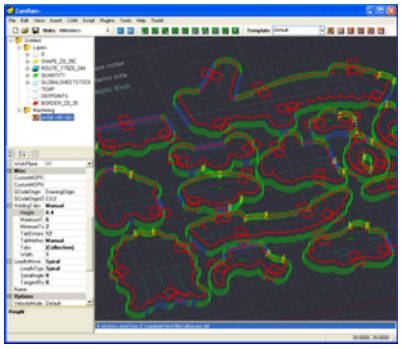
CamBam plus 0.9.8 documentation

Rev 1.17, 2013-09-25

Welcome to CamBam

CamBam is an application to create CAM files (gcode) from CAD source files or its own internal geometry editor. CamBam has many users worldwide, from CNC hobbyists to professional machinists and engineers.

CamBam currently supports the following:

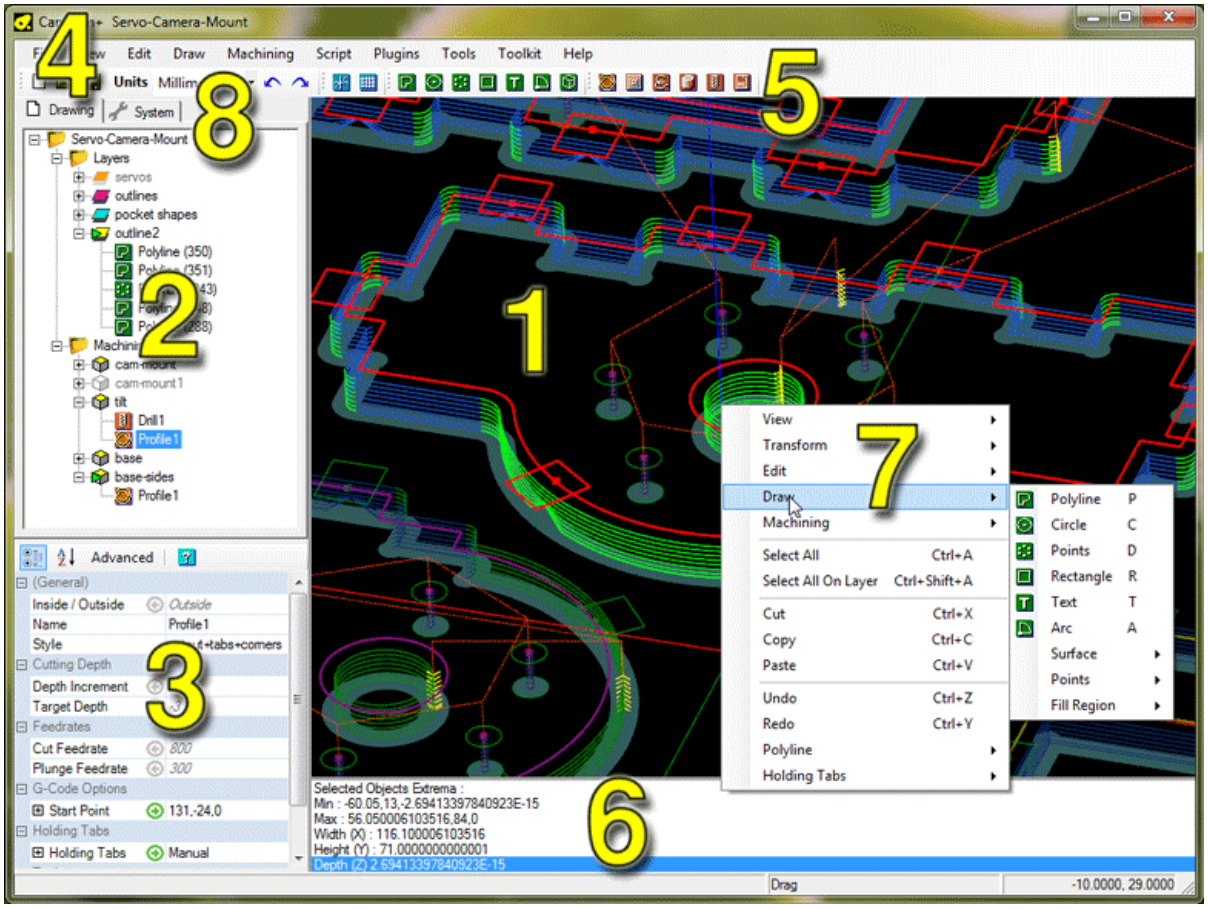


- Reading from and writing to 2D DXF files.
- 2.5D profiling machine operations with auto-tab support
- 2.5D pocketing operations with auto island detection
- Drilling (Normal,Peck,Spiral Milling and Custom Scripts)
- Engraving
- True Type Font (TTF) text manipulation and outline (glyph) extraction.
- Conversion of bitmaps to heightmaps
- 3D geometry import from STL, 3DS and RAW files
- 3D surfacing operations
- Extendable through user written plugins and scripts

Basics

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.

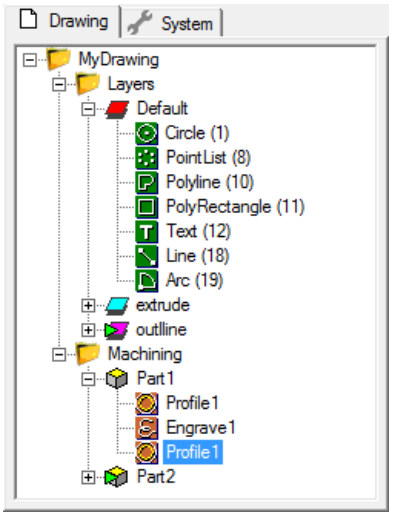
Drawing and System Tabs

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 labelled using the name of the drawing file. In the example pictured, the file is titled : 'MyDrawing'.

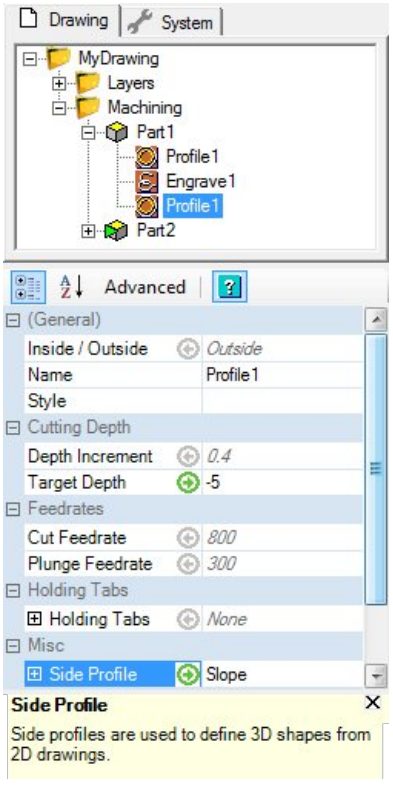
The drawing is then divided into two main sections: ***Layers*** and ***Machining***.

Layers are used to separate the drawings items into manageable sections which can be labelled, 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 customise 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.

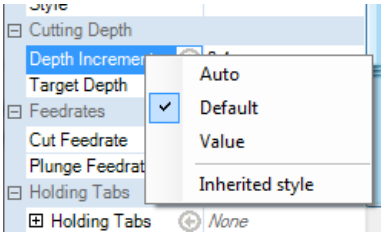
For some objects, such as machining operations, a symbol may be shown to the right of the property name. These are:

 **Auto**, indicating the value used will be automatically calculated.

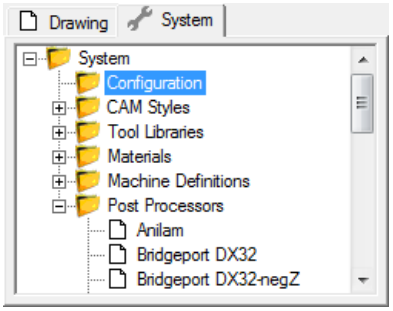
 an explicit **Value** has been entered.

 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.

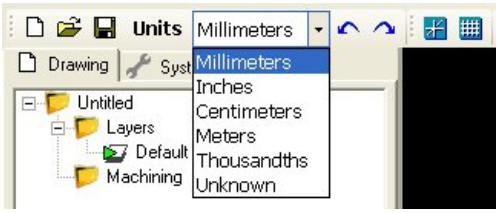
 ***3D Surfacing*** - 3D Meshes can profiled using multi pass roughing or finishing profiles.
Front back and molds are also supported.

 ***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.

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)
- GCode files (*.tap,*.nc, etc)
- Gerber file (*.gbr)

Unrecognised 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.

View Menu

Zoom

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

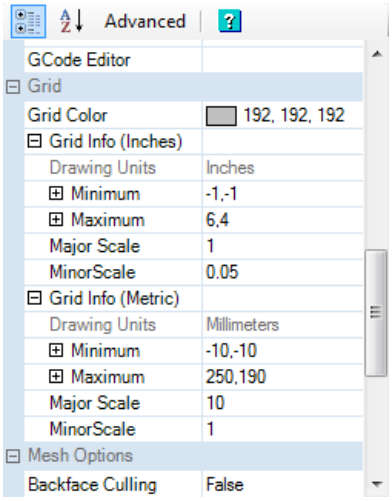
- ***Reset*** - reverts to a known position. (XY plane) and performs a ***Zoom To Fit***.
  Equivalent to **`ALT`** + double click.
  If ***Rotation Mode*** *Left_Middle* is active, the same operation can be done by a double left click while holding the middle button pressed.
  If ***Rotation Mode*** *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.

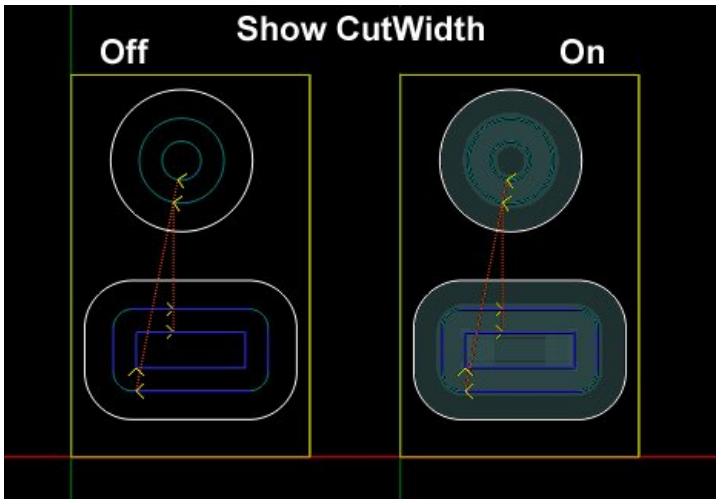


Display Setup

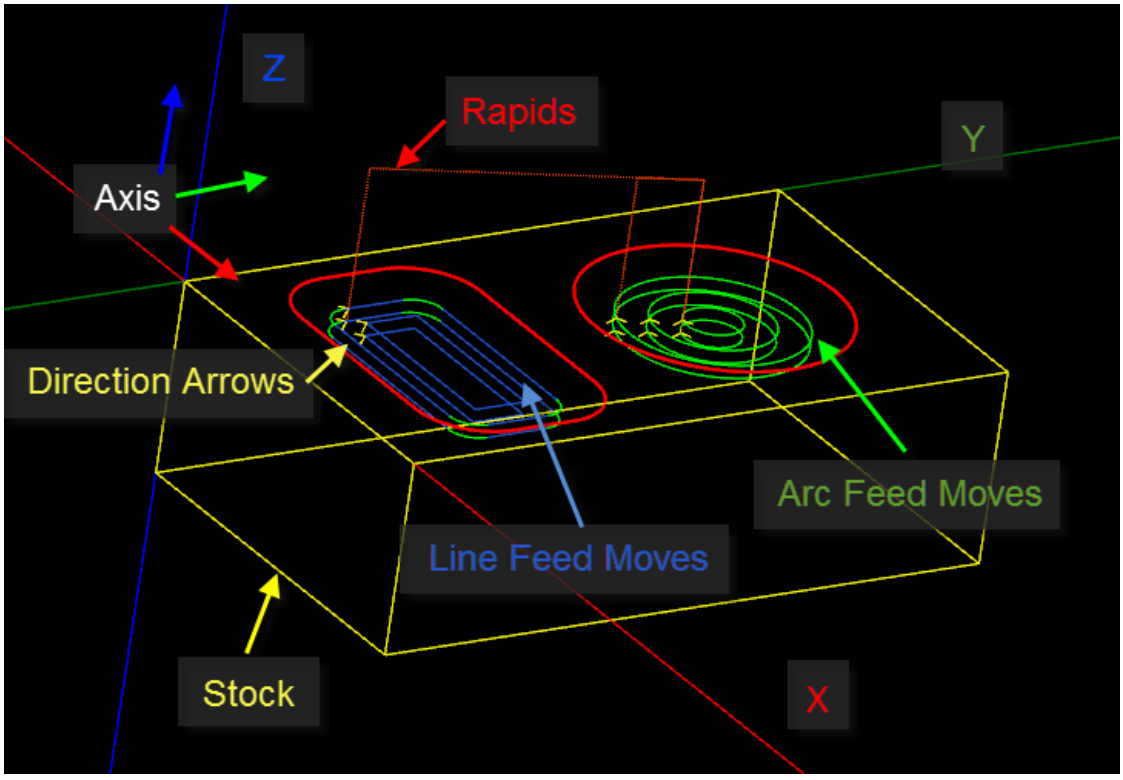
The following options enable or disable the display of graphical aids.

