining operation, the machining operation source objects will need to be reselected to choose the cleaned polyline.

Note: In the latest version of CamBam, the toolpath generation routines will automatically attempt to detect and fix any back tracks in polylines before creating offset polylines from them. In many cases the manual ***Remove overlaps*** operation will not be needed. The automatic checks can be disabled in the system configuration settings by setting ***Offset Backtrack Check*** property to *False*.

Edit Surface

This section describes the operations available from the ***Edit - Surface*** menu.

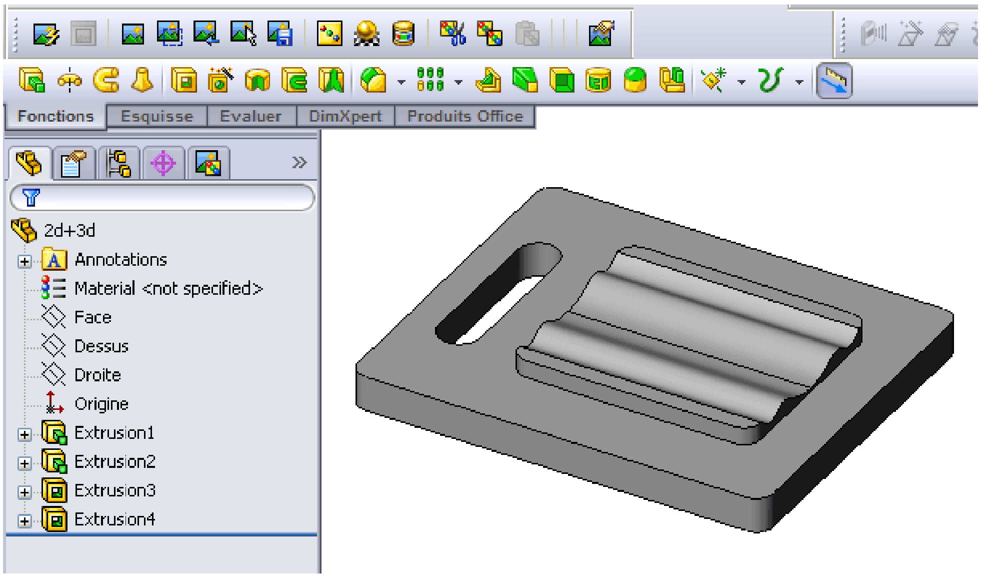
Plane Slice X, Y, Z

These functions obtain polylines from a 3D object by slicing the object along a given axis.

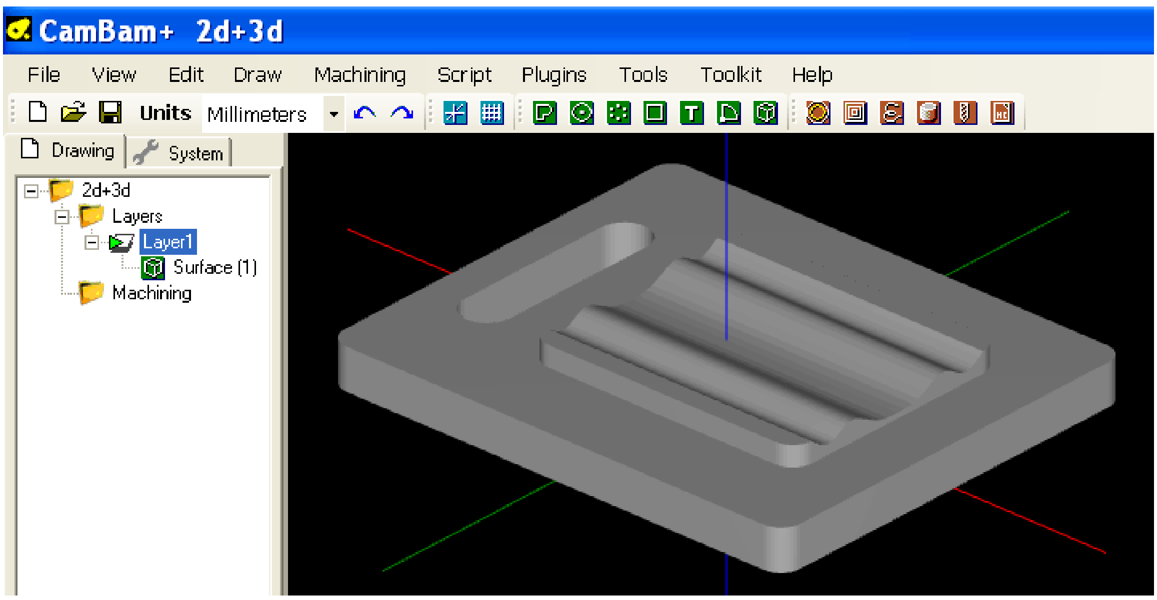
Plane slices provide a useful way of generating 2D machining operations from 3D models, without having to redraw the models in 2D. For many engineering or prismatic 3D shapes, 2D machining operations can provide much simpler, faster and more accurate operations than the 3D machining operations.

The following examples show plane slice being used to create a combination of 2D and 3D machining operations from a 3D model.

Here is the original model object, created within SolidWorks ®.

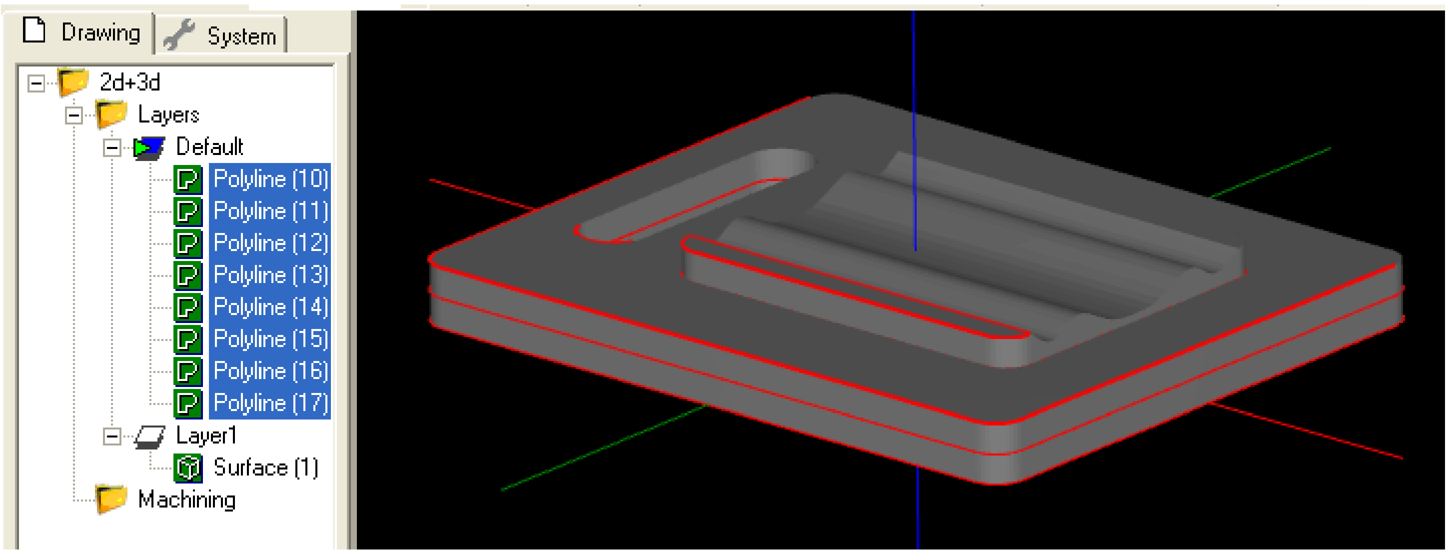


The model is then loaded into CamBam.



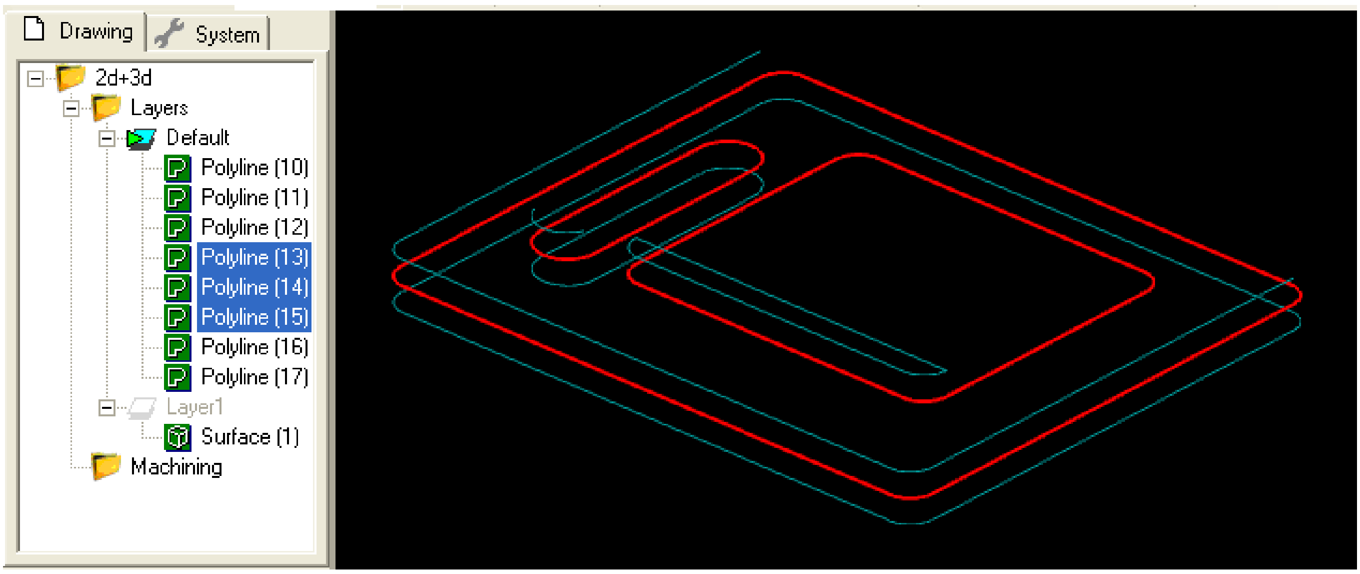
The wavy part of the model will be machined using 3D operations, but it would be more efficient to use 2D operations on the flat parts of the model.

Plane Slice Z is used to slice the model along planes, normal to the Z axis, at 5mm intervals.

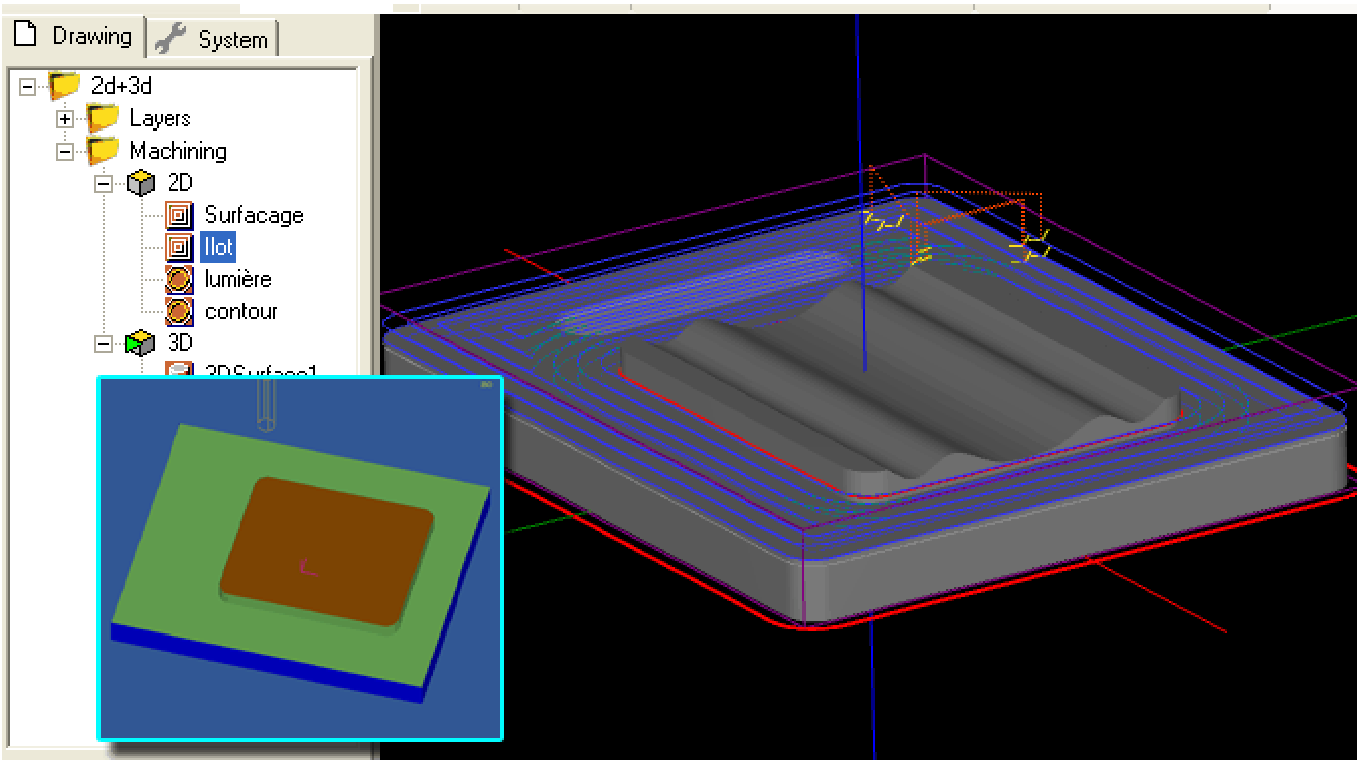


It is good practice to create a new layer to receive the plane slice polylines, to make the drawing more manageable and so that the polylines may be viewed and manipulated independently of the original 3D surface.

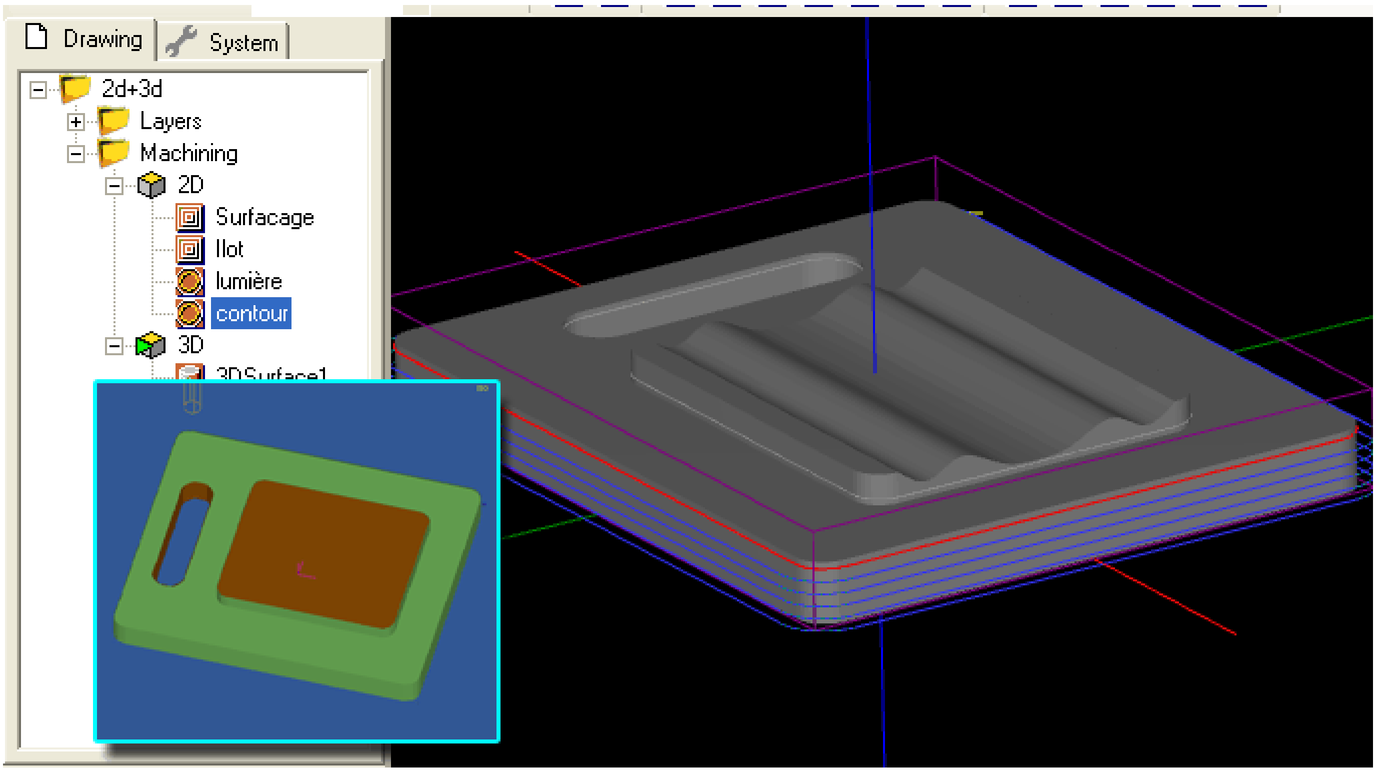
In this example, we are only interesting in a selection of the plane sliced polylines (shown highlighted below). The other polylines can be deleted. The layer containing the 3D surface has also been hidden for clarity.



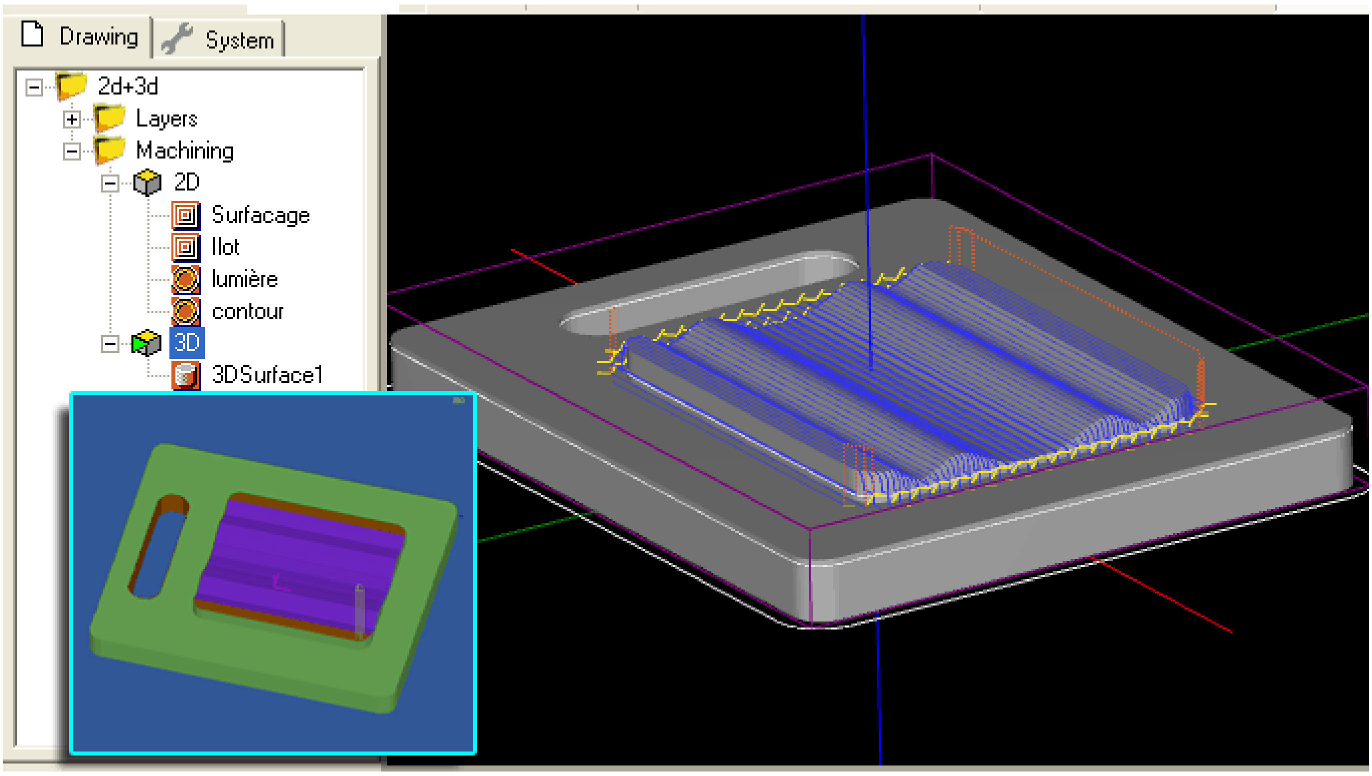
2D pocket operations are used to clear the flat areas of the mode. The inset image shows the results when simulated using *CutViewer Mill*.



2D profiles are then used for cutting slots and the outer shape.



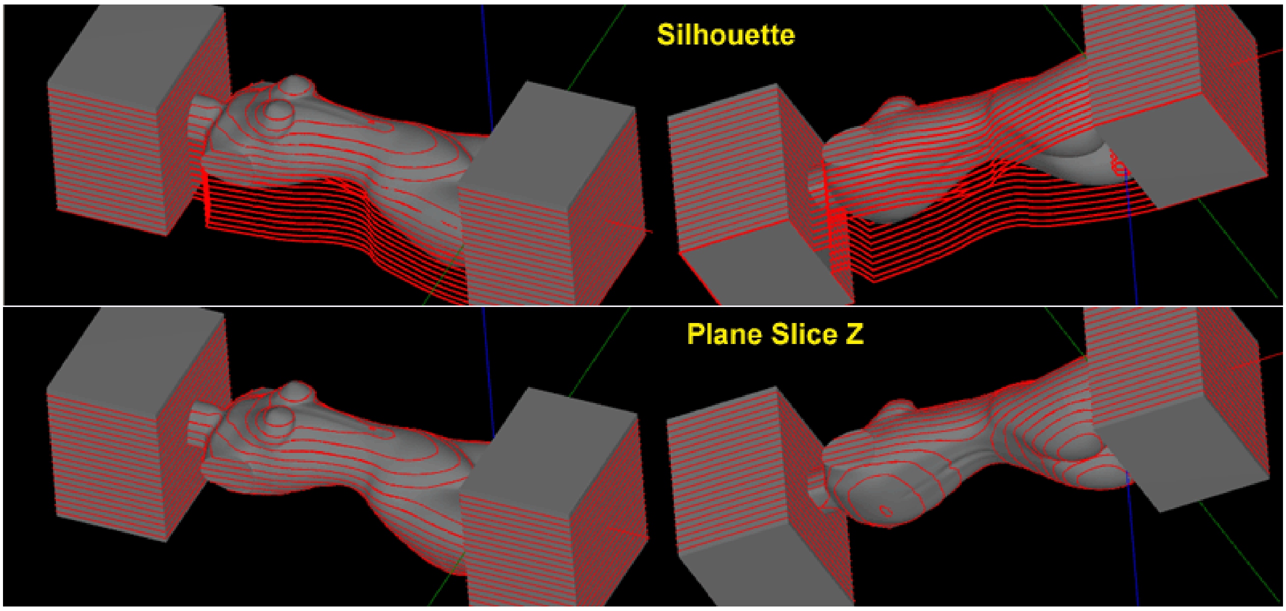
A ***3D surface*** operation is then used for the wavy area of the shape. The inner plane slice polyline is used to restrict the 3D surface by specifying a boundary shape. See the ***3D tutorial*** for more information on this operation.



Silhouette

A Silhouette is similar to a plane slice operation, except overhanging areas of the model from higher layers are projected downwards. These represent the limits of parts of the model accessible by a cutting tool. The silhouette routines can only be used for the Z axis.

A comparison of Silhouette (top) and Plane Slice Z (bottom).



Invert Faces

Invert Faces will reverse the direction of face normals for each selected surface.

3D Meshes are built from many triangular faces. Each triangle has a ‘normal’, which is a direction vector perpendicular to the triangular face. The order of the points around the triangle will determine the normal direction. CamBam uses a 'Right Hand Rule', so that if the fingers of your right hand curl in the direction that the triangle points are ordered, your thumb will point in the direction of the normal.

Normal vectors are used to determine the outside facing direction of each face. When displaying meshes, faces pointing away from the viewer are often ignored by the display routines. If meshes are wound using a 'Left Hand Rule', this can result in the models appearing dark or hard to see in the CamBam drawing display.

Edge Detect

Will dump edge lines detected from selected 3D meshes.

These are triangle faces edges that have no neighbours, or the neighbouring face forms an angle.

Project Lines to Surface

Selected drawing shapes will be projected onto any selected surface objects.

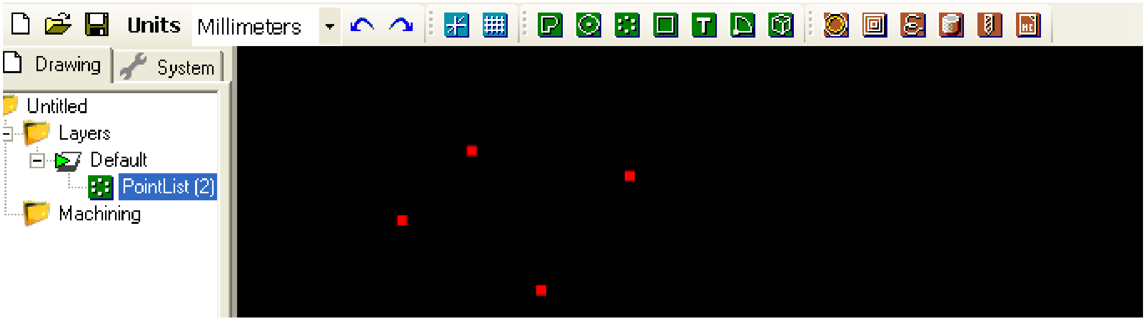
The routine prompts for a Projection Resolution. This is the distance along each drawing line at which the Z height is tested.

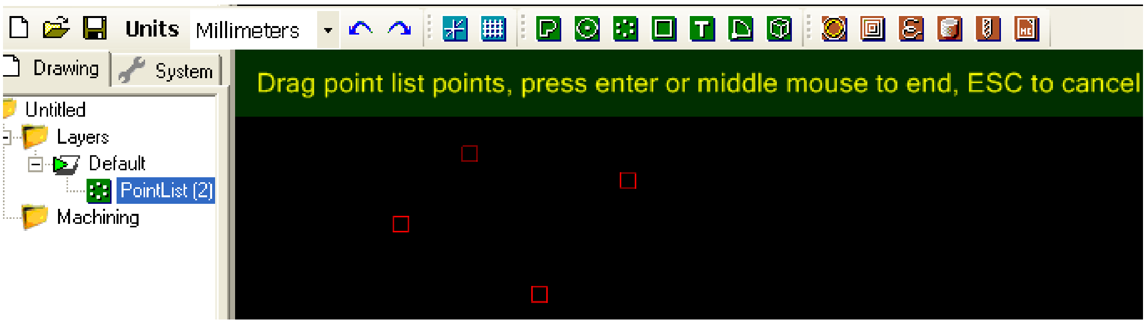
If a point along the line is outside any surfaces, the minimum depth of all selected surfaces is used.

Editing Point Lists

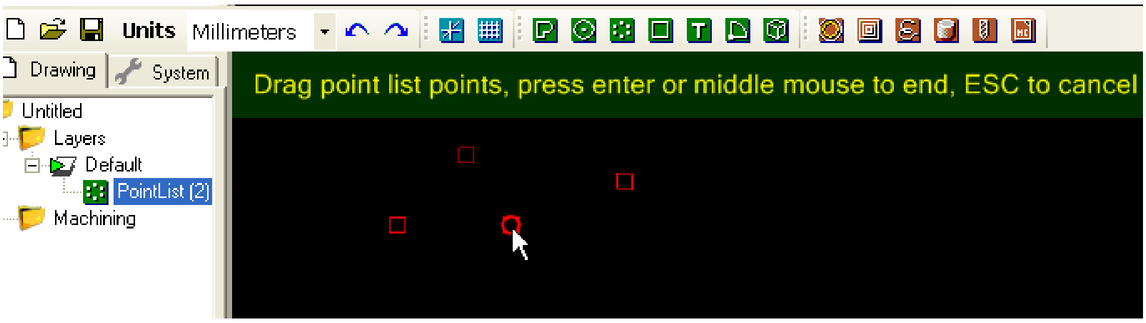
Move or add points

You can edit a point list object by double clicking any of its points in the drawing view to activate the edit mode.

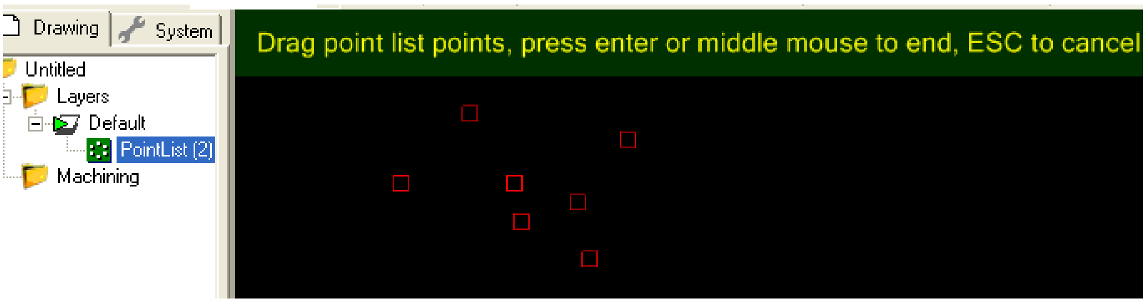




To move a point, click and drag the square point icons.



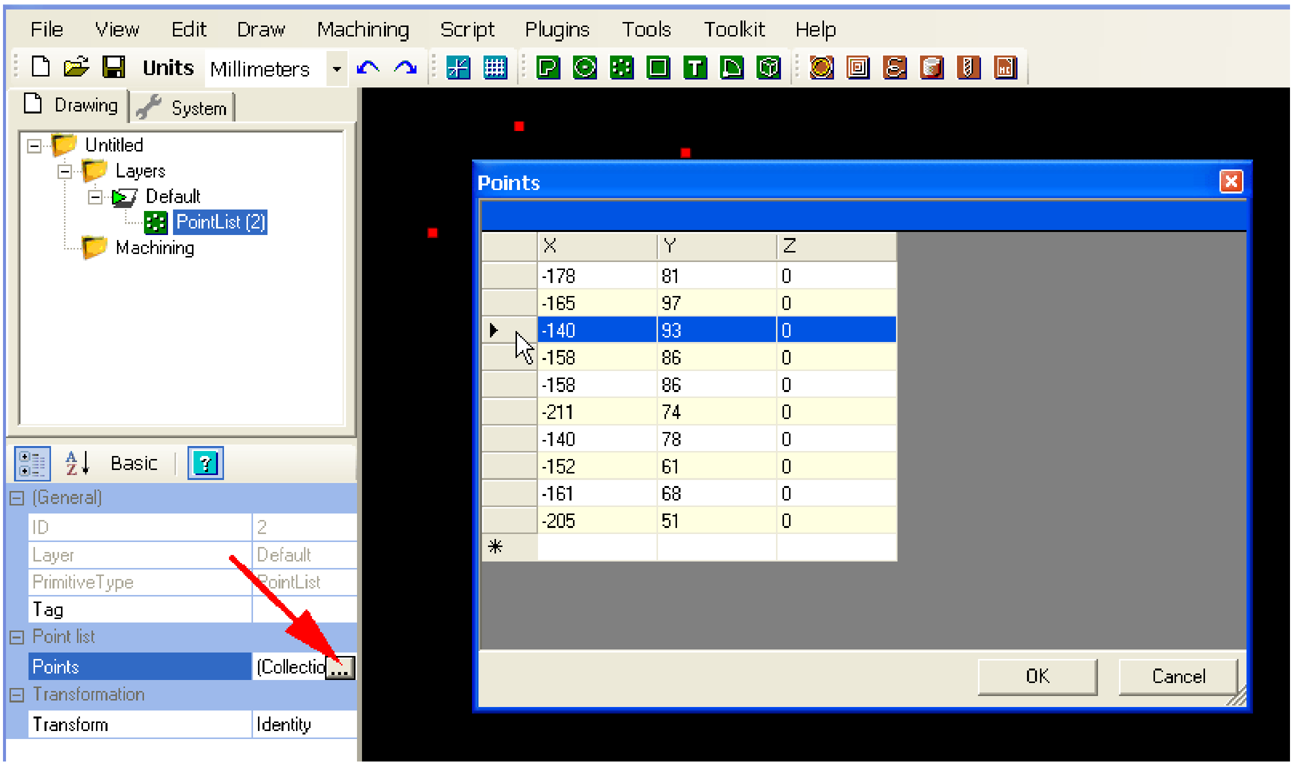
Click on empty areas to add new points to the point list.



Click the middle mouse button, or press **Enter**, to accept the changes. Press **ESC** to cancel the point list edit.

Deleting points or entering explicit coordinates.

The points in a point list can also be edited in a tabular format by clicking the [...] button to the right of the ***Points*** collection property of the selected point list.



Points can be deleted by highlighting a row and pressing the **`Delete`** key.

Exact X, Y and Z coordinates can be entered directly into the table.

Entering coordinates into the bottom line marked with an '*' will insert a new point into the list.

It is also possible to cut, copy and paste the point list data from this table as tab delimited text. This also allows cutting and pasting points to and from a spreadsheet such as Microsoft Excel.

Explode Point Lists

The ***Edit - Explode*** operation can be used to break a point list containing multiple points into the individual points.

Drawing Surfaces

The 3D surface and solid modelling functionality of CamBam is limited, but there are number of helpful 3D drawing routines. These are available from the ***Draw - Surface*** menu.

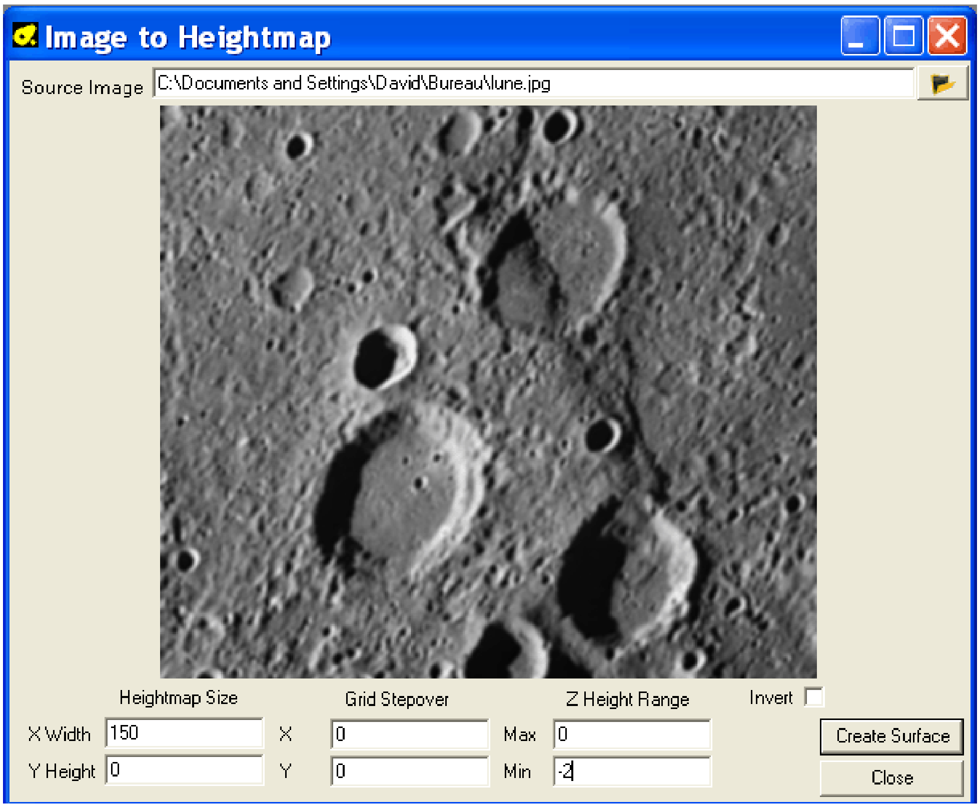
From Mesh File

This inserts a 3D surface from an STL file.

From Bitmap

Converts a bitmap image into a 3D surface by using the brightness levels to define the Z heights.

This is a similar process used by the ***Heightmap Plugin***, but where the plugin will only generate an engraving toolpath, the ***draw surface from bitmap*** routine will create a 3D mesh that can be used to create more sophisticated 3D machining operations such as waterline and scanline roughing.



Click the  button to open an image file.

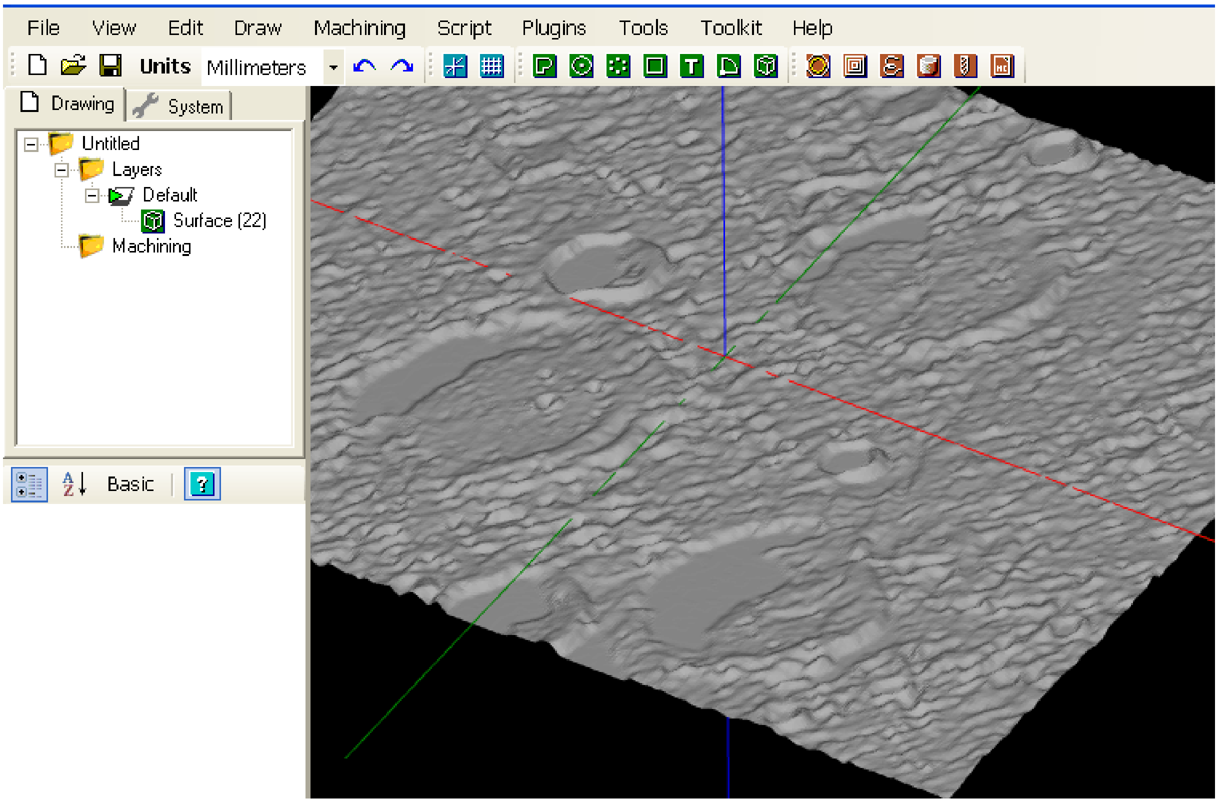
Heightmap Size: Set the X and Y dimensions (in current drawing units) of the surface to be generated. If the X or Y dimension is left at 0, this dimension will be automatically calculated so as to preserve the aspect ratio of the original image.

Grid Stepover: Controls the size of each triangular facet that will be used to build the surface. If this is set to 0, the size will be based on one image pixel size.

Z Height range: The minimum and maximum Z heights that correspond to the lightest and darkest parts of the image.

Invert: When unticked, dark areas represent low Z values and light areas higher Z areas. If invert is ticked this is reversed.

Click ***Create surface*** to generate the 3D mesh into the current drawing.



From text file

Allows you to use a plain text file (ASCII) providing a list of coordinates representing the triangle faces of a 3D object.

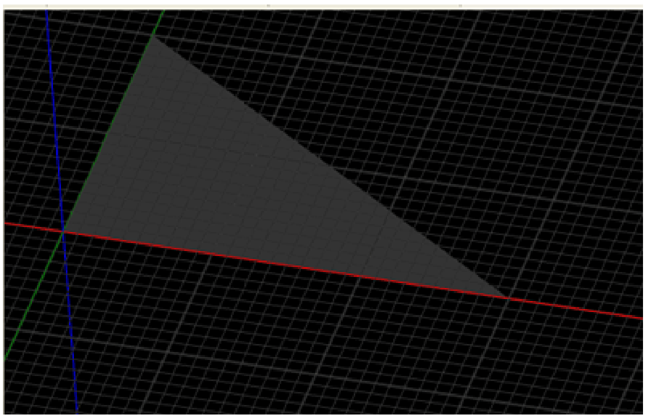
Each line consists of nine coordinates, separated by a space, corresponding to the coordinates X, Y and Z of three vertices defining a triangle.

Example:

```
0 0 0 0 20 0 30 0 0
```

This defines 3 points : Point1 (x,y,z) = 0,0,0 Point2 = 0,20,0 and Point3 = 30,0,0

This file gives the following result:



Extrude

Extrude is used to create a 3D surface from a 2D line by projecting it in the Z direction.

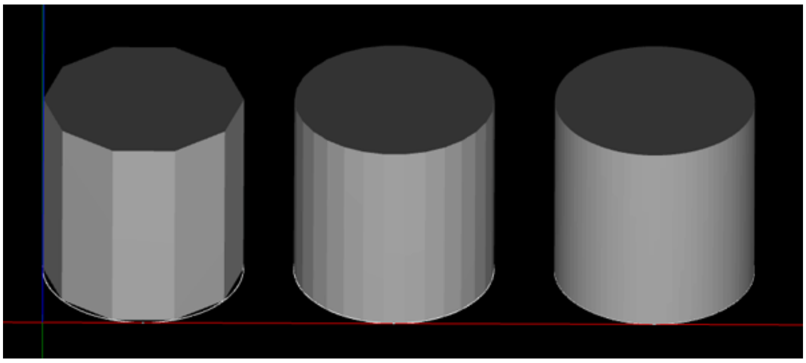
This operation was originally added to create shapes for use as holding tabs or 'sprues' on 3D machining operations.

To create extrusions along other axis, the shape must first be extruded in Z, then rotate the extruded surface object to the required orientation.

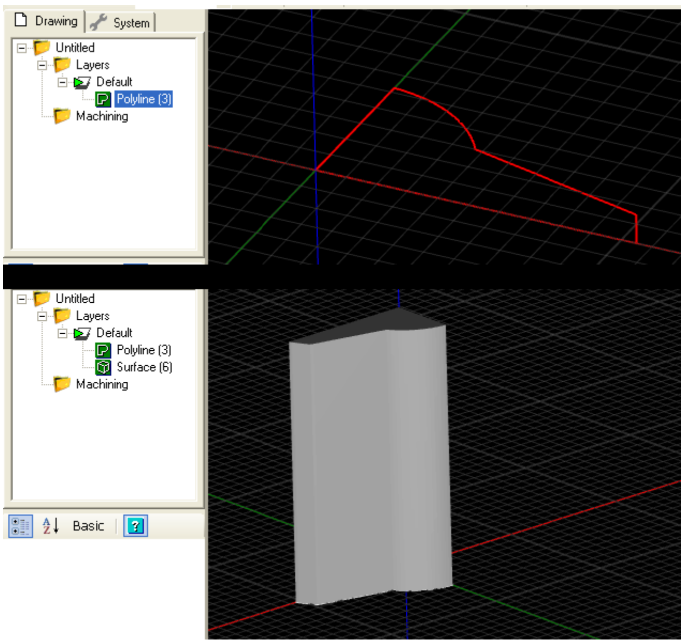
The ***Extrude*** operation will first prompt for an ***Extrusion Height***. This will be the Z height of the extruded surface. A positive height will extend toward the positive Z axis (ie toward the viewer when drawing is in the normal orientation with the XY plane parallel to the screen). A negative height will extend the surface along the negative Z axis (ie into the screen).

The routine next prompts for the ***Extrusion Steps***. This controls the number of steps around the source shape to insert faces on the extruded surface. More steps will result in a smoother surface.

This following image of a circle extrusion shows steps of 10, 30 and 100.



Another example of an extruded polyline.



Extrude solid New 1.0

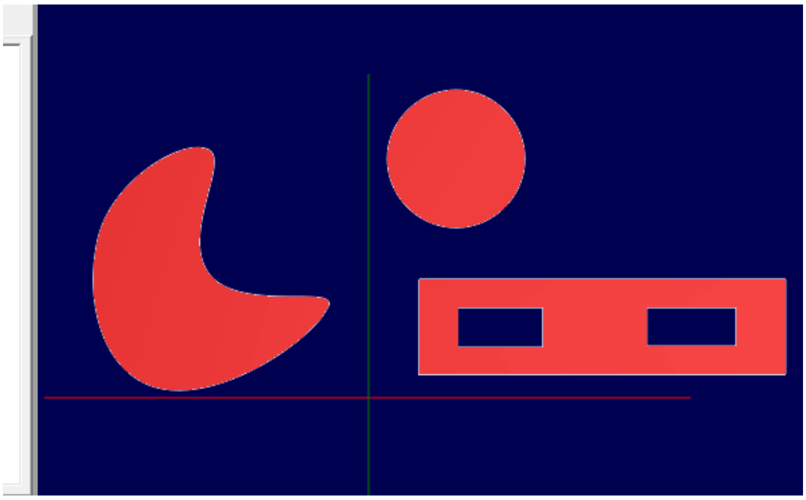
Like extrude, but the ends of the shape are closed in order to produce a "solid" object that can be exported to STL for printing.

A window will ask you to enter a thickness for the shape, then an ***Arc Expand Tolerance*** , as for simple extrusion.

Fill New 1.0

Fill the selection with a full 2D surface. (without thickness)

A window will ask you to enter an ***Arc Expand Tolerance***, as for extrusion.



Region Fill

These methods are used to fill regions, polylines and other closed shapes with various line patterns.

These fill patterns are used by machining operations such as pockets and 3D waterline roughing to generate toolpaths, but can be used independently to create interesting drawing effects.

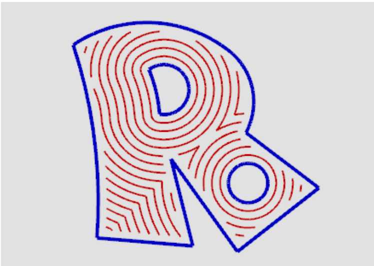
Region fillers take the following parameters:

Margin : This is the distance away from source shapes to avoid filling lines. In pocketing, this would be the same as the tool radius.

Step Over : This is the distance between fill lines. In pocketing, this would be the same as the stepover distance.

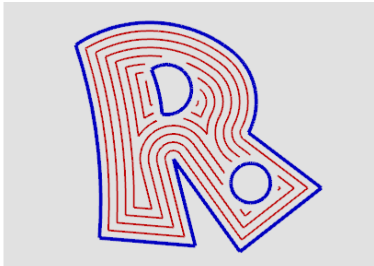
Filling Patterns

Inside Offsets



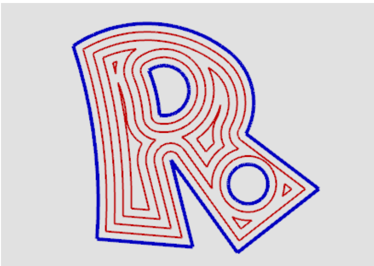
Progressive offsets from holes outwards.

Outside Offsets



Progressive offsets from outside shape inwards.

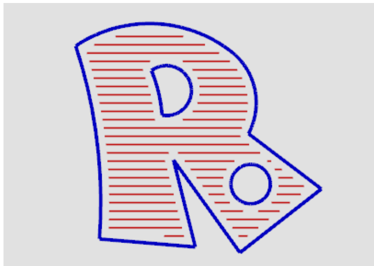
Inside + Outside Offsets



Progressive offsets from outside shape

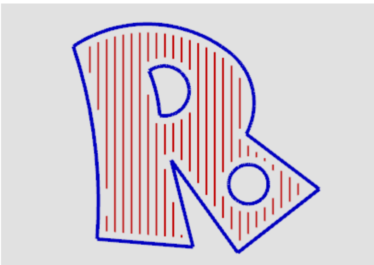
inwards unioned with offsets from holes outwards.

Horizontal Hatch



Horizontal line fill style.

Vertical Hatch



Vertical line fill style.

Tutorial: Timing Pulley Profile

This tutorial demonstrates using ***Profile*** machining operations to generate an HTD5 timing pulley.

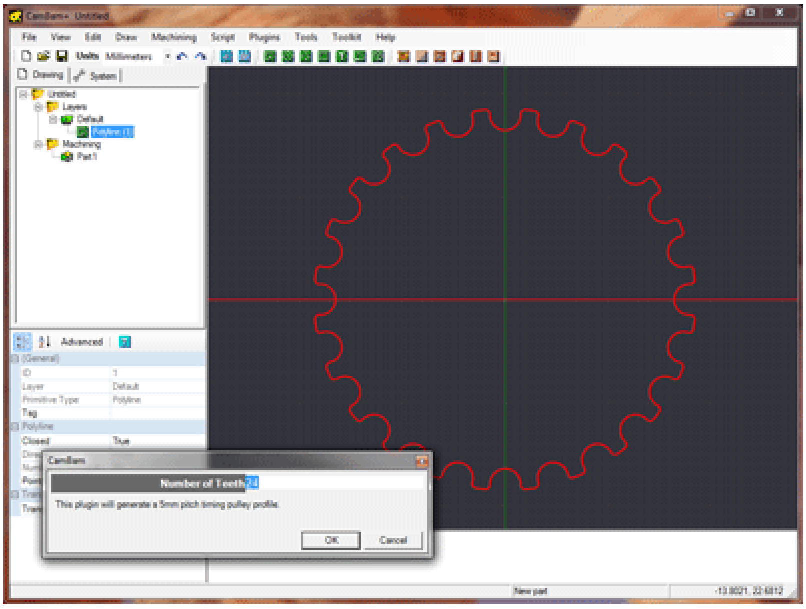
This tutorial uses the ***Plus Toolkit plugin*** to generate the timing pulley profile.

Step 1 - Insert an HTD timing pulley outline.

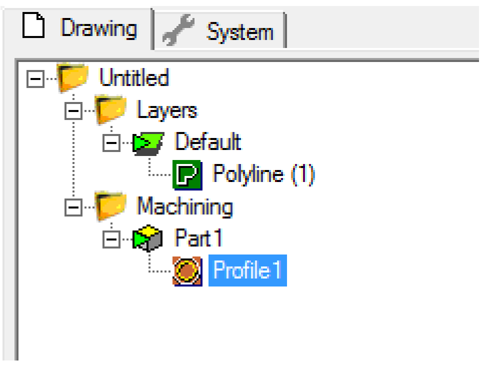
Use the new Plus Toolkit to generate a timing pulley by selecting the ***Toolkit - Timing Pulley*** menu item.

The plugin will prompt for the number of teeth for a 5mm pitch pulley, then insert a new curve with the center of the pulley about the origin.

ALT + double click will zoom the drawing to fit the view window.



Step 2 - Insert a Profile machine operation



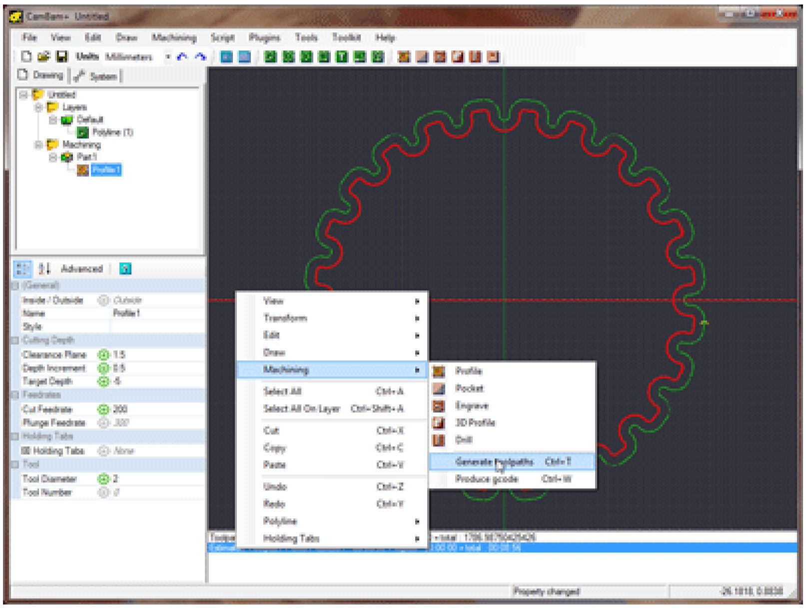
Select the pulley outline then click the ***Profile*** machining operation button from the toolbar. A new profile object will be created and displayed under the ***Machining*** folder in the drawing tree. The object property window will display the profile's properties ready for editing.

Change the Profile machine operation's properties to the following:

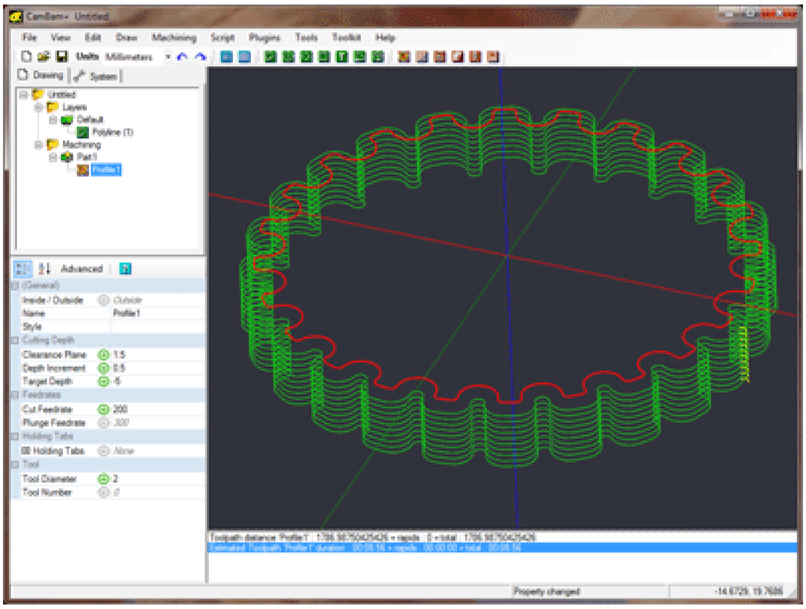
Tool Diameter	2
Depth Increment	0.5
Target Depth	-5
Cut Feedrate	200
Plunge Feedrate	100
Clearance Plane	1.5

Note: Some properties such as ***Clearance Plane*** may not be shown in the property grid.
Clicking the ***Advanced*** button at the top of the property grid will display all available properties.

Generate the resulting toolpath for the profile; right click the drawing to bring up the drawing context menu, then select ***Machining - Generate Toolpaths***.



To rotate the 3D drawing view, hold the **ALT** key then click and drag on the drawing. To reset the view, hold the **ALT** key then double click the drawing. Another rotation mode (*Left_Middle*) can be set in the ***Rotation Mode*** property of system configuration settings. If this mode is selected the view can be rotated by clicking the middle mouse button and dragging with the left. To reset the view in this mode hold the center mouse button and double click.



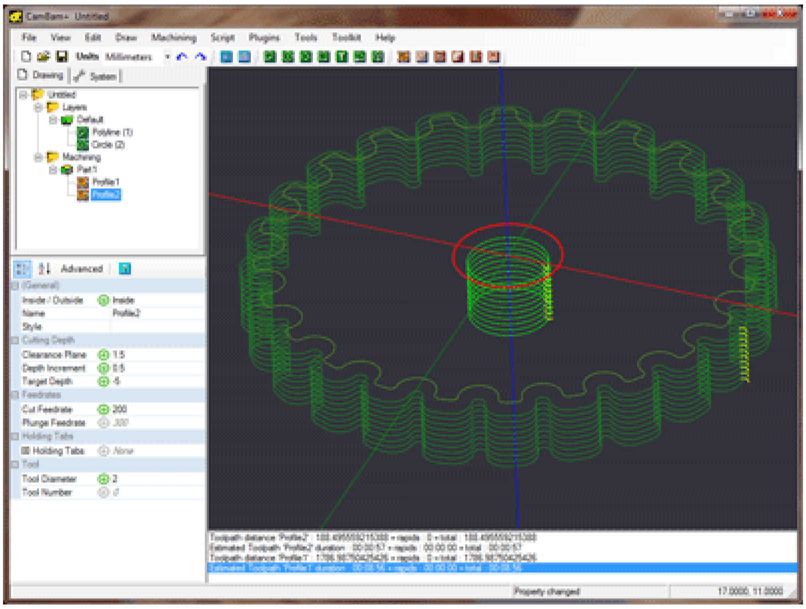
Step 3 - Creating the inner hole

First draw a circle using the circle drawing tool with the center on the origin with ***Diameter*** = 8.

Select the circle and insert another profile machining operation .
Set the target depth and other properties to match the first profile operation.

> ***Hint:*** A quick way to do this is to select ***Profile1*** and copy it to the clipboard (using the context menu or **CTRL+C**). Then select ***Profile2*** and use the ***Paste Format*** command from the context menu shown when right clicking ***Profile2***, or use **SHIFT+CTRL+V**.

Change the ***Inside / Outside*** property to *Inside*.
Again, right click the machine operation in the file tree and ***Generate Toolpaths***.



Step 4 - Creating GCode

Before producing the gcode output, now would be a good time to save your drawing.

Visually inspect the toolpaths and double check the parameters of each machining operations.

To create a gcode file (post), right click to get the drawing menu then select ***Machining - Produce GCode***.

CamBam will then prompt for the location of the gcode file to produce. If the drawing file has been saved, the default gcode file will be in the same folder as the drawing file, with a .nc extension.

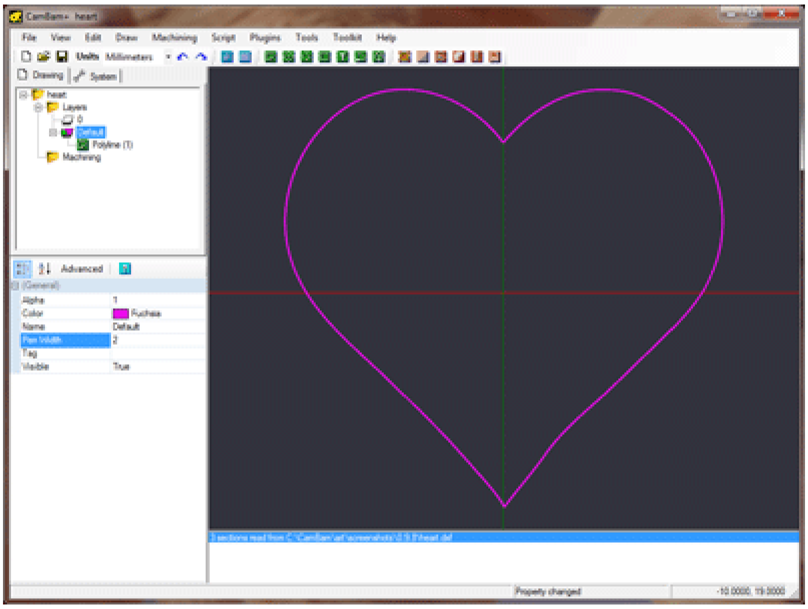
If the destination file already exists you will be asked to confirm whether to overwrite it.

To control how the gcode file is produced, select the ***Machining*** folder from the drawing tree. The machining properties for this drawing will then be displayed in the object properties window.

Tutorial: Pocketing and Island Pocketing

This tutorial will describe using the ***Pocket*** machining operation and will also cover - Loading DXF files, CAD drawing, object transformations and Automatic Island Pockets.

Step 1 - Load a DXF file



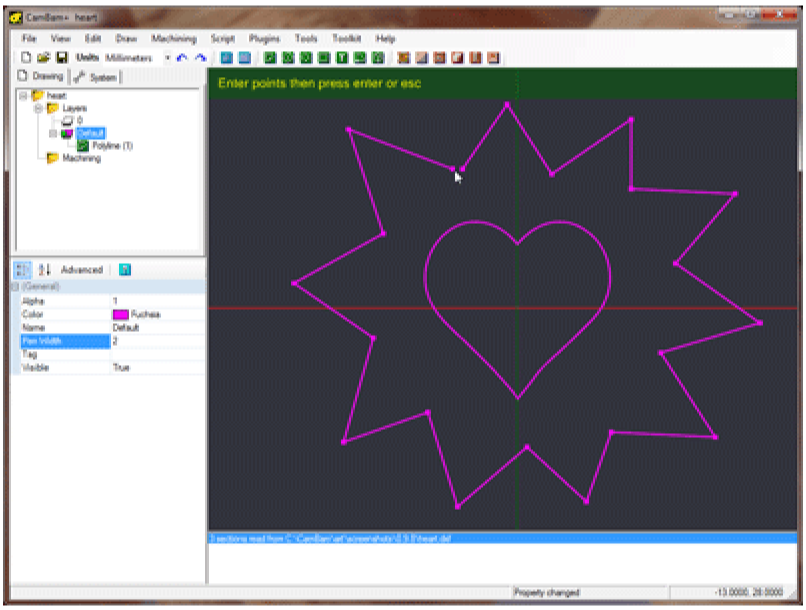
The sample file above, includes a heart shape DXF file. If you are married and fanatical about CNC, this shape can come in very handy indeed!

This shape is a nice and clean, closed polyline. If your DXF files contain many small segments or uses non polyline objects you should tidy the drawing before creating any machining operations.

To convert objects to polylines, select them, then select ***Convert To - Polyline*** from the drawing's context menu, or when the drawing window has focus, use the **CTRL+P** shortcut key.

Step 2- Free hand CAD drawing

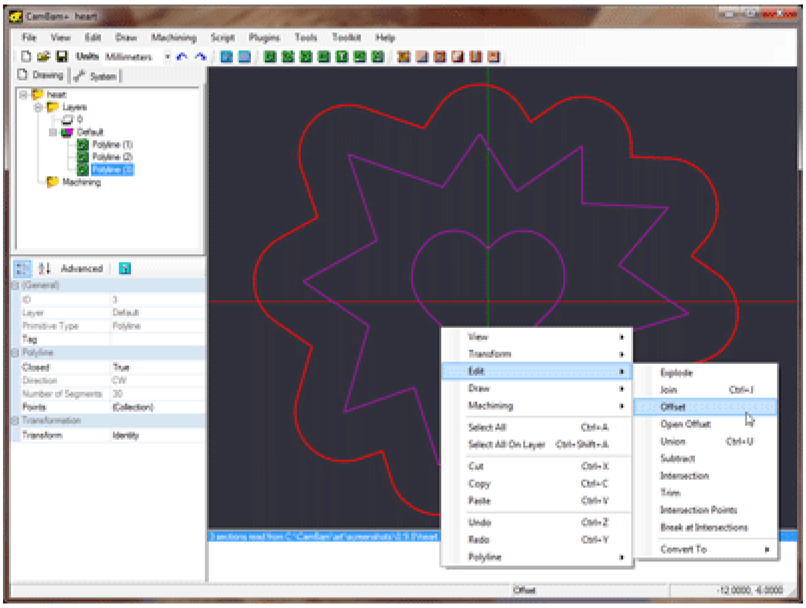
Use the polyline drawing tool to draw a random shape around the heart. This will form the outer boundaries of an island pocket. For the last point, press the **C** key to close the shape, or click on the first polyline point (the cursor should snap to it), then press **ENTER** or click the middle mouse button.



If the polyline does not sit evenly around the heart, you can free hand move objects by selecting them, then hold the **SHIFT** key and drag objects with the left mouse button. To position objects more accurately, use the ***Transform - Translate*** drawing context menu. This will translate an object given an origin and destination point.

To make the shape a bit rounded create an offset shape. Select the polyline, then click ***Edit - Offset*** from the drawing context menu. This will prompt for a distance to draw an offset polyline. Positive offsets will be outside the polyline shape and negative offsets will be inside.

To rotate a shape, select it then use the ***Edit - Transform - Rotate*** menu option or **CTRL+R** keyboard short cut to enter rotation mode. Select a center of rotation point, a starting angle then move the mouse around the start point to the desired position.



Step 3 - Pocket the heart

Select the heart shape then insert a pocket machine operation using the pocket tool . For pocketing basics, please see the ***Stepper Mount tutorial*** in Getting Started.

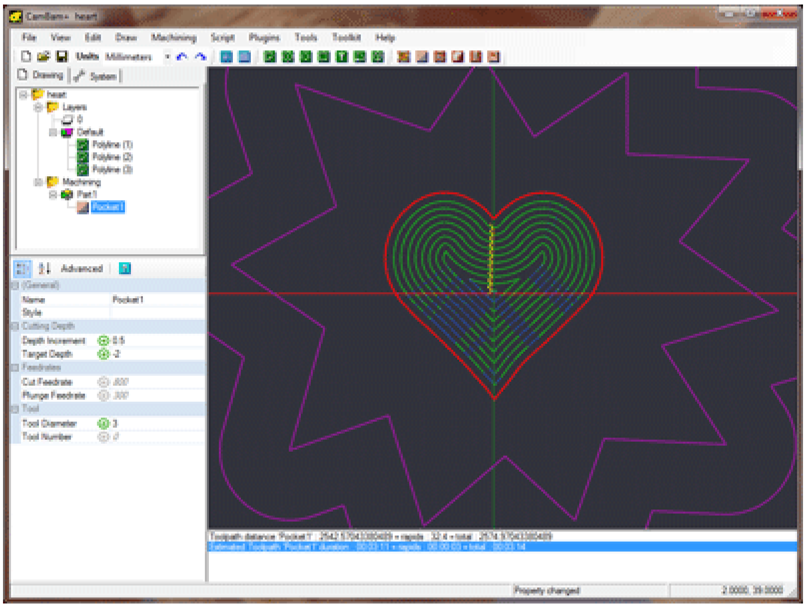
The important thing to remember is that the ***Target Depth*** should be lower than ***Stock Surface***. If stock surface is at zero then the **target depth should be negative**.

CamBam can cut deep pockets by generating toolpaths at progressively deeper cutting levels. The distance between each cut level is specified in the ***Depth Increment*** property.

To ensure a final light finishing pass at the lowest level cut, enter a small value in the ***Final Depth Increment*** property (0.1mm , 0.004”). This will be the depth of stock removed at the bottom pass of the pocket.

Another useful parameter is ***Roughing Clearance***. Enter a small value here to specify how much stock to leave remaining between the walls of the pocket and the target geometry. This stock can then be removed using a later finishing profile.

If a **negative** ***Roughing Clearance*** value is specified, the geometry will be over cut by this amount. This is very useful when making inlays or die cutting. The ***Roughing Clearance*** can be adjusted so the positive and negative shapes fit very closely. Roughing clearance can be adjusted while the stock is still mounted in the CNC machine. The pockets can then be tested with a previously cut inlay for a good fit.



Step 4 - Creating an Island Pocket

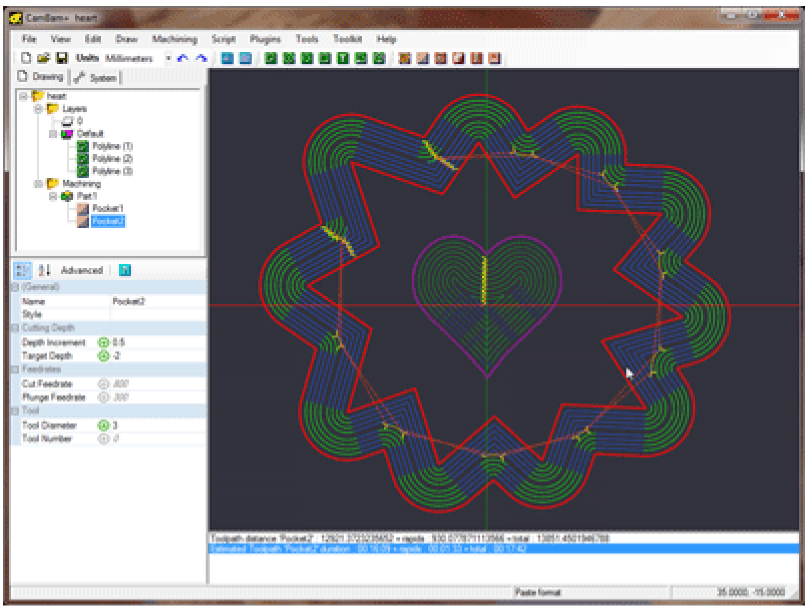
Island pockets can be created automatically by selecting inner and outer polylines then inserting a pocket as usual. Shapes within other shapes will be detected and toolpaths will be excluded from these inner 'hole' shapes.

If there are 3 concentric shapes selected for a pocket, the pocketing routines will interpret this as a pocket within an island within a pocket. In this tutorial we could have used just 1 pocket from all 3 polylines, but for clarity 2 separate pockets were used.

With the 2 outer polylines selected, insert another pocket .

To save entering in all the pocketing parameters for the second pocket, right click the first pocket operation in the drawing tree then select ***Copy*** from the context menu, then right click the second pocket operation and select ***Paste Format*** from the context menu. This will copy all the properties from the source operation into the destination, apart from information such as the operation's name and lists of source objects.

Generate the toolpaths again. If all is well, the routines should detect that you intend to do a island pocket and will generate toolpaths in between the 2 curves.

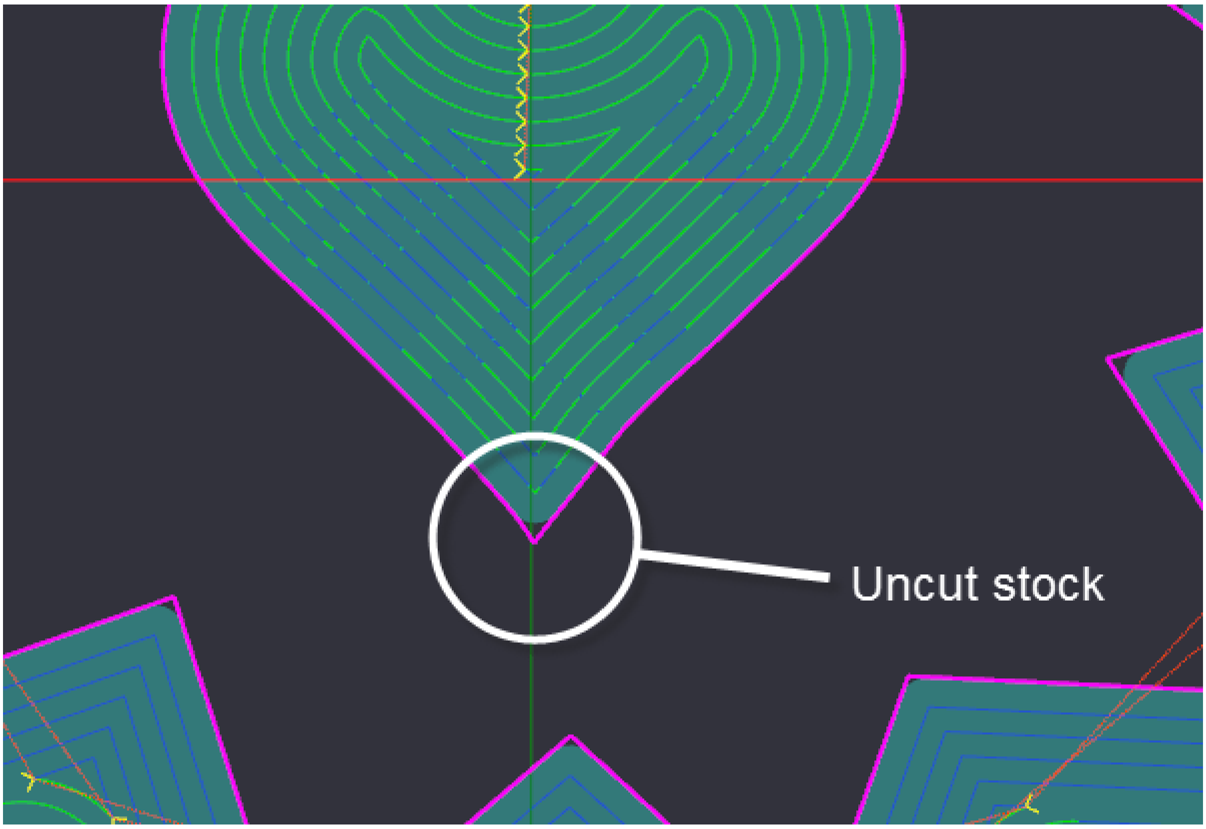


Step 5 - Show Cut Widths

Before we continue, we will turn on the ***Show Cut Widths*** option to indicate the areas that will be machined. From the main ***View*** menu, or the ***View*** sub menu of the context menu shown when right clicking the drawing, tick the ***Show cut widths*** option, if it is not already ticked.

Show cut widths will shade the areas that will be cut. It should be easy to spot any areas that are not shaded and will therefore have stock remaining.

In the image below there will be uncut stock in the inside corner shapes where the tool radius cannot reach without overcutting.



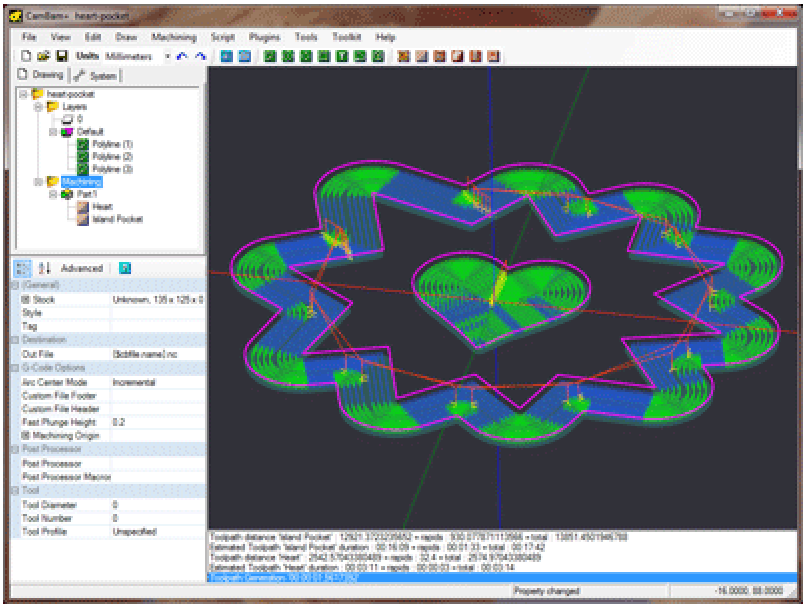
Step 6 - Machine operation renaming.

The drawing is basically complete and ready to save and generate gcode, but first we will do some cosmetic changes to help manage the drawing.