- ***Show toolpaths*** - Enable / disable the display of lines representing the toolpaths.

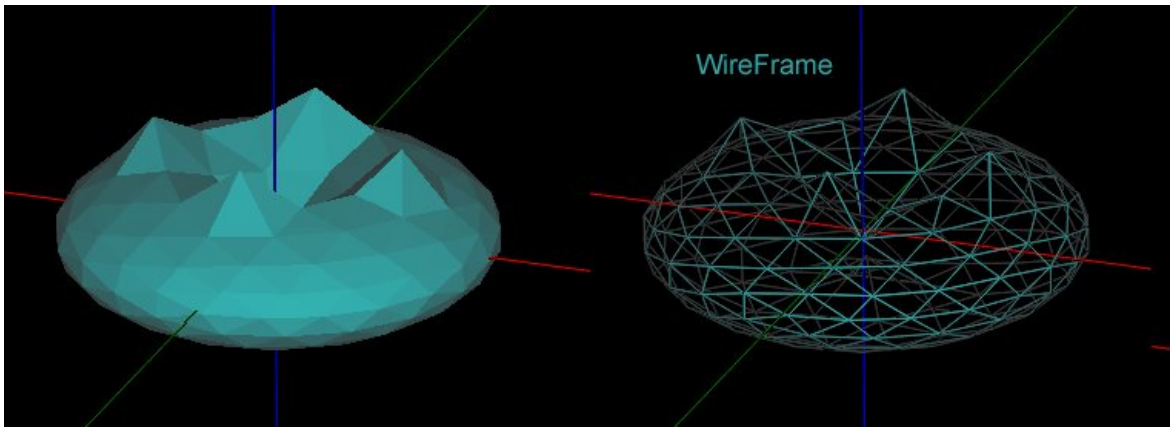
- ***Show cut widths*** - Enable / disable the display of a shaded area depicting the width of cuts along the toolpaths based on the specified tool diameters.



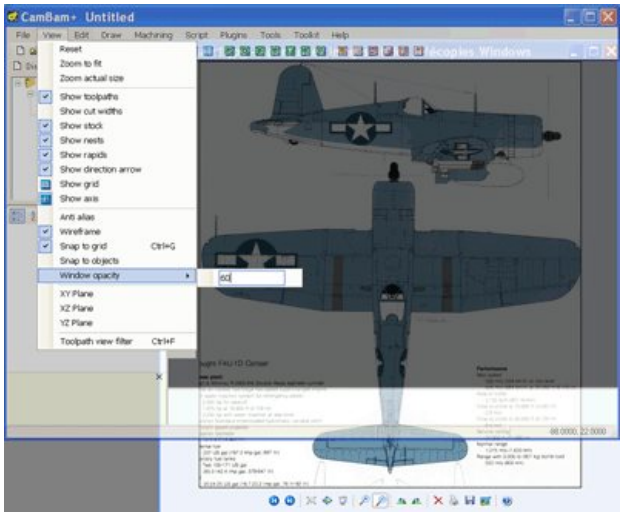
- ***Show stock*** - Enables / disables the display of the 3D representation of the block of material to be machined.
- ***Show nests*** - Enables / disables the display of arrays of machining operations, defined in the Part ***Nest*** properties.
- ***Show rapids*** - Enable / disable the display of 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.

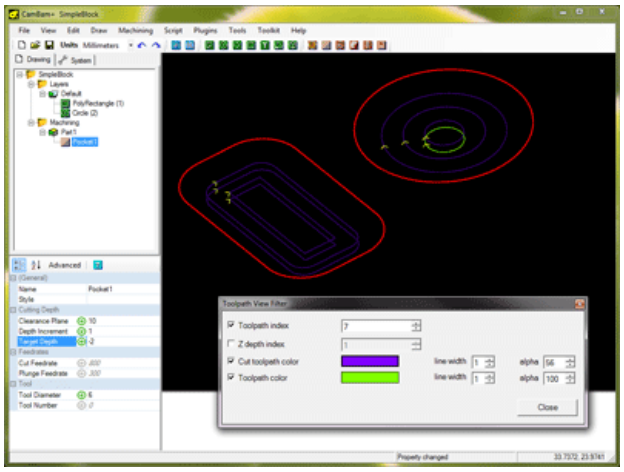


- ***Snap to grid*** - Enable / disable snap to grid.
- ***Snap to object*** - Enable / disable snapping to other drawing objects.
- ***Windows opacity*** - A value between 0 and 100% (opaque) which allows tracing over reference drawings in windows behind the CamBam drawing.



- ***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.

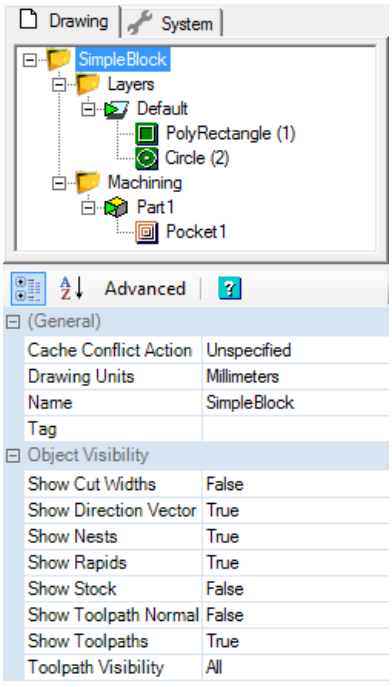
Click on the colored rectangles to change the display color of the toolpaths.

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

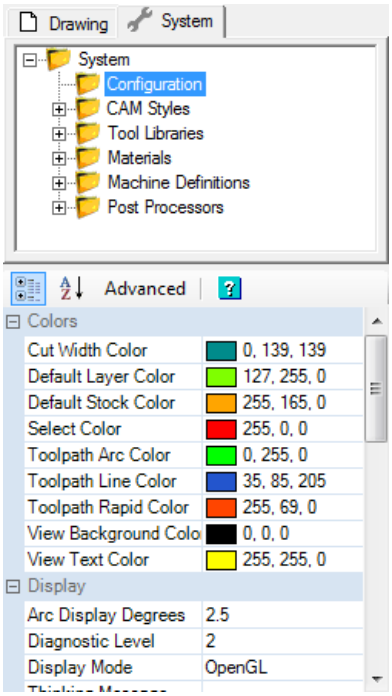
The ***Toolpath View Filter*** window can be kept open while manipulating the drawing, such zooming and panning the display.

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 section.



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***

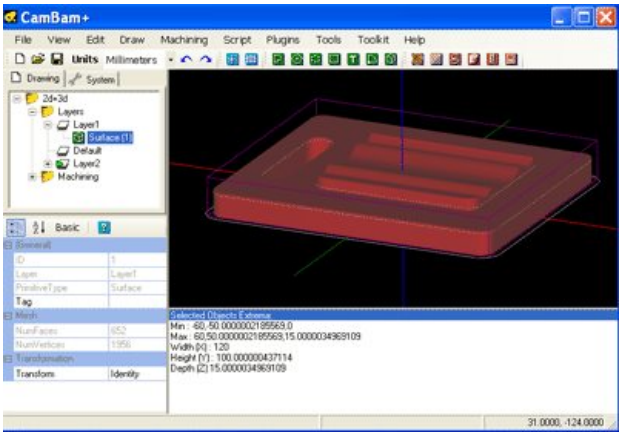
  Determines whether there are any newer CamBam updates available from the CamBam website.

- ***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

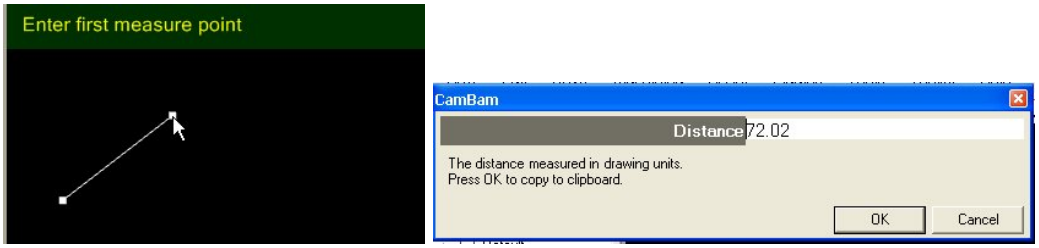
Max: maximum coordinates of the object in X, Y and Z are separated by a comma.
Example: X=60, Y=50.000..., Z=15.000...

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.

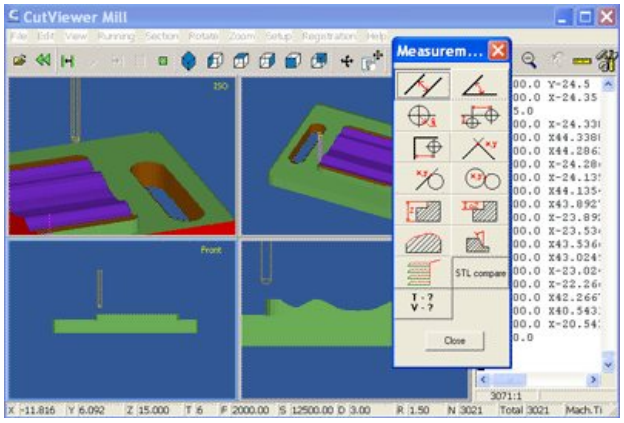


- ***Reload post processors***

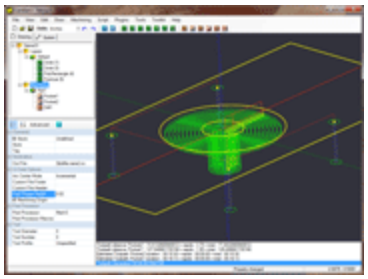
  Reloads all the post processor definitions from disk. This may be needed if a post processor has been modified from another instance of CamBam.

- ***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.



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.

Download the files used in this tutorial

Step 1 - Create and set up a new drawing.

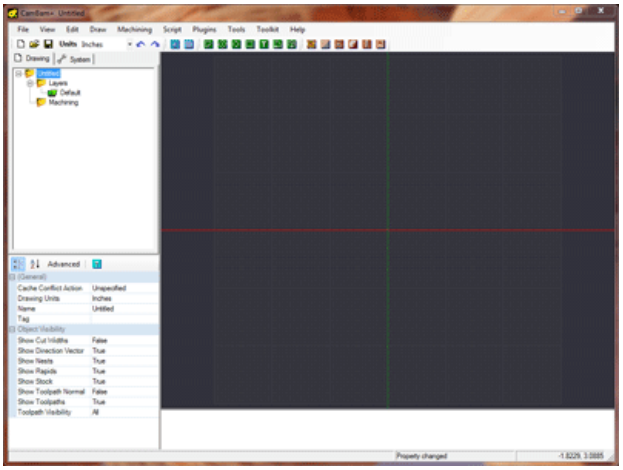
Start with a empty drawing, use ***File - New*** or the new file icon from the toolbar.

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).

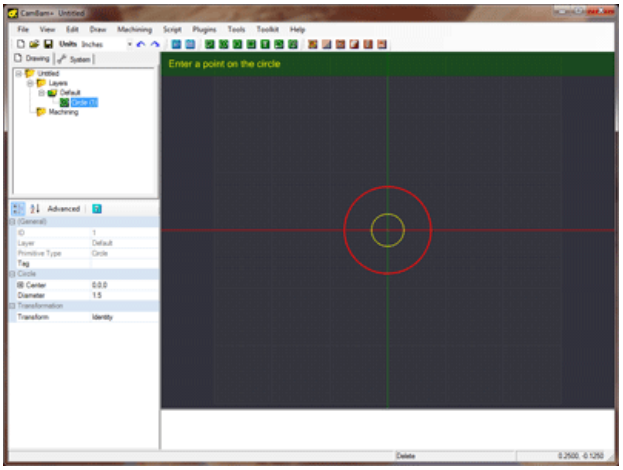
Select the circle drawing tool button from the tool bar. A prompt will be displayed at the top of the drawing window to guide the current drawing operation.

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.

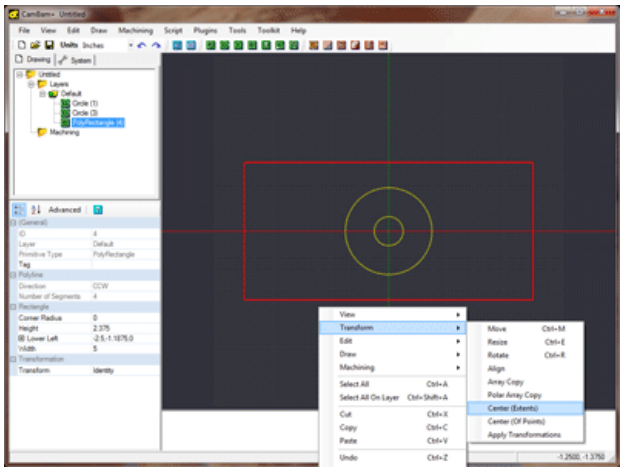
Select the rectangle drawing tool button from the tool bar. Once again, a prompt will be displayed at the top of the drawing window to guide the current drawing operation.