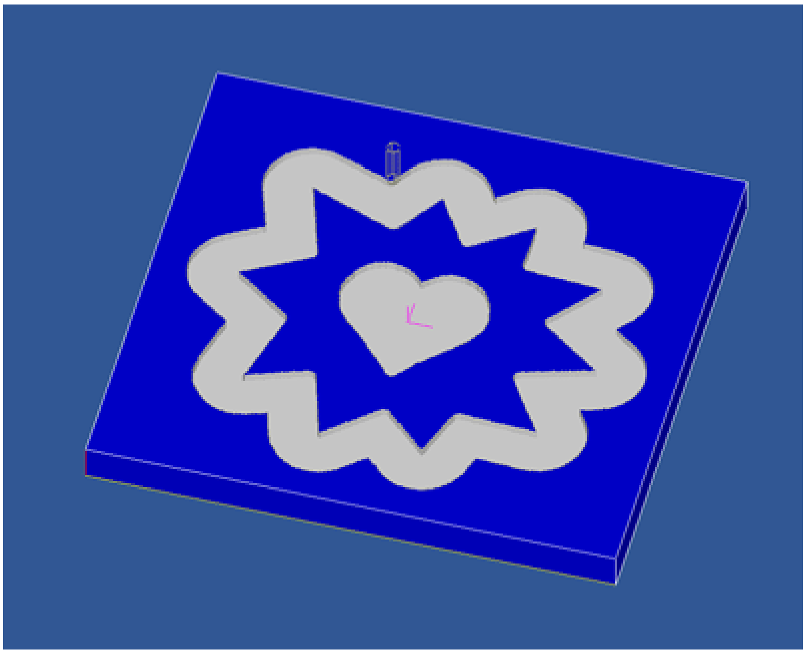
Machine operations can be given a more meaningful name, to help with readability and debugging. To rename a machine operation, select it in the drawing tree and press **`F2`**, or click the name a second time. Avoid using special characters in the name such as parenthesis as could cause problems due to nested comments.

To change the order of machine operations, click and drag them to the desired position within the drawing tree.

Create the gcode as normal. The new machine operation names will be present in comments within the gcode file. This is very useful for diagnostic purposes.



The following image shows the pocket simulated using *CutViewer Mill*



Tutorial: Drilling

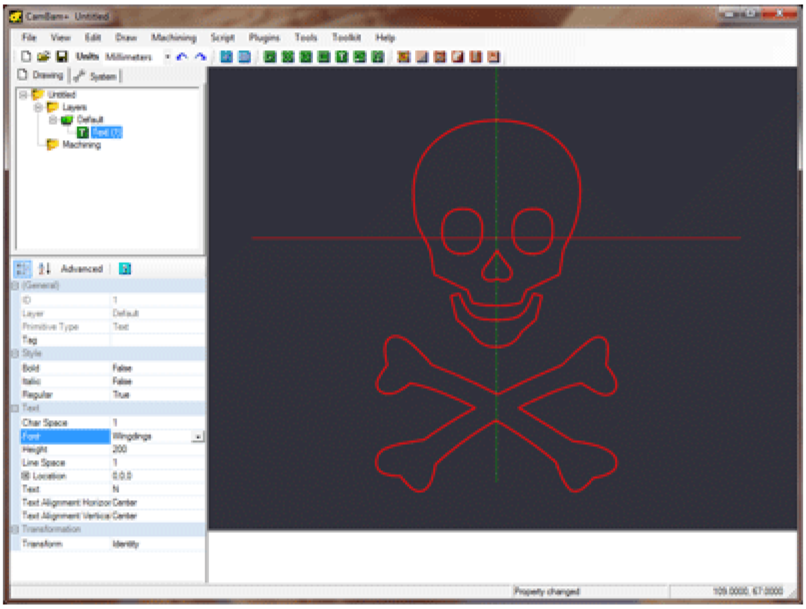
Creating drilling patterns is very easy. Here a Wingding font character ‘N’ is used to create a drilling pattern for an external hard disk enclosure.

Step 1 - Insert text

Drilling machine operations are based on point lists or circle centers. There are a number of routines in CamBam to generate point lists that can give interesting effects.

In a new CamBam drawing, insert a text object. The Wingdings capital ‘N’ character happens to be a natty jolly roger.

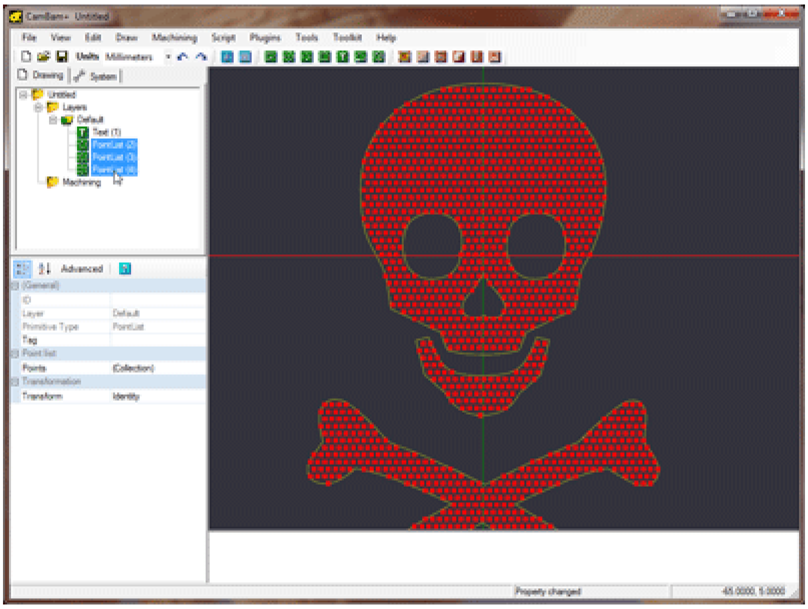
Set the text ***Height*** to something large like 200 (I am working in mm here), and change the ***Font*** property to *Wingdings*.



Step 2 - Fill text object with points

Select the text object then chose ***Draw - Points List - Offset Fill Geometry*** from the drawing context menu. This will prompt you for a step distance. Enter 2 here and press enter.

You should now have created a set of points which fill the selected geometry (excluding any holes in regions and text).



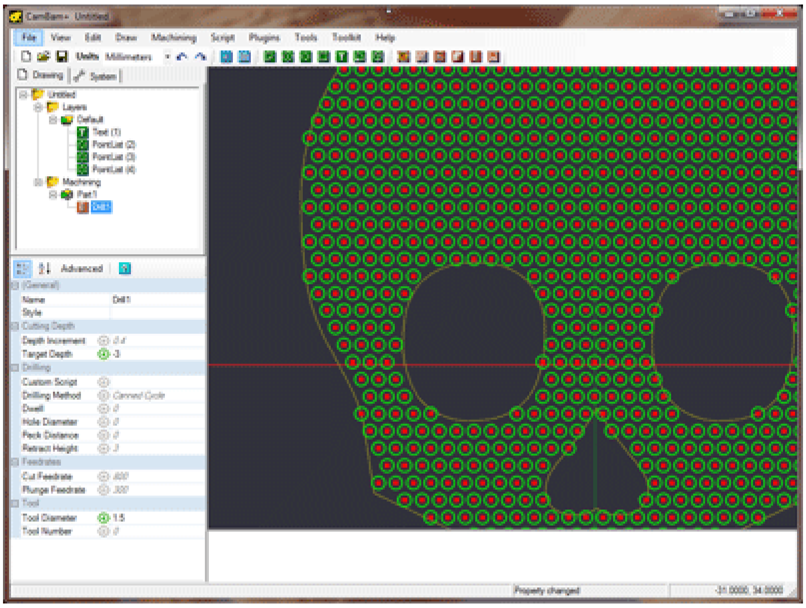
Step 3 - Insert a drilling machine operation

With the point lists selected, insert a drilling machine operation .

Under the drill mop properties, set ***Tool Diameter*** to 1.5 and the ***Target Depth*** to -3.

That’s pretty much it! To make things clearer, you can right click on the Default layer in the file tree and select ‘hide’.

You should now just see a bunch of circles indicating the drilling hole sizes.



Right click on machining in the file tree to generate the gcode.

Here’s one I prepared earlier.

This is the aluminium cover off an ICY BOX external USB hdd enclosure. Should look neat with some LEDS behind it.

Most geometry can be used to generate point lists. Try experimenting with the other ***Draw - Point List*** options.

Tutorial: Bitmap Heightmaps

This tutorial describes using the Heightmap plugin to generate pseudo 3D profiles from bitmaps. The same routine can also be used to generate photo engravings in two-toned materials and lithopanes. The source code for the heightmap plugin is also provided with CamBam for the adventurous.

Warning: The Heightmap plugin can produce gcode that plunges the full depth of your heightmap in one go. The engraving machining operation now supports a ***Depth Increment*** property that can be used to machine deep height maps in multiple passes. Another alternative is to use the ***Draw - Surface - From Bitmap*** operation to create a 3D surface mesh which can be used with the 3D machining operations.

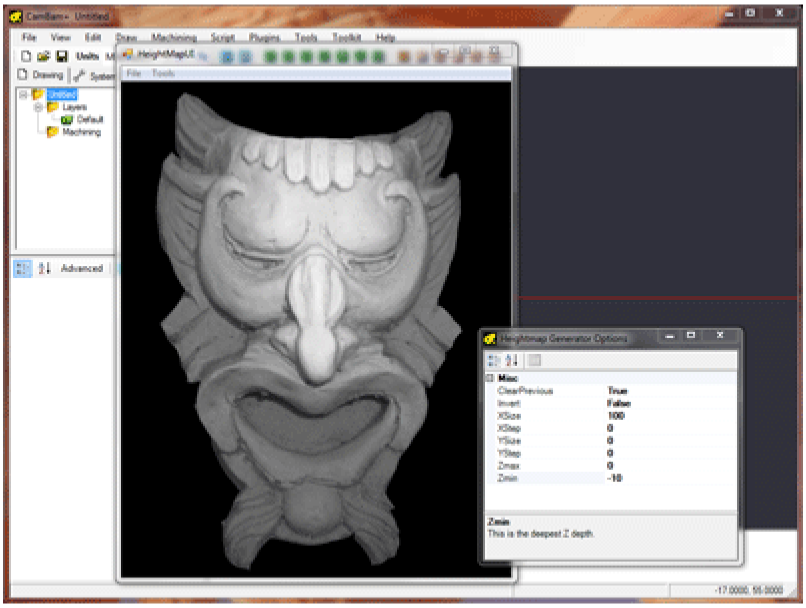
Step 1 - Open the Heightmap plugin

The height map plugin is accessed from the ***Plugins - HeightMap Generator*** option from the top menu.

CamBam Plugins are .NET class library .DLLs and are located in a plugin subfolder in the CamBam application folder; Typically:

```
C:\Program Files\CamBam plus 0.9.8\plugins
```

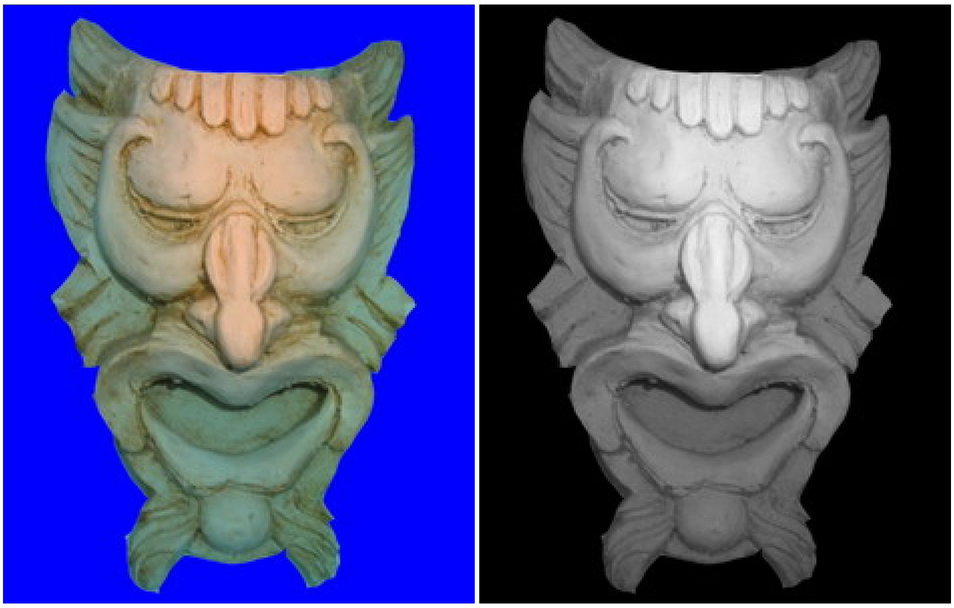
The source code for the Heightmap generator can also be found in a zip file in this folder.



Step 2 - Select a Bitmap File.

The success of a Heightmap depends largely on the quality of the source bitmap. Front lit objects with even shading usually work best.

Inspired by the inimitable greybeard’s experiments in the cnczone **3D for Crazies** thread, I photographed an object submerged in a tray containing water and blue food colouring. I then used a drawing program to filter the bitmap to just show the red channel as a greyscale image. In theory, the further the item is from the surface of the liquid, the more blue it will appear. This worked much better than I expected although care must be taken to avoid surface reflections and air bubbles. This is perhaps not such a good idea for making heightmaps of people.



Object in blue food colouring and object with Red Filter applied.

With the Heightmap generator window open, select ***File - Open*** from the top menu and select the source image.

Step 3 - Heightmap Options

Change the heightmap options from the Heightmap plugin's ***Tools - Options*** top menu.

Here is an explanation of the properties:

ClearPrevious	The ***Tools - Generate Heightmap*** menu option from the heightmap form can be called multiple times. If this option is set True, the heightmap previously created will be removed before a new heightmap is generated.
Invert	If True then darker colours are higher (larger Z values), otherwise lighter colours are higher.
XSize / YSize	Width (X) and Height (Y) of the heightmap in the same units as the current CamBam drawing. These values control the actual physical size of the resulting heightmap. If the ***YSize*** is set to 0, the aspect ratio of the bitmap will be applied to the ***XSize*** value to determine the Y height. Examples: ***XSize*** = 100 (mm), ***YSize*** = 0 ***XSize*** = 4 (inches), ***YSize*** = 0

XStep / YStep	A heightmap creates a series of scan lines, much the same as how a television image is created. The ***YStep*** value controls how far apart the horizontal scan lines are and the ***XStep*** values determines how far apart each point in the line is in the X direction. If either is set to 0, the height will be calculated at each pixel point. Examples ***XStep*** = 0, ***YStep*** = 0 (calculate height at each bitmap pixel) ***XStep*** = 0, ***YStep*** = 0.75 (mm) (calculate height at each pixel in one scan line, with each horizontal scanline 0.75mm apart) ***XStep*** = 0, ***YStep*** = 0.001 (inches) (calculate height at each pixel in one scan line, with each horizontal scanline 0.001in apart).
Zmax	This is the highest Z depth. If the stock surface is used to zero the Z axis, then ***ZMax*** would typically be zero as well.
Zmin	This is the deepest Z depth and represents the Z coordinate of the deepest cuts in the heightmap. Examples ***ZMax*** = 0, ***ZMin*** = -10 (mm) The heightmap heights will range from -10mm at the deepest to 0mm at the highest points. ***ZMax*** = 0.125 (inches) ***ZMin*** = -0.125 (inches) The heightmap heights will range from -0.125in at the deepest to 0.125in at the highest points.

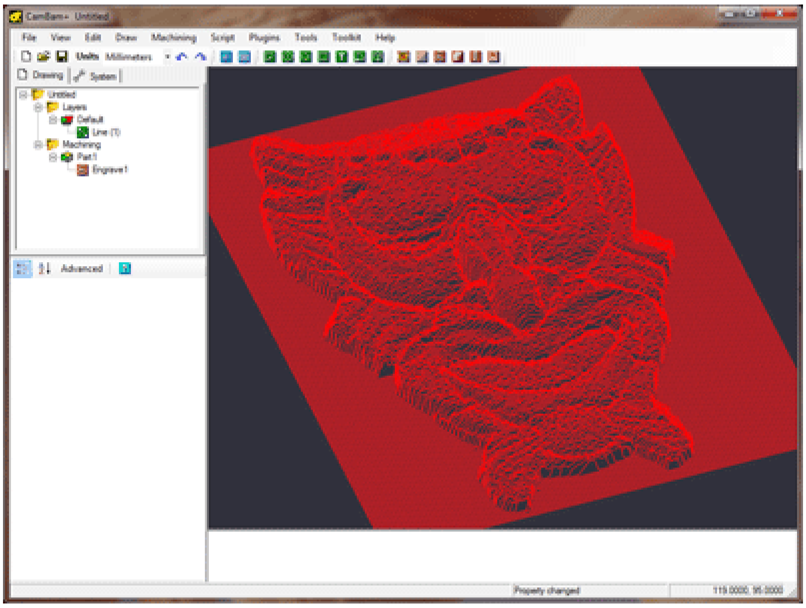
Step 4 - Generate Heightmap

Close the options window and select Tools - Generate Heightmap.

You should see some lines appear in the underlying CamBam drawing. Leave the Heightmap generator window open and rotate and scale the CamBam drawing to get a better idea of the Heightmaps dimensions.

More information on rotating, panning and zooming the drawing view can be found in the ***Rotating and panning the drawing*** section

Here is a screenshot of the resulting heightmap.



As well as generating a 3D Line object that contains the resulting heightmap, the plugin also creates an engraving machine operation linked to this line. An engraving operation is used as these are designed to 'follow' the associated geometry. In effect it is using the 3D line as a toolpath.

Change the Engraving machine operation parameters such as cutting feedrates.

NOTE: Do not change the engraving target depth value, the cutting depth is taken from the source line.

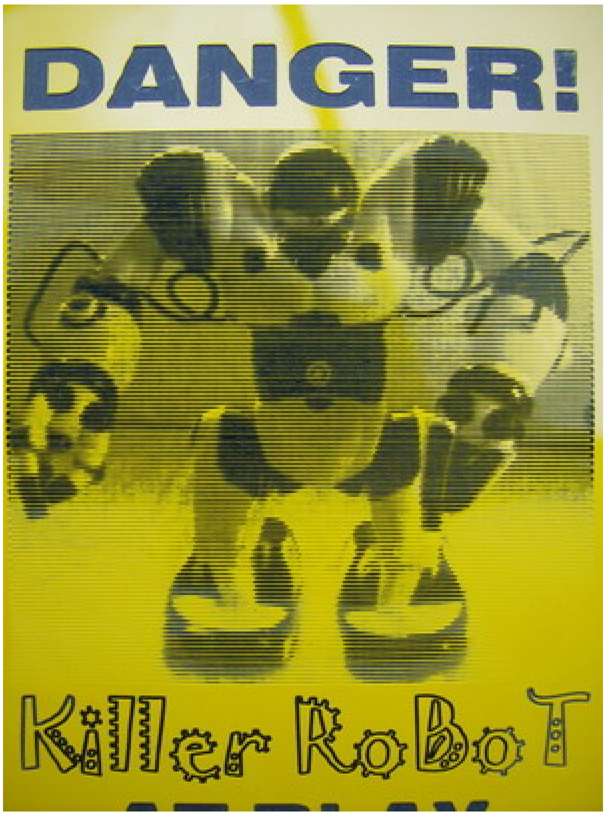
To convert the heightmap into gcode for your machine, right click the ***Machining*** group in the CamBam tree view then select the ***Create GCode File*** menu option.

Here is the very first Heightmap I produced from CamBam. The image is 120mm X 90mm using a 2mm flat bit in plywood. Not fantastic to look at but at least there were no disasters.

Photo Engraving

The heightmap process can also generate shaded engravings from bitmaps.

A V cutter is used, usually into a 2 tone engraving laminate. The deeper the cutter, the wider the cut and darker it will appear (if using a light on dark laminate). The Z depth needs only be small (~0.5mm, 0.02in). The YStep stepover should be increased so the 'scan lines' do not overlap and spoil the shading effect. This distance will vary depending on the V cutter angle and depth. For a 60 degree cutter at 0.5mm I use a 0.7mm YStep.



A lithopane is another variation on this theme, where an image is engraved into a thick translucent material and viewed with back lighting. Lithopanes are typically inverted with the deeper cuts resulting in thinner material and more light shining through.

Creating a Point Cloud From a Heightmap

Here is a method to generate a DXF point cloud.

Generate a heightmap polyline as per usual and select the line if it is not already.

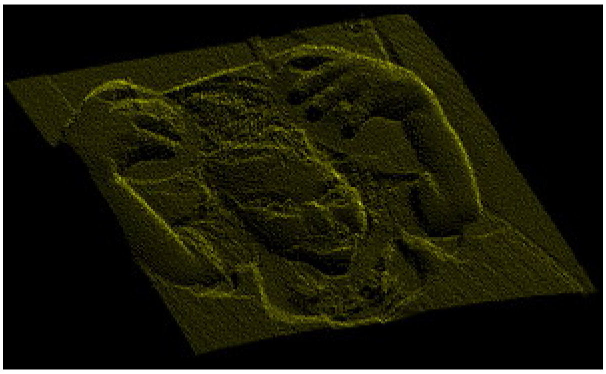
Now do ***Draw - Point List - Step Around Geometry***. from the drawings context menu.

This will insert a point along the line every N step distance.
By default, the heightmap will do 1 bitmap pixel = 1 drawing unit (This can be changed in the heightmap

options).
I entered 1 for the step distance then pressed OK.

CamBam currently displays points using biggish squares so it will look cluttered, but don't worry about that. The line object can now be deleted.

The drawing can now be exported to a DXF file. Here a heightmap pointcloud is viewed in AutoCAD.



Tutorial: Text Engraving

This tutorial describes inserting text into CamBam and generating an engraving operation from it.

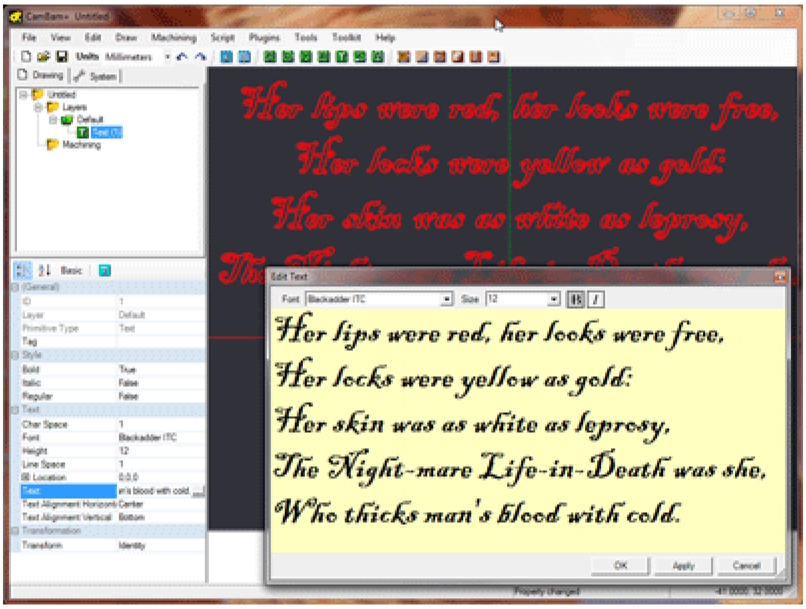
Inserting Text

To insert text into a drawing, use the ***Draw - Text*** menu option or the tool bar icon.

A multi-line text editor will be displayed. Insert the text the press OK. You will then be prompted for the location of the text by clicking on the drawing.

Note: By default, after entering a text item, the draw text command will be repeated and the text entry screen shown again. Press the ***Cancel*** button to end the text entry commands. This behaviour can be turned off by setting the ***Repeat Commands*** option in system configuration to *False*.

Text can be modified at a later stage by double clicking the text object in the drawing tree, or by clicking the ellipsis [...] button to the right of the Text object's ***Text*** property.



Refer to the Text drawing section for details on the text drawing object. (CAD Entities)

Creating Engraving GCode

To create some engraving gcode, select the text then select the ***Machining - Engrave*** menu option or use the toolbar icon.

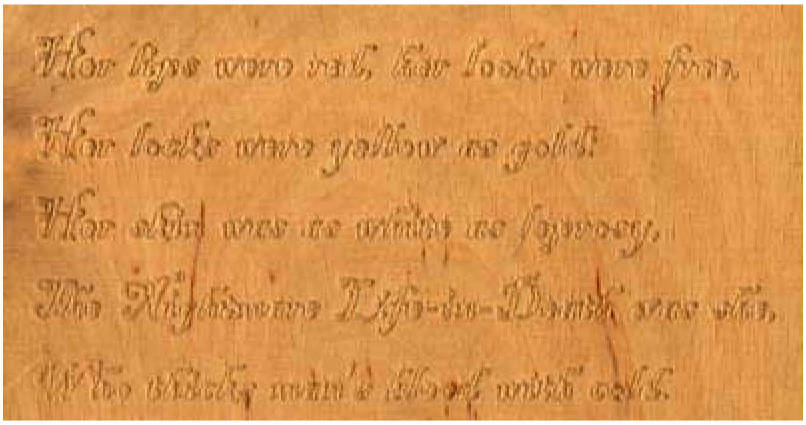
For shallow engraving (0.3mm), try these properties:

Depth Increment = 0.3

Tool Profile = V-Cutter

Note: The default CAM Style used an *Auto* setting for ***Target Depth***. For engraving operations using a V-Cutter, the target depth is automatically calculated to be one depth increment below the stock surface.

Sample Engraving…



Not exactly high art, but the letters are quite small (3-6mm) and plywood isn't the best precision engraving material.

Single Stroke (Stick) Fonts

To create thin engraving, ideally a 'stick' font should be used, that is a font with no thickness. Unfortunately, True Type Fonts (TTF) do not support open shaped fonts so engraving results with TTF fonts can be hit and miss.

GeorgeRace has created some excellent 'Stick Fonts' and has made them available on the CamBam user forum here. (and they are also available on the plugin Website)

Tutorial: 3D Profile

This tutorial gives an introduction to ***the 3D Profile*** operation, and covers:

- Loading 3D models, sizing and positioning.
- Front face waterline roughing.
- Front face scanline finishing.

Loading 3D models, sizing and positioning

Loading

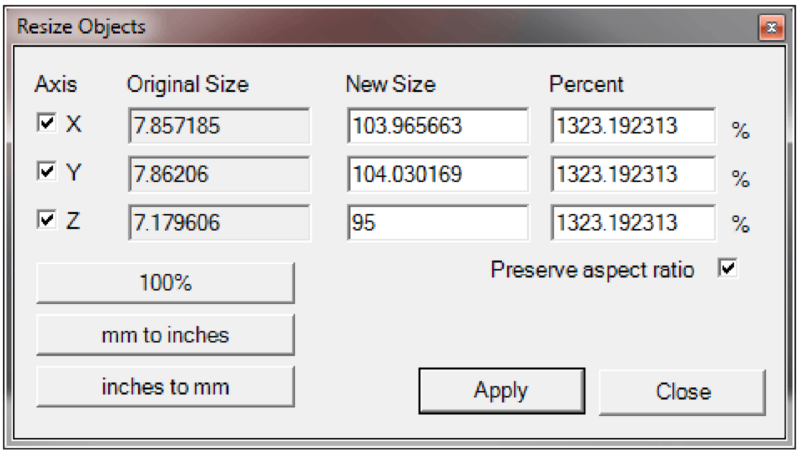
CamBam can read .3DS file, .STL and .RAW 3D mesh files (and STEP since V1.0). These can be loaded using the ***File - Open*** menu option or by dragging files onto the CamBam drawing window.

If an imported object is not immediately visible, it may be because its default dimensions are very small compared to the currently display stock object. If this is the case, temporarily hide the stock using ***View - Show stock***, then use ***View - Zoom to fit***.

To machine successfully, the 3D model needs to be aligned within the intended machining area. This may involve combinations of the following transformations.

Sizing

To make the model a fixed size, the ***Transform - Resize*** command can be used.
This will open the Resize window which will show the existing object dimensions and allow them to be resized to a specific dimension, or by a scaling percent.



Rotating

The model should be rotated so that it is facing toward the screen (i.e. in the positive Z direction).

Transform - Rotate can be used to rotate selected objects. First select a rotation point, and then move the mouse around this point to select a rotation angle. Press the X, Y or Z keys to change the current axis of rotation. If snap to grid is enabled, the rotation angle will snap to multiples of 30 and 45 degrees.

Selected objects can also be rotated by using the ***transformation property editor***.
Rotations follow a right hand rule, so to visualise this, point your right thumb in the direction of the positive axis of rotation.
A positive rotation is then in the direction that your fingers curl around the axis.

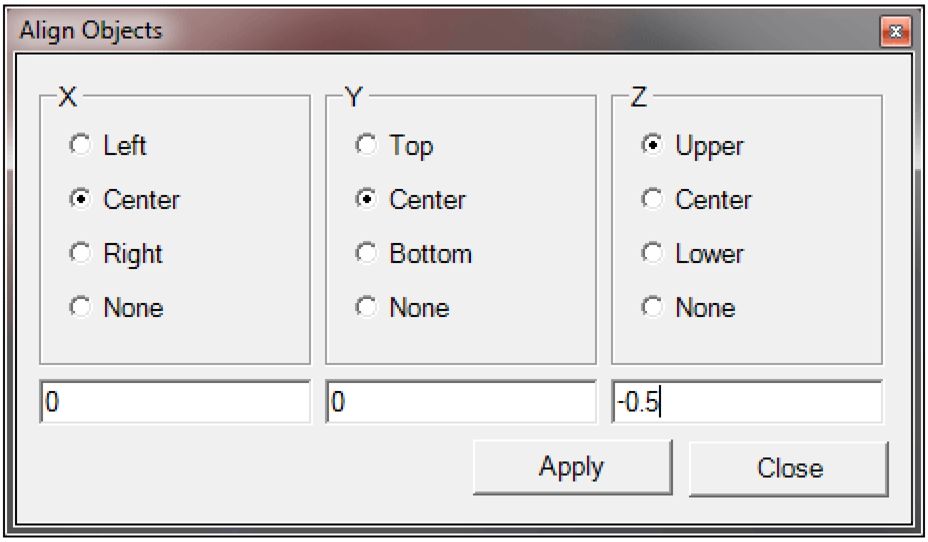
Another alternative is to use free-hand rotation. This is done by selecting objects, then holding the **SHIFT** key while using the view rotation key+mouse combination (ie **ALT** + left mouse drag or Center mouse + Left mouse drag, depending on your configuration settings).

Positioning

Transform - Align can be used to position selected objects. This will display a form with 3 columns, one for each axis. Select the point of the selected axis to align, or none to leave the current axis position intact. Enter the drawing coordinate underneath which will be the new location of the alignment point, then press ***Apply***.

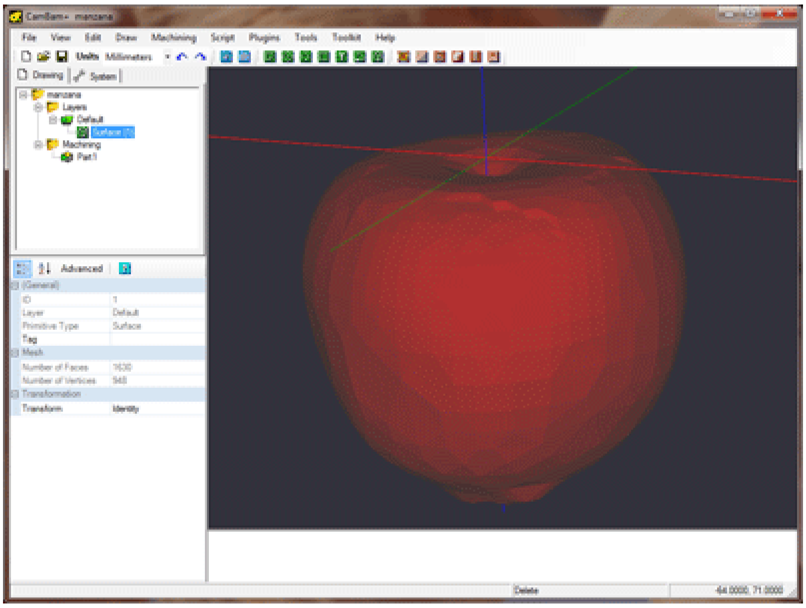
For example, to position an object so that it's lower left corner is at the drawing origin and the highest Z point is just below the stock surface (if using Z=0), use the following alignment values:

```
X - Left, Value = 0
Y - Bottom, Value = 0
Z - Upper, Value = -0.5
```



It may be more convenient to reference the machine's Z=0 to the work table, then use a ***Stock Surface*** value that is the Z height of the stock. This works well when the stock used has an uneven surface or is difficult to reference a tool to (particularly after a roughing pass). This can also simplify back face machining. If using this method, use the following Z alignment options:

```
Z - Center, Value = 0
```



This image shows a 3D model loaded, sized and positioned

Front face waterline roughing

Waterline roughing is an efficient way to clear the bulk of the stock around the 3D model.

Create a 3D Profile operation

Select the 3D surfaces to machine, then insert a 3D Profile machining operation (***Machining - 3D Profile***) or select the icon from the toolbar.

If a ***Stock*** object has been correctly defined, some properties may be automatically calculated, such as ***Stock Surface*** and ***Target Depth***, as the default CAM Style has these values set as Auto values.

Basic Properties

Note: dimensions shown here are metric.

Property	Value	Notes
Profile 3D Method	*Waterline Rough*	
Depth Increment	*3*	The maximum Z depth per cut of each machining layer.
Lead In Move	*Spiral* *3 degree*	As well as making life easier for the cutter, this also gives the Fast Plunge Height behaviour a reference point which helps avoid slow plunges.
Roughing Clearance	*1*	Leave a small amount of stock to be cleared away in the finishing pass, to avoid waterline machining marks being visible.
Stock Surface	*0*	In this example, Z=0 is referenced to the stock surface.
Target Depth	*-50*	If the model is 100 units tall, this will machine the top half of the model.
Tool Diameter	*6*	To increase roughing speed, use a larger tool.
Tool Profile	*End Mill*	Scanline (horizontal /vertical) and waterline finishing

		methods will adjust tool paths for bull/ball nose cutters. For waterline roughing operations, the tool profile does not affect the tool path.

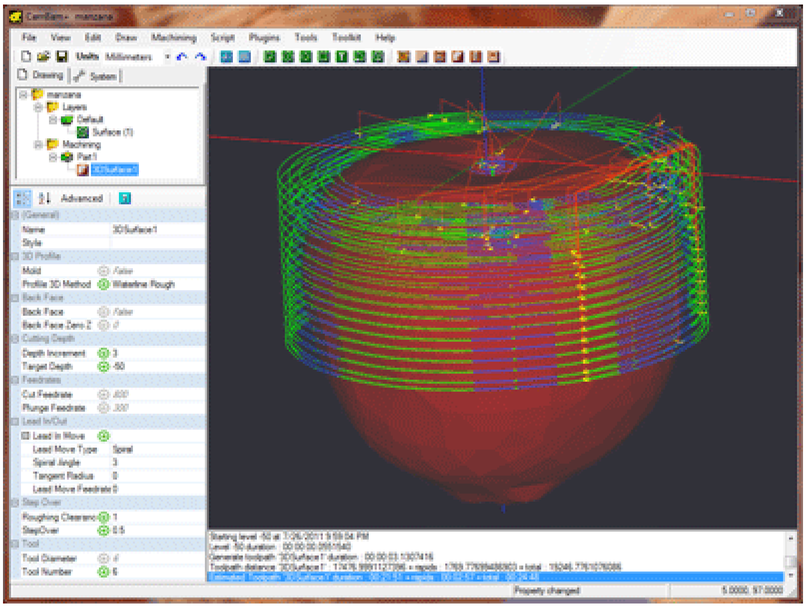
Advanced Properties

Property	Value	Notes
StepOver	*0.5*	Distance between toolpaths expressed as a fraction (0-1) of cutter diameter.
Plane Slice Only	*False*	CamBam's waterline routines have been designed to work best with natural / curved shapes. Engineering shapes with perpendicular sides can potentially cause problems. If problems are encountered, setting Plane Slice Only to True can help but will only work with shapes that do not have any overhangs.

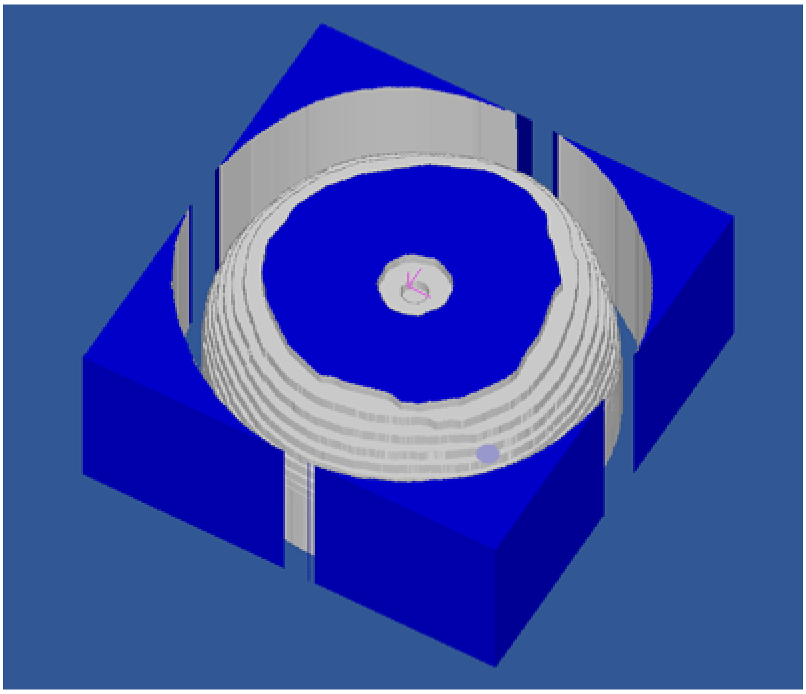
General Settings

There are some properties under ***Machining*** that are useful when working with 3D files.

Property	Value	Notes
Rebuild Toolpath Before Post	*Prompt*	3D Toolpath generation can take many minutes. This option will prompt whether to regenerate toolpaths before creating gcode. If 'No' is specified, the post processor will use the last generated toolpaths.
Fast Plunge Height	*0.2*	A small value here allows the post processor to rapid down to the fast plunge height distance above the last cut stock depth and can speed machining times considerably. ***Warning!*** Care should be taken with this setting, especially for machines with flex or backlash. Setting ***Fast Plunge Height*** to a little larger than ***Depth Increment*** should be safest. .
Toolpath Visibility	*Selected Only*	Having front roughing, finishing and back face toolpaths visible is very confusing. This option will only show the toolpaths for the machining operation currently selected in the drawing tree. ***NOTE:*** From version 0.9.8 this option is now set in the file's properties (the first object in the drawing tree).



This image shows the waterline roughing toolpaths.



Simulation of the roughing pass in *CutViewer Mill*

Scanline finishing

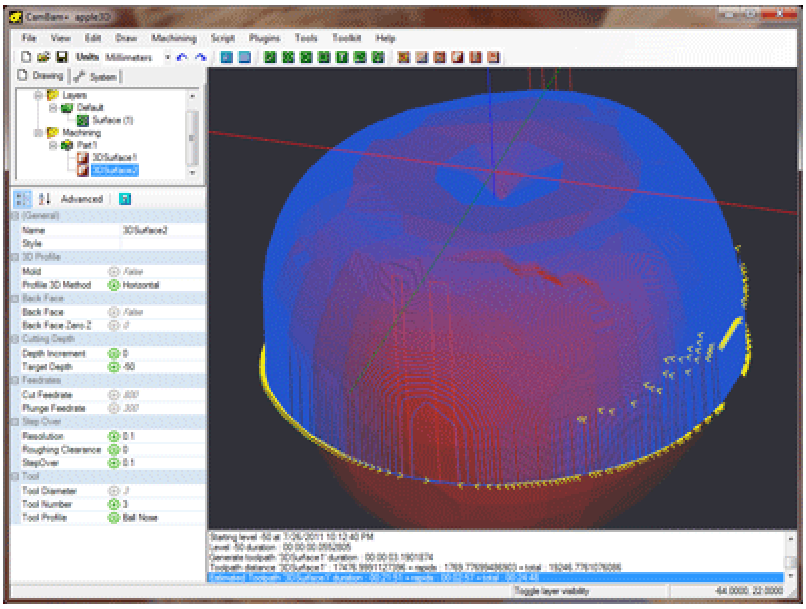
Once the bulk of the material has been cleared by roughing, a scanline finishing pass can be applied. Select the 3D surface and insert a second ***3D Profile*** operation.

This time, set the ***Profile 3D Method*** to *Vertical* or *Horizontal*. To attain a finer finish, with less toolmarks, a horizontal pass followed by a vertical finishing pass can be used.

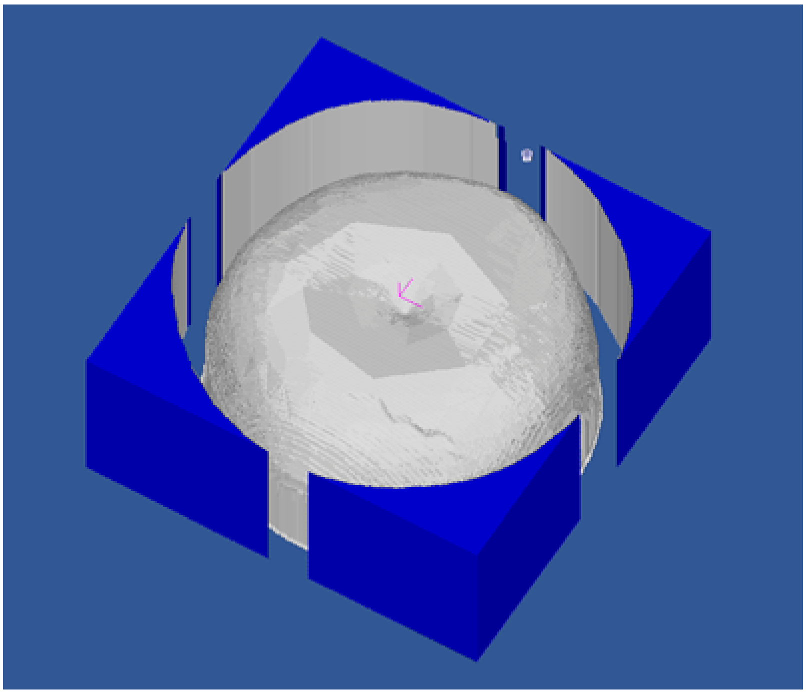
Basic Properties

Note: dimensions shown here are metric.

Property	Value	Notes
Profile 3D Method	*Horizontal or Vertical*	
Depth Increment	*0*	Depth increment should be 0 to do a single finishing pass.
Target Depth	*-50*	Use the same target depth as the roughing pass.
Roughing Clearance	*0*	No roughing clearance - will clear off clearance stock from the roughing pass.
StepOver	*0.1*	Distance between toolpaths expressed as a fraction (0-1) of the cutter diameter. Smaller stepovers will give a nicer finish but take longer to machine.
Resolution	*0.1*	This is the distance along toolpaths expressed as a fraction (0-1) of the cutter diameter, at which the height of the model is tested. 0.1 should be adequate, but a smaller value could be used if inaccuracies occur (especially around small features or perpendicular edges).
Tool Diameter	*3*	A smaller tool will result in more detail but takes longer to machine.
Tool Profile	*Bull Nose*	The horizontal and vertical scanline, as well as waterline finish, methods will adjust tool paths for ball nose cutters.



This image shows the scanline finishing toolpaths



The finishing pass simulated in CutViewer Mill

Adjusting the machining boundary

The 3D profile operation will machine the minimum area around the objects. To change this behaviour, a number of boundary options can be defined.

Property	Value	Notes
Boundary Margin	*2*	Adds a small extra margin around the shape outline boundary.
Boundary Taper	*3*	Tapers the boundary edge which helps give cutter clearance at lower depths.

Back face machining

Please refer to the 3D Profile - back face tutorial.

Tutorial: 3D Profile - back face machining

This tutorial explains some more advanced concepts of the ***3D profiling operation*** and covers:

- Back face machining.
- 3D Holding Tabs.

Back face machining.

Back face machining is very similar to the front face roughing and finishing passes, with a few extra parameters to control the back face machining behaviour.

The front and back faces may be machined in a single piece of stock, by flipping the stock over after the font face has been machined. Alternatively the front and back faces may be machined in separate pieces of stock which can then be fixed together.

The ***Back Face Zero Z*** parameter is a key concept to understand. The 3D model is in effect, flipped over to machine the reverse side. ***Back Face Zero Z*** determines the current Z coordinate that will become Z=0 when the model is flipped.

Referencing Z=0 to the machine's work surface and setting a positive stock surface value will result in the model being rotated about Z=0. In this case ***Back Face Zero Z*** should be set to 0.

If the top of the stock is referenced as the machine Z=0, ***Back Face Zero Z*** would be the deepest Z coordinate of the model. When the model is flipped, this point will then be at or ideally just below the stock surface (Z=0) plane.

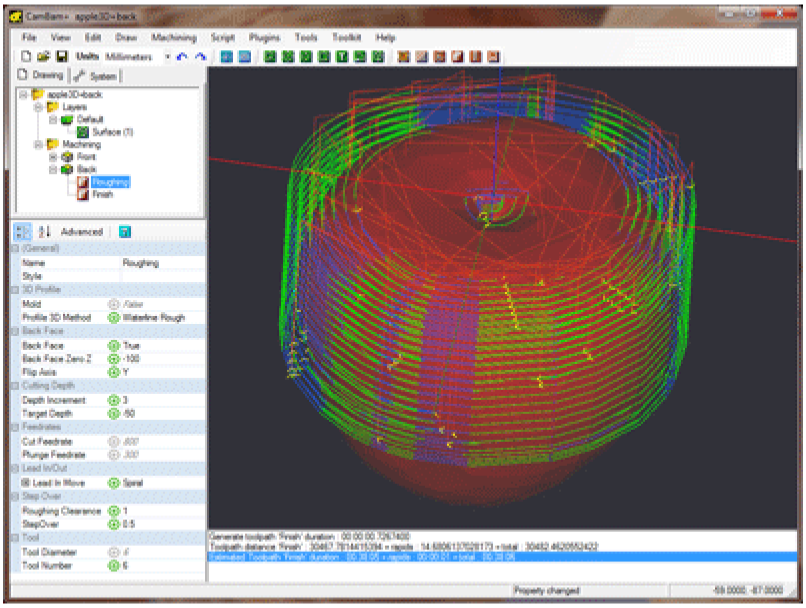
Basic Properties

Hint: If a Part has already been created containing the roughing and finishing passes for the front face, creating the back face can be simplified by copying and pasting the Part used by the front face and then altering the properties specific to back face machining.

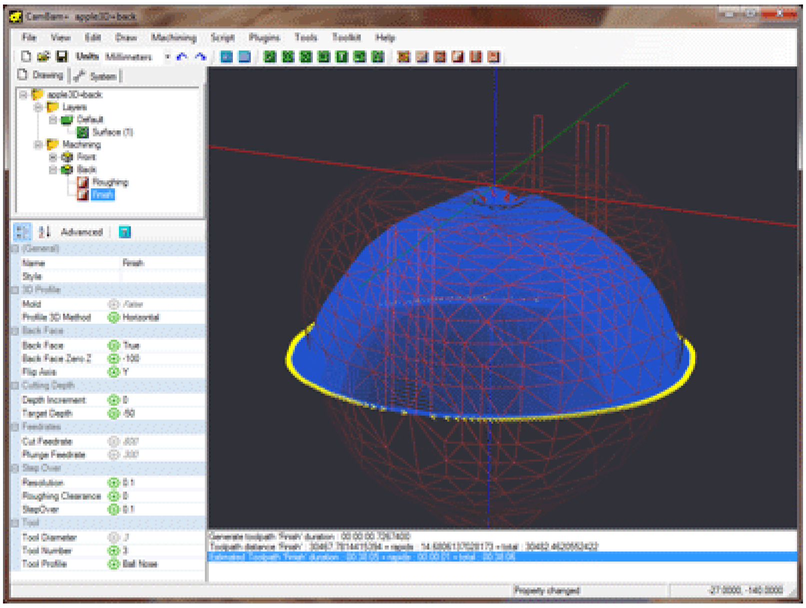
Property	**Value**	**Notes**
Back Face	*True*	
Back Face Zero Z	*0*	In this example, the table surface Z=0 is used, so the model is rotated about Z=0 to machine the back face.
	-100	In this example, the stock surface Z=0 is used, the model is around 100 units tall and is aligned so that the highest Z point is at or just below the stock surface (Z=0).
Flip Axis	*X*	The stock will be rotated around the X axis (top to bottom), when the back face is to be machined.
	Y	The stock will be rotated around the Y axis (left to right), when the back face is to be machined.

The back face toolpaths will be displayed in the orientation they will be machined, which may overlay the top face of the surface.

Hiding the drawing layer containing the 3D surface, or setting wireframe view mode makes viewing the toolpaths easier.



Back face roughing toolpaths



Back face finishing toolpaths (with wireframe view)

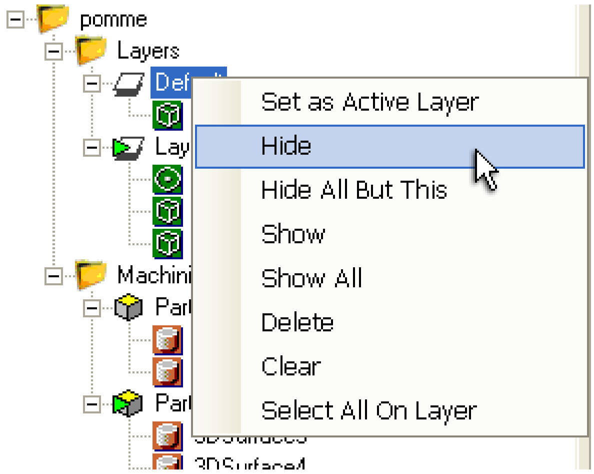
3D Holding Tabs.

There is currently no automatic 3D holding tabs functionality, but this feature is planned for a future release.

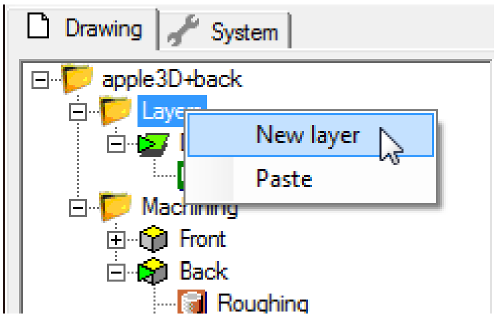
Here is a method to manually create 3D holding tabs or sprues using cylinder meshes.

Extrude a circle

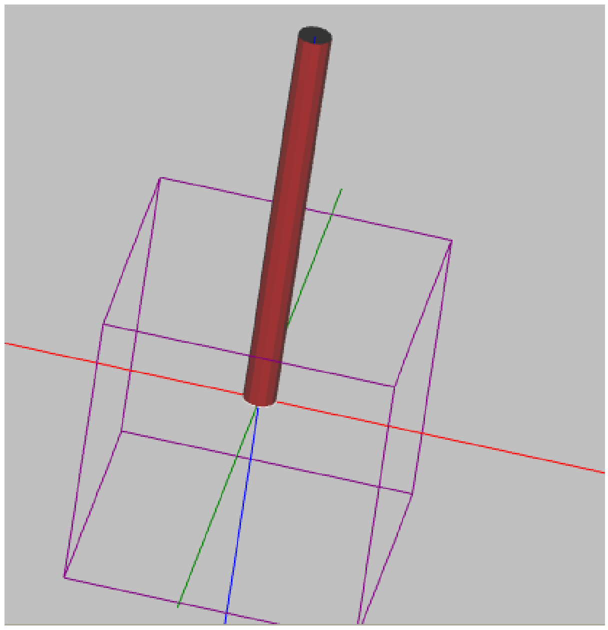
Hide the drawing layer containing the 3D mesh.



Create a new layer to contain the holding cylinders ('tabs').



Draw a 2D circle with a diameter of the holding tabs to be used. Place the center of the circle at the drawing origin (0,0).



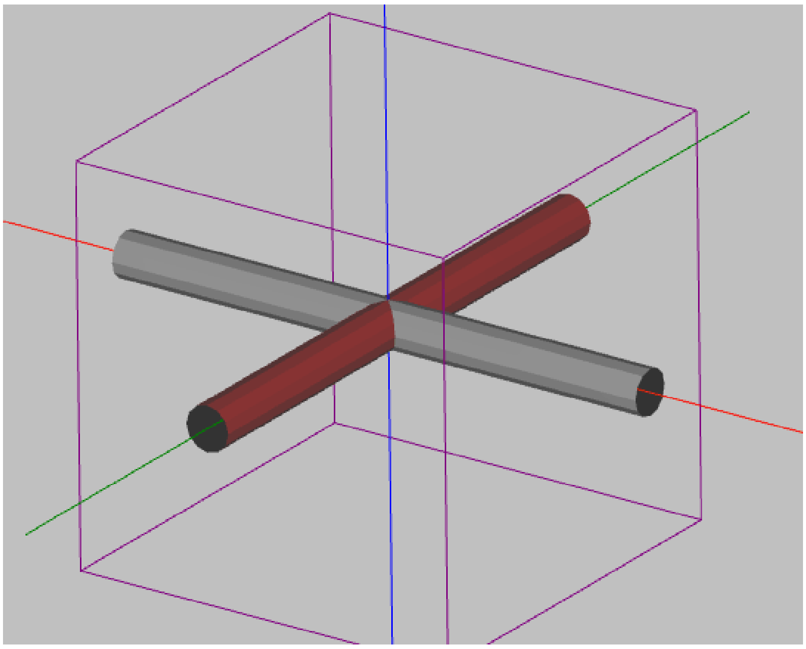
With the circle selected, select ***Draw - Surface - Extrude***. Enter an extrusion height large enough to span the largest width of the model plus an extra margin to allow for tool diameters.
You will also be prompted for the `Arc Expand Tolerance`. Curves will be converted to flat facets when extruding. This tolerance setting controls how much error or deviation is allowed from the curve (measured in drawing units).

Rotating the drawing view should show a 3D cylinder extending in the positive Z direction.

Position the cylinder

First, center the cylinder (***Transform - Center (Extents)***).

Use a combination of cut and paste and transformation rotations to position the cylinders at required positions around the model.

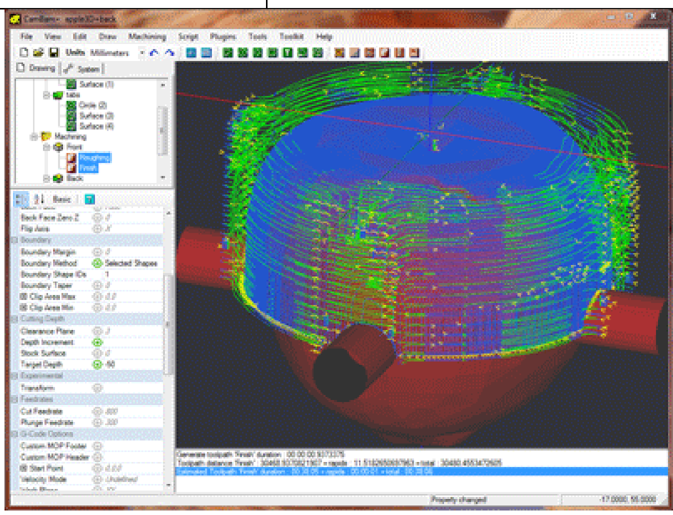


Adjust the machining boundary

The holding tab shapes need to be added to the 3D profiles list of surfaces to machine. To do this, right click on the machining operation in the drawing tree and select ***Select Drawing Objects***. `Ctrl`+click to select the extruded cylinders.

To prevent the machining operation machining around the ends of the cylinders, we need to reduce the boundary shape.
This is achieved by specifying *Selected Shapes* in the ***Boundary Method*** property and selecting only the main surface object, excluding any holding tab cylinders.

Property	Value	Notes
Boundary Method	*Selected Shapes*	This will create a trimming boundary just from selected shapes.
Boundary Shape IDs	*1*	Enter the ID of the main 3D surface to machine, excluding any holding cylinders. The [...] button to the right of the property can be used to select the shapes.



Front roughing toolpath with manual holding tabs.

CamBam V1.0 and use of the 4th rotary axis.

Basically, *CamBam* is not designed to be used in 4-axis machining, but with some manual "additions" in the machining operations and some plugins, it is possible to use it for machining with one A or B rotary axis.

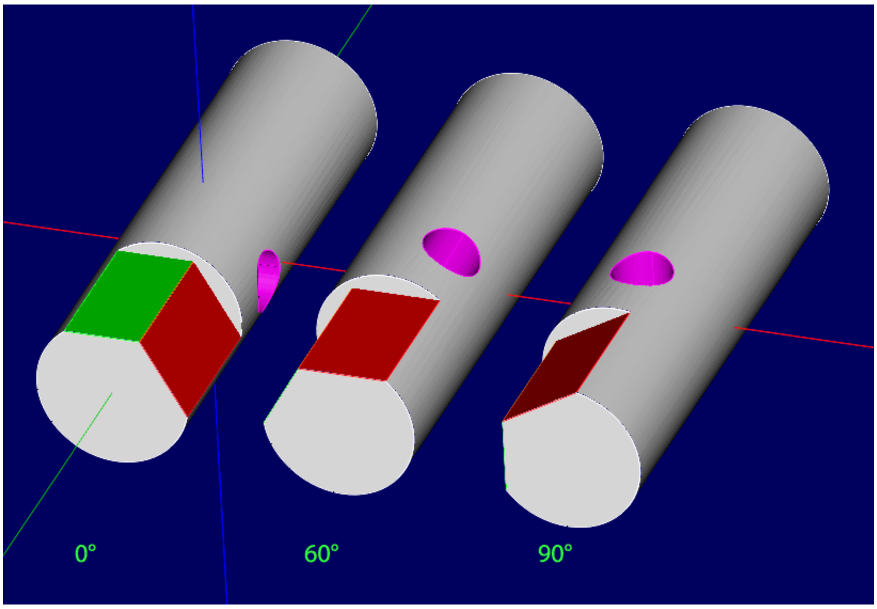
CamBam files as well as 3D objects are available in this archive

1. Positional machining.

a) Simple positional machining.

The principle is simple, the rotary axis only serves to position the part at a desired angle, then locks in position; then we can machine in XYZ as usual.

An example with the following part:



We will start with the rotary axis at 0° and we will machine the green area ; in this case a simple surfacing can be done with a pocket or profile operation.

We will then rotate the rotary axis by 60°, then we will surface the red area in the same way.

And finally, we will rotate the axis another 30° to reach the 90° position and we will use a drilling operation to machine the hole.

The rotation information will have to be added manually in the machining operations concerned. We will see later how to do it.

Note that the 3D object is only there to make things more readable ; in reality it is not necessary until 3D profiling operations are used.

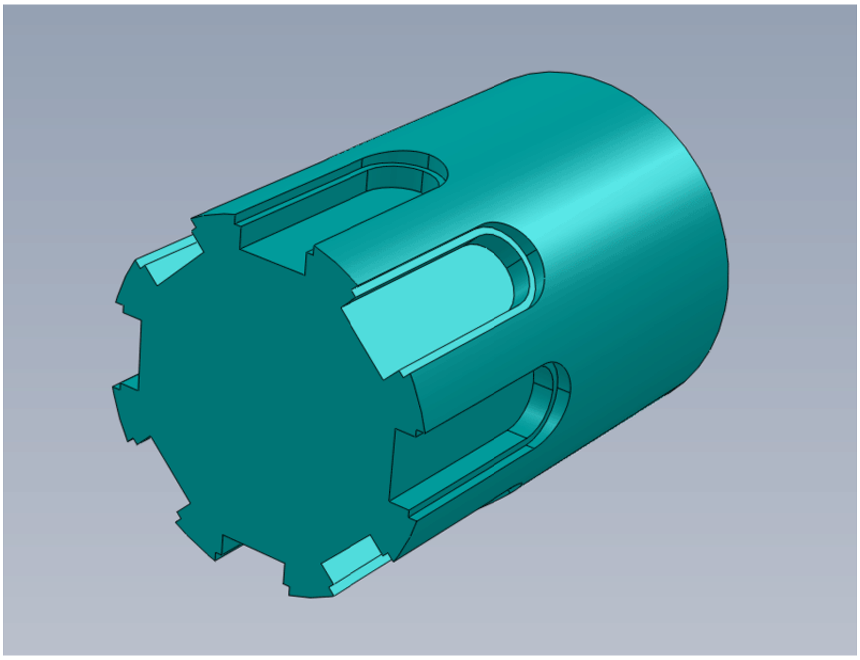
This method can be used with all types of machining operations, including 3D machining operations: in this case it will of course be necessary to have a 3D model and it will be possible to use the method of double-sided machining of 3D machining operations to make the top/bottom faces.

b) Positional machining with repetitions

In the previous case, each machining is different, whether by its type (surfacing, drilling), its machining parameters (the depth to be machined is not the same on the green face and the red face) and the angle positioning is not repetitive, which makes it necessary to define a specific rotation for each machining.

One can however have a part on which one wants to repeat the same machining in a regular way on the periphery of the part ; example machining a 6-sided or a splined shaft. In this case, there will be one or more machining operations (surfacing, pocket, drilling, or even 3D machining, etc.) repeated 6 times at 60° intervals for a 6-sided. We will then use the *CamBam* nesting function to repeat the machinings to which we will add an axis rotation command. (a rotation value relative to its last position).

Example of a part that can be made by repetition at regular angular intervals. (here every 45°)



Note that these two methods can also be used with a non-motorized 4th axis; in this case it will be up to you to turn and lock the 4th axis to its new machining position ; it will suffice to insert an M0 (pause) in the Custom MOP Footer property of the last machining operation to stop the machining so you can turn the axis to its new position befor to continue the machining.

2. Wrapping GCode

a) Simple engraving

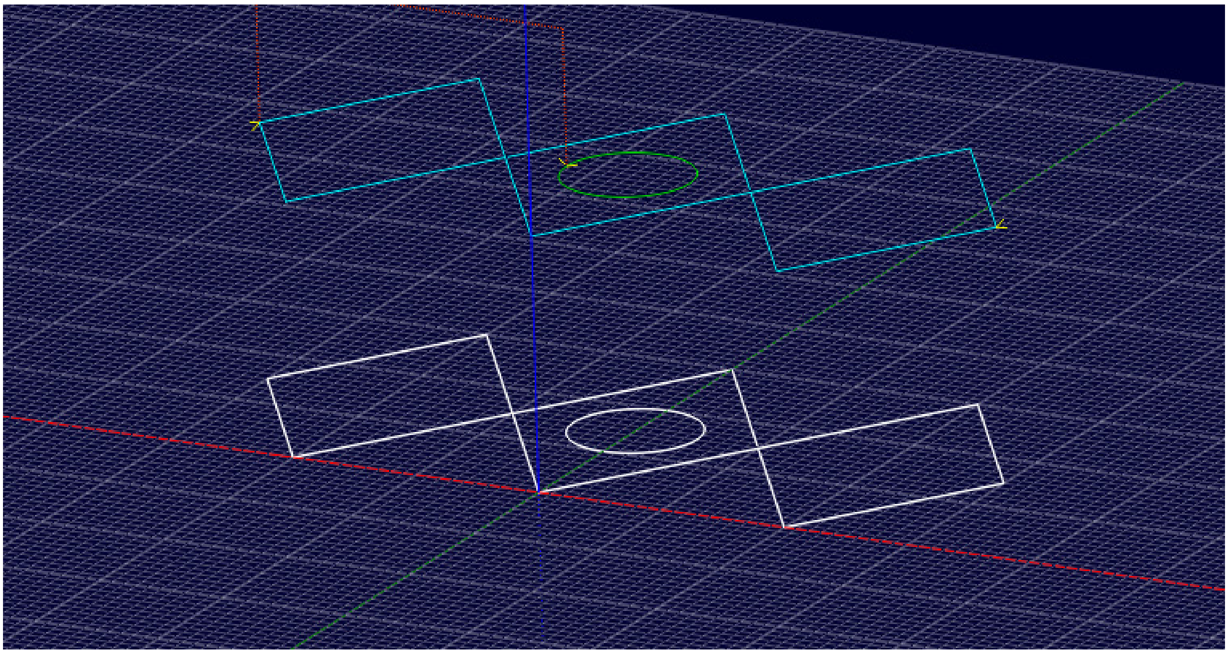
Another method, suitable for other types of machining, consists of working "flat", in other words, from *CamBam*'s point of view, the machining is done in XYZ, then one of the coordinates X or Y, depending on the orientation of the 4th axis on the machine, is converted into an angular position. It's like wrapping the machining around a cylinder. On *CamBam* V1.0, the post-processor will convert the X or Y coordinates into A or B coordinates.

An example with text written "flat" and then wrapped around a cylinder.

In this case, the GCode produced is a simple 3-axis XYZ GCode, once produced, the GCode then undergoes a modification to convert the cartesian coordinates of the X or Y axis into polar coordinates for an A or B axis. This conversion can be done using a plugin (Wrapper), or if you are using version 1.0 of *CamBam*, it can be done directly by the post processor (which must have been selected for this).

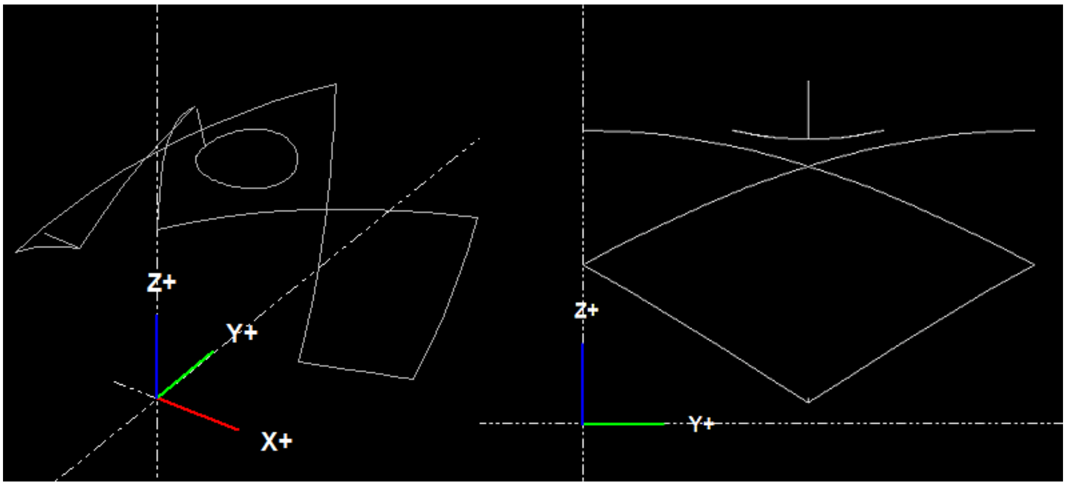
Example, engraving of a simple logo on a 40mm diameter cylinder.

The "flat" toolpaths in *CamBam*.



Note that the toolpaths are at radius in Z (blue toolpath), **the drawing itself is at Z=0**; it is important with the *CamBam* built-in engraving opeation.

Once the GCode is done and wrapped, we get this result.



All types of machining operations can be wrapped.

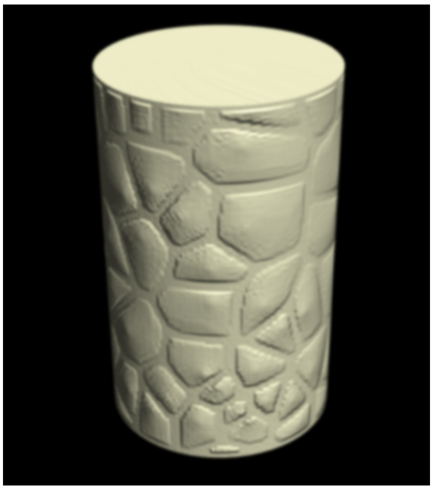
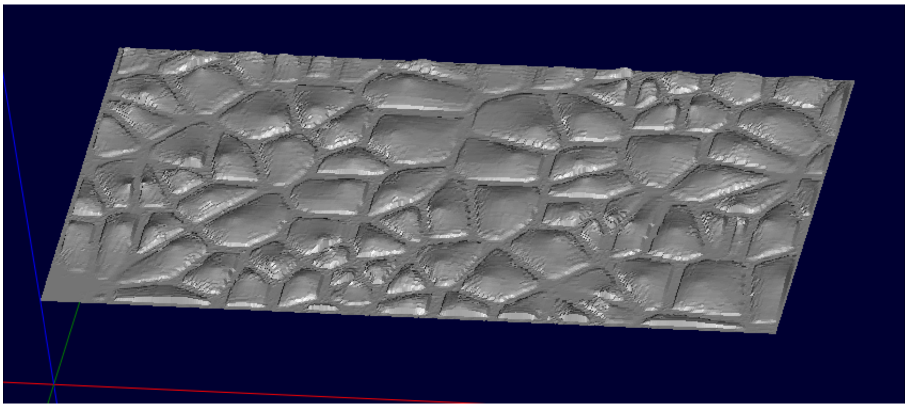
Engraving and 3D machining operations will need to take into account their specificities, in particular the way the Z is managed in engraving (*), which differs from other machining operations.

(*) With the engraving operation, the Z coordinates for ***Stock surface*** and ***Target depth*** are relative to the Z position of the lines to be engraved and are not given in absolute coordinates as in other operations. I'm talking about the standard *CamBam* engraving operation ; the V-Engrave plugin works like other machining operations, ie. the Z position of the lines to be engraved has no effect, the values of ***Stock surface*** and ***Target depth*** are in absolute coordinates.

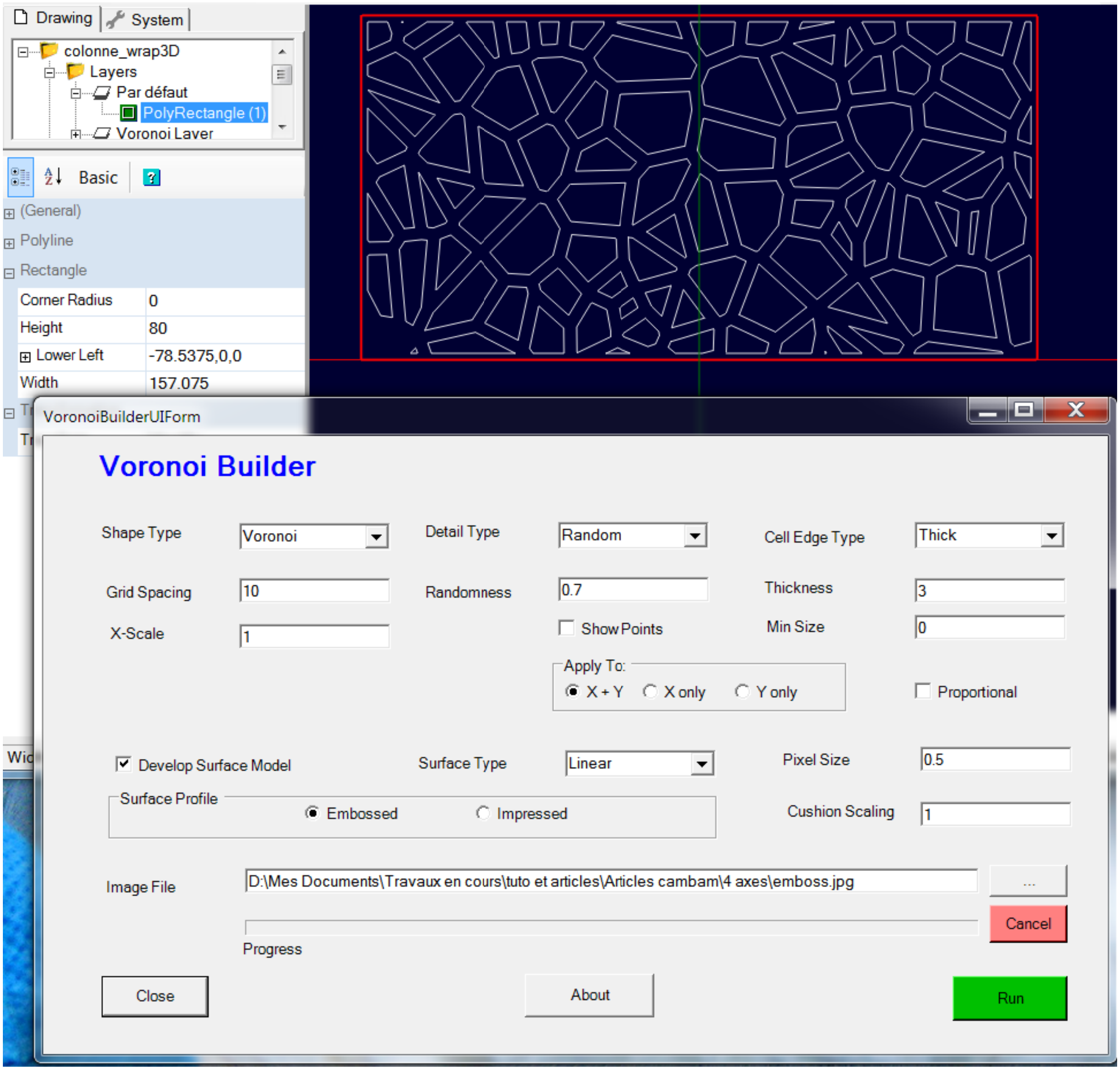
With *CamBam* V1.0 and wrapping via the post-processor (*RotaryY* or *RotaryX*), it is the ***Stock surface*** value that is used to define the wrapping radius, with the engraving operation it is therefore necessary that the drawing be at Z=0 and that ***Stock surface*** = the radius because unlike other machining operations where the value of ***Stock surface*** is an absolute value, independent of the Z position of the drawing, this is not the case with engraving where the value of ***Stock Surface*** is relative to the Z position of the drawing.

b) GCode wrapping used with 3D machining operations.

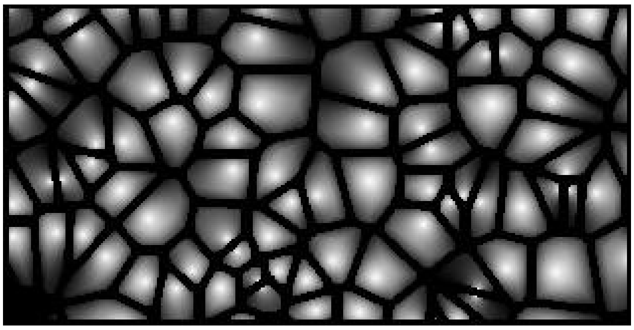
It is also quite possible to wrap 3D machining operations, such as this column which can be obtained following a wrap of a 3D profiling operation on a surface object (a mesh) representing a 3D stone texture.

The 3D texture above was made in *CamBam* using the Voronoi Builder plugin, which allow the creation of 2D cells (stones) as below



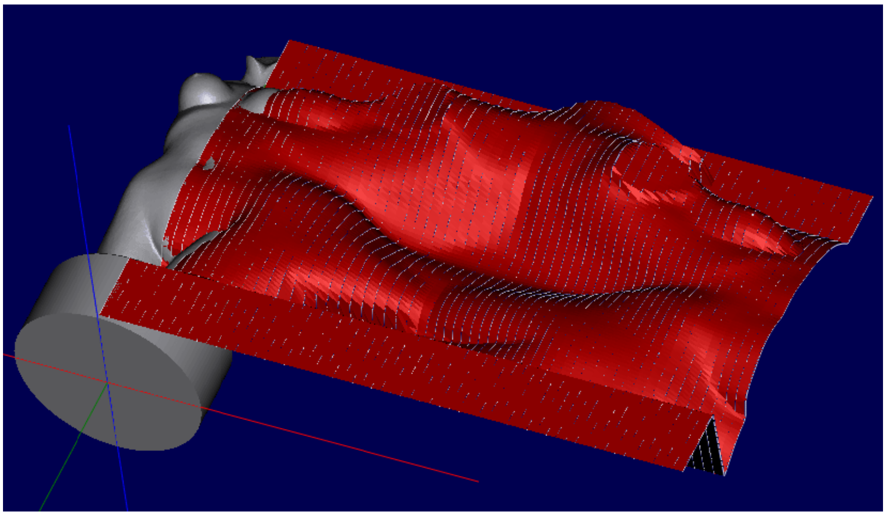
Then I used "Develop surface model" in the plugin, to generate the grayscale image below, representing a relief with bulging areas at the "stones".



This image was then used to produce the 3D surface using: ***Draw/Surface/From Bitmap***.

Here, we start from an already "flat" (unrolled) 3D shape, but if we have a "normal" 3D object, it will be necessary to unroll it in order to obtain the surface that will be used with the 3D machining operations

This unrolling of the source object can be done using my plugin Unroll 3D model.



Here, the Venus statuette (in grey) has been unrolled to obtain the unrolled shape in red.