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

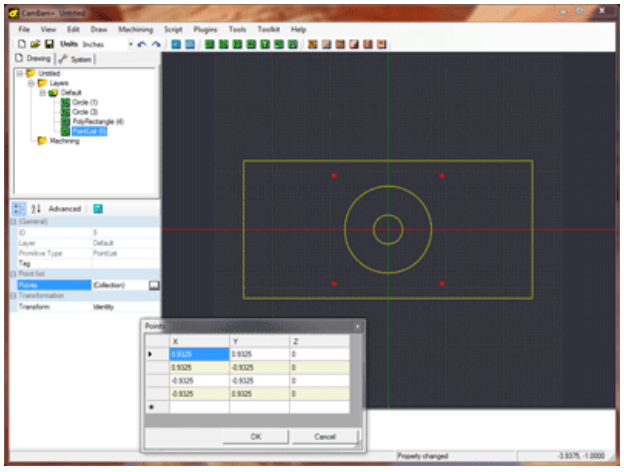
The Nema 23 stepper motor has 4 bolt holes arranged in a square 1.856" (47.14mm) apart. We will be adding a drilling machining operation later to generate these holes so to prepare for that we need to insert 4 center points at the hole centers.

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

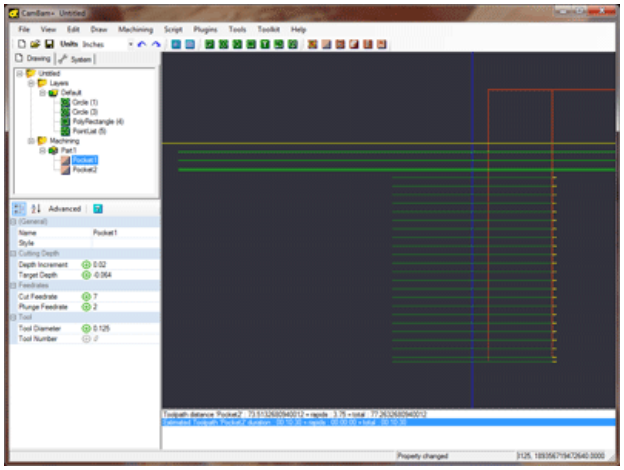
Change the pocket machine operation's properties to the following:

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 ***Machinining - 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***.

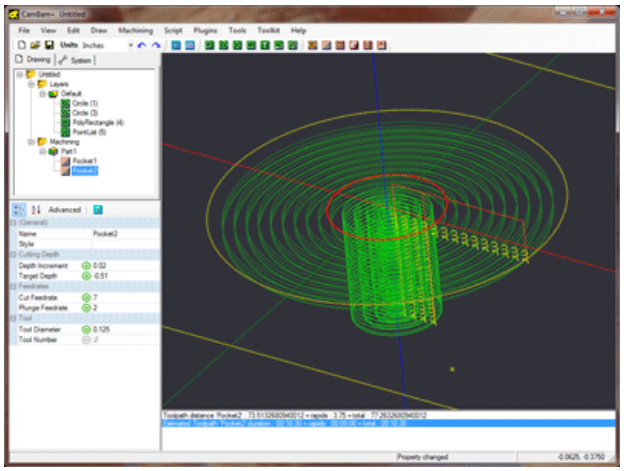


To rotate the 3D drawing view, hold the **ALT** key then click and drag on the drawing. To reset the view, hold the **ALT** key then double click the drawing. Other rotation modes can be set in the ***Rotation Mode*** setting of the sytem

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.

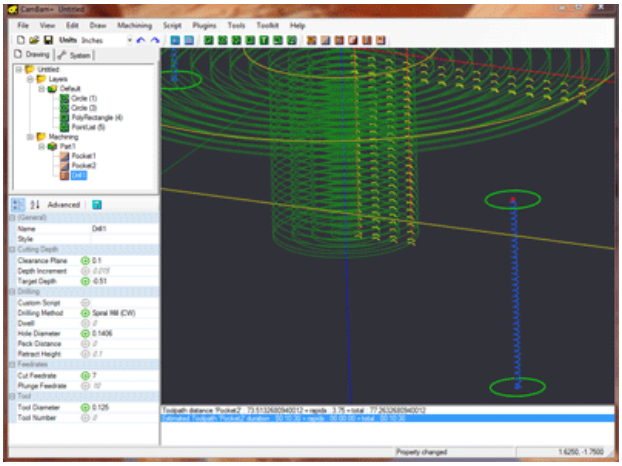
Custom Scripts which allow snippets of gcode to be inserted at each drill point.

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 wiill 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***.

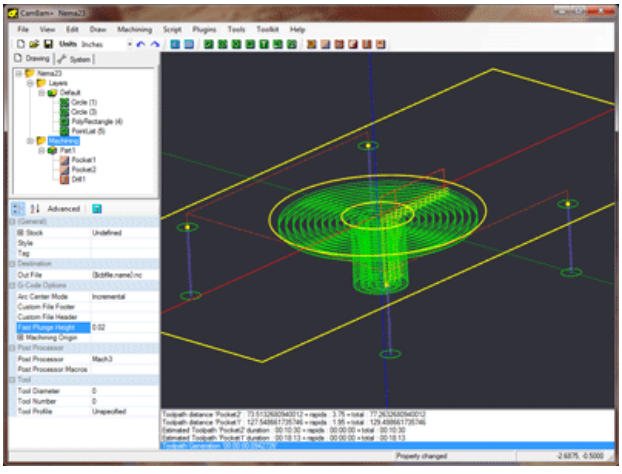
CamBam will then prompt for the location of the gcode file to produce. If the drawing file has been saved, the default file will be in the same folder as the drawing file with a .nc extension.

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 EMC2, 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+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 object
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
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
Page Up or **Num Pad -**	Zoom out
Page Down or **Num Pad +**	Zoom in
Home	Reset view

Machining (CAM)

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.
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

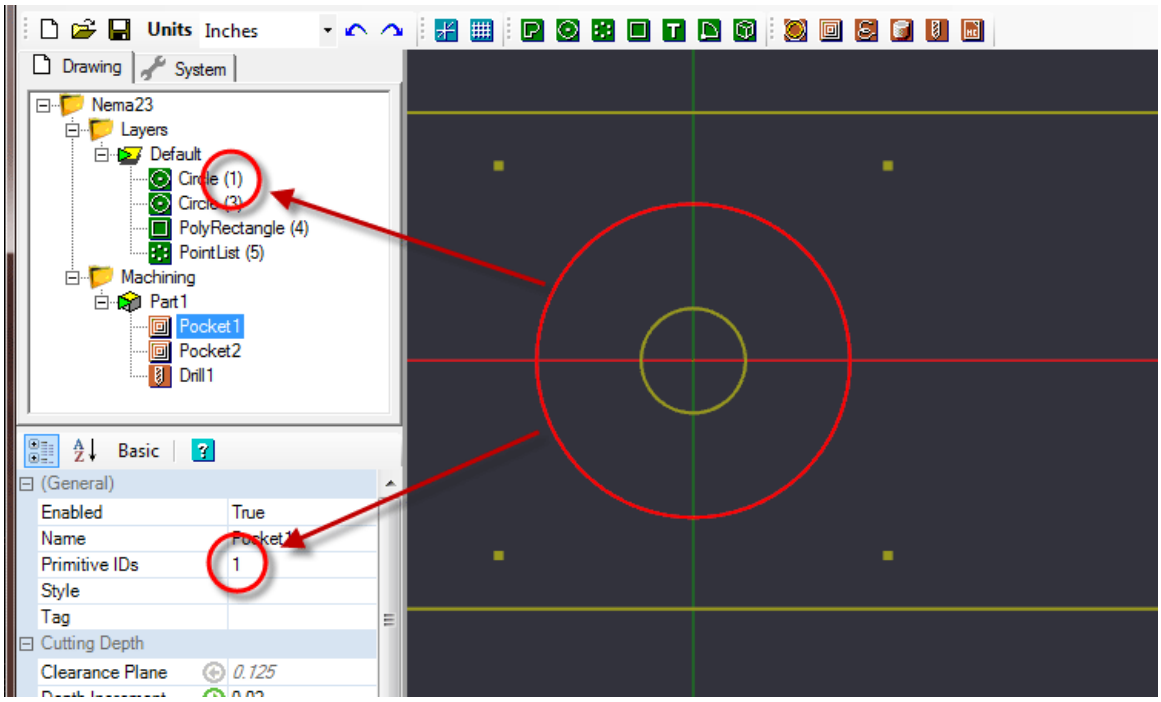
To add a machining operation, select one or more drawing objects (2D or 3D depending on the type of operation to be inserted), then click on the toolbar icon that corresponds to the desired operation, or choose from the ***Machining*** menu.

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.

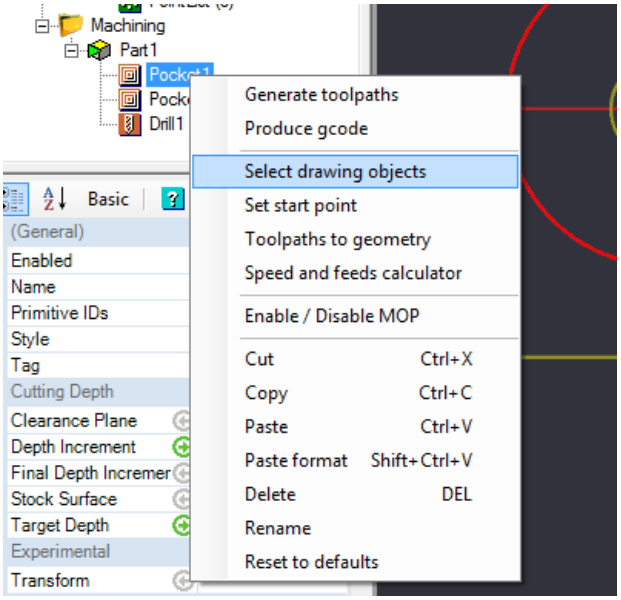
Changing machining operation source objects.

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.



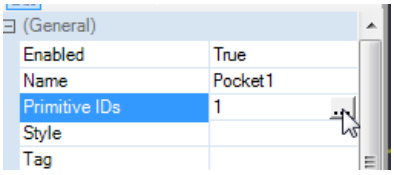
The drawing window displays the objects already assigned to the operation in red. All the object selection methods can be used to alter the current selection. Holding the **`Ctrl`** key and left clicking objects will add and remove objects from the selection.

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.

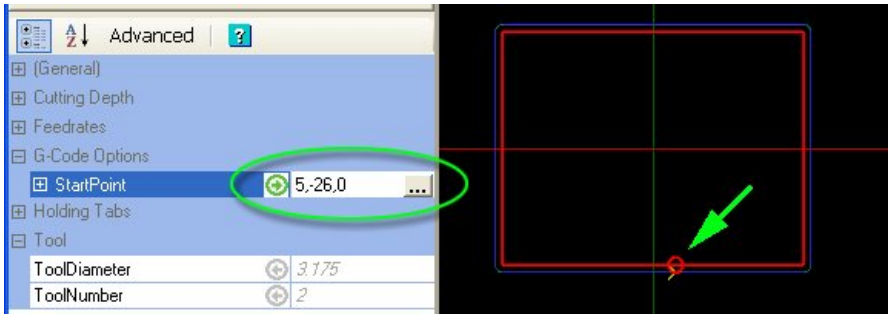
Managing machining operations

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.

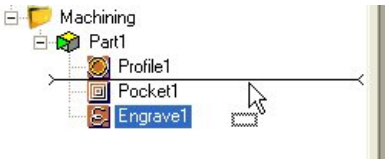
Set start point: Sets the starting point of a machining operation by clicking on the drawing at the desired start point. The operation will start at the closest point possible to the select start point.

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.

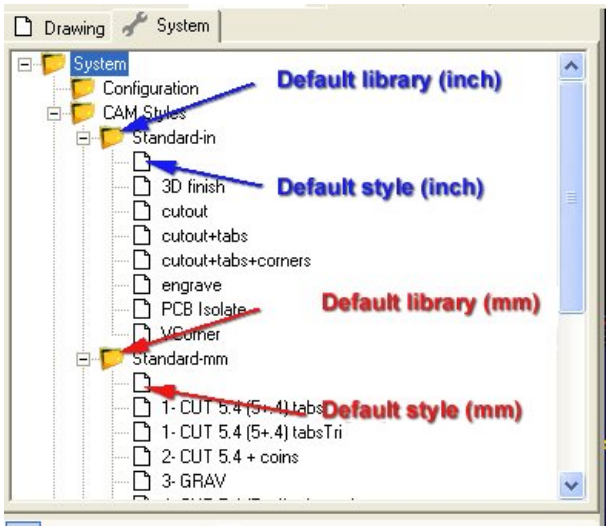
Paste format: This function copies most of the properties of a machining operation that has been copied to the clipboard using the ***Copy*** command, into the selected target machining operation. The target's name and source drawing objects are left intact.