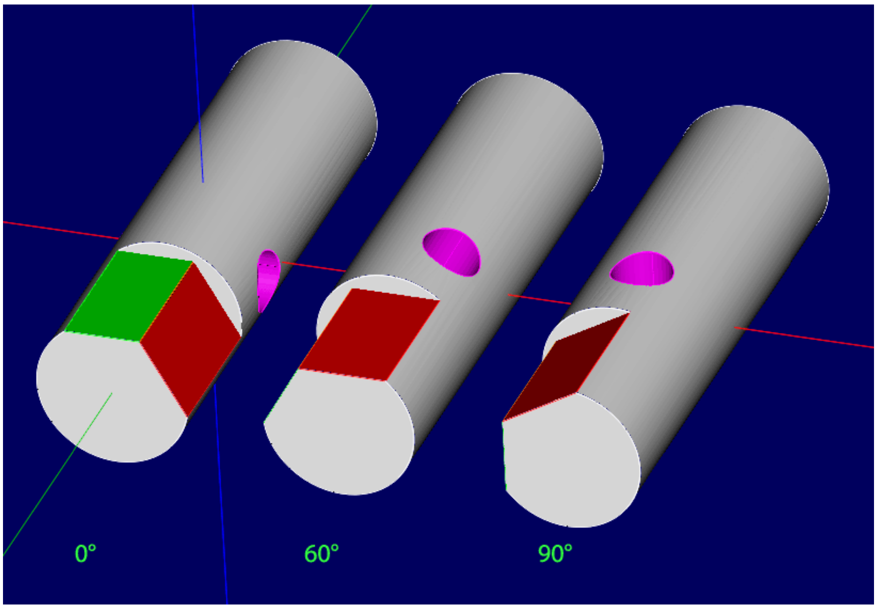
One or more 3D machining operations can be applied to the unrolled shape, then the GCode obtained will then be wrapped for machining with the 4th rotary axis.

See HERE for the plugin allowing to unroll a 3D object, as well as its documentation.

3. Practical examples

a) Simple positional machining

We are going to create the machining operations to create the three machining on the part below.



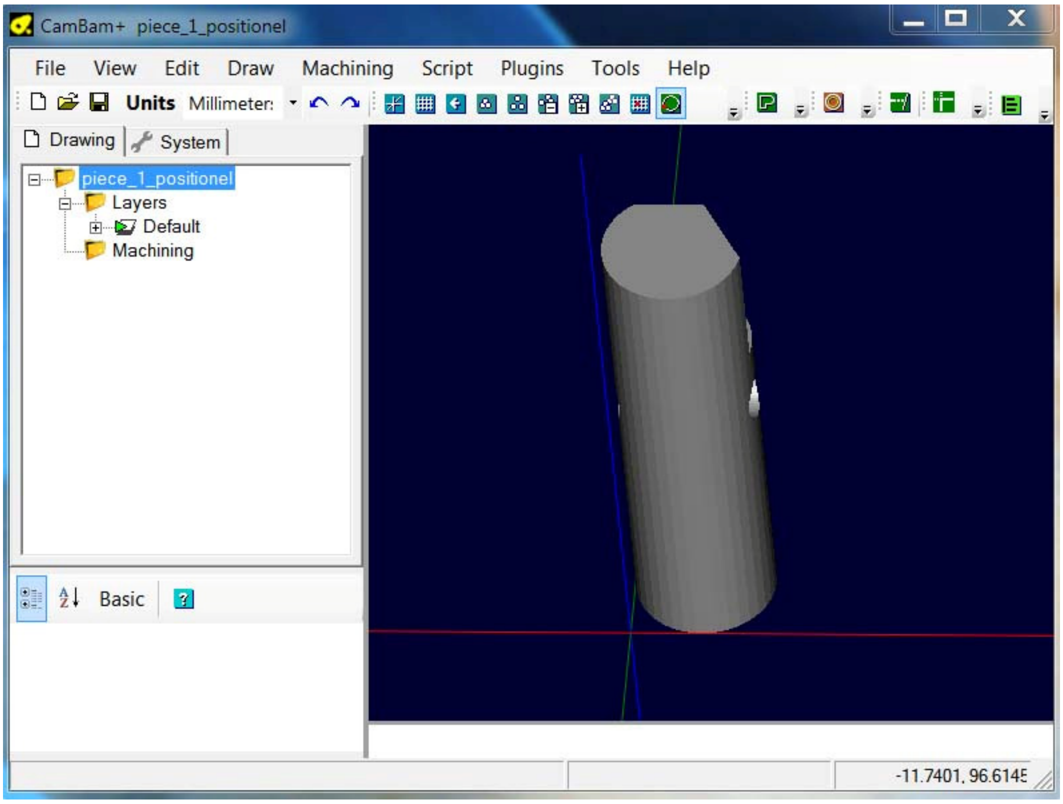
Opening and positioning the 3D object

This tutorial assumes that the Numerical move/rotate plugin is installed.
It is based on an A axis aligned with the Y of the milling machine.
The Z zero must be done at the rotation axis of the 4th axis.

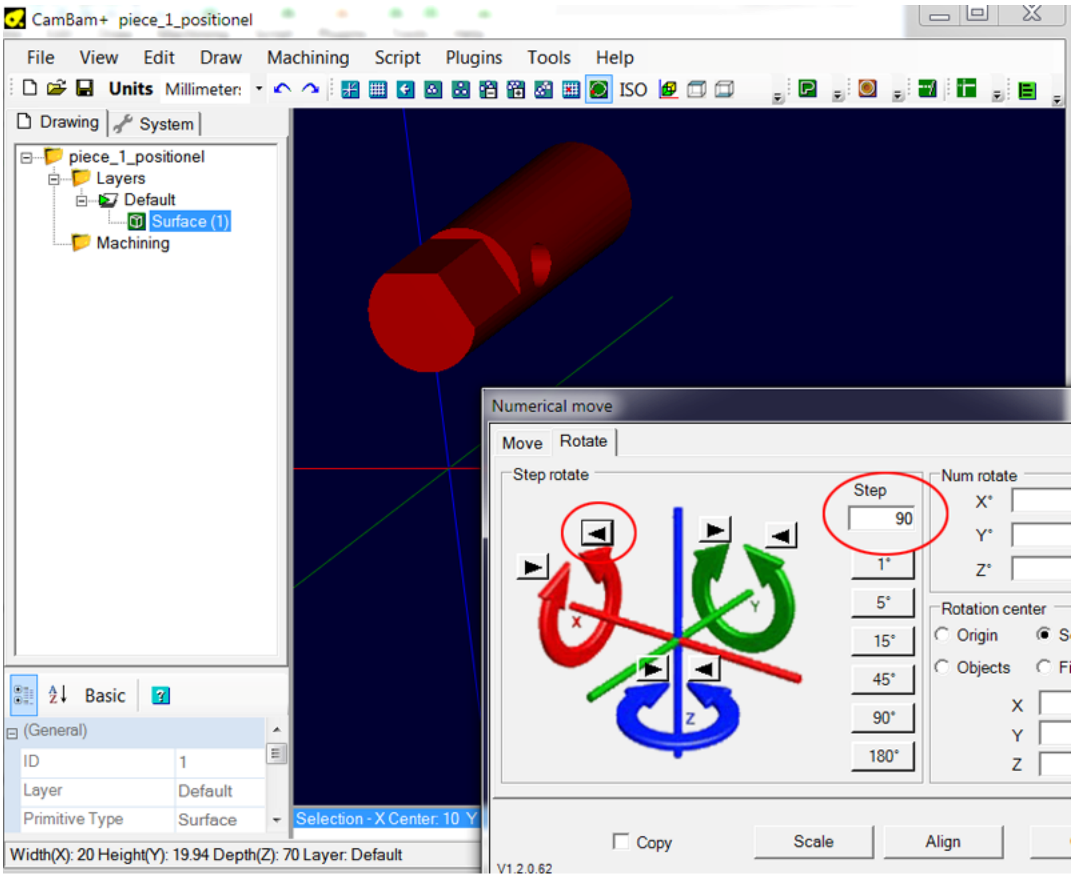
The 4th axis installed on my machine, aligned with the Y axis

Open the **Piece_1_positionel.stl** file contained in the archive.

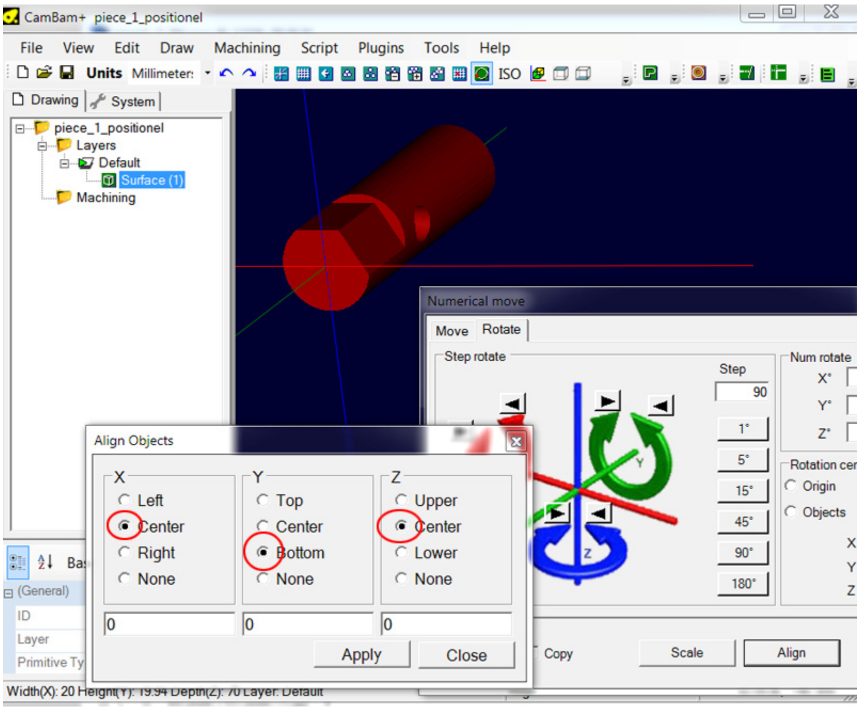
The surface object obtained will have to be positioned correctly because it is badly oriented.



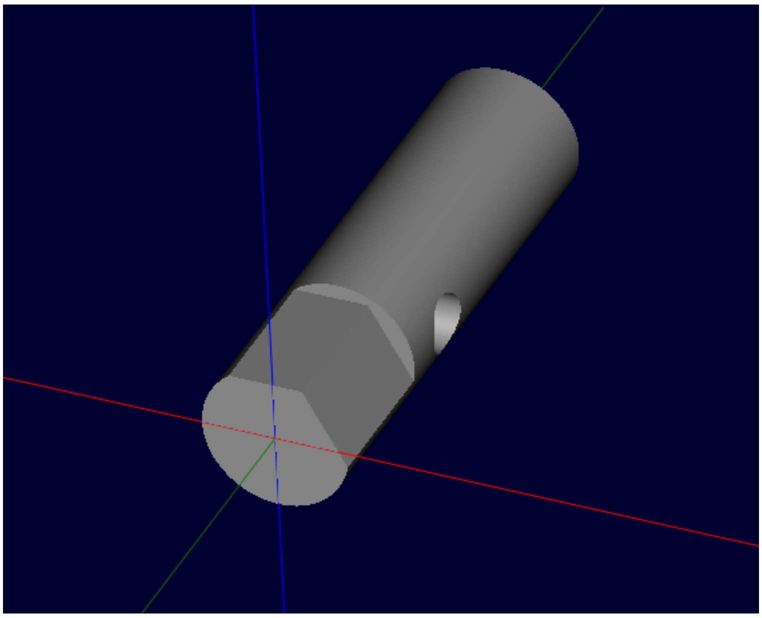
Select the surface object, open the [Numerical move/rotate]() plugin (**`ctrl+shift+M`**) and rotate the object 90° around X counter-clockwise to get the following result.



Then click the ***Align*** button to open the alignment tool and adjust the settings as pictured, then click ***Apply*** to position the 3D object.



It should now be centered in Z and X, and the front face should be aligned with the 0 in Y as shown below.

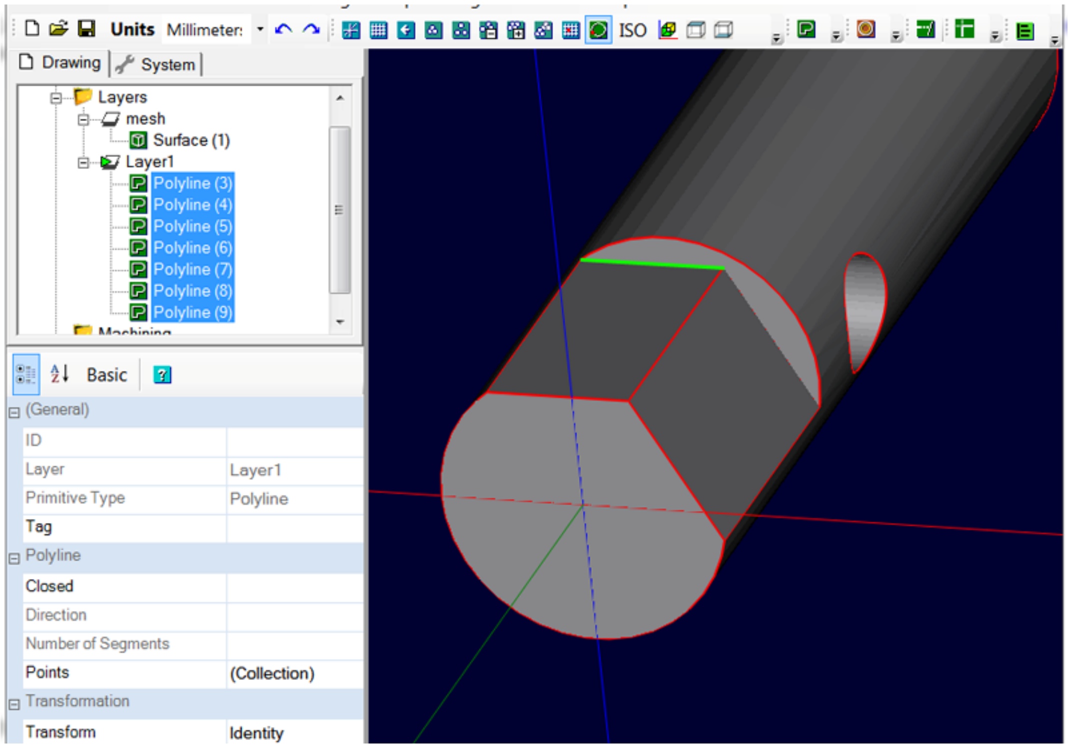


Obtaining 2D lines for machining

We are now going to create the 2D line that will be needed for machining the flat area ; This machining will be done with a profile operation + Cut Width enabled.

If you are using *CamBam* version 1.0, just use object snapping to be able to draw a line directly on the 3D object (green line in the image below). If you are using an older version, you will need to go through the intermediate step below.

Create a new layer (here Layer1), select the surface object, then use the ***Edit/surface/edge detect*** menu. You should get this result.

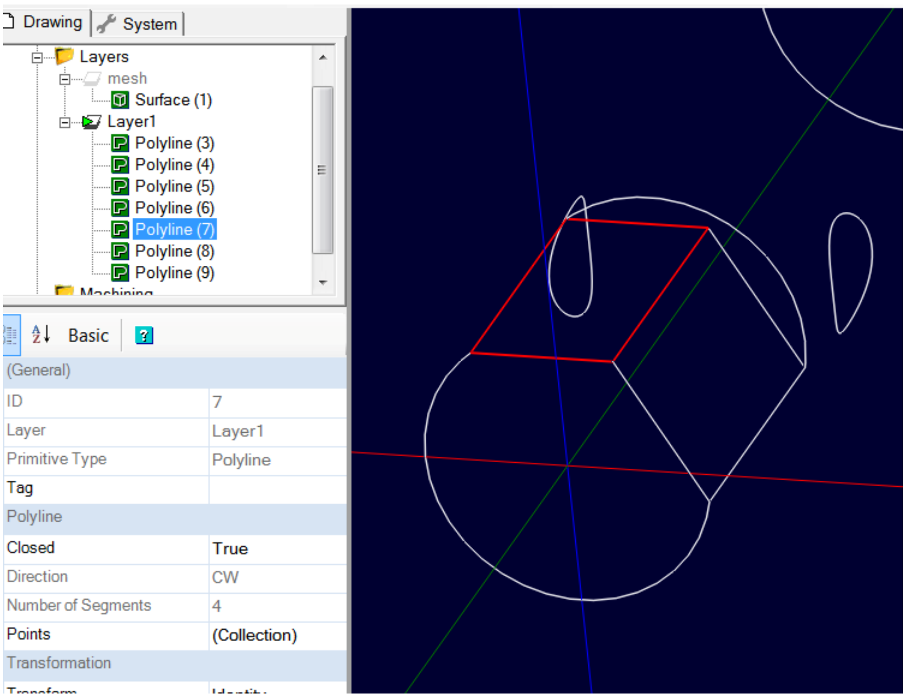


I have highlighted in green the line that interests us.

From there, the goal is to recover the green line alone. If drawn on *CamBam* V1.0 with object snapping, it's already done. Here, I will simply break down the polylines that make up the contours and retrieve only the line that interests me, but I could also draw the line directly since now, thanks to edge detection, I can snap to the ends of line.

I chose to show you the most complex way, because if here, we just have a straight line, we could have a more complex outline, and in this case, it is easier to recover an existing line than to create a new one.

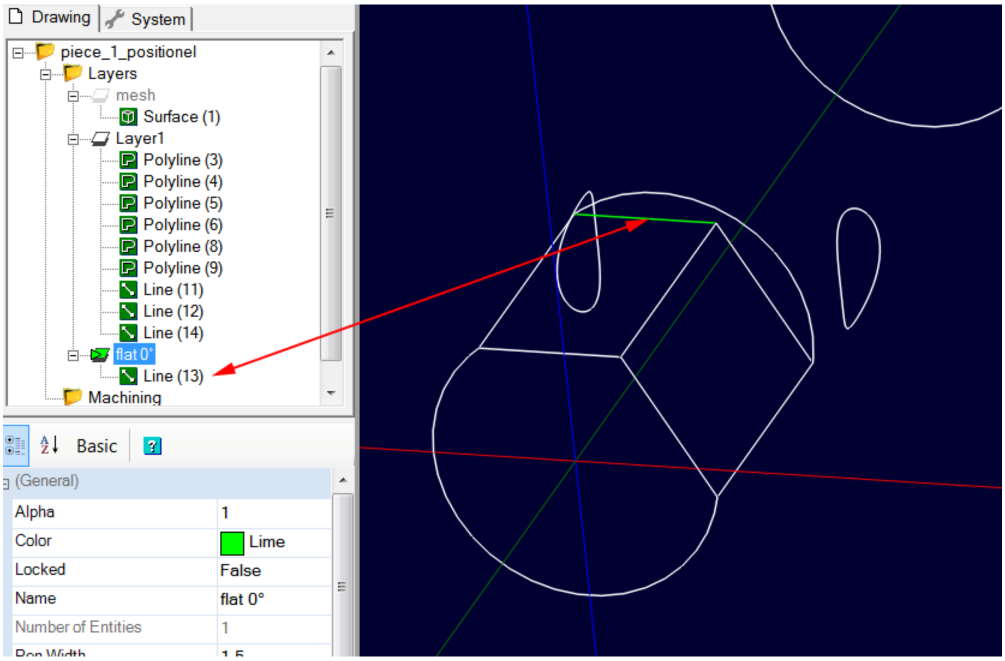
To clearly see what the lines in question look like, you can hide the layer on which the 3D object is located to see only the lines; as we can see, polyline 7 contains the line we are interested in.



I will now explode (***Edit/Explode***) polyline 7 and keep only the line that interests me, which I would put on another layer called "flat 0°"; the other lines will no longer be used and can be deleted.

To put line 13 obtained after exploding on the new layer, just drag/drop it in the tree structure.

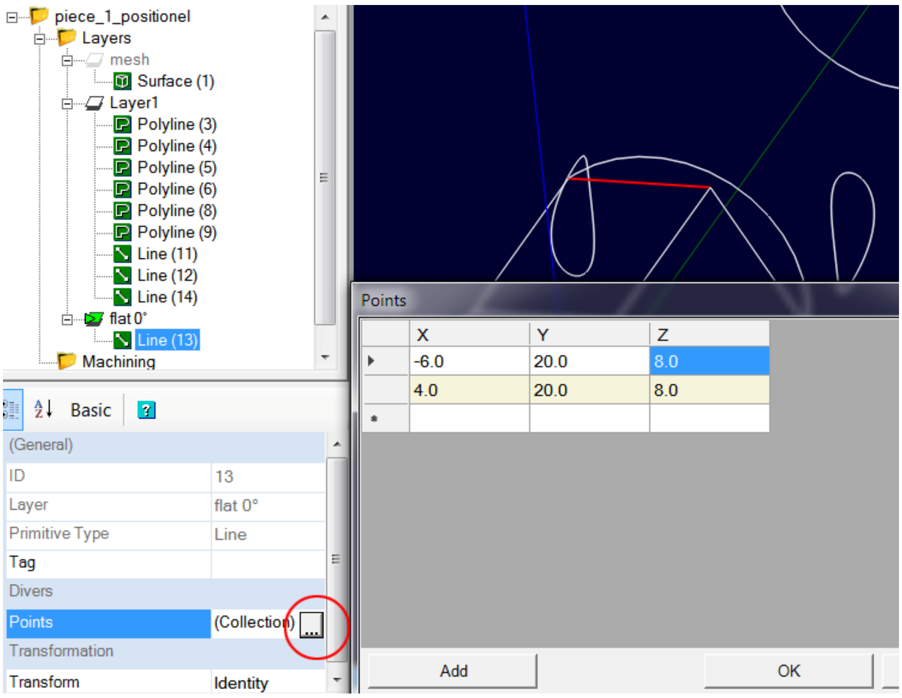
Note: The Entity numbers (ID) shown in this tutorial can, of course, be different in your own project or in the example file.



This line will not only be used to define the machining position, but it will also allow us to know the Z position of the flat area to be machined.

To know its Z position, select line 13, then click on the |...| of the ***Points*** property of the line. As we can see, the surface is at Z = 8

The part is 20mm in diameter, the upper surface of the cylinder is therefore at Z = 10



Note: When using an object in .STEP format instead of .STL (*CamBam* V1.0 only); the contour lines will already be available when opening the STEP file and it will not be necessary to perform Edge detect.

Setting up of the first machining operation.

We are now going to create the ***Profile*** operation which will allow us to surface the first flat area.

Select the line we created (Line 13 in the case of the example) then assign it an ***Profile*** operation.

Set the following properties as follows:

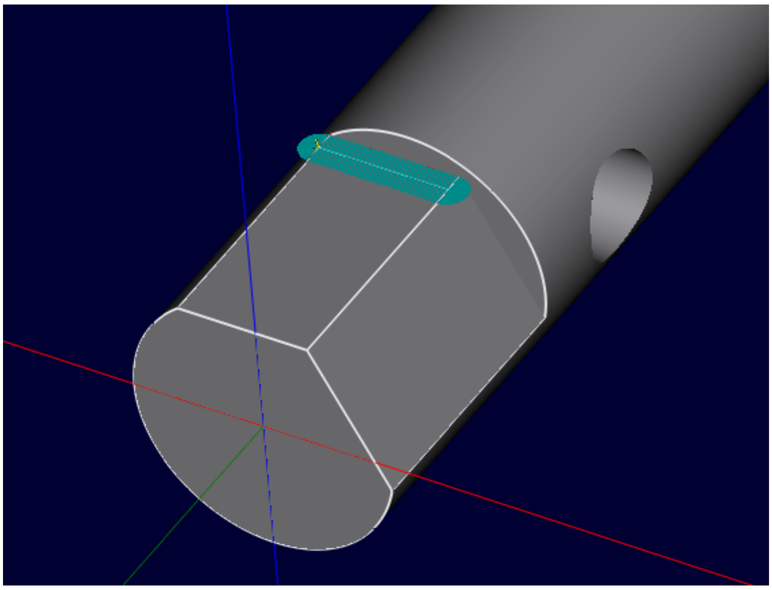
- ***Tool Diameter*** = 3
- ***Tool Profile*** = End Mill
- ***Stock Surface*** = 10 (the radius)
- ***Target Depth*** = 8
- ***Clearance Plane*** = 15
- ***Depth Increment*** = 2
- ***Milling Direction*** = mixed

Adjust the other properties such as ***Cut Feedrate***, ***Spindle Speed***, etc to suit the material you are machining and the performance of your machine.

Activate the display of the cutting width (***View/Show cut width***)

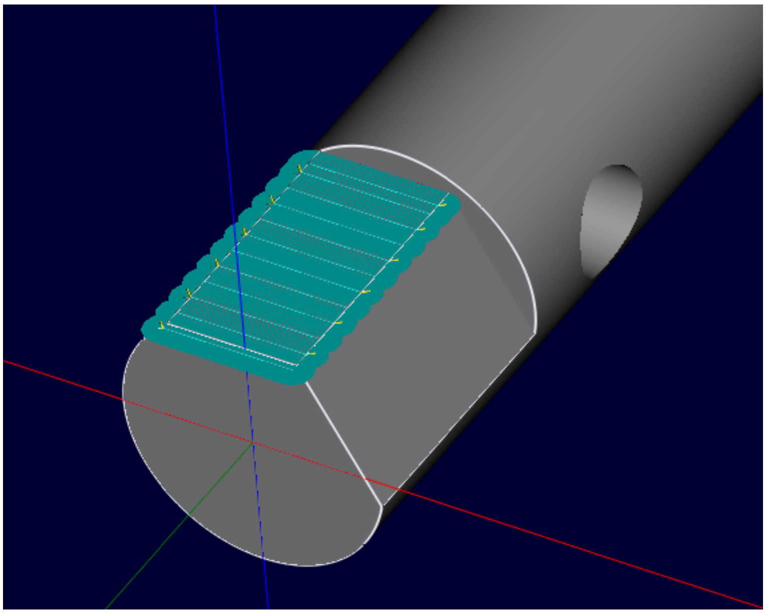
Generate the toolpaths for this operation.

You should get the following result



If the machining is not done on the right side of the line, invert the value of the ***Inside/Outside*** property and regenerate the toolpaths in order to have the machining on the right side.

It only remains to give a value for ***Cut Width*** ; in this case, the flat area is 20mm long, we are going to give a slightly larger value so that the cutter does not cut the entire pass width at once; enter 22 for ***Cut Width*** then regenerate the toolpaths. You should get the following result:



We are done with this operation for now, save your work before to continue.

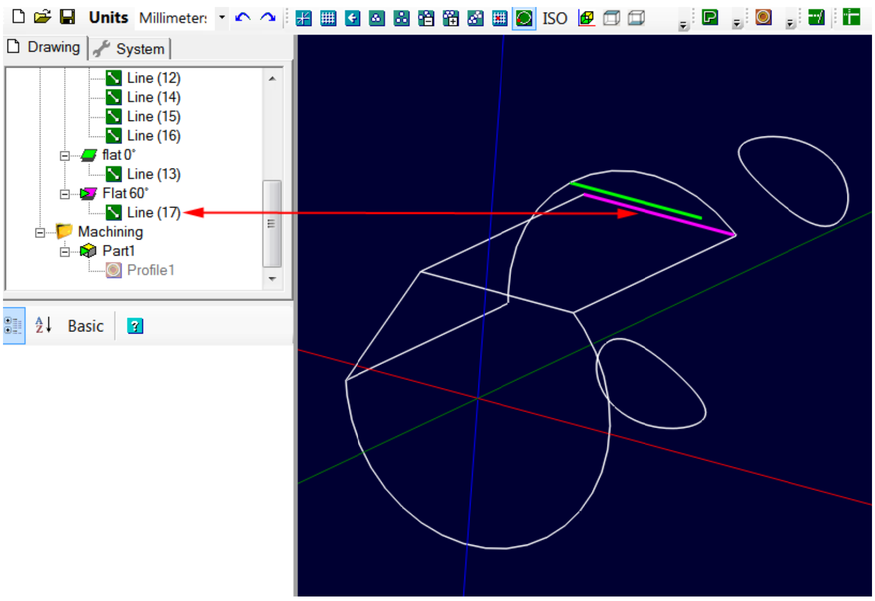
Setting up the second machining operation

We will repeat the same process to create the second flat area.

We are going to rotate the contour lines of layer 1 to put them in the new position.

- Hide the layer that contain the 3D surface object, it is not nedeed that the 3D object is rotated too because now we will only use the lines that we have obtained with ***Edge Detect***.
- right-click layer 1 and choose ***Select All on Layer*** from the context menu to select all contour lines.
- Using the Numerical Move/Rotate plugin, rotate the contour lines selection 60° left around Y (CCW) to bring the 2nd flat area to horizontal.
- To have a clearer display, deactivate the previous machining operation (right click on the machining operation > ***Enable/Disable MOP***)

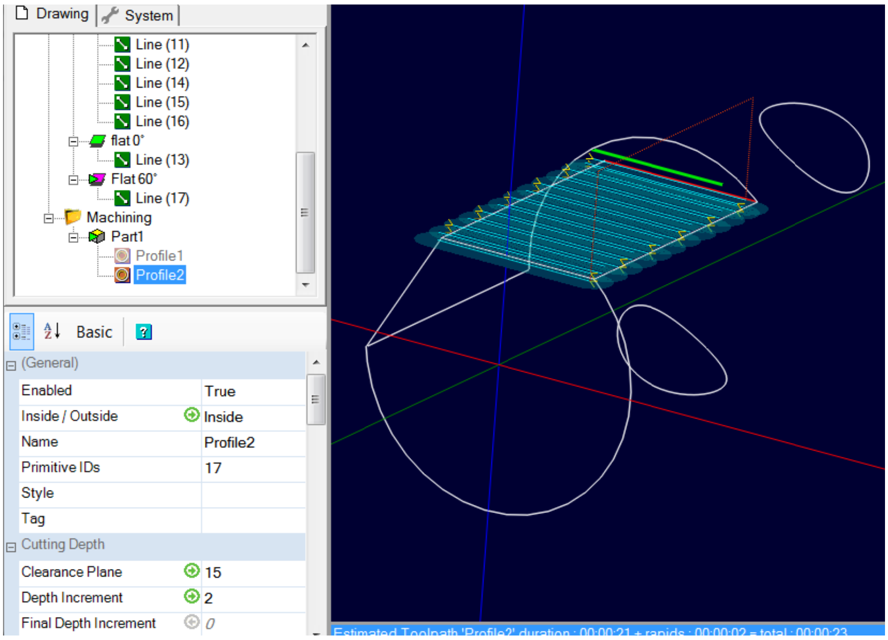
From there, do the same manipulation as before to retrieve the line that interests us and place it in a layer that we will call "Flat 60°"; we get the line in purple on the image. (line 17)



Copy/Paste the Profile1 operation, then assign line 17 to this new operation (by dragging and dropping line 17 onto the Profile 2 operation, or using the context menu of the Profile 2 machining operation: ***Select drawing objects***)

Activate the Profile 2 operation, and change ***Target Depth*** to match that of the flat area ; here 7.464 and generate the toolpaths. If the cut is not on the right side, invert the value of ***Inside/Outside***.

You should get the following result

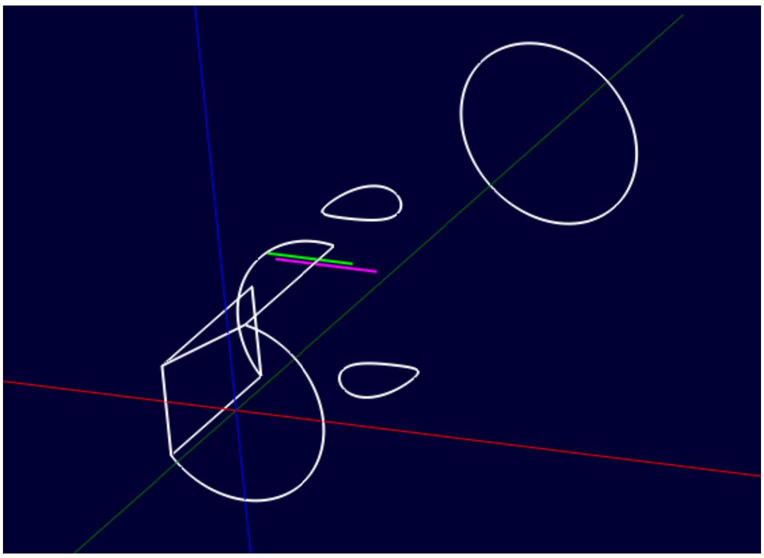


Setting up the third machining operation

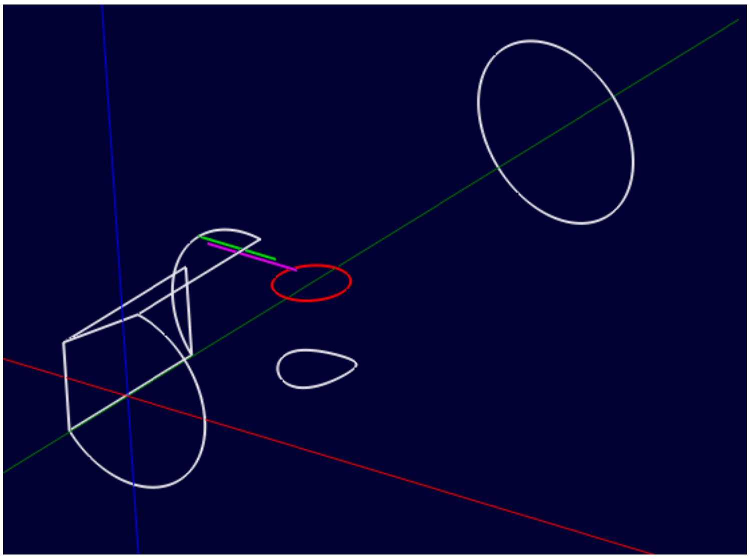
We will now machine the hole using pocket machining.

- Disable Profile 2 to get a clearer display.
- Select all the content of layer 1 (the outlines)
- Using the Numerical Move/Rotate plugin, rotate the contour selection 30° left around Y (CCW) to bring the hole into a vertical position.

You should get the following result.

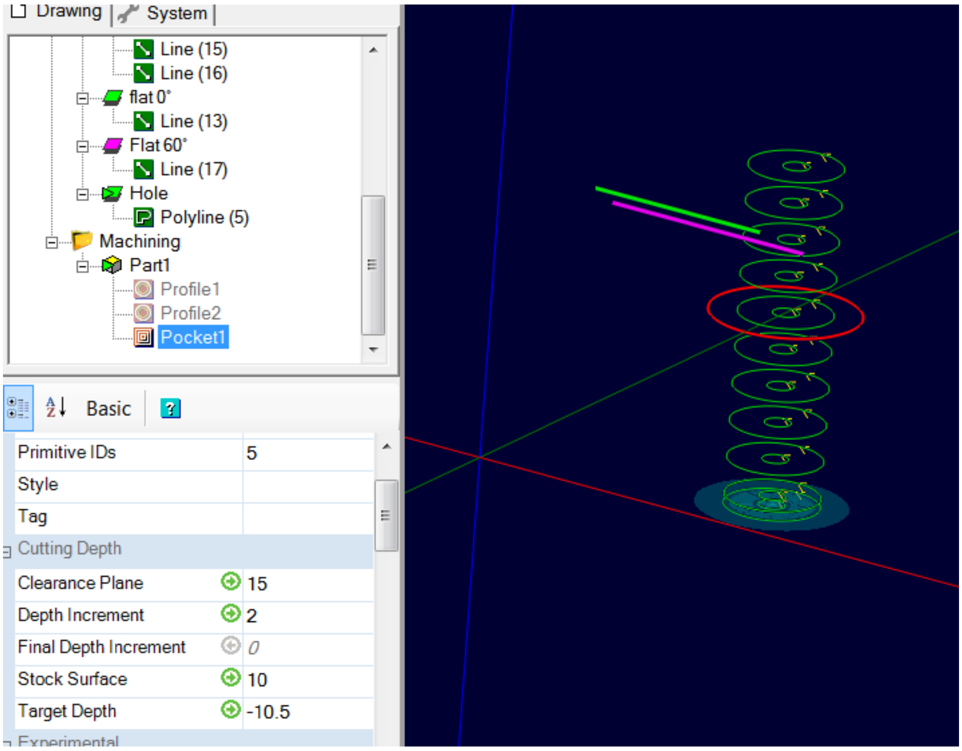


Select the top circle, then use the ***Edit/Polyline/Arc Fit*** menu with a value of 0.02 ; This will create a "flat" circle made of arcs from the selected shape. Note that the shape is positioned at Z=0 ; it does not matter.



As before, create a new layer, which we will call "Hole" and move the circular polyline we just created there. We can now delete or hide layer 1; the outlines will no longer be useful to us.

- Select the polyline we just created, then assign it a ***Pocket*** operation.
- Right click on profile 1 (or Profile 2) > ***Copy***.
- Right-click on the pocket operation we just created > ***Paste Format*** to copy the settings from profile 1 operation into the pocket operation.
- Enable pocket operation and change ***Target Depth*** to -10.5
- Generate toolpaths



We are done with the machining operations.

Implementation of the 4th axis rotation instructions

We are now going to insert the codes which will make it possible to rotate the 4th axis to position it before each machining.

To do this, we will directly write the GCode instructions by hand in the machining operations ***Custom MOP Header*** property.

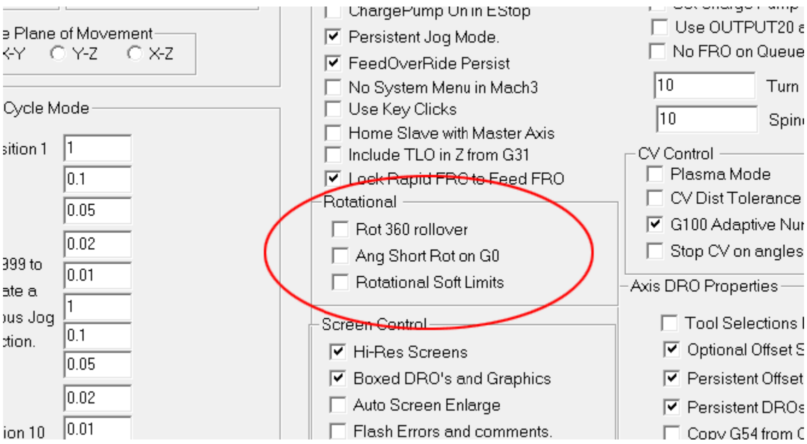
Activate the Profile 1 and Profile 2 operations which are currently inactive if you have followed this tutorial to the letter.