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 **[New 0.9.8]**	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 **[New! 0.9.8]**	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	This property is currently used only by the ***Lathe*** machining operation.
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 **[New! 0.9.8]**	Select a CAM Style for this machining operation. All default parameters will be inherited from this style.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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 **[New! 0.9.8]**	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 **[New! 0.9.8]**	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	This property is currently used only by the ***Lathe*** machining operation.
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 **[New! 0.9.8]**	Select a CAM Style for this machining operation. All default parameters will be inherited from this style.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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 **[New! 0.9.8]** $n - tool number $p - Dwell $q - Peck distance $r - Retract height **[New! 0.9.8]** $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 **[New! 0.9.8]**	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 **[New! 0.9.8]**	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 **[New! 0.9.8]**	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 **[New! 0.9.8]**	For peck canned cycles, retract to this value after each peck.
Roughing / Finishing	This property is currently used only by the ***Lathe*** machining operation.
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 **[New! 0.9.8]**	Select a CAM Style for this machining operation. All default parameters will be inherited from this style.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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 **[New! 0.9.8]**	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	This property is currently used only by the ***Lathe*** machining operation.
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 **[New! 0.9.8]**	Select a CAM Style for this machining operation. All default parameters will be inherited from this style.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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 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 **[New! 0.9.8N]**	The tolerance used when automatic arc fitting is applied. Zero will use an automatically calculated value.
Auto Arc Fitting **[New! 0.9.8N]**	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 **[New! 0.9.8]**	A 2D Point, used with ***Clip Area Min*** to define a clipping area.
Clip Area Min **[New! 0.9.8]**	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 **[New! 0.9.8]**	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	When *Waterline Rough* 3D profile method is selected, this option controls the pattern used to fill the pockets at each waterline layer. 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	This property is currently used only by the ***Lathe*** machining operation.
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 **[New! 0.9.8N]**	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. For horizontal and vertical 3D profile methods, this is the distance between each scan line. For waterline roughing, this is the distance between fill offset lines. For waterline finishing, this value is not used.
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 **[New! 0.9.8]**	Select a CAM Style for this machining operation. All default parameters will be inherited from this style.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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.

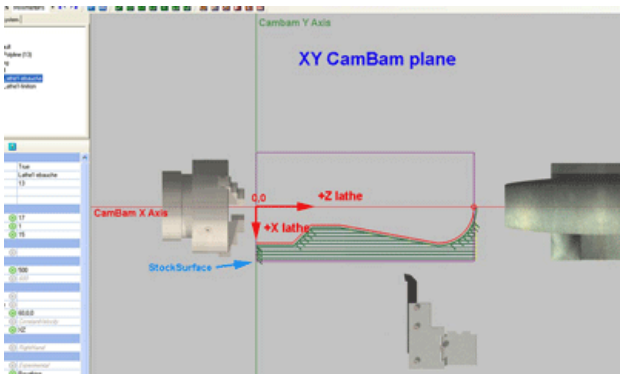
The Lathe machining operation has been provided as a plugin. In this way the plugin can be developed and updated independently of the main CamBam application. It is also a demonstration of the ability to extend CamBam's machining capability using user written plugins.

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

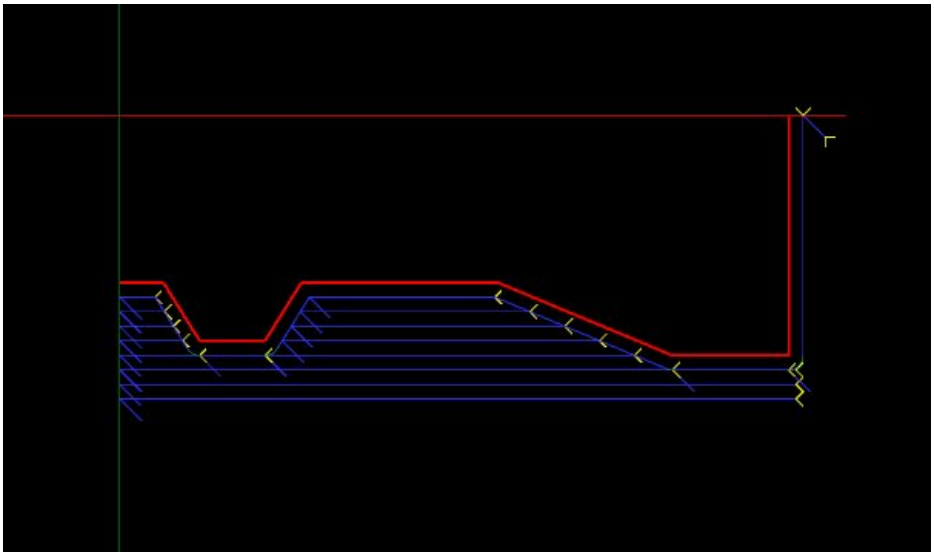


A lathe profile can be generated from a 2D line representing the shape to machine. The shape should be drawn so that:
The lathe **+X** axis is drawn in the **-Y** direction and
The lathe **+Z** axis is drawn in the **+X** direction.
This is so that the drawing will appear in the same orientation as when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

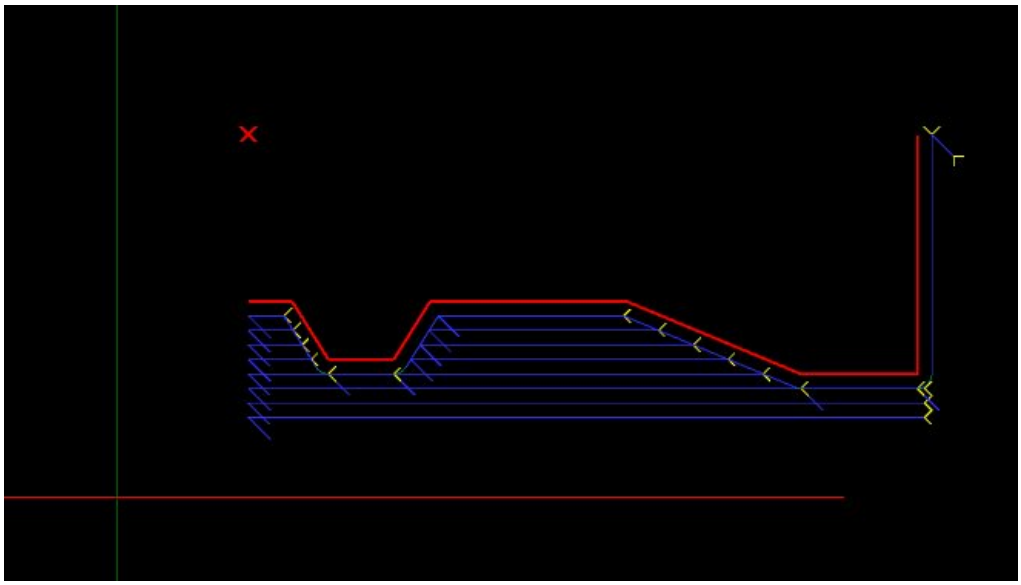
Only draw the profile line to be cut. Do not draw closed polylines, mirrored lines on the opposite side of the turning axis or lines along the turning axis as the lathe operation will try to cut these as well which will cause problems.

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.



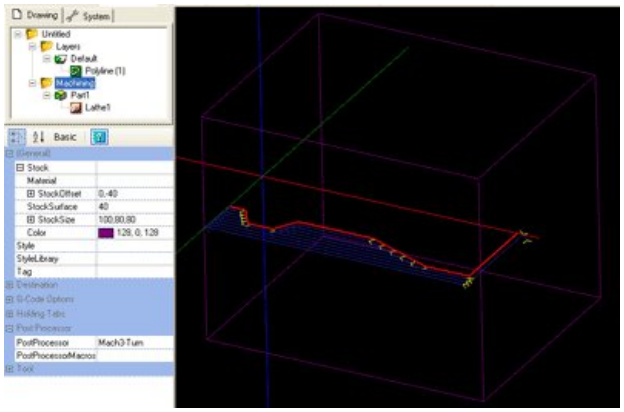
You can set the machine zero by setting MachiningOrigin property of the machining or part objects. Click the  button to the right of the MachiningOrigin property to select the machine zero point on the drawing.

Stock Object

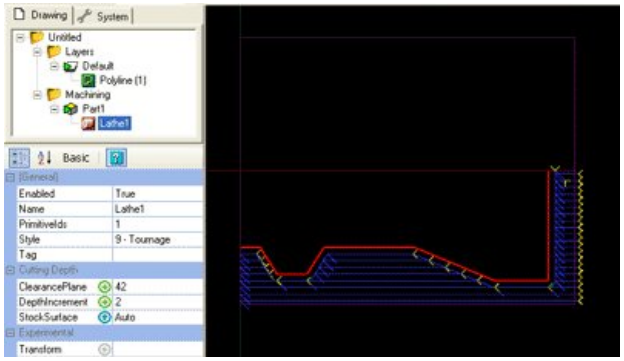
The lathe operation can use information from the stock object if one is defined, to determine properties such as stock surface and the machining envelope.

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

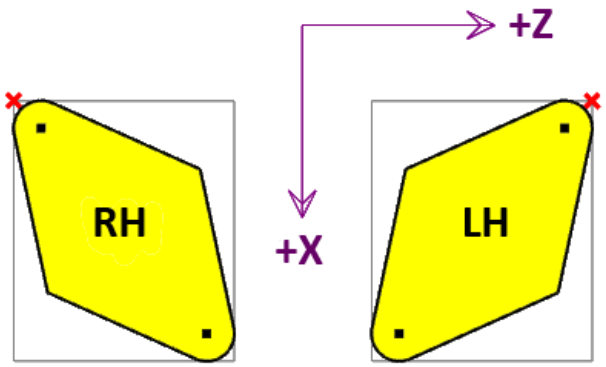
Clearance Plane	The safe **X** lathe coordinate to avoid any stock. The clearance plane value should always be expressed as a **radius**.
Custom MOP Footer	A multi-line gcode script that will be inserted into the gcode post after the current machining operation.
Custom MOP Header	A multi-line gcode script 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 **[New! 0.9.8]**	Select a CAM Style for this machining operation. All default parameters will be inherited from this style.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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 EMC2)
X Mode Radius	Code to use to set X radius mode (eg G8 for EMC2)
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}
```

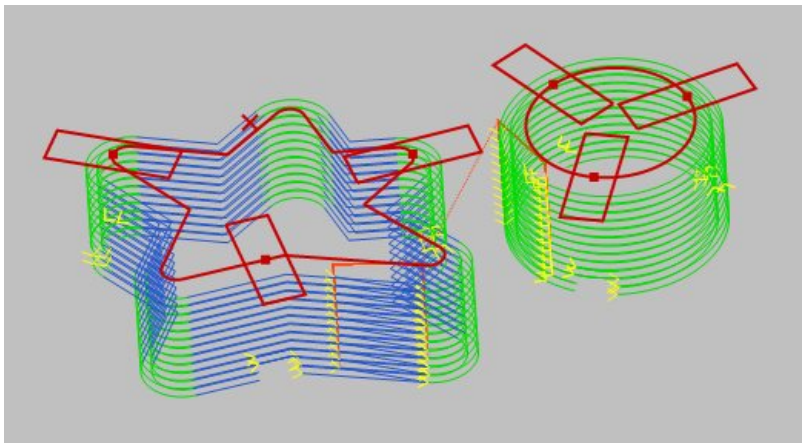
Creating GCode

The basic work flow for creating Gcode files is:

- 1. Create or Import drawing objects
- 2. Select drawing objects and define Machining Operations
- 3. Generate Toolpaths and visually inspect
- 4. 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.



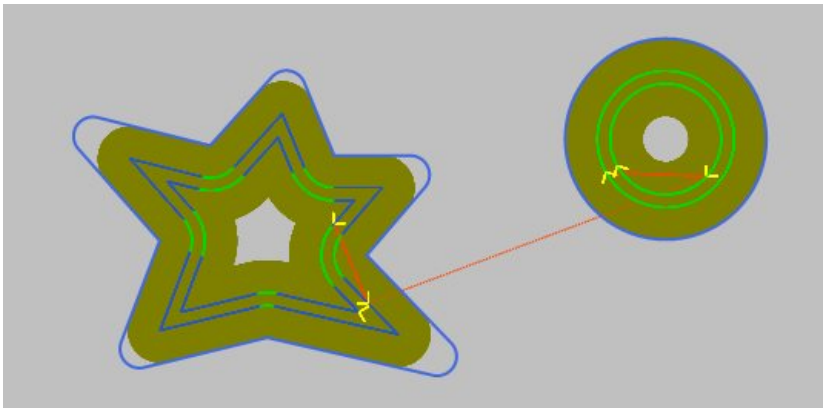
CamBam provides a 3D drawing view. Rotate the view (using **ALT**+*Drag*) to see more detail of the toolpaths including the different depth levels.

The toolpaths indicate the path that will be travelled 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 **[0.9.8] moved from machining to first item in the drawing tree.**	*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 **[0.9.8] moved from machining to first item in the drawing tree.**	*True* | *False*. Controls the visibility of a small arrow at the start point of each toolpath that indications the direction of machining.

Show Rapids [0.9.81 moved from machining to first item in the drawing tree.	*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 [0.9.81 moved from machining to first item in the drawing tree.	*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-CV* 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.>/p>
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 **[0.9.81 moved from machining to first item in the drawing tree.**	*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.

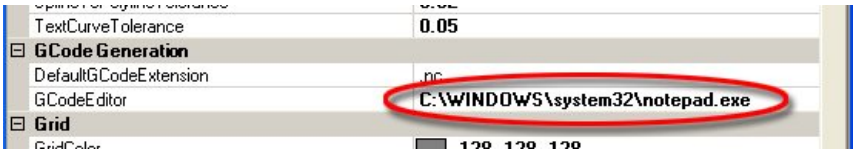
To invoke the gcode editor, use the ***Machining - Edit gcode*** menu option, or from the context menu presented when right clicking the machining folder in the drawing tree.

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.

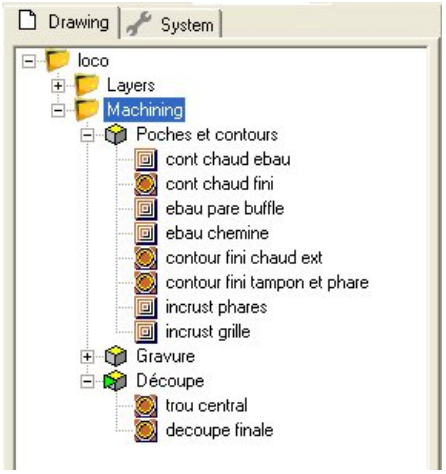


Gcode files can also be view and their toolpaths displayed (or back plotted) using the ***NC File*** operation  . Double clicking the ***NC File*** object in the drawing tree will invoke the gcode editor on the source gcode of the ***NC File*** operation.

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.

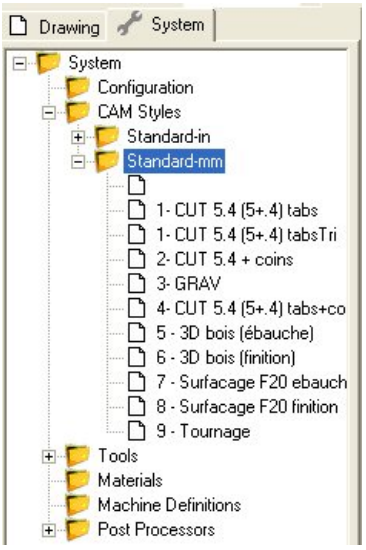
Properties

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*. **New [0.9.8f]** ***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.

	New [0.9.8f] ***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. **New [0.9.8f]** ***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. **New [0.9.8f]** ***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.
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-CV* 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.>/p>
Style Library	This property is used to locate the style definitions used in the Part or machining operations.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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

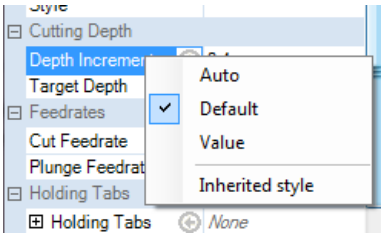
Machining operation and Style properties can have multiple states, indicated by the icon to the right of the property name.

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.

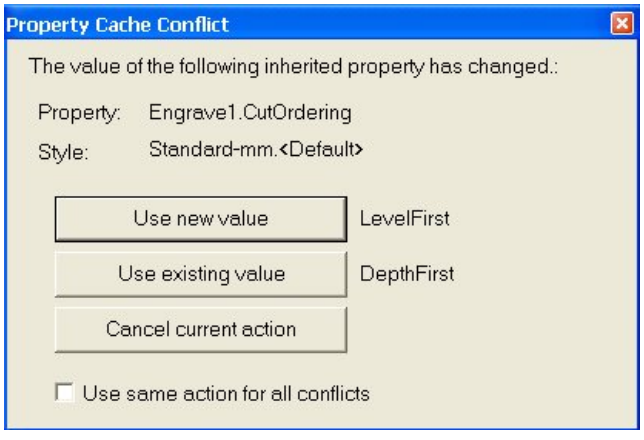
Value - The property value has been entered explicitly. In this way, the property is overridden from the value stored in its parent style.

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 cancelled.

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.

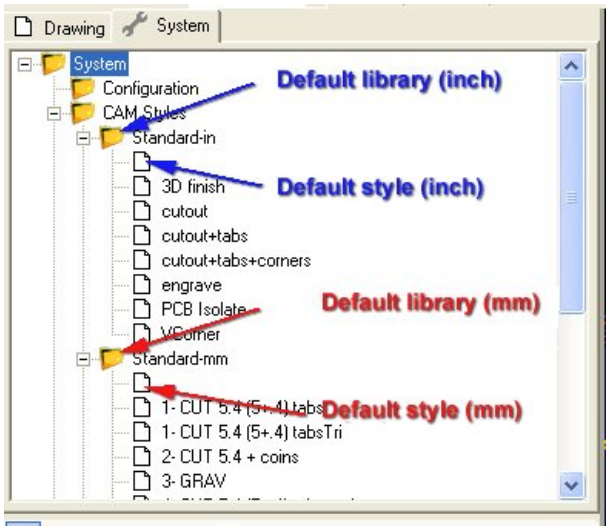
Machine operation, Part and Machining CAM styles

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:

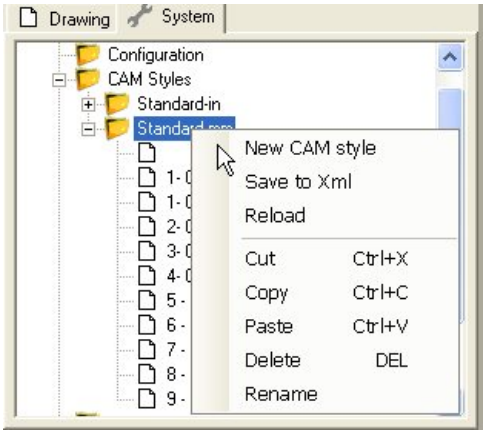
{$Material} This will be expanded to the name of the material used in the stock object.

{$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}

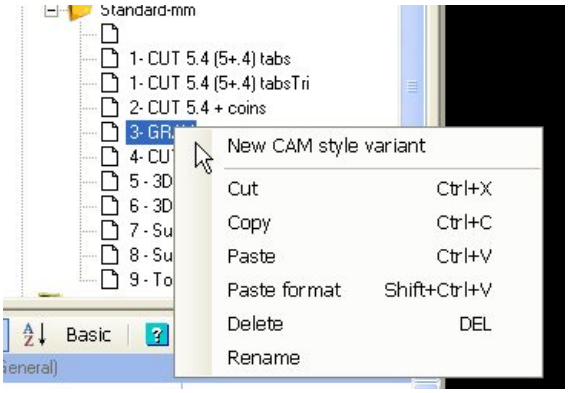
Styles and style libraries can be cut, copied, pasted, deleted and renamed within the System tree. A right click context menu gives access to many of these commands.



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.

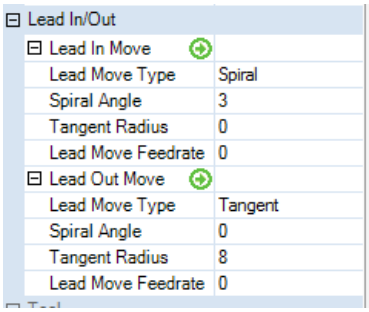
The context menu shown when right clicking a style also contains a ***New CAM style variant*** option. This will create a new style that inherits its default parameters from the selected 'parent' style.



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

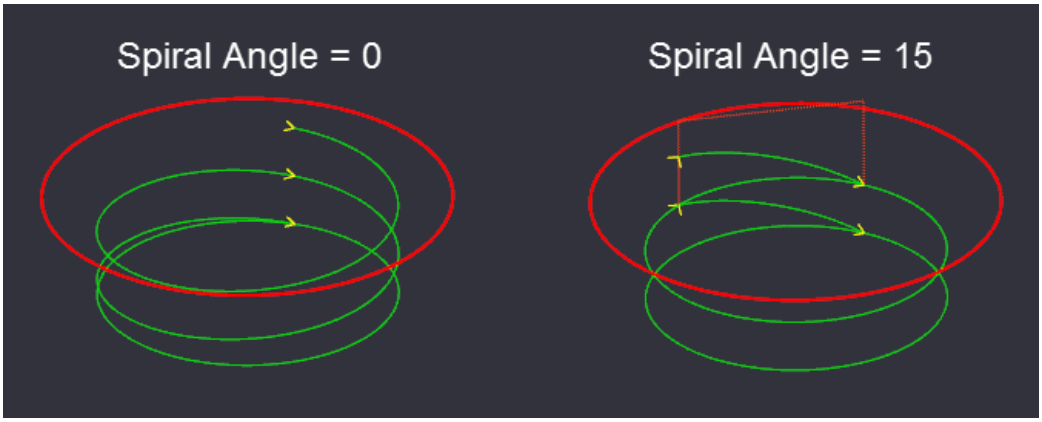
The entry move will follow the path of the underlying toolpath in X and Y, while decreasing the cutter Z from the previous stock level, down to the next target depth.

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.