The GCode of the examples was produced by the *Mach3* post-processor provided with *CamBam*, if you use another post processor, the GCode may have some variations.

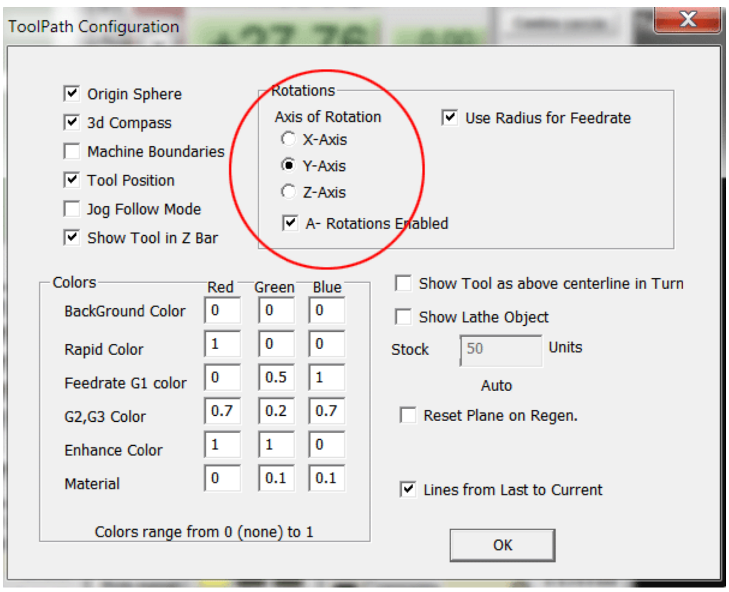
Note that the direction of rotation under *Mach3* is positive clockwise and negative counterclockwise.In the case of this example, as we are rotating the part counter-clockwise, it will therefore be necessary to provide negative rotation values so that the display is in the right direction on Mach3.

My settings in Mach3 are:

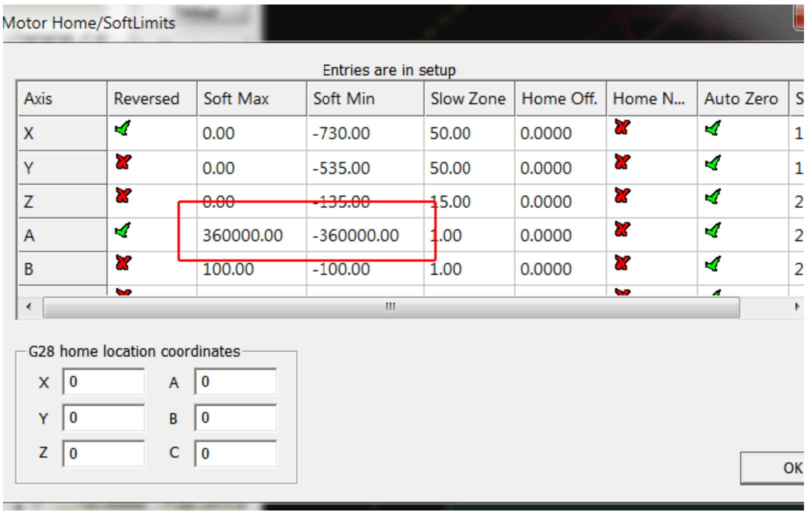
Menu **Config/General Config**



Menu **Config/ToolPath** for an A axis aligned with the Y axis



Menu **Config/Homing-Limits** (allows 1000 turns in positive and negative)



If we generate the GCode directly, without having put the rotation instructions, we will obtain the following result:

```
( Made using CamBam - http://www.cambam.co.uk )

( positionel_simple 8/29/2019 9:37:52 PM )

( T5 : 3.0 )
G21 G90 G91.1 G64 G40
G0 Z15.0
( T5 : 3.0 )
T5 M6
( Contour1 )
G17
M3 S12500
G0 X-6.0 Y-0.5
G0 Z11.0
.....
.....
G1 F600.0 Y18.5
G1 F800.0 X-6.0
( Contour2 )
S12500
G0 Z15.0
G0 X6.6549 Y-0.5
G0 Z11.0
.....
.....
G1 F600.0 Y18.5
G1 F800.0 X6.6549
( Poche1 )
S12500
G0 Z15.0
G0 X0.29 Y29.3642
G0 Z11.0
G1 F600.0 Z8.0
G3 F800.0 X0.6268 Y30.3108 I-0.3276 J0.6498
.....
.....
G3 X1.1003 Y27.7569 I-1.0237 J2.3111
G0 Z15.0
M5
M30
```

To insert the rotation instructions, we'll use the ***Custom MOP Header*** property of each operation that requires positioning the A axis at a given angle, and insert the following GCode into it:

G0 Z15
G0 A-xx

The **G0 Z15** raises the Z in rapid to position 15 so the tool is in safe position
The **G0 A-xx** rotates the A axis to the angular position -xx°

It is possible to replace the G0 Z15 by a macro which will directly return a G0 to the clearance plane and which will use the value of the Clearance Plane property of the machining operation.

For example, to raise the Z axis to the current clearance plane and rotate the A axis to the -60° absolute position I would write

{$clearance}
G0 A-60

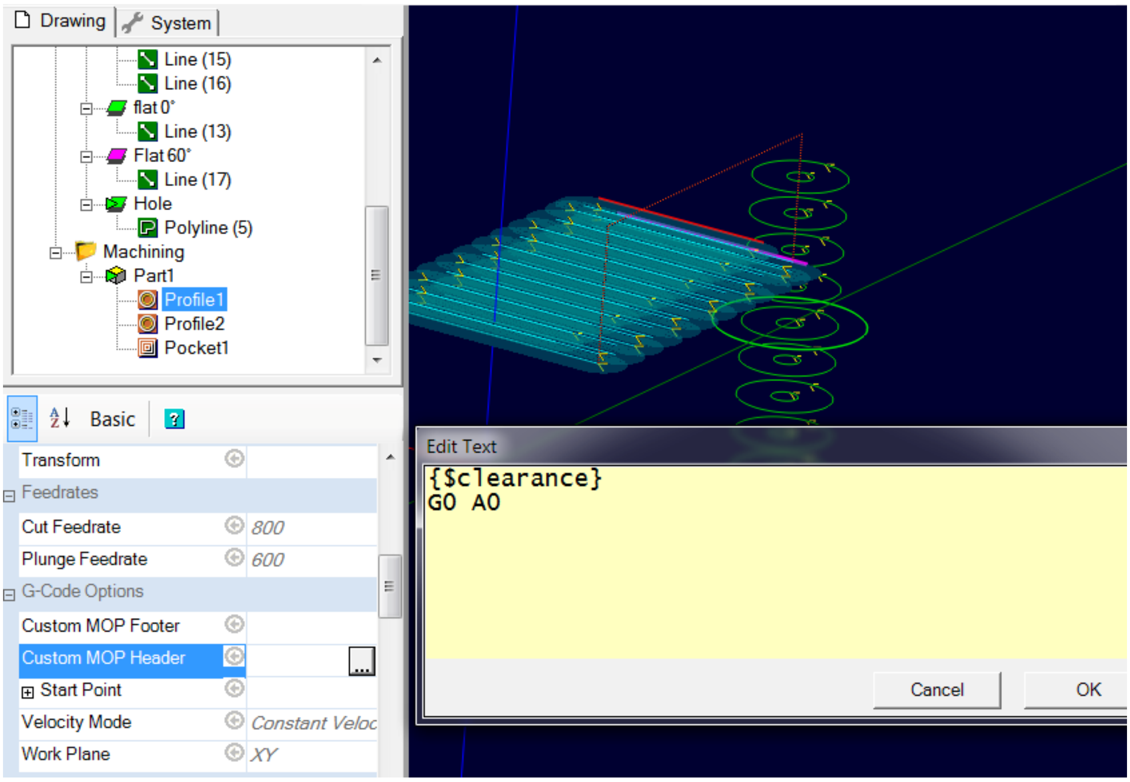
The **{$clearance}** macro will return G0 Z15 in the case of this example.

Note that this macro will only produce a G0 Zxx if necessary, ie. if the tool is not already at the clearance plane or above.

It is vital to program this Z move to the clearance plane, either by a G0 command or via the macro, otherwise the A axis risks rotating before the tool is outside of material.

Select the profile 1 machining operation, then click the |...| button of the ***Custom MOP Header*** property ; in the edit window, enter the following text then click OK to validate.

{$clearance}
G0 A0



Select the Profile 2 machining operation and enter the following text in the ***Custom MOP Header*** property

{$clearance}
G0 A-60

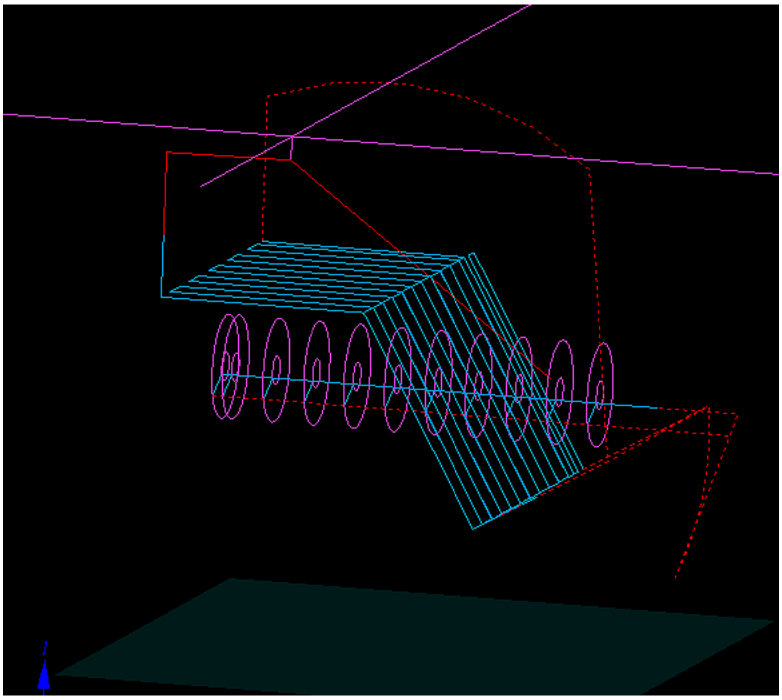
And finally do the same for the Pocket 1 operation and enter the text:

{$clearance}
G0 A-90

Save your project then generate the GCode for the whole project ; you should get the following result where you can see that the rotation instructions have been added.

```
( Made using CamBam - http://www.cambam.co.uk )
( positionel_simple 8/29/2019 10:12:08 PM )
( T5 : 3.0 )
G21 G90 G91.1 G64 G40
G0 Z15.0
( T5 : 3.0 )
T5 M6
( Contour1 )
G17
G0 A0
M3 S12500
G0 X-6.0 Y-0.5
G0 Z11.0
G1 F600.0 Z9.0
.....
.....
G1 F600.0 Y18.5
G1 F800.0 X-6.0
( Contour2 )
G0 Z15.0
G0 A-60
S12500
G0 X6.6549 Y-0.5
G0 Z11.0
G1 F600.0 Z9.0
G1 F800.0 X-4.9282
.....
.....
G1 F800.0 X6.6549
( Poche1 )
G0 Z15.0
G0 A-90
S12500
G0 X0.29 Y29.3642
G0 Z11.0
G1 F600.0 Z8.0
.....
.....
G3 X1.1003 Y27.7569 I-1.0237 J2.3111
G0 Z15.0
M5
M30
```

On the *CamBam* display, all the operations remain superimposed because the software does not know how to display angular positions, but on a simulation under *Mach3*, we can clearly see that the different operations are performed at the correct angular position.

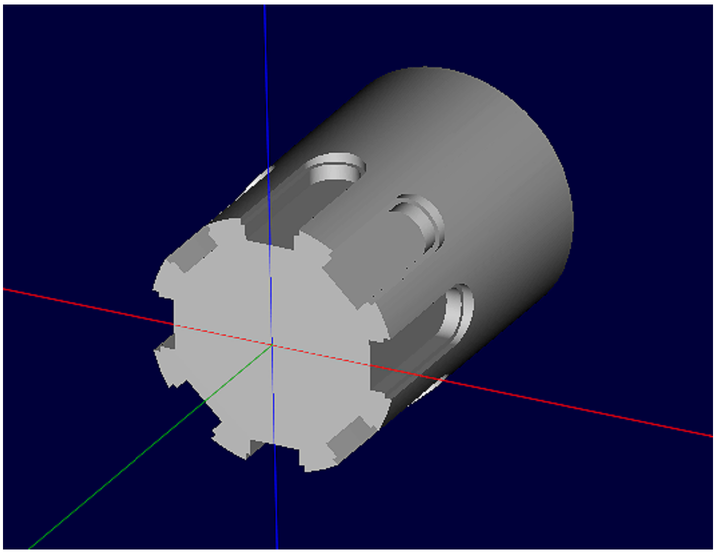


If it is necessary to perform several operations in a row at the same angular position, then of course only the first operation at this angular position will have to contain a rotation order.

The *CamBam* example file is available in the archive under the name: ***positionel_simple.cb***

b) Positional machining with repetitions

Open the ***canelures.stl*** 3D object, then, if necessary, rotate it and position it so as to align it with the Y axis and to put its front face at 0 in Y using alignment functions such as we have already done this for the previous piece.

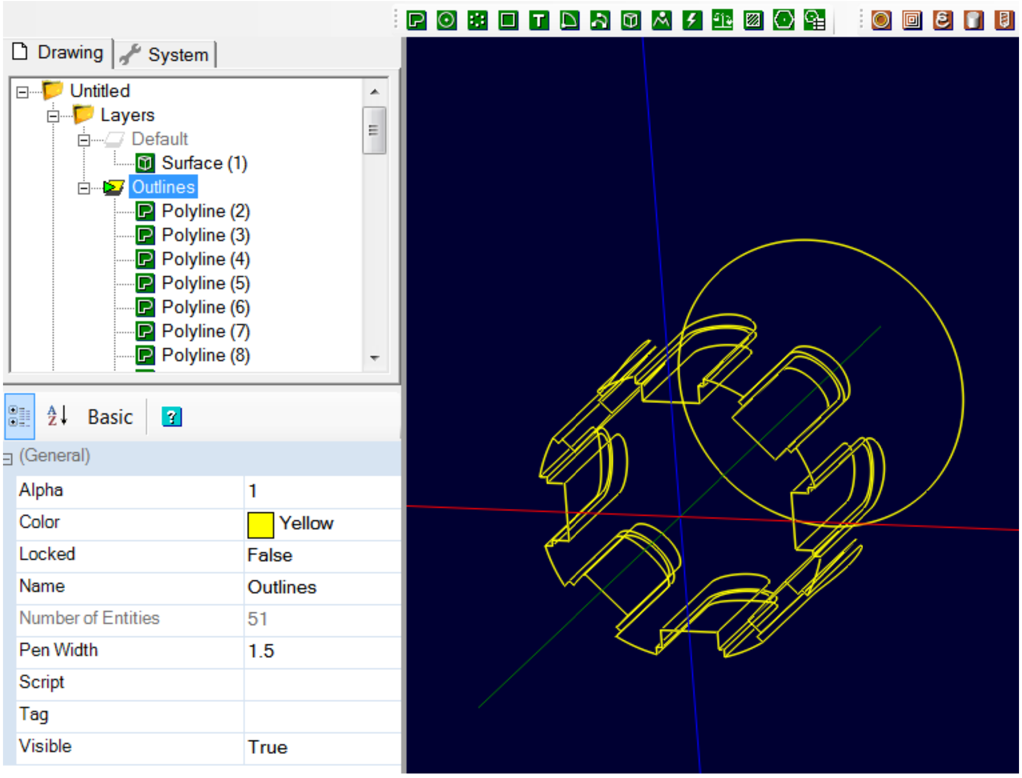


Getting 2D lines for machining

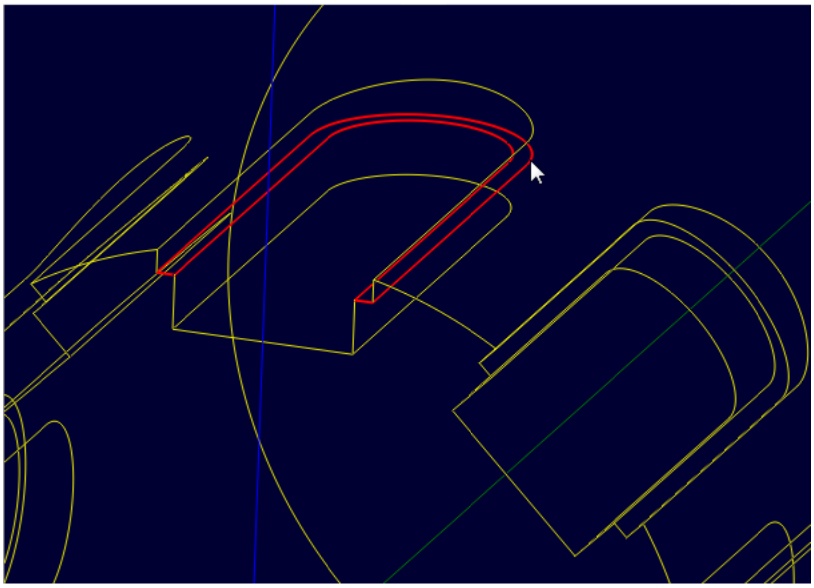
We will now extract the outlines to a new layer.

Create a new layer called "Outlines" (for example), then select the 3D object and use ***Edit/Surface/Edge detect***.

Here, the layer containing the 3D object is hidden in order to better see the resulting lines.



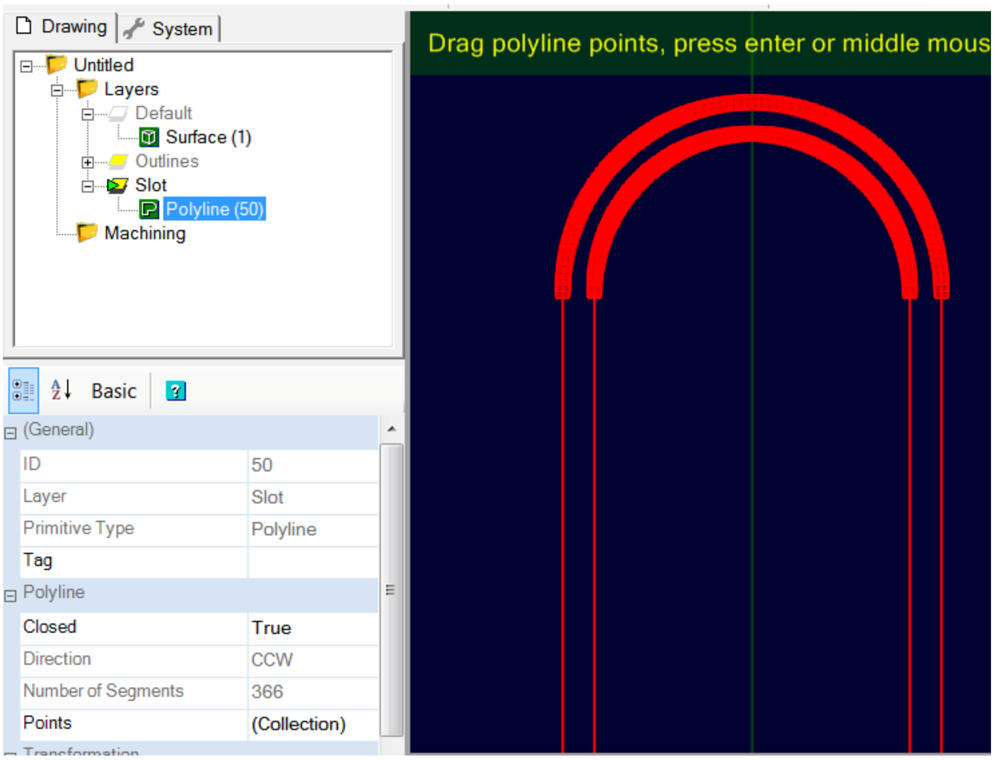
On the top slot, we'll select the polylines that will give us the outline of the wide slot and the outline of the narrow slot ; the height position of the polylines does not matter ; in this specific case, the easiest way is to select the polyline shown in the following image because it contains the 2 contours that interest us in the same polyline.



Create a new layer called "Slot" (for example), select the polyline and move it to this new layer.

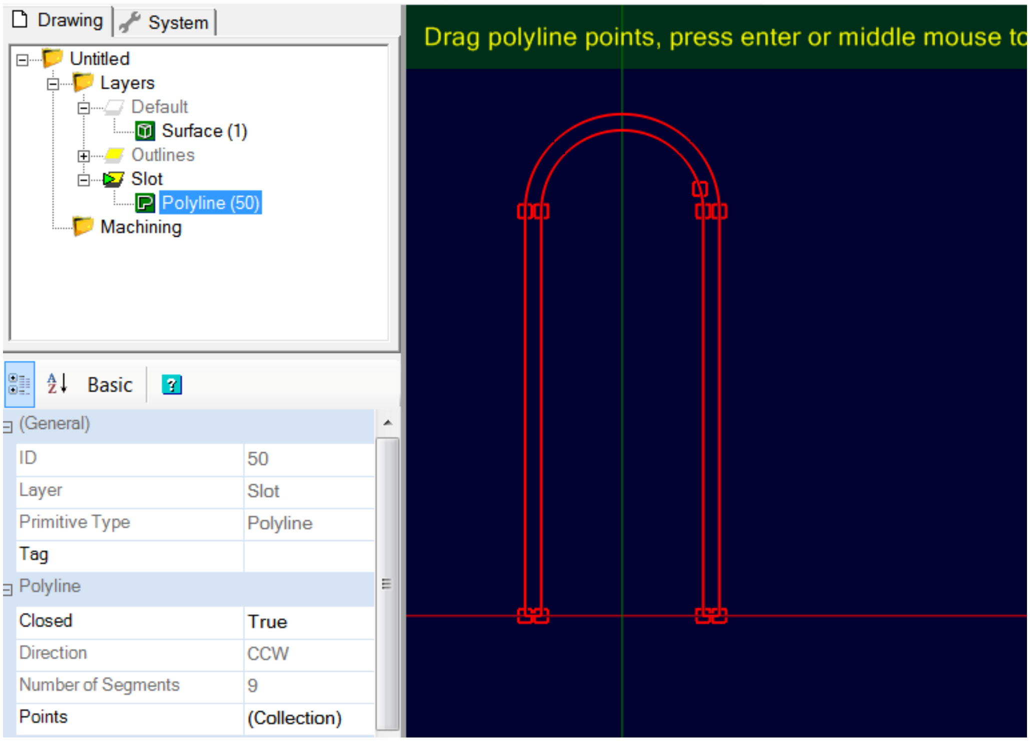
Hide or delete the "Outlines" layer which will no longer be useful to us.

Switch to plan view (XY plane) with an ALT + double left click in the drawing window (or via the "views" toolbar if it is installed), then double click on the polyline to switch to edit mode. As you can see in the image, because the path was extracted from a mesh object, you don't get an arc on the rounded parts, but a succession of small segments.



With the polyline selected, use ***Edit/Polyline/Arc Fit*** with a tolerance of 0.02 (mm)

We then obtain a result as below.



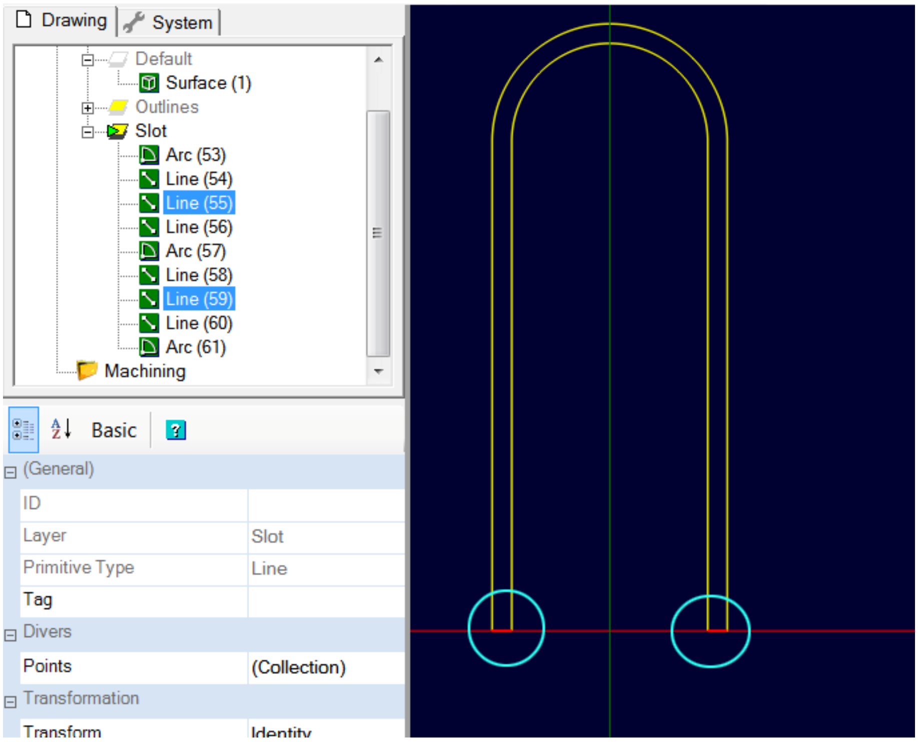
Also note that the polyline has been aligned with the 0 in Z.

Of course, it would have been possible to draw these polylines directly in *CamBam*, it would even have been easier.

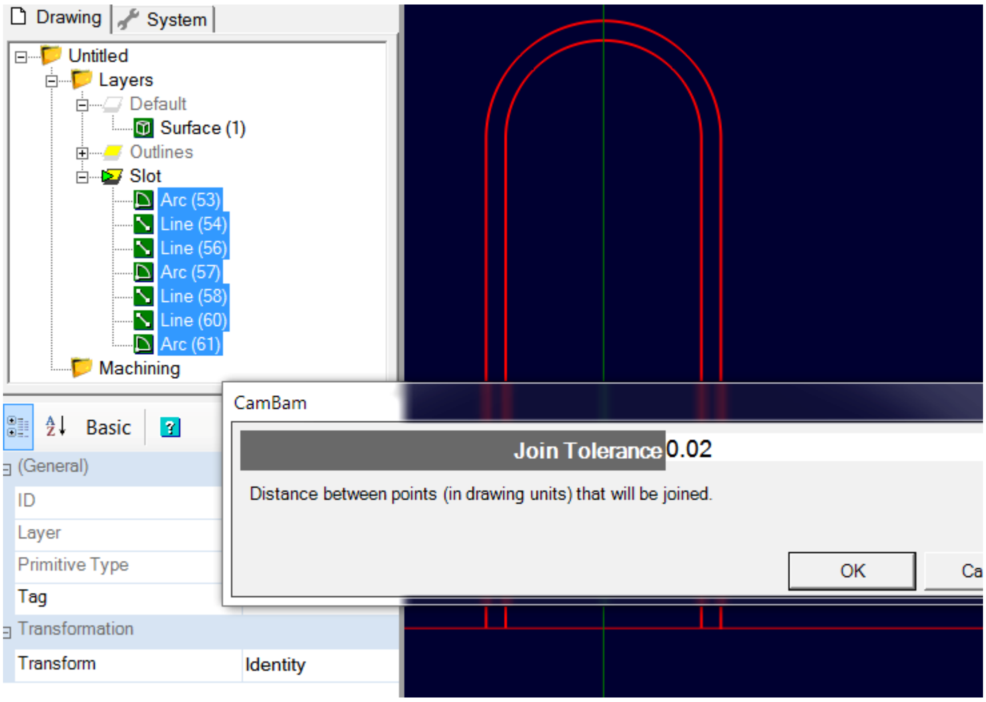
To do the machining, we need to separate the 2 polylines, and extend them a little downwards, so that the tool can go down out of the material. We also need to delete the 2 small segments at the bottom, which join the 2 outlines.

Select the polyline, then use the ***Edit/Explode*** menu to obtain the polyline components.

Select the 2 bottom lines (circled in blue on the image) and delete them.



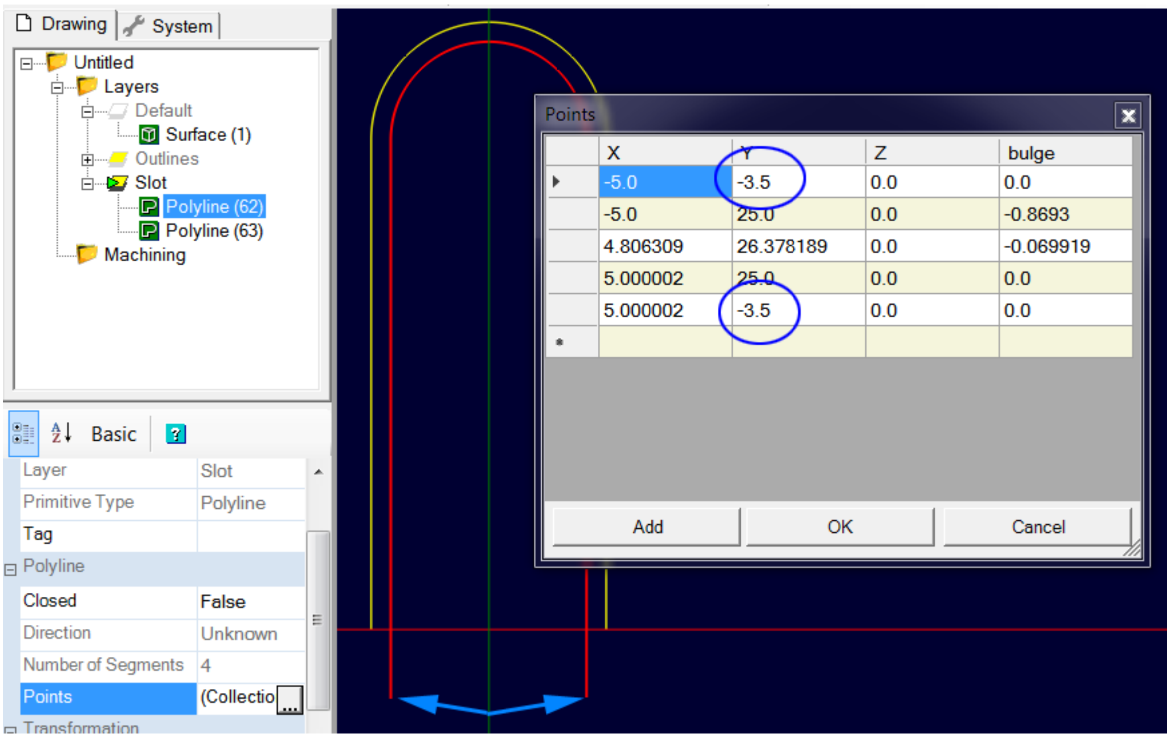
Then, select all the objects of the "Slot" layer then use the ***Edit/Join*** menu with a tolerance of 0.02 to reconstitute 2 complete polylines.



We are now going to extend the 2 polylines downwards ; in this case, the easiest way is to edit the collection of points of each polyline and to modify the value of the 2 points which are aligned on the X axis ; their Y coordinate is 0 ; we will change it to -3.5 in Y.

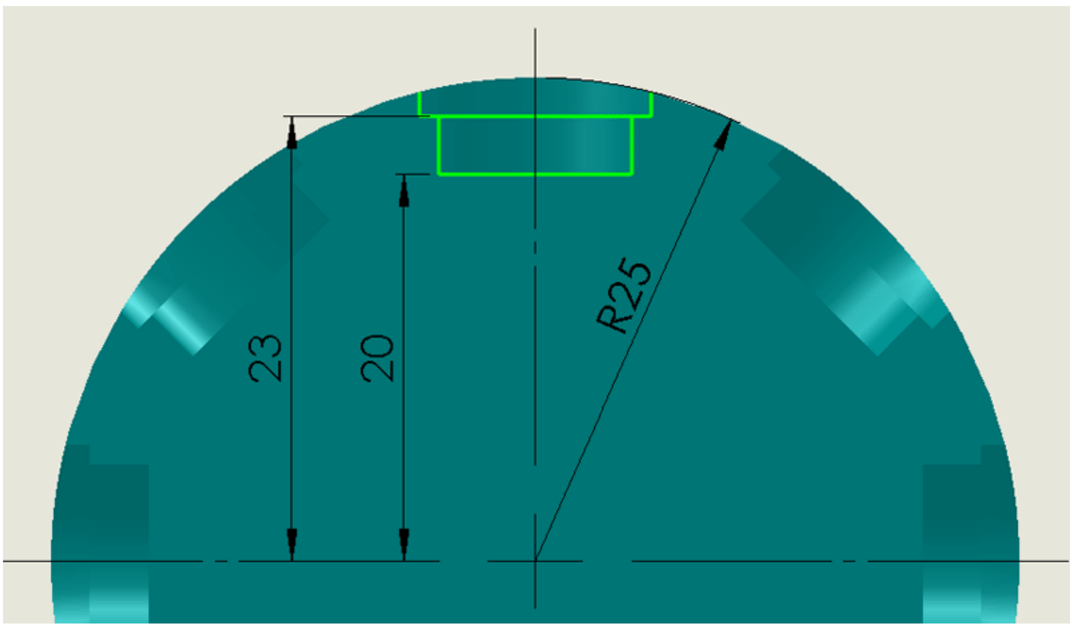
For each of the two polylines:

- select it
- click the |...| to the right of the ***Points*** property, and change the values that are 0 in Y to a value of -3.5
- confirm with OK



Once the two polylines have been extended, we will now be able to add two machining operations to make the two slot levels. As before, we will use a Profile operation with a ***Cut Width*** value <> 0 if necessary.

Here are the dimensions of the part:



Setting up of machining operations

We will start by machining the deep groove using the inner polyline, then using a second machining operation we will machine the rabbet at the top of the groove using the outer polyline.

Select the inner polyline, then assign it a ***Profile*** machining operation with the following settings:

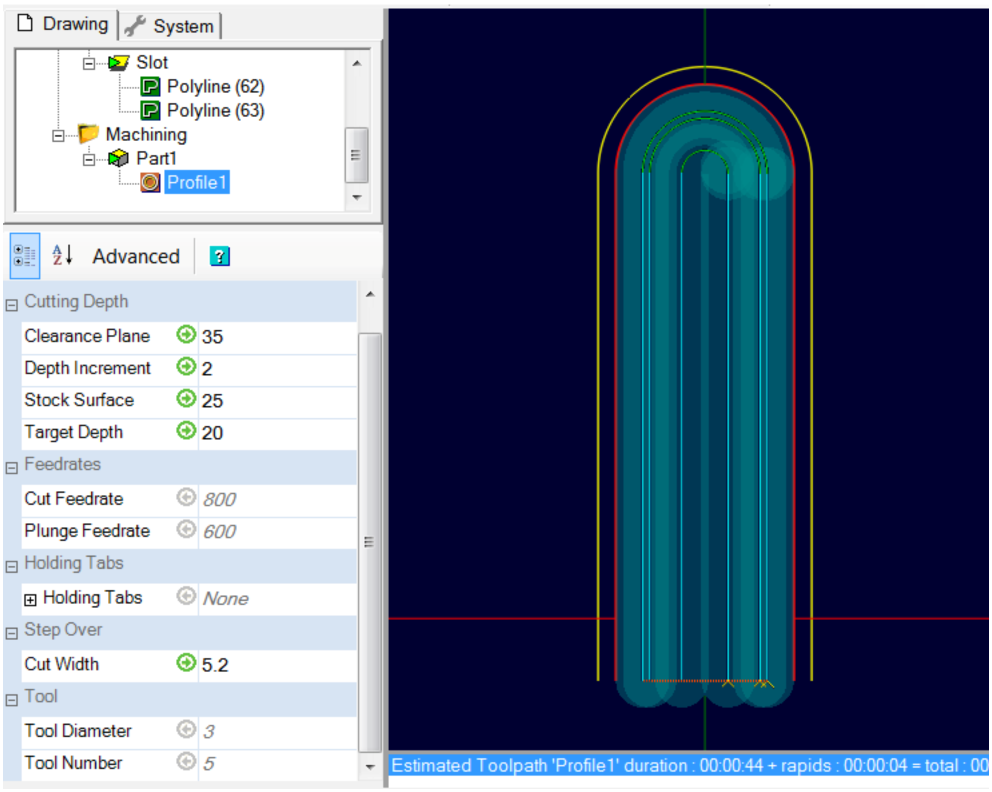
- ***Tool Diameter*** = 3
- ***Tool Profile*** = End Mill
- ***Stock Surface*** = 25 (the radius)
- ***Target Depth*** = 20
- ***Clearance Plane*** = 35
- ***Depth Increment*** = 2
- The groove being 10mm wide, set the ***Cut Width*** to 5.2 for example, to have an overlap of the passes in the center.

Adjust the other properties such as ***Cut Feedrate***, ***Spindle Speed***, etc. to suit the material you are machining.