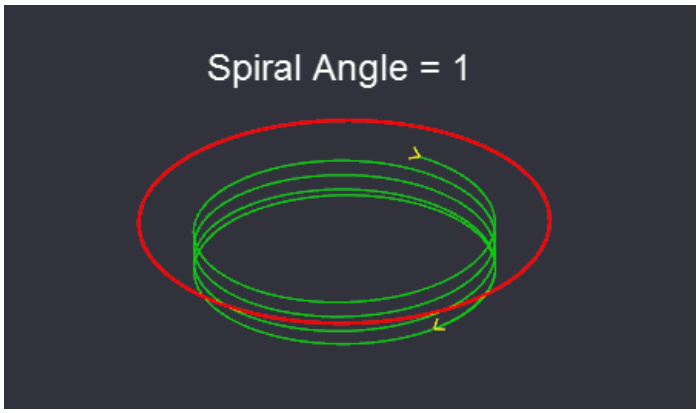
> **Hint:** The plunge down to each depth increment can slow machining times considerably. To reduce this, a ***Fast Plunge Height*** value can be set in the ***Machining*** options. This allows a rapid move to be used down to the next cut level.

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.

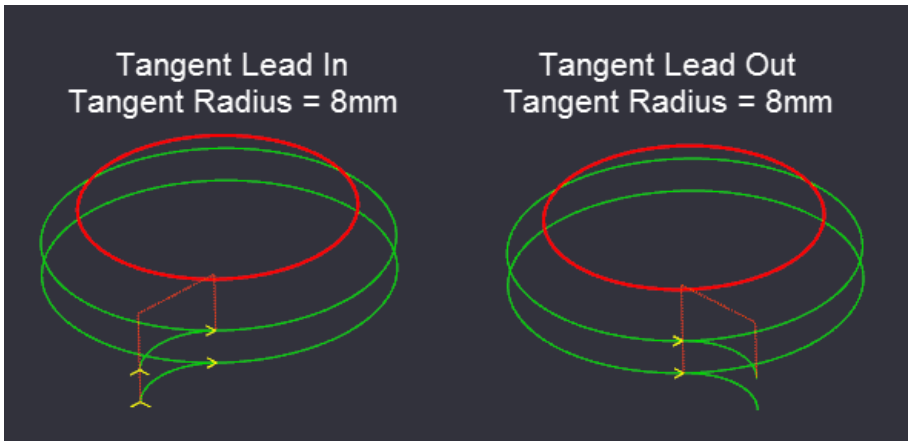


Tangent

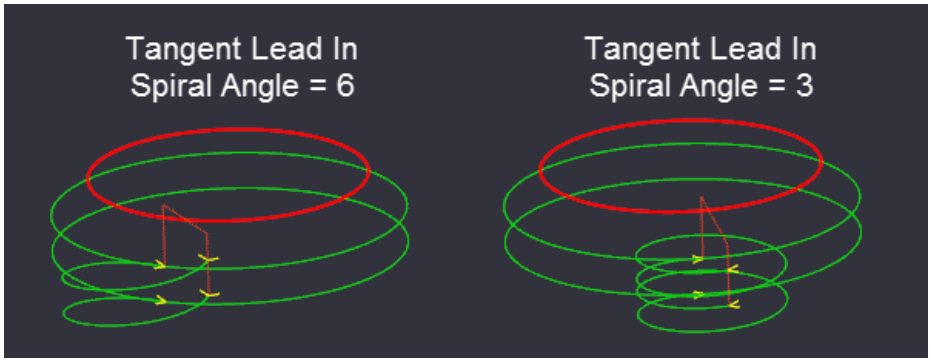
The tangent lead move will use a circular arc move to enter or exit the stock, meeting the target toolpath start point at a tangent.

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

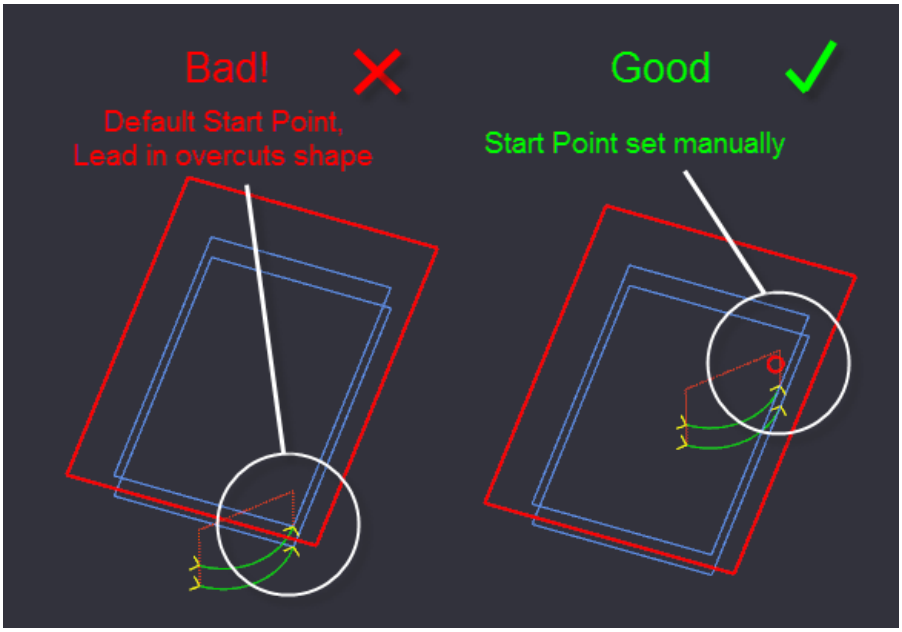
Tangent moves are particularly useful for lead out moves, to avoid tool marks as the cutter moves away from the stock.



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.



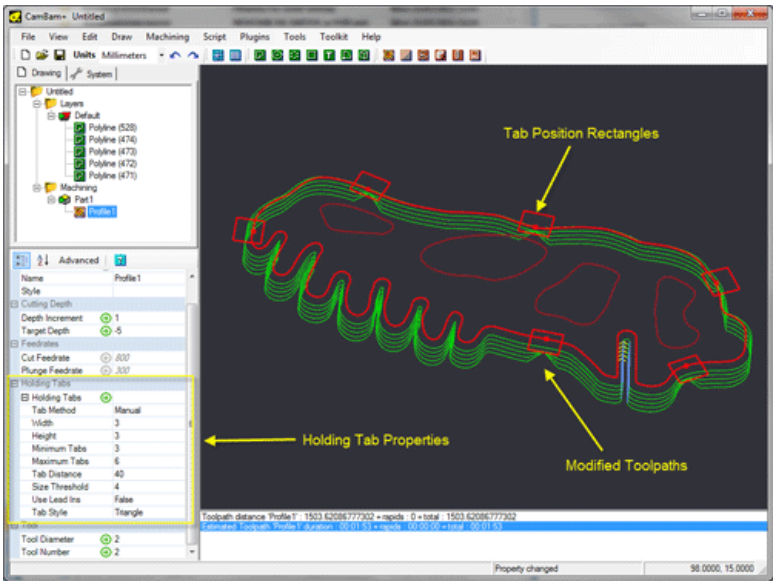
In some cases, such as an inside profile cut with internal corners, the default toolpath start point may lead to problems when using tangent lead moves. In these cases, the machining start point should be modified to move it to a more sensible location, away from inside corners.



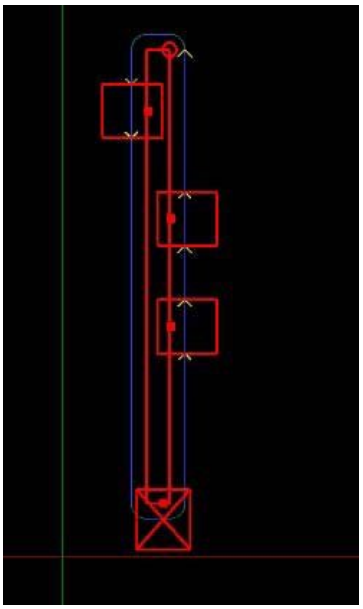
Holding Tabs

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.

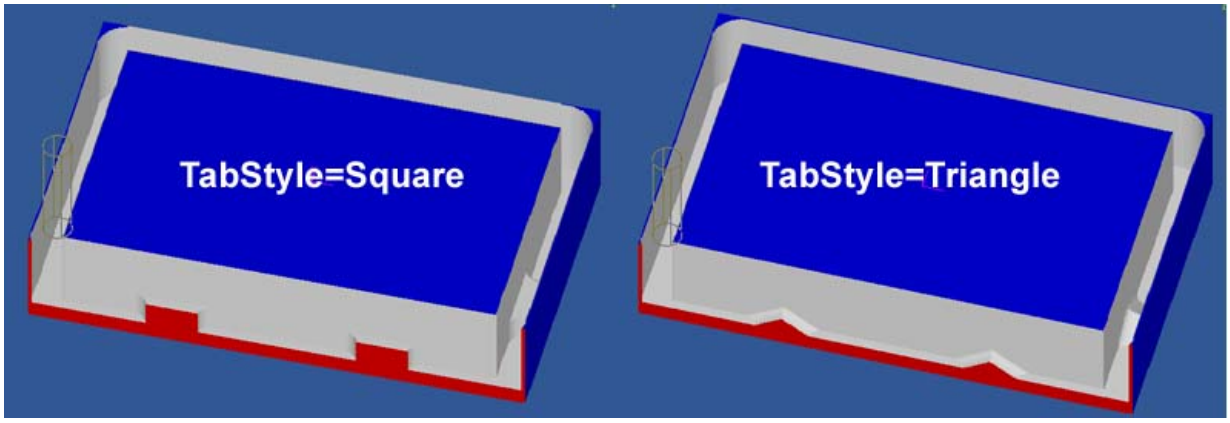
If ***Tab Distance*** is set to 0, the minimum number of tabs is always presumed.

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.



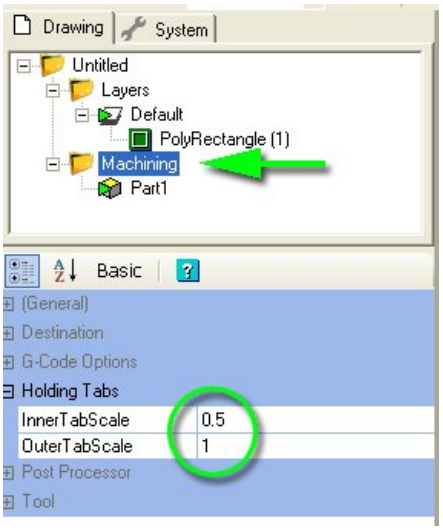
Properties

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 **[New! 0.9.8]**	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 modied or set manually. **[New! 0.9.8N]** *Automatic (Outer | Inner)* - Similar to *Automatic*, except tabs will only be added to the outside (or inside) shapes of regions.

Tab Style **[New! 0.9.8]**	*Square* - Square cross section tabs. *Triangle* - Triangle cross section tabs. **[New! 0.9.8N]** *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.

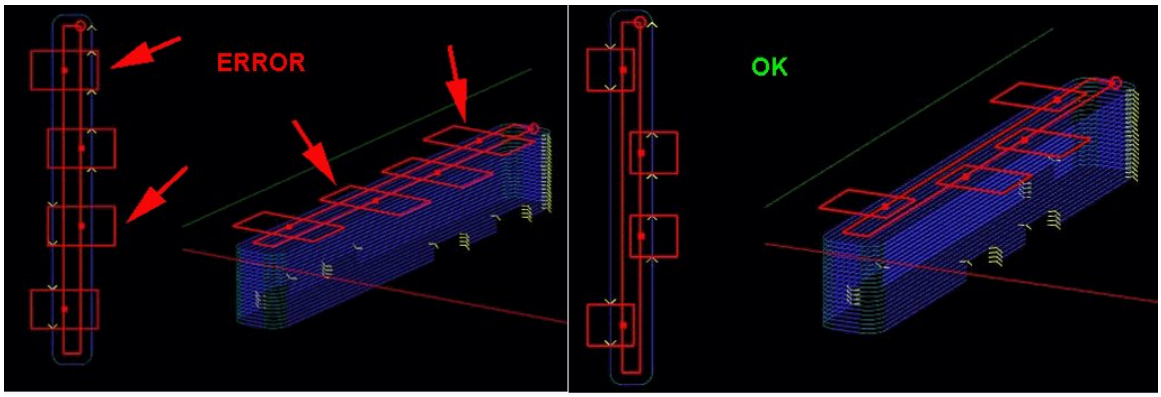
Advanced Settings

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

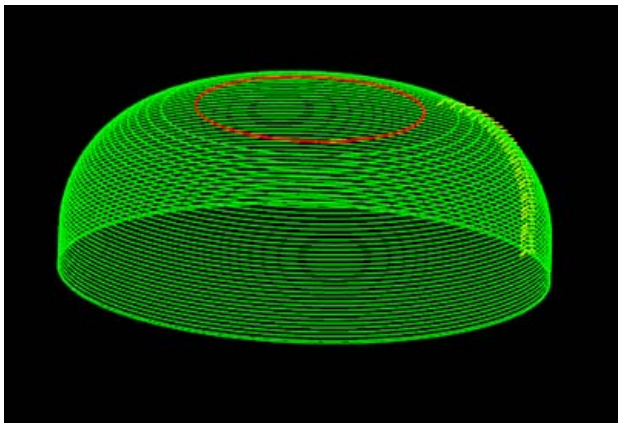
Side profiles are a method of producing 3D contours from 2D shapes, by creating radii and slopes. Side profiles are created by manipulating the ***Side Profile*** composite property of the ***2D Profile*** machining operation.

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

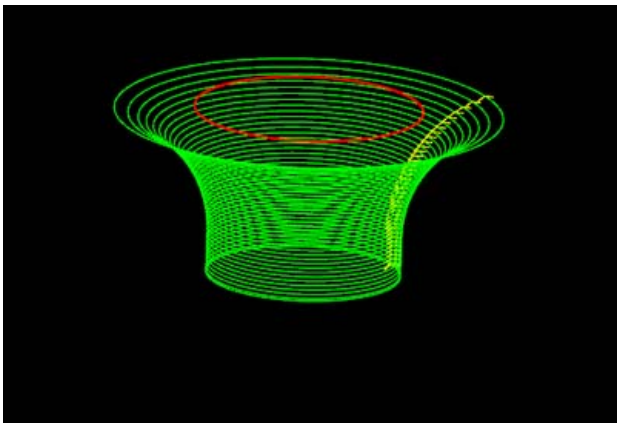
Properties

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.

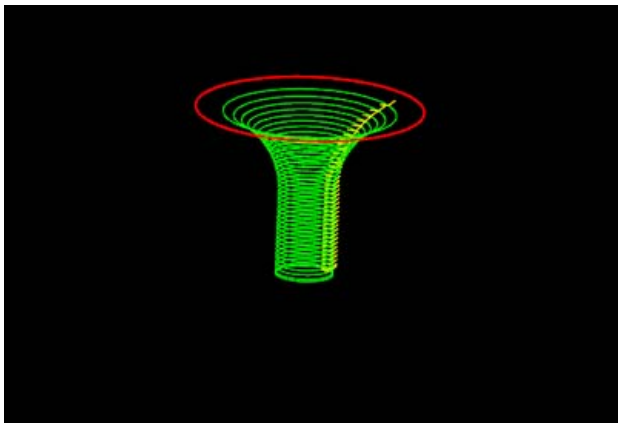
The sign of the ***Value*** parameter is significant and reversing the sign will result in different effects. Below are some examples of various combinations of side profile methods, value signs and profile inside/outside settings. These images were created from the **side profiles.cb** sample file.



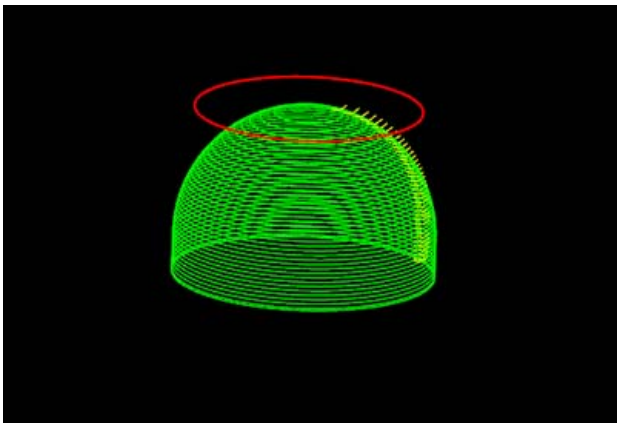
Method=CovexRadius, Value=+Ve, Profile=Outside



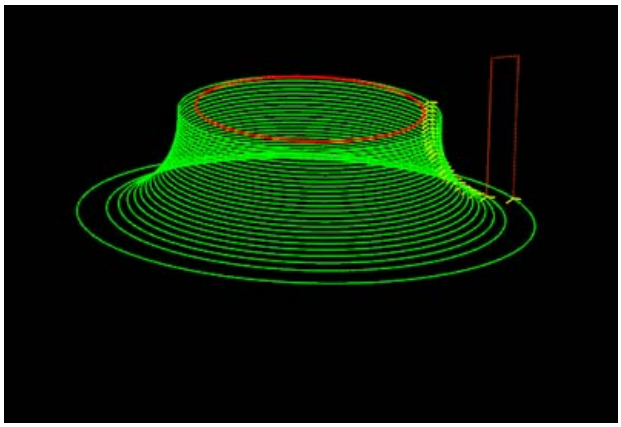
Method=CovexRadius, Value=-Ve, Profile=Outside



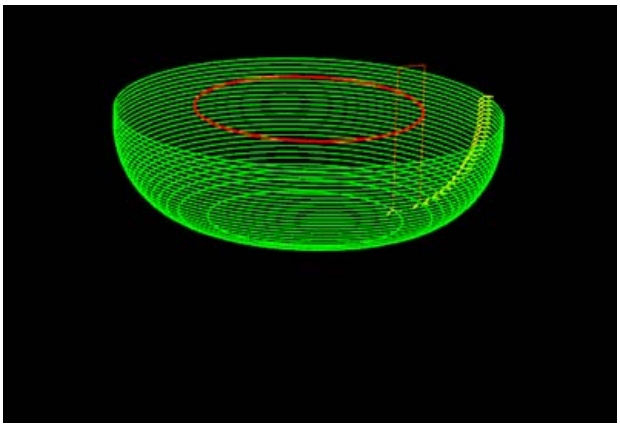
Method=CovexRadius, Value=+Ve, Profile=Inside



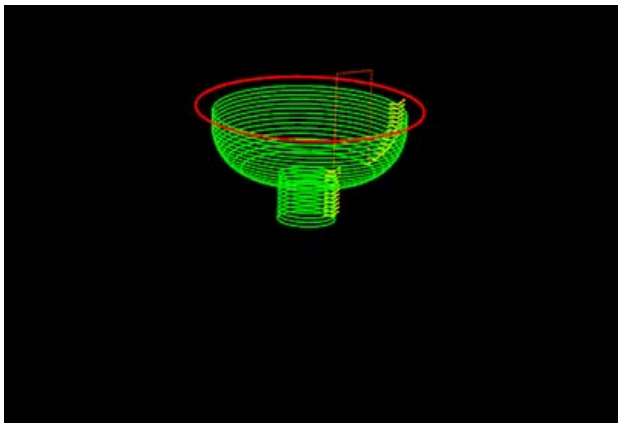
Method=CovexRadius, Value=-Ve, Profile=Inside



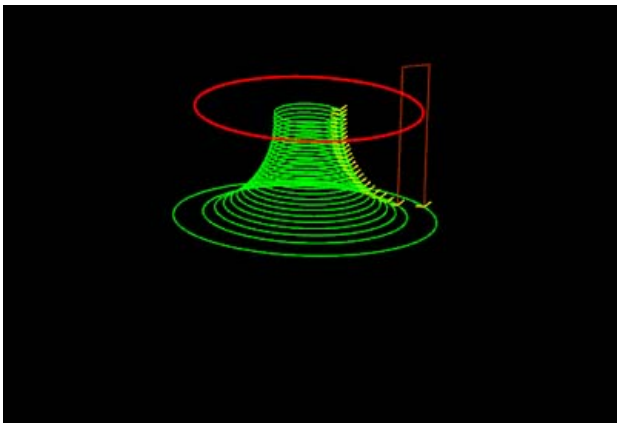
Method=ConcaveRadius, Value=+Ve, Profile=Outside



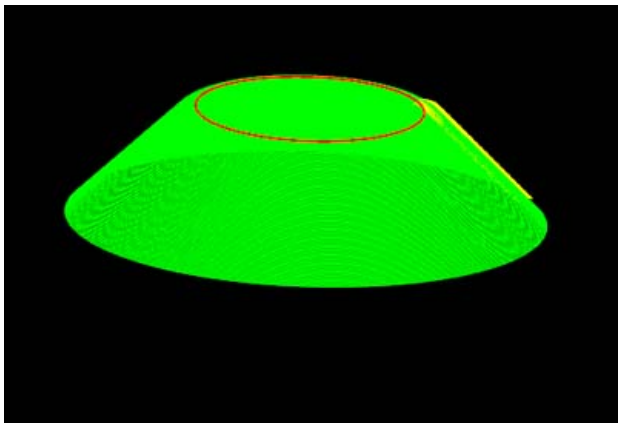
Method=ConcaveRadius, Value=-Ve, Profile=Outside



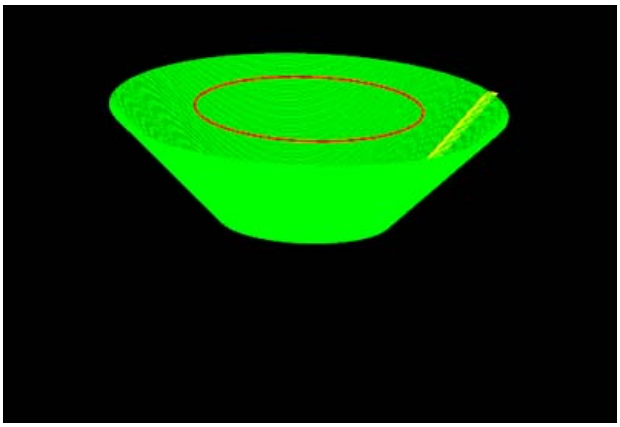
Method=ConcaveRadius, Value=+Ve, Profile=Inside



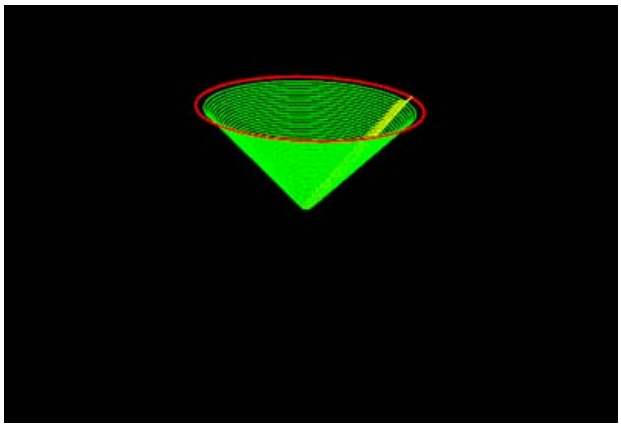
Method=ConcaveRadius, Value=-Ve, Profile=Inside



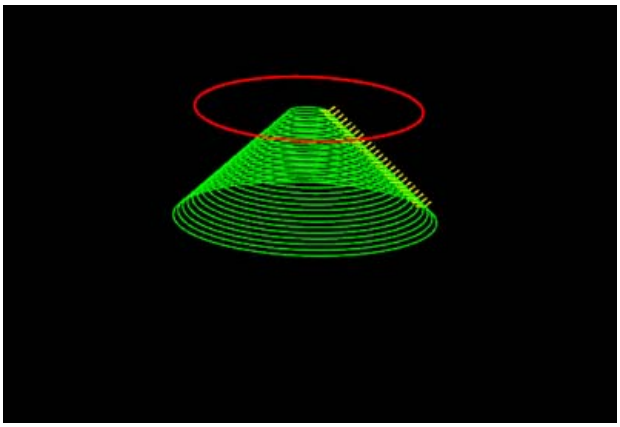
Method=Slope, Value=+Ve, Profile=Outside



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.

[New! 0.9.8N]
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

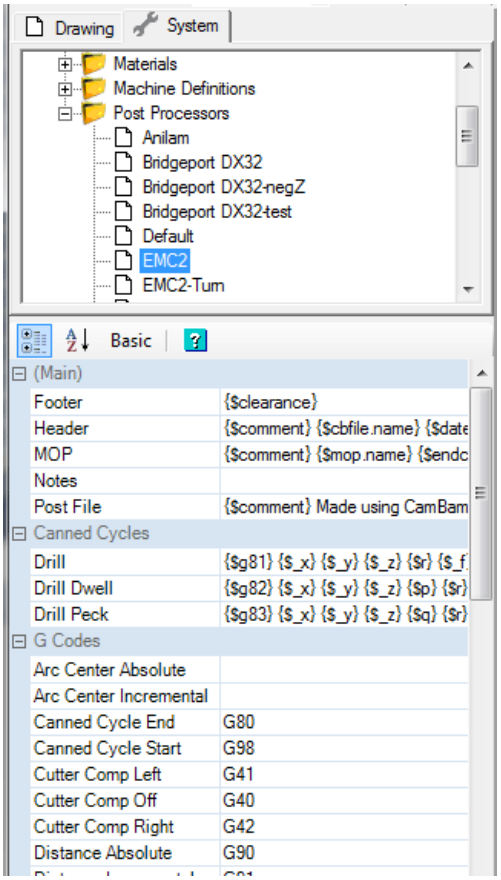
Post Processor Management

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.



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

[New! 0.9.8L]
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

[New! 0.9.8L]
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

[New! 0.9.8N]
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

[New! 0.9.8h]
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

[New! 0.9.8h]
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

[New! 0.9.8h]
Used in the {$cuttercomp(off|L|R)} macros. Typically Off=G40, Left=G41, Right=G42.

G Codes - Distance Absolute, Distance Incremental

[New! 0.9.8h]
Typically absolute=G90, incremental=G91. **NOTE!** Incremental distance mode is not currently supported.

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

[New! 0.9.8h]
Typically inches=G20, millimeters=G21.

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

[New! 0.9.8h]
Typically exact stop=G61, constant velocity=G64.

G Codes - Workplane XY, Workplane XZ, Workplane YZ

[New! 0.9.8h]
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

[New! 0.9.8h]
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

[New! 0.9.8h]
Typically M47.

M Codes - Spindle CW, Spindle CCW, Spindle Off

[New! 0.9.8h]
Typically CW=M3, CCW=M4, Off=M5.

M Codes - Stop

[New! 0.9.8h]
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}
```