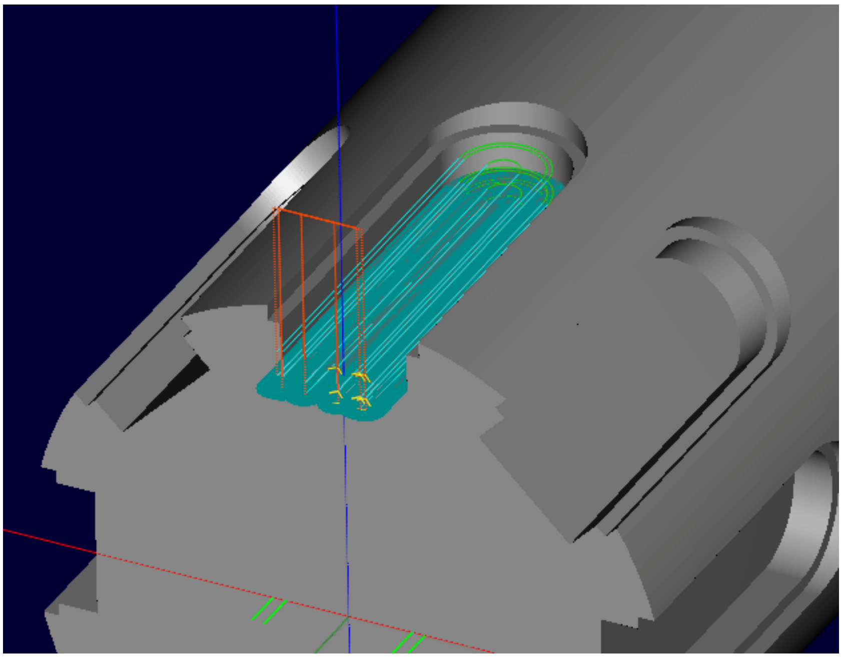
Activate the display of the cutting width (***Menu View/Show cut widths***)

Generate the toolpaths for this operation.

You should get the following result in top view ; if the machining is not done on the right side, modify the Inside/Outside property (on an open line, the machining side depends on the drawing direction of the polyline)



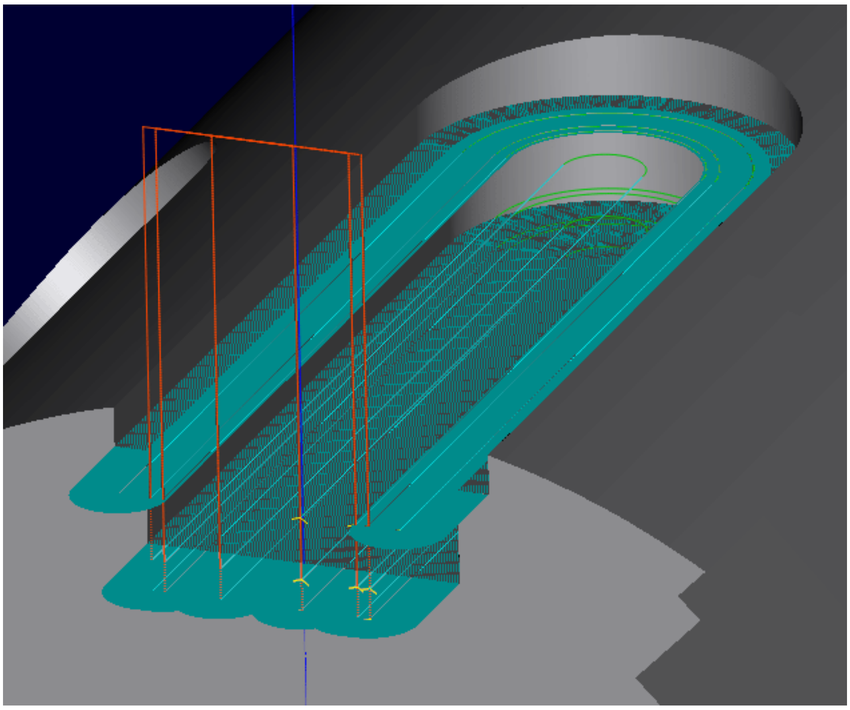
And in perspective view, you can see that the machining is done at the right position and at the right depth by displaying the layer containing the 3D object.



Select the outer polyline, then assign it a ***Profile*** machining operation with the following settings:

- ***Tool Diameter*** = 3
- ***Tool Profile*** = End Mill
- ***Stock Surface*** = 25 (the radius)
- ***Target Depth*** = 23
- ***Clearance Plane*** = 35
- ***Depth Increment*** = 2
- ***Cut Width*** = 0

Generate the toolpaths and you should get the result below.



We are done with setting up the machining operations.

Setting up rotation and repeat instructions

We are now going to set up the rotation instructions as well as a machining repetition in order to reproduce these two machining operations eight time with a 45° offset.

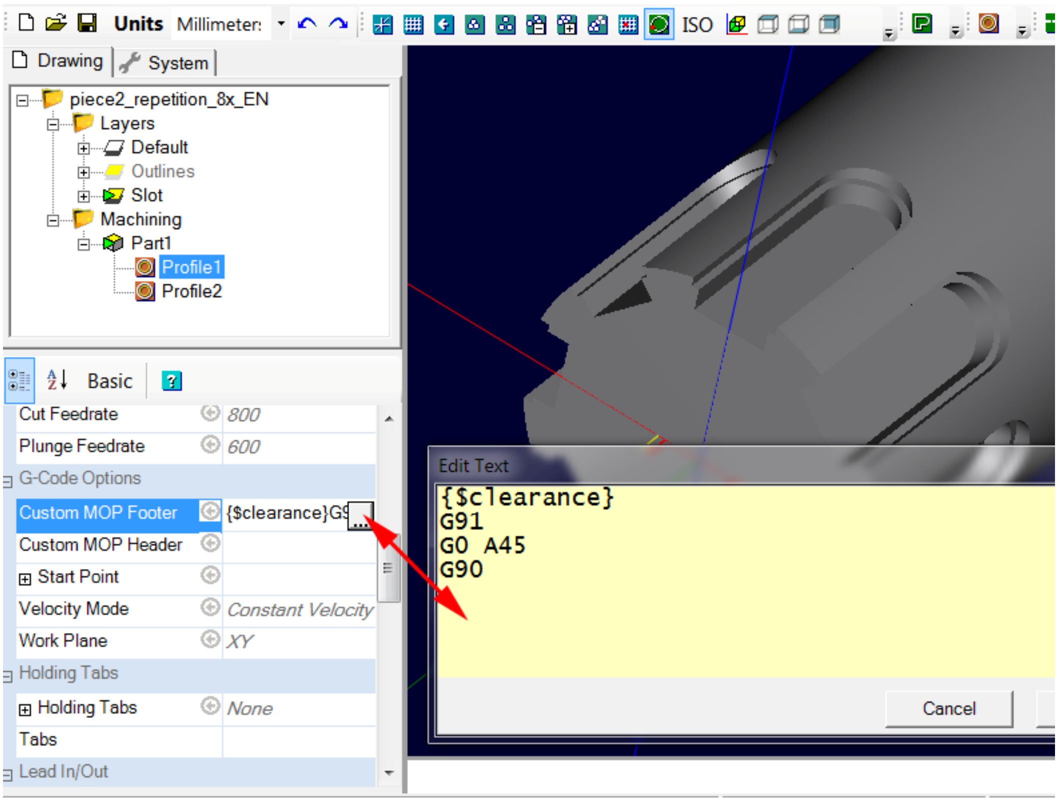
In the case of simple positional, we wrote the rotation instructions in the machining operations, and this for each machining position, and these instructions were in absolute position. (0°, -60°, -90°)

In the present case, on the other hand, we are going to repeat the same machining series several times, but without creating as many operations as rotational positions ; we will therefore have to give relative rotation instructions (incremental mode) at the start of the first machining operation ; in the case of this piece, there are eight positions out of 360°, so it will be necessary to turn the rotary axis by 360/8=45° at each repetition.

In the ***Custom MOP Header*** of the first machining operation (Profil 1) of the ***CAMpart*** **Part1** we are therefore going to write:

{$clearance} >go up to the clearance plane
G91 > use relative coordinate mode.
G0 A45 > rotate axis A by 45°
G90 > return to absolute coordinates for the rest of the machining

Each time this code will be executed, A axis will rotate by 45°



In this case, the direction of rotation of the axis does not matter and you can use -45° as well as 45°

Now we are going to have to repeat the set of machining operations in the ***CAMPart*** **Part1** eight times.

The idea is to use *CamBam*'s ***Nesting*** function ; this function makes it possible to repeat all the workings of a ***CAMPart*** a certain number of times. In general, this function is used to repeat machining operations at different places on a stock.

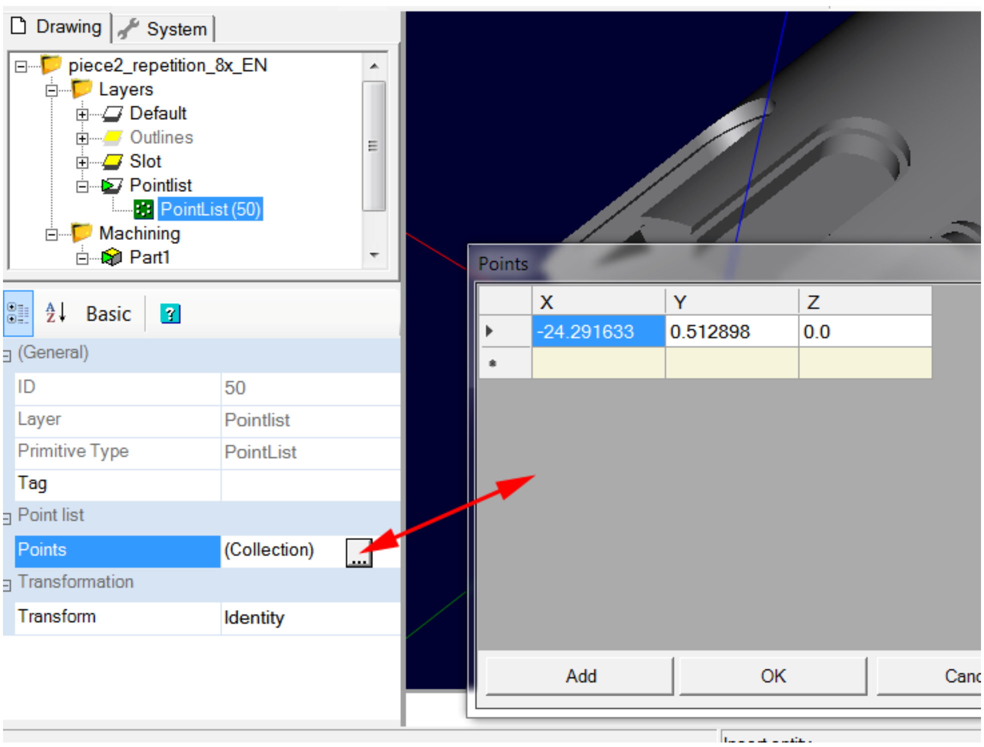
In this case, we are going to create a repetition that reproduces the workings of **Part1** always at the same place (X and Y therefore do not change) but due to the rotation GCode included in the header of the first operation, at each repetition, the A axis will first rotate 45° before the two machining operations of part1 are executed.

For this we are going to use the *Points List* option of the ***Nest Method*** property of the ***CAMPart*** **part1**. This method makes it possible to repeat the contents of the ***CAMPart*** at each position defined by a point in the point list used as a model.

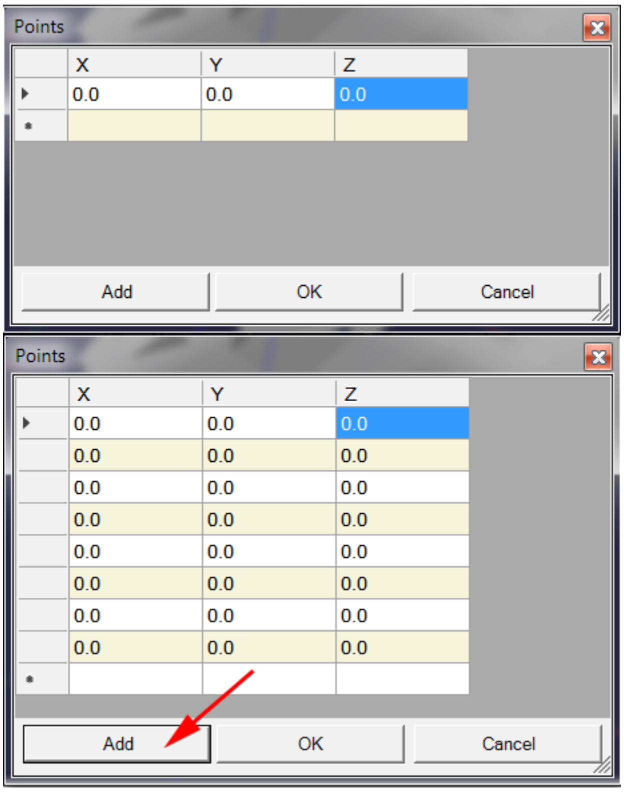
As we want the machining operations to be all repeated at the same place, we simply need to create a point list containing 8 points (for 8 repetitions) whose coordinates will all be at 0,0,0 in XYZ.

Use the ***Draw Point List*** tool from the toolbar, and click once in the drawing area to create a point. (anywhere)

You will get a ***Point List*** object on the current layer ; select it and in its properties, click on (Collection) in the ***Points*** property, then on the |...| on the right to edit this collection.



Replace the values that are not at 0 by editing them, then click the ***Add*** button 7 times to obtain 8 lines, all at 0,0,0



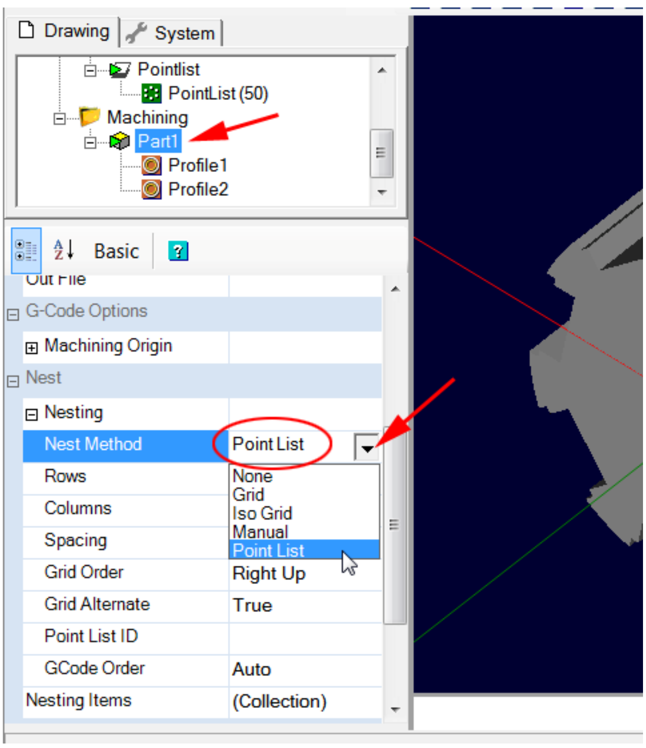
If you need to delete a row, select it by clicking in the left column and use the **DEL** key on the keyboard to delete it.

Click ***OK*** to finish.

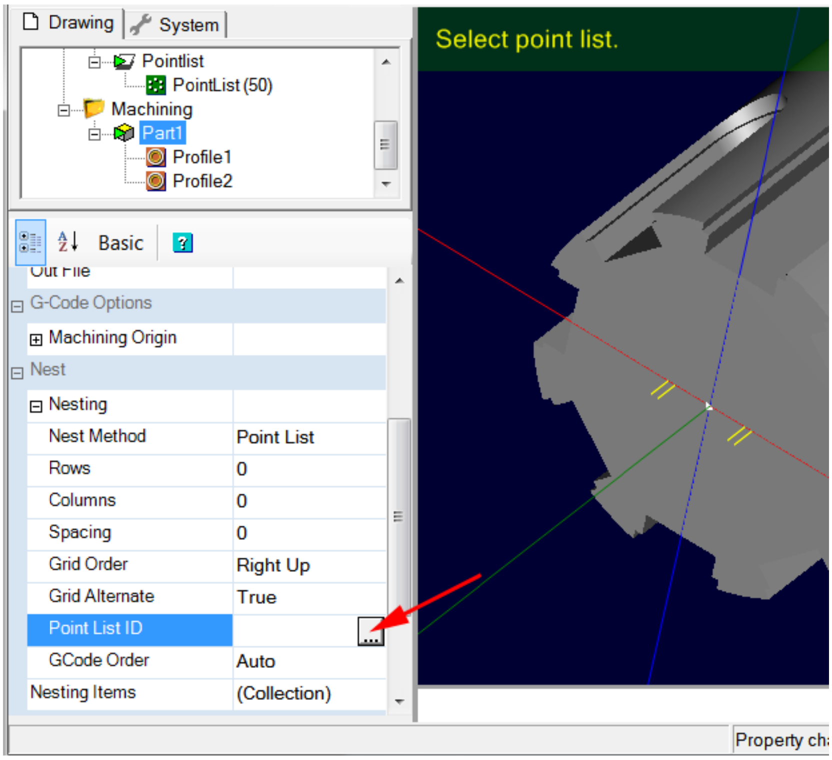
We are now going to define the nesting and assign to it the ***Point List*** that we have just created.

Select the ***CAMPart*** containing the operations to be repeated (part1), unfold the ***Nesting*** property and set:

Nest Method: *Point List*



Then, to select the ***Point List*** to use, click on the ***Point List ID property***, then on the |...| that appears on the right, *CamBam* then asks you to select the list of points to use.

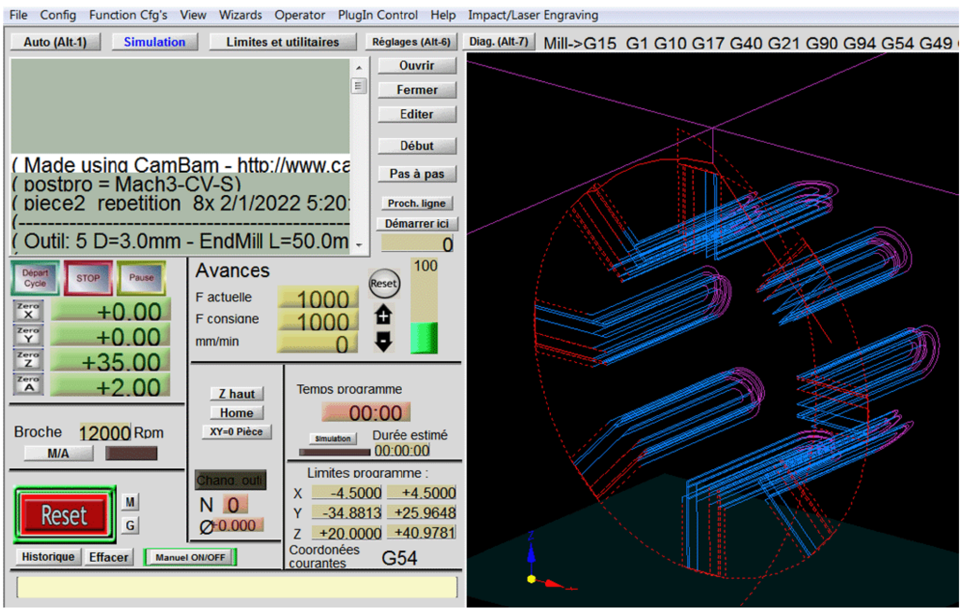


Then click on the ***Point List*** object in the list of drawing objects to select it, and validate with the **Enter** key on the keyboard (don't forget to validate with **Enter**, otherwise it will not be taken into account). In the case of this example, the ***Point List ID*** property should then contain the value 50 (the ID number of the ***Point List*** object)

You can also manually enter the ID number in the ***Point List ID*** property if desired.

We are now done with the parameterization and we can produce the GCode and check it on *Mach3* or on a 4-axis machining simulator like NCnetic for example.

Seen under Mach3



- ***Note:*** if you need to create a list containing a large number of points, another method is to:
- Create a point at 0,0,0 with the ***Draw Point List*** tool (use the grid to make sure the point is created at 0,0,0)
- This point being selected, use the command ***Edit/Transform/Array Copy*** and give a value equal to number of points necessary less 1 (-1 because there is already an existing point) in **Number of Copies**, click on OK
- In **Offset per step** enter a value of 0,0,0 so that all points are created at the center of the *CamBam* universe.
- You will get as many Points List objects as points requested.
- With all these Point List objects selected, use the ***Edit/Join*** command to obtain a single ***Point List*** object containing all the points.

See also the documentation about Nesting.

The *CamBam* example file is available in the archive under the name: ***piece2_repetition_8x.cb***

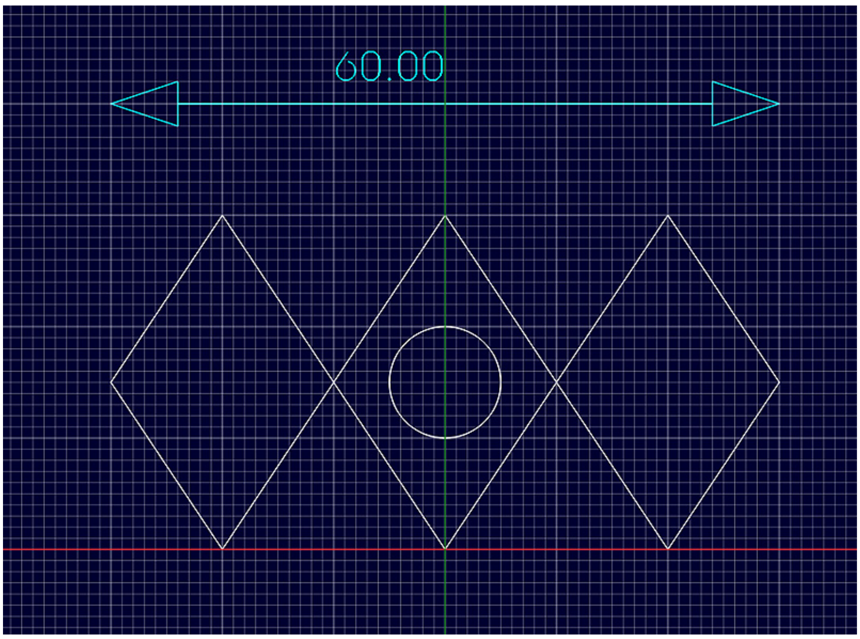
c) 2D engraving on a cylinder by wrapping the GCode

We will now move on to another 4-axis machining method ; this time, we will work with a "flat" machining, then, thanks to a specific post processor, we will produce a GCode which will "roll up" on itself ; it is in fact a question of replacing the Cartesian coordinates in X or Y by polar coordinates, the X (or Y) axis will be replaced by the A axis.

We will start with a simple, very basic engraving that will have to engrave on a 40mm diameter cylinder.

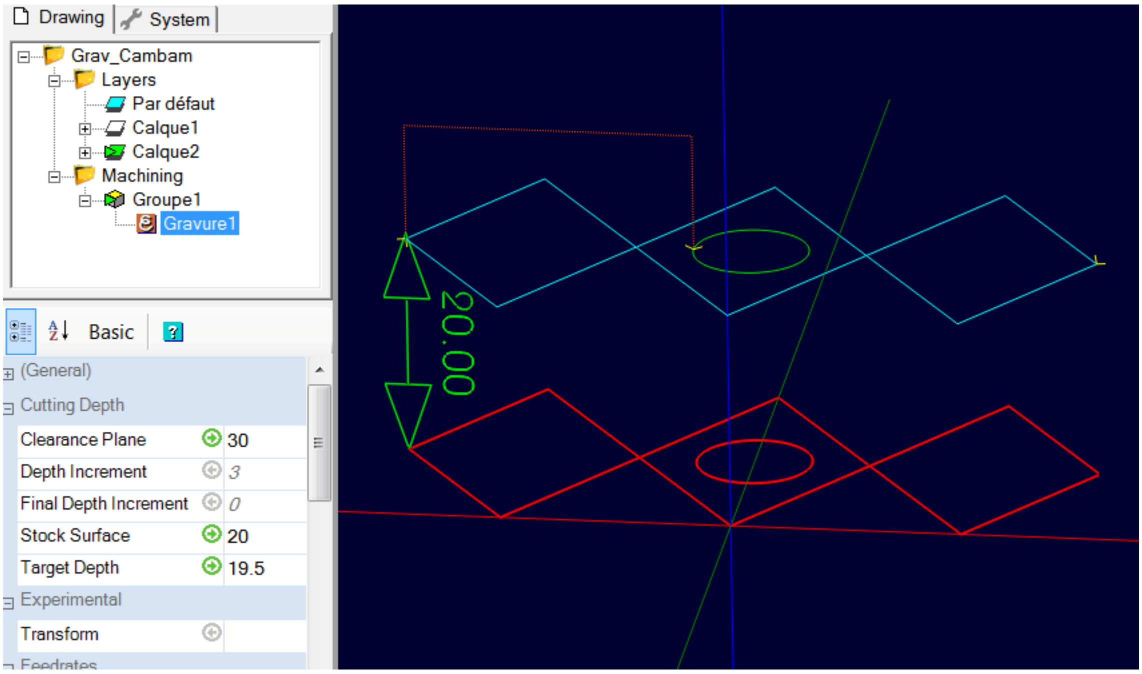
At first, we draw the design to be engraved ; in this case, my A axis is aligned with the Y axis of the machine and it is the X axis that we are going to wrap around the cylindrical shape.

Here is the drawing

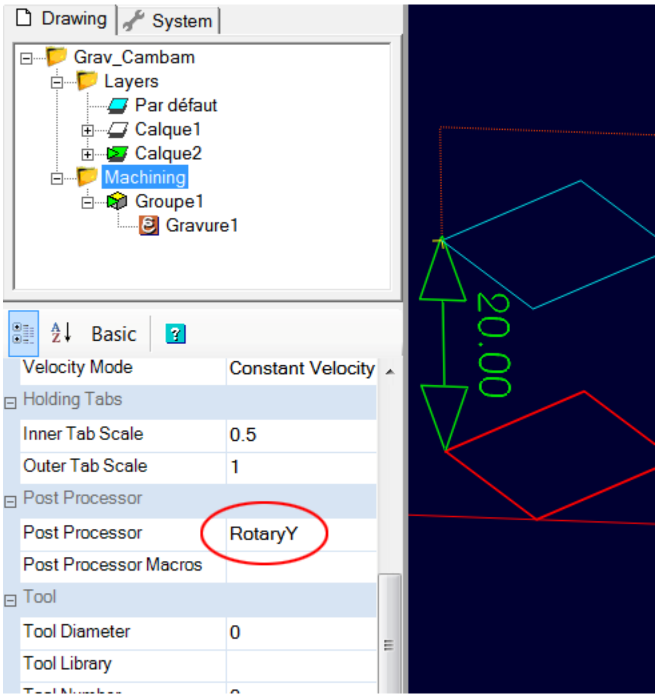


As we can see, the drawing is 60mm wide (in X therefore), and it is this X axis that will be wrapp. On a 40mm diameter cylinder, a complete "turn" would be 40 * Pi, so about 125.66 mm ; here my drawing is only 60mm, so it will not make a complete turn of the cylinder but only 60/125.66 = 0.477 times the circumference, so 0.477*360° = 171.9°

As the post processor uses the ***Stock Surface*** value as the wrap radius, we have to leave the drawing at Z=0 and set the ***Stock Surface*** value to the value of the wrap radius, here 20mm. On the image we see the drawing (in red) and the toolpaths (in blue) at 19.5mm in Z i.e. an engraving of 0.5mm depth compared to the radius of 20mm.

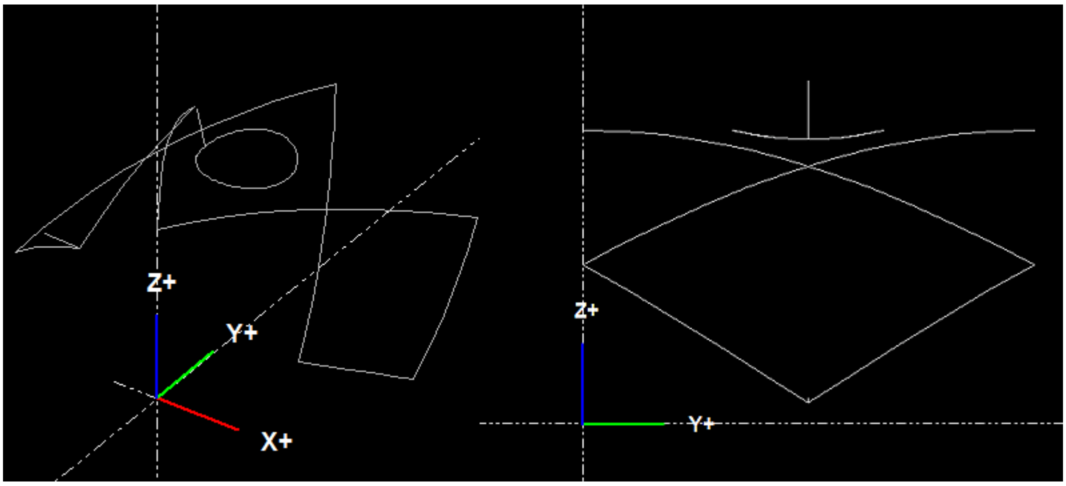


The work flow is quite similar to a flat engraving, except that we will select a specific post-processor for the wrapping. In *CamBam* V1.0, this post processor is called *RotaryY* for wrap X around Y, or *RotaryX* for wrap Y around X. In the case of this example, the rotary axis is aligned with the Y axis , so I'm going to choose the *RotaryY* post processor in order to wrapp X around Y.



Now just produce the GCode as usual. The result cannot be simulated on *CamBam*, it will be necessary to examine the GCode produced on a software capable of displaying 4-axis GCode such as *Mach3*, NCnetic, *NCplot*, etc ...

Here is the result of the toolpaths seen under *NCplot*.



We can wrap any type of GCode, whether it is an engraving, a pocket, a profile or a 3D machining.

The *CamBam* example file is available in the archive under the name: ***grave_cylindre.cb***

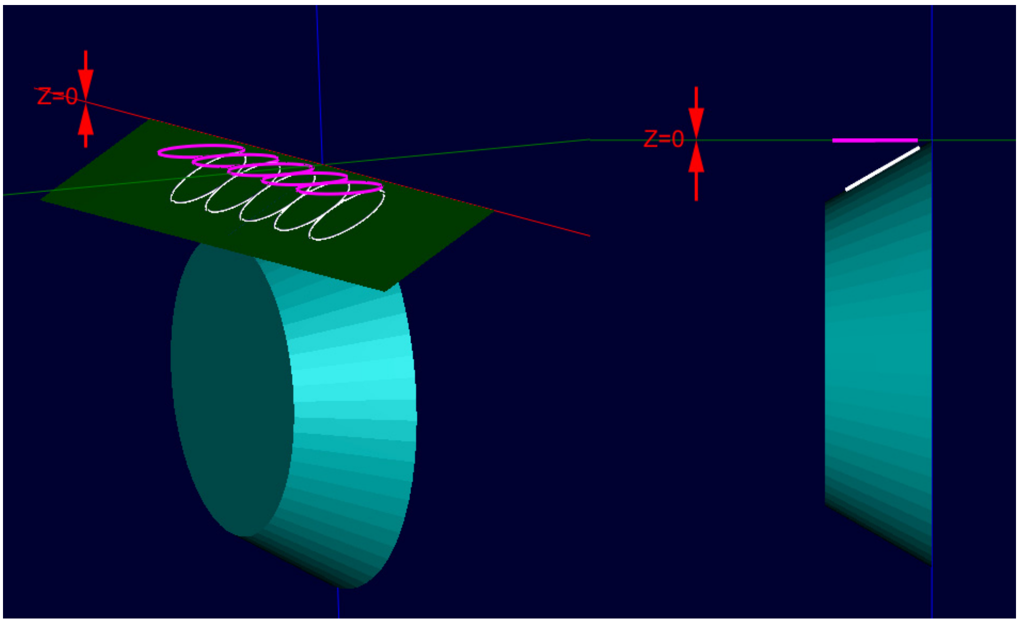
d) Engraving on a cone or other non-cylindrical form of revolution

It is also possible to engrave on a conical shape or any other form of revolution.

1. **Engraving on a cone**

The principle is the same as for engraving on a cylinder, except that the design to be engraved must be inclined at the same angle as the slope of the cone.

As for a cylinder, where the position of the drawing to be engraved must be at Z = 0 and ***Stock surface*** = at the wrapping radius, here, it is the large diameter of the cone which must be at Z = 0, the drawing to be engraved, will therefore be below Z=0.

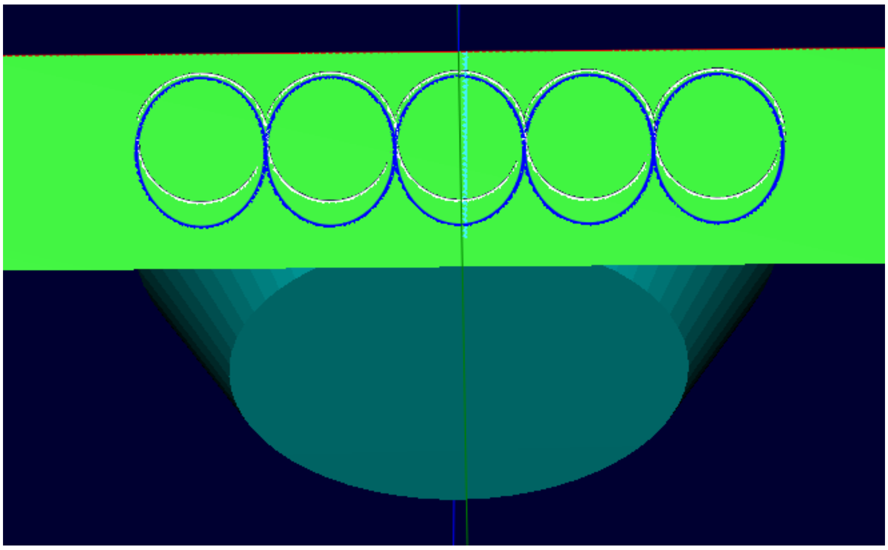


On this image, the shapes in purple have been tilted to follow the slope of the cone, which gives the circles in white ; the 3D shapes (the cone and the inclined plane) are not necessary but they help to check that everything is in the right place.

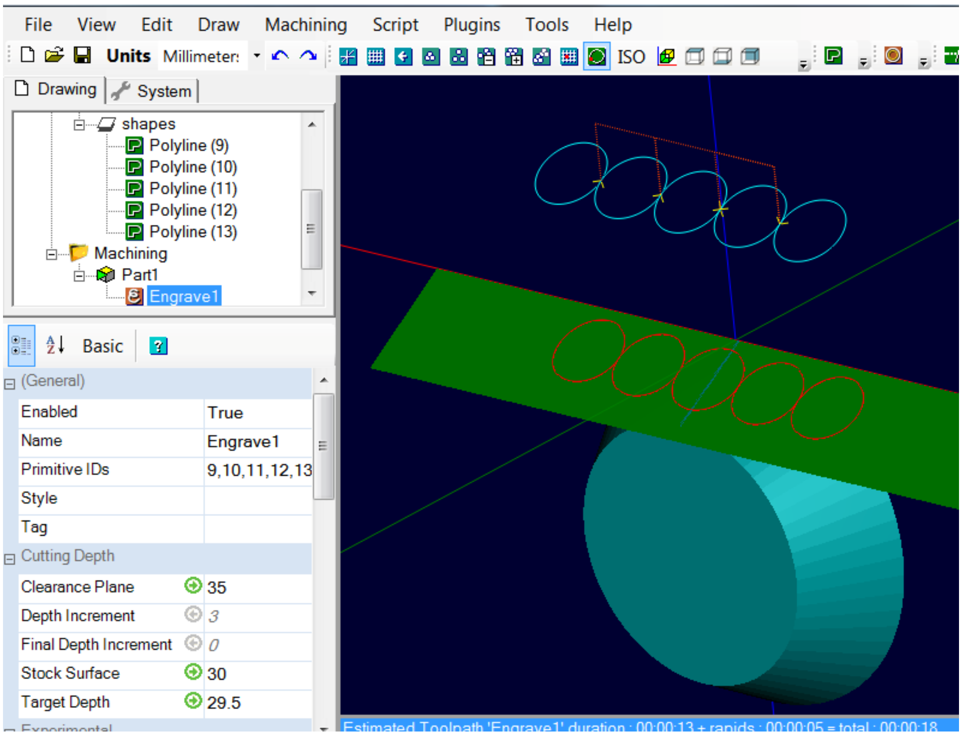
> ***Note:*** *CamBam* cannot rotate/skew shapes containing arcs, so it is imperative to remove the arcs in order to have a dawing containing only straight segments ; this can be done using ***Edit/Polyline/Remove Arcs***.

It is also possible to create inclined shapes by projecting planar shapes onto an inclined surface, however this will induce a deformation ; in the case of a projection, the inclined plane is therefore necessary. To do this, use ***Edit/Surface/Project Lines To Surface***

As can be seen in this image, a projection of the circles (in blue) on the inclined plane produces oval shapes unlike an inclination which does not distort the drawing (in white)



As with engraving on a cylinder, ***Stock Surface*** should have a value equal to the radius. (the major radius in the case of a cone)



Only the standard *CamBam* engraving operation can be used for this type of machining, as it is the only one that works in 3D (a pocket or profile would only produce "flat" toolpaths and would not follow the inclination of shapes)

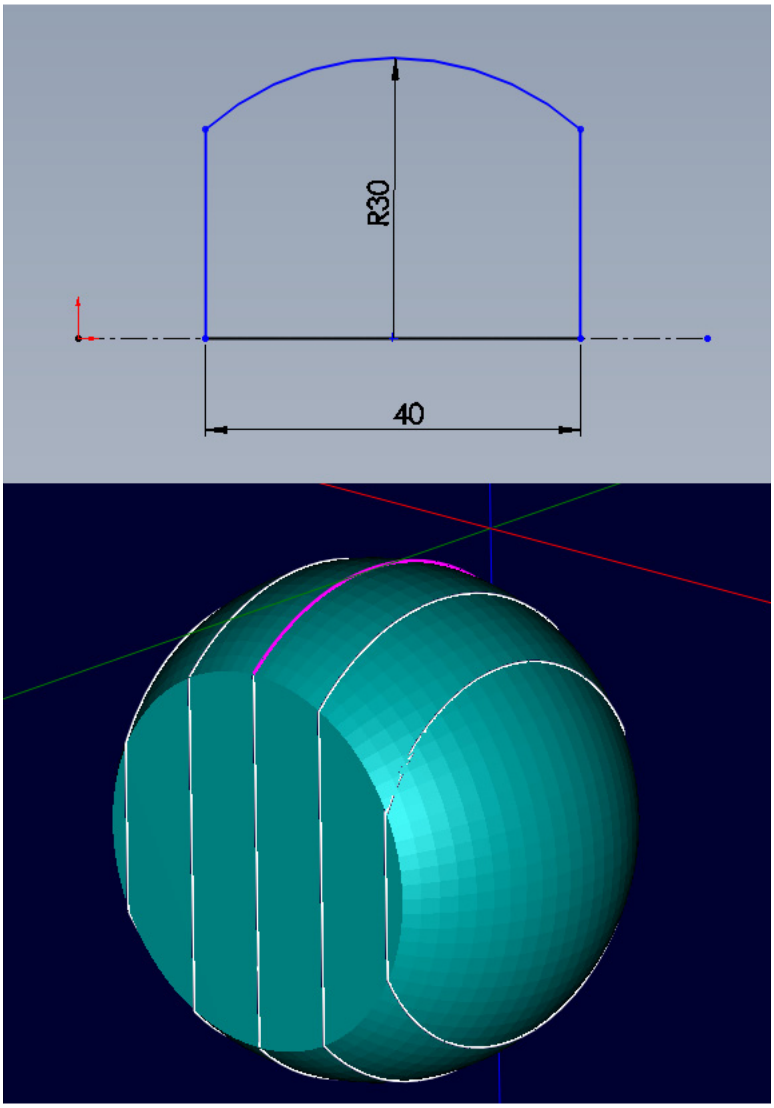
Don't forget to set the ***Clearance Plane*** to a value greater than the radius of the part.