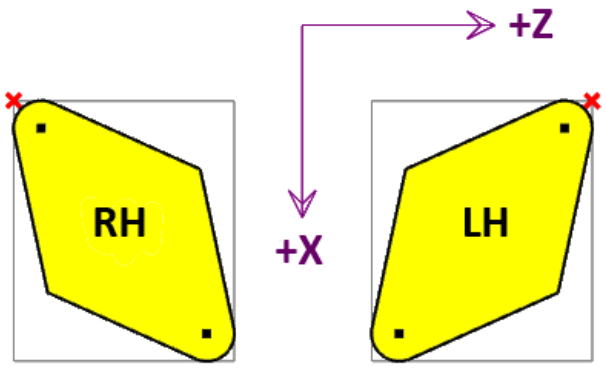
[p]Drill Peck[p]

```
{$g83} {$_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

[New! 0.9.8N]

Line Numbering - Add Line Numbers

If *True* then line numbers will be inserted at the begining 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

[New! 0.9.8N]
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

[New! 0.9.8h]
Character sequence used at the end of a line. Escape code \r and \n can be used.

Options - Invert Arcs

Controls the behaviour od 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 postive 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

[New! 0.9.8i]
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

[New! 0.9.8L]
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

[New! 0.9.8j]
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"

Post Processor Macros

$arc.i ***$arc.j*** ***$arc.k*** **[New! 0.9.8L]**	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 **[New! 0.9.8L]**	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*** **[New! 0.9.8N]**	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.
$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 **[New! 0.9.8L]**	Outputs the ***Clearance Plane*** value of the current machining operation.
$mop.cutfeedrate **[New! 0.9.8L]**	Outputs the ***Cut Feedrate*** value of the current machining operation. The 'F' register code prefix is not output.
$mop.depthincrement **[New! 0.9.8L]**	Outputs the ***Depth Increment*** value of the current machining operation.
$mop.dwell **[New! 0.9.8L]**	Outputs the ***Dwell*** value of the current drilling operation.
$mop.first.x ***$mop.first.y*** ***$mop.first.z*** **[New! 0.9.8N]**	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 **[New! 0.9.8L]**	Outputs the ***Hole Diameter*** value of the current drilling operation.
$mop.name	Inserts the current machining operation's ***Name*** property.
$mop.peckdistance **[New! 0.9.8L]**	Outputs the ***Peck Distance*** value of the current drilling operation.
$mop.plungefeedrate **[New! 0.9.8L]**	Outputs the ***Plunge Feedrate*** value of the current machining operation. The 'F' register code prefix is not output.
$mop.retractheight **[New! 0.9.8L]**	Outputs the ***Retract Height*** value of the current drilling operation.
$mop.stocksurface **[New! 0.9.8L]**	Outputs the ***Stock Surface*** value of the current machining operation.
$mop.tag **[New! 0.9.8L]**	Outputs the ***Tag*** value of the current machining operation.
$mop.targetdepth **[New! 0.9.8L]**	Outputs the ***Target Depth*** value of the current machining operation.
$move.x ***$move.y*** ***$move.z*** **[New! 0.9.8L]**	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 **[New! 0.9.8N]**	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.
$set(x|y|z|a|b|c|f|p|q|r, <value>) **[New! 0.9.8L]**	Sets the current value of the specified X, Y or Z register. No gcode will be output. Example: `$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 **[New! 0.9.8N]**	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 **[New! 0.9.8N]**	Inserts the current tool's ***Name*** property (from tool library) or T(tool number) if there is no tool library entry.
$tool.profile **[New! 0.9.8L]**	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 **[New! 0.9.8N]**	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, $zneg, ... **[New! 0.9.8h]**	The same as the other register macros ($x, $_y etc), but with the value sign reversed.
$xabs, $yabs, $zabs, ... **[New! 0.9.8h]**	The same as the other register macros ($x, $_y etc), but with the value always positive.

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.

New [0.9.8]
As of version 0.9.8, 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 **[New! 0.9.8]**	Select a CAM Style for this machining operation. All default parameters will be inherited from this style.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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 labelled '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.

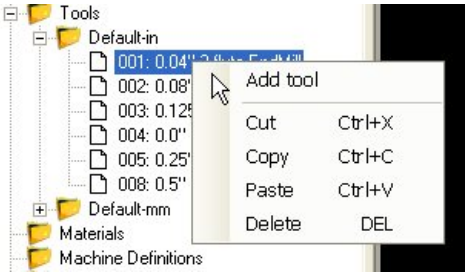
Tool numbers can also be set at the ***Machining*** and ***Part*** levels. If a tool number is set at the machining level, this will be the default tool, used by all parts and machining operations, unless explicitly set in the part or machining operation. The tool selected for the part will override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

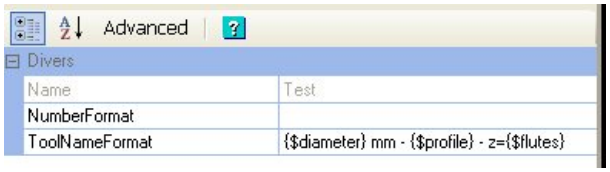
Tool libraries and definitions are a relatively new addition to CamBam. Some of the properties available in the tool definitions are intended for future functionality, but for the current release can be considered for informational use only.

Axial Depth Of Cut *Informational*	The maximum (Z) depth of cut for this tool.
Coating *Informational*	
Comment **[New! 0.9.8N]**	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 signalling 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.



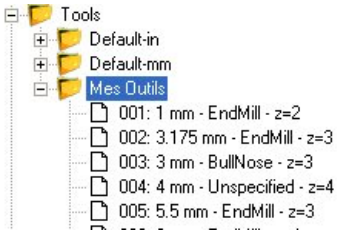
It is also possible to automatically rename using an expression entered into the tool libraries ***Tool Name Format*** property.
The expression can contain the following macros:

{$diameter} = Tool diameter
{$profile} = Tool shape / profile
{$flutes} = Number of teeth / flutes
{$index} = Tool number
{$length} = Tool length
{$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}
```