The *CamBam* example file is available in the archive under the name: ***grave_cone.cb***

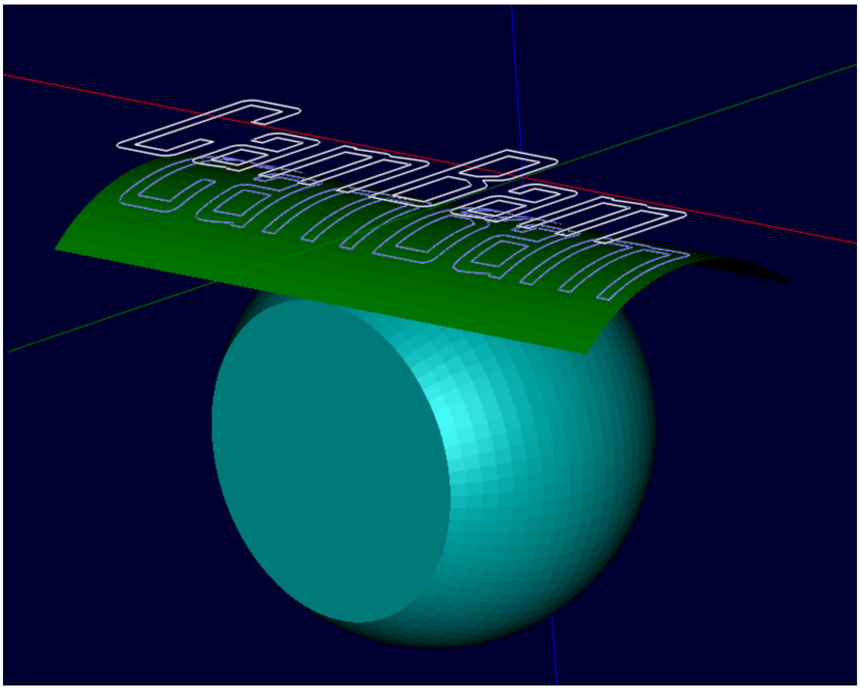
2. Engraving on any form of revolution.

The principle remains the same, except that the projection of the drawing to be engraved will be done on a shape having the profile of the development of the final shape and not on a simple inclined plane.

After slicing the part with ***Edit/Surface/Plane Slice X***, I keep the purple curve, which I will extrude to obtain an developed shape. *CamBam* can only extrude in Z, so this will require some manipulations. (you can also use my Unroll 3D Model plugin to create the developed)

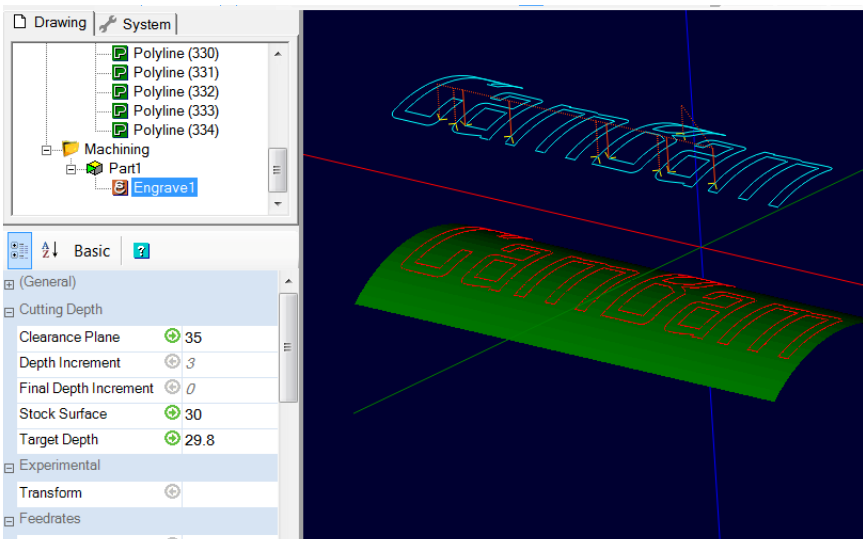


I then project the drawing to be engraved on the developed.



And I use the projected lines as a design to engrave.

The rest happens as with an engraving on a cylinder/cone, ***Stock surface*** = at the maximum radius of the shape.



The *CamBam* example file is available in the archive under the name: ***grave_tonneau.cb***

e) 3D machining with wrapping the GCode

As for the engraving above, wrapping a 3D model does not pose any problem and is done in exactly the same way ; the "flat" 3D object will have to be positioned at the radius and the toolpaths will be produced as for any "standard" 3D machining, it is the *RotaryX* or *RotaryY* post processor that will generate a wrapped GCode.

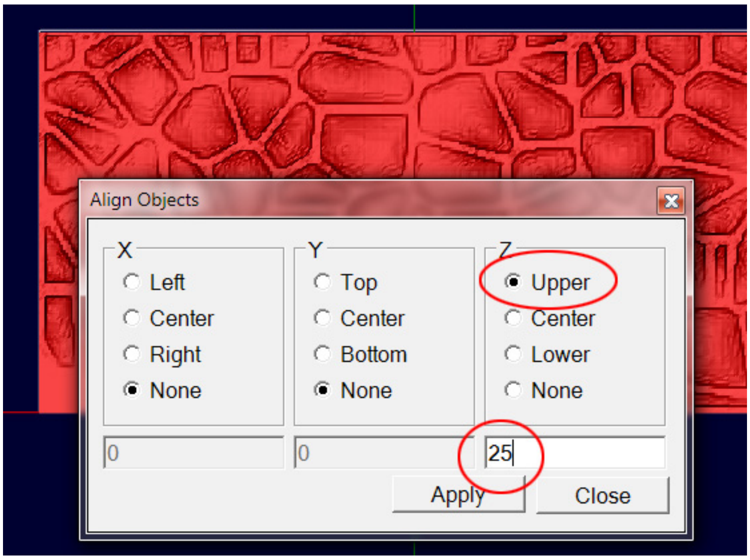
The only particularity in the case of a 3D model is that it will be necessary to finely manage the area to be machined using the ***Boundary*** because the machining must not "overflow" on the sides, otherwise it would make a "hole" in the part and they also need to come right to the edge of the part.

In this example, the 3D model needs to be wrapped around a 50mm diameter cylinder, and I want it to go all the way around, so I need a 3D surface of 50 * Pi = 157.075mm wide.

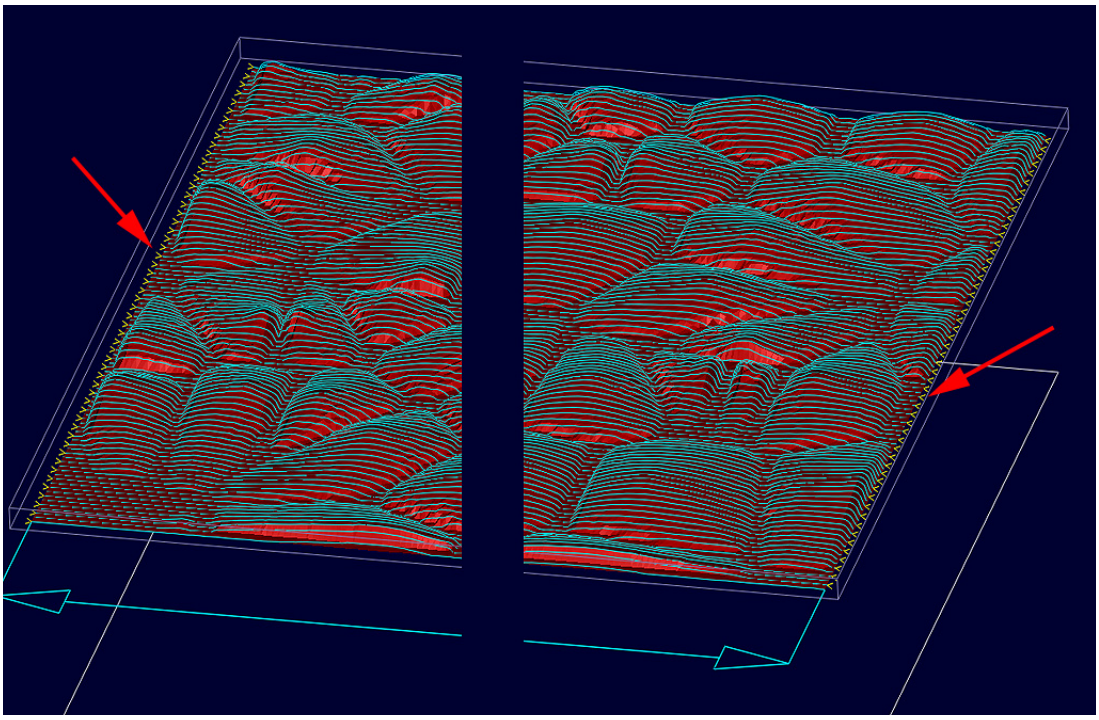
The top of the 3D model should be at Z=25 (the radius) ; I use the alignment tool to position the top surface of the 3D model at 25mm in Z. (the shape is centered on X, but it is not a requirement)

The 3D model is available in the archive under the name: ***colonne_wrap3D.stl***

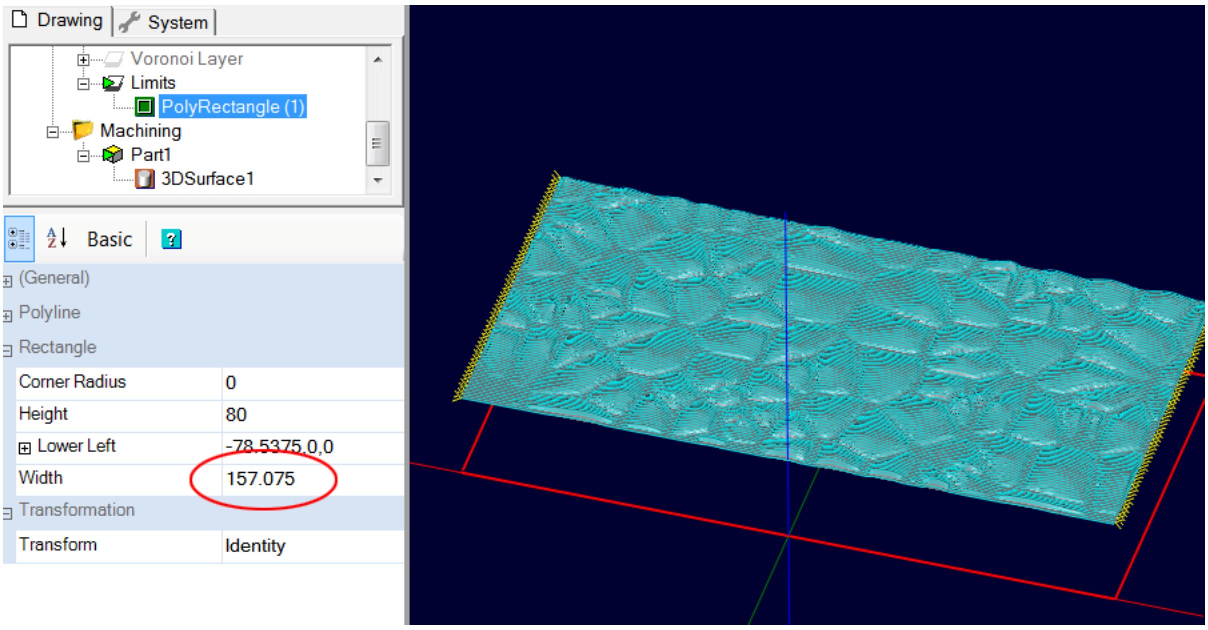
Positioning of the upper face at Z = 25



As seen in the following image, the toolpaths should stop right at the edge of the 3D model (red arrows)



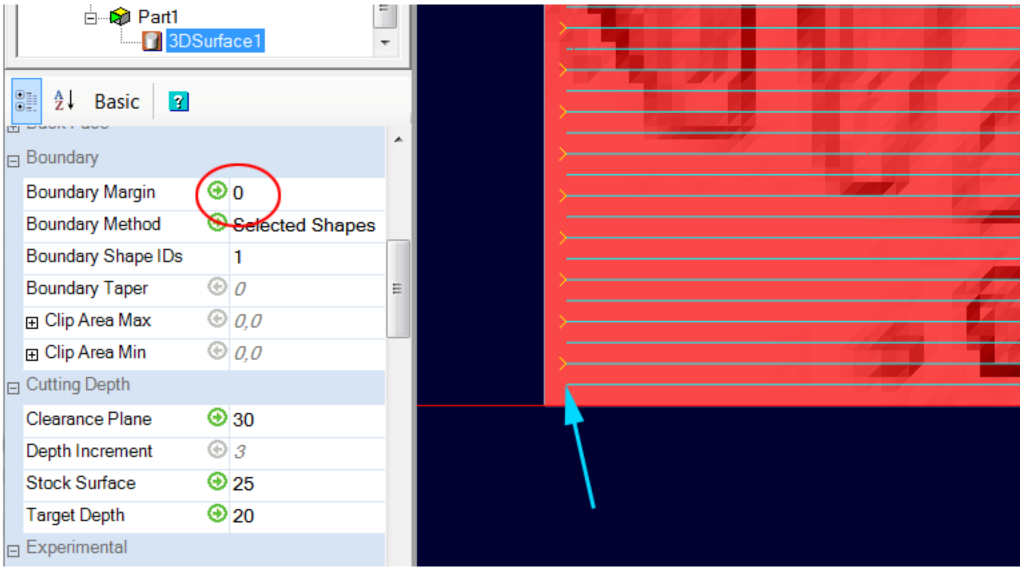
To make sure that the toolpaths will not down on the sides of the model, which would produce a hole in the part, or that the toolpaths will not stop before the end of the model, which would leave an unmachined area, I uses a rectangle of the exact width to be machined, here 157.075, which will be used as the machining limit.



In the ***Boundary***, I set ***Boundary Method*** to *Selected Shapes*, use my rectangle (Rectangle1) as ***Boundary Shapes IDs***, and set ***Boundary Margin*** to a value equal to the radius of the tool being used ; here my tool is 1.5mm in diameter, so I set my ***Boundary Margin*** to 0.75

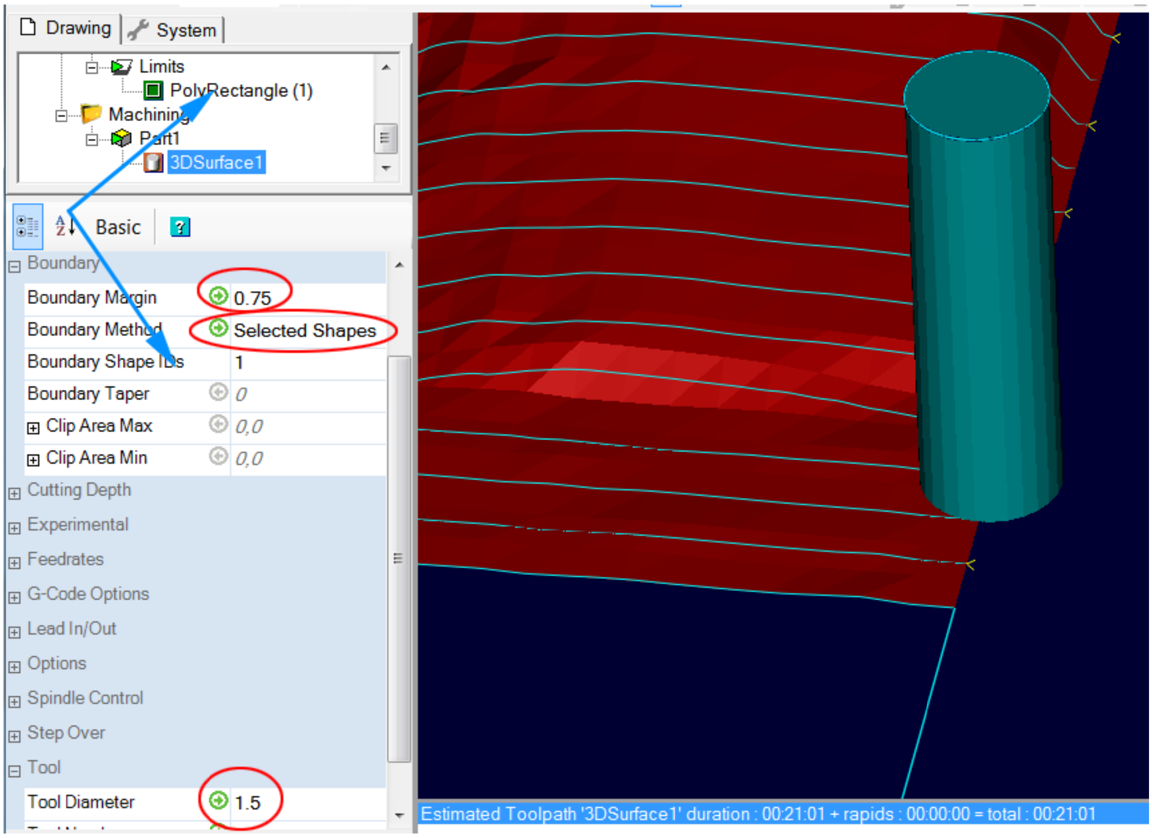
Why add tool radius? ....because *CamBam* manages the boundaries in such a way that the tool stays **inside** the boundaries, therefore the toolpaths stop at a distance equal to the radius of the tool from the boundaries.

On this image, with ***Boundary Margin*** to 0, we see that the toolpaths do not reach the end of the model bounded by the rectangle used to define the limits.



The goal is to ensure that the center of the tool arrives exactly at the end of the boudary shape, no more otherwise we go down the sides, but no less otherwise the model will not be machined to the end.

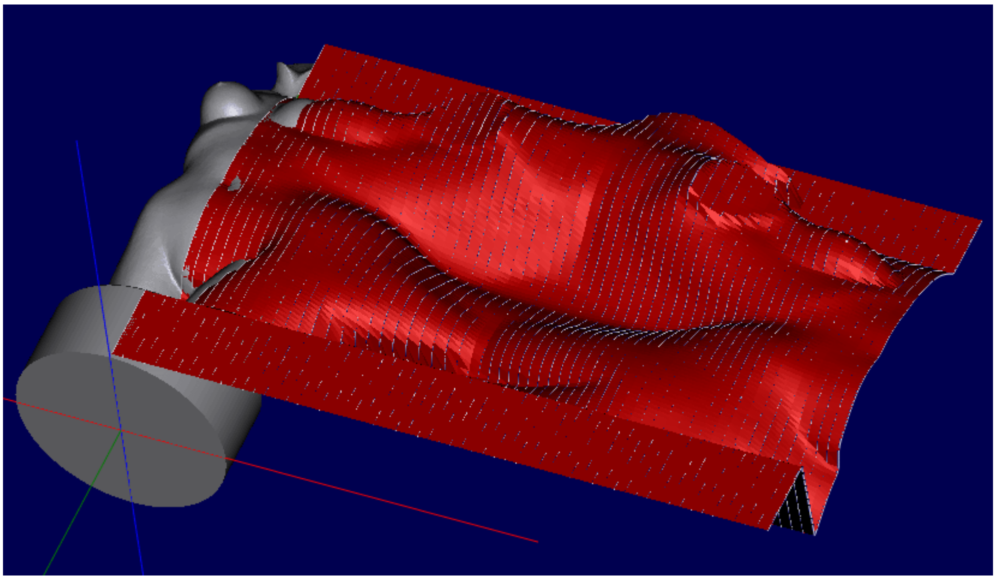
With ***Boundary Margin*** equal to tool radius, the center of the tool comes right up to the boundary of the model



Now all that remains is to produce the GCode, of course using a *RotaryY* post processor in this case.

The example *CamBam* file is available in the archive under the name: ***colonne_wrap3D.cb***

In the case where the 3D object is not a developed shape, you must use my Unroll 3D Model plugin to obtain the unrolled shape, the rest of the procedure is then identical to what we have just seen with the 3D "stone texture" above.



Here, it is the unrolled shape in red that would be used to produce the GCode.

The scanline direction (horizontal/vertical) of the 3D profile operation will determine whether the shape is machined by turning, or by parallel strips along the axis of rotation.

Automation

CamBam supports two forms of automation : Scripting and Plugins.

Scripting

Some example scripts are provided in the \scripts sub folder of the CamBam installation directory.

Refer to these forum sections for more information and examples of scripting.

Scripts and Plugin Help

Resources - Scripts and Plugins

Plugins

Plugins are .NET .dll files which can be written in a variety of supported .NET languages such as C#, Visual Basic, C/C++ etc.

Some example plugins are provided in the \plugins sub folder of the CamBam installation directory.

For a fantastic introduction to writing user plugins refer to MrBean's thread on the CamBam forum:
How to write a CamBam plugin

Configuration

Tools - Options

Arc Display Degrees	Arcs are displayed using multiple line segments. This setting defines the angle between each segment. Smaller numbers make smoother curves but slower display rates.
Arc Fit Tolerance	The tolerance used when automatic arc fitting is applied in some drawing operations. Zero will use an automatically calculated value.
Auto Arc Fitting	Whether to apply arc fitting in certain drawing operations (such as plane slicing)
Auto-Apply Transformations	If *True*, transformations such as rotations, moving, resizing and array copies will automatically apply transformations and reset the transformation matrix. In some cases, such as rotating a circle around the Y or X axis, this is not possible so the original entity plus transformation matrix is retained.
Backface Culling	When displaying surface meshes, faces with back facing normals (using right hand rule) are not displayed. This can speed up displaying meshes considerably and also make the wireframe 3D view clearer.
Check Version At Start	*True* | *False* If *True*, the program will check for updates from the internet when it loads. Set this option False if you are not connected to the internet. The version check only downloads a tiny text file from the CamBam web site containing the latest version number. No other information is transferred.
Cut Width Color	The color used to display toolpath cut widths.
Default Font Family	This is the font used when no font is specified for text drawing objects.
Default GCode Extension	A default file extension used when gcode files are produced.
Default Layer Color	The color to use when new layers are inserted into a drawing.
Default Layer Pen Width	Default line width used to draw objects within a layer
Default Post Processor	Post processor definitions can be used to customise the GCode output. This post processor will be used if the drawing does not specify one.
Default Script Entity	The default script to insert whenever ***Script Objects*** are inserted..
Default Stock Color	The default color to use to display stock objects.

Diagnostic Level	An integer number used to control the number of information messages displayed in the message pane at the bottom of the CamBam interface. Typical values are 0 to 4, where 0 displays little or no messages and 4 displays reams of diagnostic information.
Display Mode	Controls the method used to display the 3D drawing view. OpenGL is a fast, preferred method but may cause problems with some graphics drivers. OpenGL_Legacy uses immediate mode which was the standard OpenGL method up to 0.9.8M. GDI is a slower but potentially less susceptible to driver problems. Use this mode if the drawing display seems very slow or corrupted. Changing the DisplayMode option requires CamBam to be restarted. Holding the **SHIFT** key while starting CamBam will force the use of GDI mode.
Drawing Template	This property can contain the filename of a CamBam drawing (.cb file) to be used as a template for new drawings. Whenever a new drawing is created, or a non CamBam (such as DXF, 3DS etc) is loaded, the basic format and properties of the drawing template will be used for that document. This is useful for setting default values for properties stored in documents, such as ***Post Processors***.
Drawing Units	This sets the drawing units to be used for new drawings. This property may be overridden by the drawing units of the ***Drawing Template***, if one is supplied.
Export 3D polylines	If *true*, all polylines will export as 3D polylines.
Field Of View	Angle of view in degrees.
File Backups	When saving CamBam (.cb), library or post processor files, a backup of the existing file is created before overwriting. The backup file is of the format 'filename.b#', where # is a number. The number of backups to keep is specified in the ***File Backups*** property.
File Open Worker Thread	If *True*, open files using a worker thread
GCode Editor	Specify an external command used to edit gcode files. If no command is specified, the internal editor is used. GCode files can be edited by invoking the Machining, ***Edit gcode*** menu option. Example: %windir%\system32\notepad.exe.
Gerber - Flatten	If *True*, flatten all layers to a single (unioned) layer.

Gerber - Subtract Layers	If *True*, '*Clear*' layers will be subtracted from previous layers.
Gerber - Union Layers	If *True*, all shapes on each layer will be unioned together.
Gerber - Union Traces	If *True*, each trace will be unioned together. If *False* the traces will be left as line and arc sections.
GLSL Shader Version	The version of the OpenGL shader language to use. 110 is older but should be more compatible with older display drivers. 330 may only be supported by newer display drivers.
Grid Color	The color of the drawing grid.
Grid Info (Inches)	Information that defines the drawing grid when Inches drawing units are used. • ***Drawing Units*** - drawing units used by the grid. • ***Minimum*** - X, Y location of the lower left point of the visible grid. • ***Maximum*** - X, Y location of the upper right point of the visible grid. • ***Major Scale*** - Number of units in the grid's major scale. • ***Minor Scale*** - Number of units in the grid's minor scale.
Grid Info (Metric)	Information that defines the drawing grid when Metric drawing units are used.
Holding Tab Drag Toolpath Refresh	*True* | *False* If *True*, holding tabs are automatically applied to toolpaths when tabs are moved. If *False* tabs will be applied when the toolpaths are regenerated.
Language	The desired language to use for the CamBam user interface. CamBam will need to be restarted for this change to take effect. Language translation files will need to be downloaded from the internet for the translations to function. The translation files may be periodically updated. Use the ***Tools - Download latest translations*** menu item to download the latest versions from the CamBam website. See www.cambam.info/ref/ref.lang for more details.
Marker Color	Color used for markers such as selection bounds.
Max Line Width	The maximum line width allowed (in pixels). A zero value will use an automatic setting.
Offset Backtrack Check	If True, back track drawing glitches in polylines are detected and removed by the offset routine used in toolpath generation. Back tracks can cause the offset routines to produce unexpected results.

Out File Prompt	When to prompt for a GCode filename. *If Exist* - if out file already exists. *Always* - whenever GCode is generated *Never* - overwrite existing file
Print Options	Settings to control printing ***Print Color*** - *Black and White* - *Monochrome* - *Color* ***Page Orientation*** - *Portrait* - *landscape* ***Margins*** - *Bottom* - *Left* - *Right* - *Top* ***Printer Units*** - *Millimeters*|*Inches*|*Centimeter*|*Meter*|*Thousandths*|*Unknow* ***Border Size*** - Line thickness used to draw a border around margins (0 no border) ***Line Width Scale*** - Scale line thickness by this amount when printing
Repeat Commands	If *True*, drawing commands will be repeated. To end the current drawing mode, press ESC or click the middle mouse button.
Rotation Mode	*ALT+Left* | *Left+Middle* | *Left+Right* The key and mouse combination used to rotate the drawing view. *ALT+Left* - the view is rotated by holding down the **ALT** key and dragging with the left mouse button. *Left+Middle* - the view is rotated by pressing the middle mouse button and dragging with the left mouse button. The middle mouse button can be released while dragging. *Left+Right* - the view is rotated by pressing the right then left mouse button and dragging. The right mouse button can be released while dragging.
Select Color	The color used to select paint selected shapes.
Selected Entity Focus	If *True*, when selecting a drawing object, the drawing tree layer is expanded and scrolled to display the item. **ALT+TAB** can also be used to toggle the focus of selected drawing objects in the drawing tree.
Select Fade	Controls how much unselected shapes are faded (as a percent).
Show Grid	Sets whether the drawing grid is displayed. Alternatively use the show grid button on the toolbar.
Show Licence Form	If *True*, CamBam will display the licence entry dialog if no license file is detected when the program start.
Show Selection Corners	If *True*, display corner markers of selected shape bounds.
Snap to Grid	*True* | *False* If True, drawing points will snap to the minor grid units. This option can also be changed from the ***View - Snap to Grid*** menu option or toggled using **Ctrl+G**.

Snap to Points	*True* | *False* If *True*, drawing points will snap to shape control points, circle centers and other significant points.
Spline Curve Steps	When splines are displayed, their shape is approximated by line segments. This setting controls the number of segments used to display. A larger number will give a smoother appearance but may slow display performance. This setting does not affect the resolution of geometric operations based on splines, such as toolpath generation.
Spline to Polyline Tolerance	Splines are converted to polylines internally before they can be used for some operations, such as toolpath generation. This setting controls the degree of error allowed in this conversion, measured in drawing units. A smaller value will result in more accurate spline conversions but can hinder performance considerably.
STEP Resolution	Resolution to use when converting STEP surfaces to meshes in millimeters.
System Path	The system path is the root folder where CamBam library (styles, tools etc), post processor and drawing templates are stored. The following macros can be used: *{$common}* - Common application data folder (%ALLUSERPROFILE%). In Windows XP this is typically: \Documents and Settings\All Users\Application Data\CamBam plus 1.0\ And in Windows 7 \ProgramData\CamBam plus 1.0\ *{$user}* - User application data folder (%USERPROFILE%).
Text Curve Tolerance	Text objects are converted to polylines internally before they can be used for some operations, such as toolpath generation. This setting controls the degree of error allowed in this conversion, measured in font units (0-2048). A smaller value will result in more accurate text conversions but can hinder performance considerably.
Thinking Message	Message to display when CamBam is busy calculating. Displayed in full, unexpurgated technicolor! :-)
Toolpath Arc Color	The color of arc segments in toolpaths.
Toolpath Line Color	The color of line segments in toolpaths.
Toolpath Rapid Color	The color used to display toolpath rapids.
Use Surface Selection Color	If *True*, selected 3D meshes will be displayed using the selection color.

View 3D Wireframe	If *True*, 3D meshes will be displayed in wireframe mode.
View Background Color	The color of the drawing background.
View Focal Length Scale	For *Perspective* projection mode, drawing objects are set back from the view point by a distance based on the height of the view port, multiplied by the ***View Focal Length Scale***. Smaller values will give a greater perspective effect but may result in some clipping when viewing larger objects.
View Projection Mode	*Orthographic* - drawing dimensions are consistent regardless of depth. *Perspective* - drawing dimensions are perspective scaled based on depth. Drawing objects are set back from the view point by a distance base on ***View Focal Length Scale***.
View Text Color	The color used to display text in the drawing view.
Waterline Safety Check	If *True*, prevents gcode creating if errors are suspected in 3D waterline toolpaths.
Worker Threads	Number of simultaneous worker threads to use.

Change the location of CamBam's "system" folder

By default, the "system" of CamBam is located in ***C:\ProgramData\CamBam plus 1.0***, and it contains the folders and definition files of the tool libraries, styles, post processors, template files, scripts as well as plugin configuration files and the translation file.

It may be desirable to move this "system" folder to another location for the following reasons:

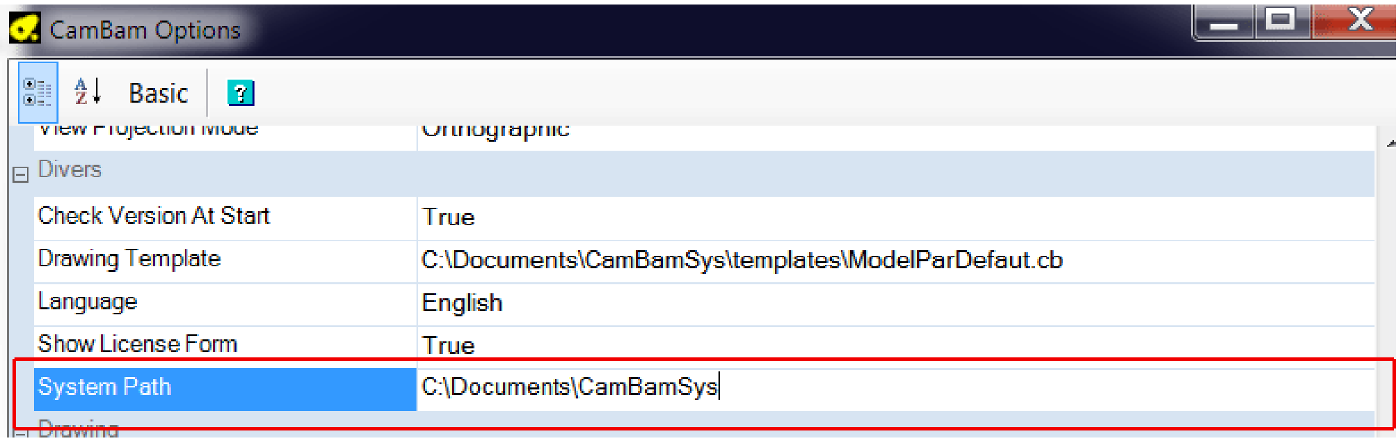
- The *ProgramData* folder is a hidden folder, not easily accessible if *Show hidden files, folders and drives* is not enabled in Windows
- If you want to share the CamBam system over the network, it is undesirable to share a Windows system folder.
- When reinstalling/updating CamBam, the default system files and folders are overwritten by the installer, which will not be the case if they are in a custom location.

To move CamBam's system folder:

1. Create a new folder in a location where you have access rights, for example in the "Documents" folder, for the example I would call this new folder *CamBamSys*, in this example its path is: ***C:\Documents\CamBamSys***
2. In CamBam, use the ***Tools/Browse System Folder*** menu to open the current system folder, then copy all the folders/files there.
3. Paste what you just copied into the new system folder (CamBamSys) you just created.

Now we are going to define this new location in the options so that CamBam knows where to find the files.

Use the ***Tools/Options*** menu, then locate the ***System Path*** property and enter the new path.



Close the options window; if ***Tools/Save settings on exit*** is checked, these new settings will be saved when CamBam is closed; if the option is not checked, use ***Tools/Save settings***.

Create a drawing template that will be used by default for new files

Some settings are not saved in the configuration file or in the ***CAMstyles***, this is the case, for example, of the ***Fast plunge height*** property of the machining folder, the definition of the ***Stock***, the ***Styles/Tools libraries*** to be used, etc..

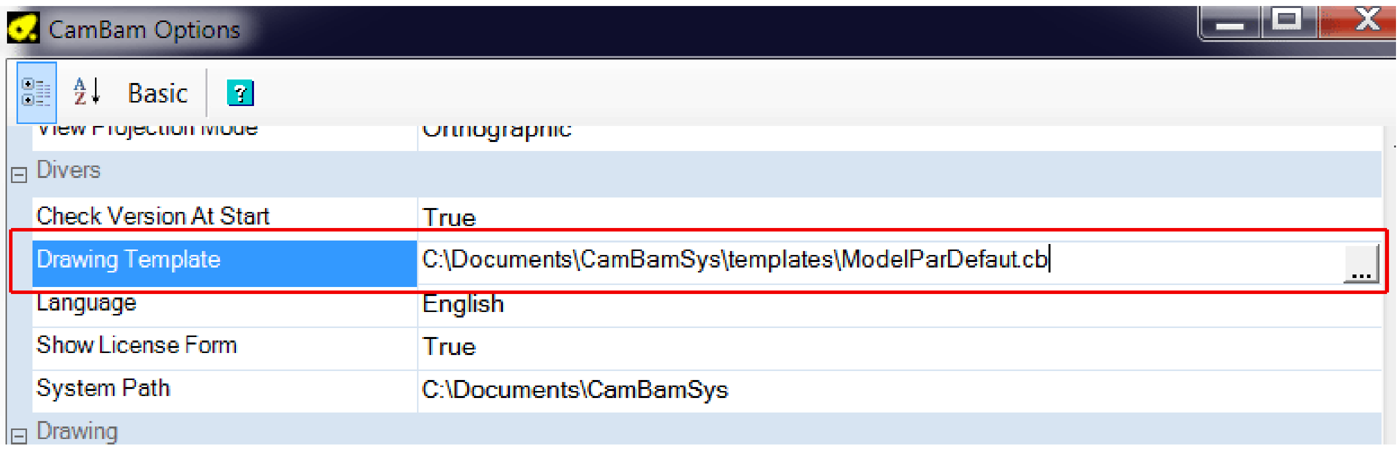
It can also be useful to have predefined layers, machining operations or even a drawing (for example a rectangle defining the machining area of your machine)

To create a template that will be used by default:

Create a new project, define the settings you want, then save this project in the ***Template*** folder of the CamBam system with a name of your choice, for example *ModelParDefaut.cb* (see above to locate the CamBam System folder).

In ***Tools/Options***, in the ***Drawing Template*** property, click the |...| which appears to the right of the property and find the template file you just created.

This file will be used as the base template each time you use the ***File/New*** menu to create a new project.



Reset CamBam to default settings.

There may be times when you need to reset CamBam to its default settings (grids, colors, etc.)

To do this, simply delete the ***CamBam1.0.config*** file located in the default system folder (*C:\ProgramData\CamBam plus 1.0*) then restart CamBam, a config file will be recreated automatically

Note: Even if you use a custom System Folder, the ***CamBam1.0.config*** file will still be in the default location and not in your custom System Folder. By default, under Windows, the *ProgramData* folder is a hidden folder, so you will have to enable the display of hidden files and folders in Windows.