Speeds and Feeds Calculator

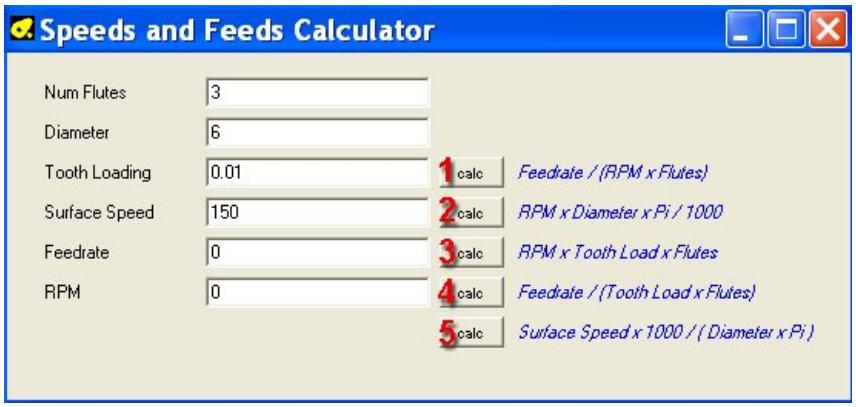
The speeds and feeds calculator is accessed from the context menu shown when right clicking a machining operation. This can be used to calculate feed rates, spindle rpms and other machining parameters.

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).

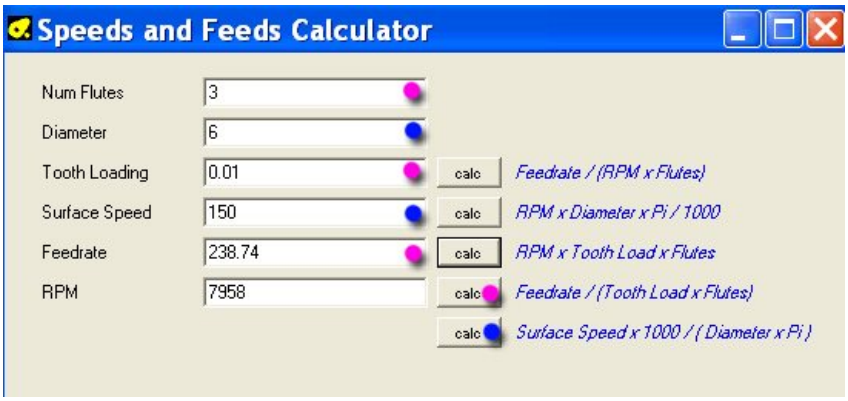
Warning: When the speed and feeds calculator is opened, it will contain feedrate and spindle speed information from the selected machining operation. These values may need to be recalculated to attain accurate values given the current tooth load and surface speeds.

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.



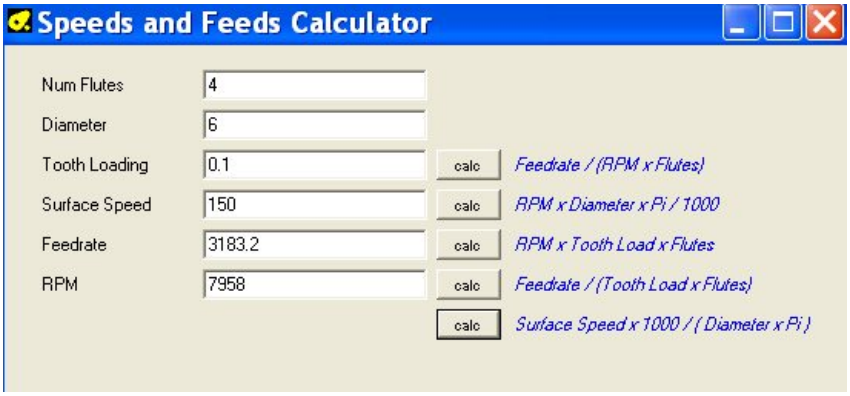
Example 1

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

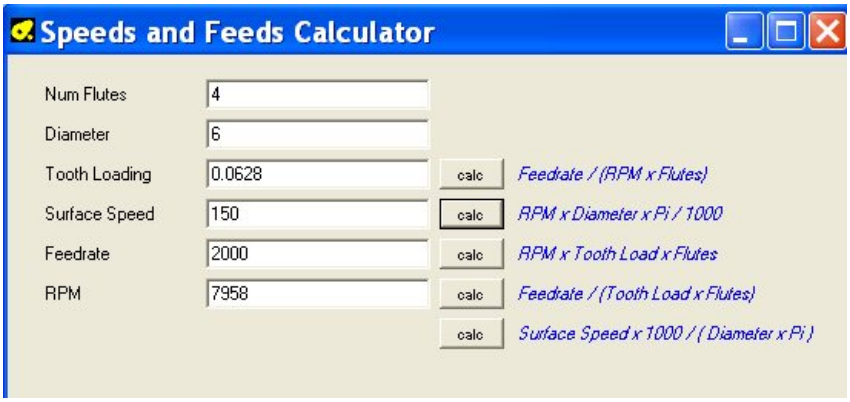


The button 5) calculation provides a spindle speed of **7958 rev/min** and button 3), a feedrate of **3183.2 mm/min**.

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.



Drawing (CAD)

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.

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 **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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+R**

Circle

Properties

Center	The coordinates of the center of the circle
Diameter	The diameter of the circle.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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.

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 **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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

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 **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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 **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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

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.
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.

Surface

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

Spline

Splines (or NURBS) can currently only be imported from DXF files. Spline drawing is not yet supported.

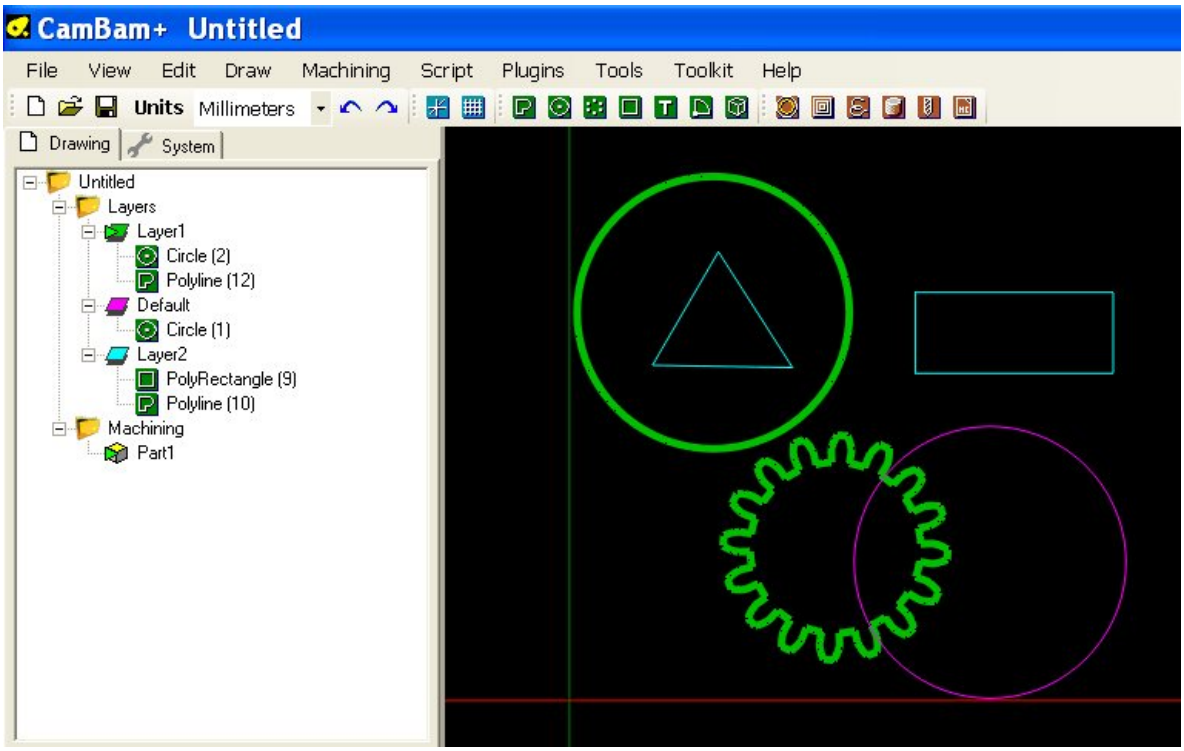
Layers

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.

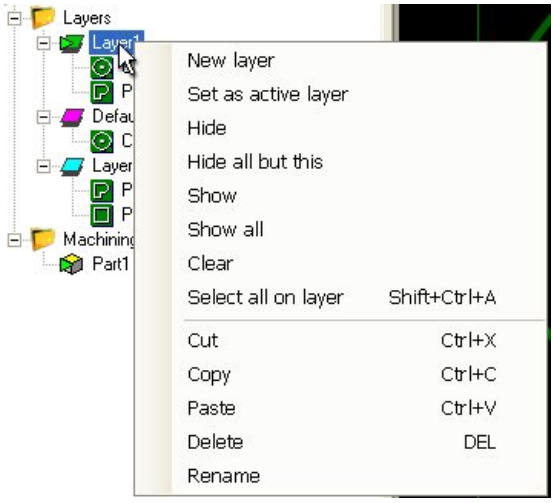
Selecting a layer in the drawing tree allows its properties to be altered in the property grid. The appearance of the layers, such as colour and line widths, can be set in this way.



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.



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.

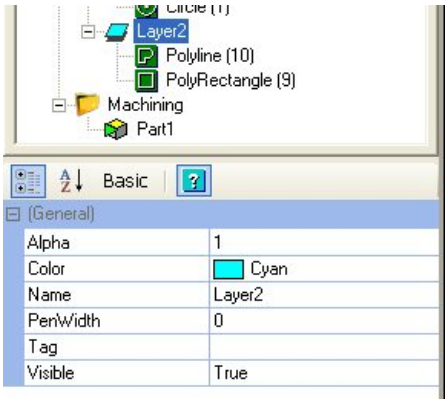
Delete

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

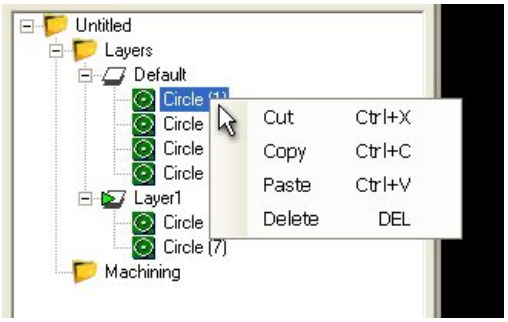


Alpha	The level of transparency of the drawing objects in the layer. 0 to 1, 1=opaque, 0 = completely transparent.
Color	Color used to display the drawing objects.
Name	The name of the layer.
Pen Width	Thickness of the drawing lines.
Tag **[New! 0.9.8]**	A general purpose, multiline text field that can be used to store notes or parameters from plugins.
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.

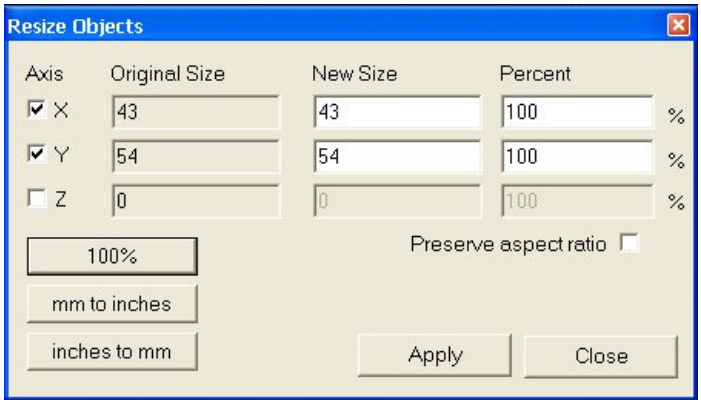
Using the keyboard only, selected objects can be moved holding the **SHIFT** key and using the arrow keys. This will move the object one minor grid unit in the arrow key direction (If using millimeters, this will be 1mm, if using inches then this will be 1/16"). If **CTRL+SHIFT** keys are held down,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

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 revert 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).

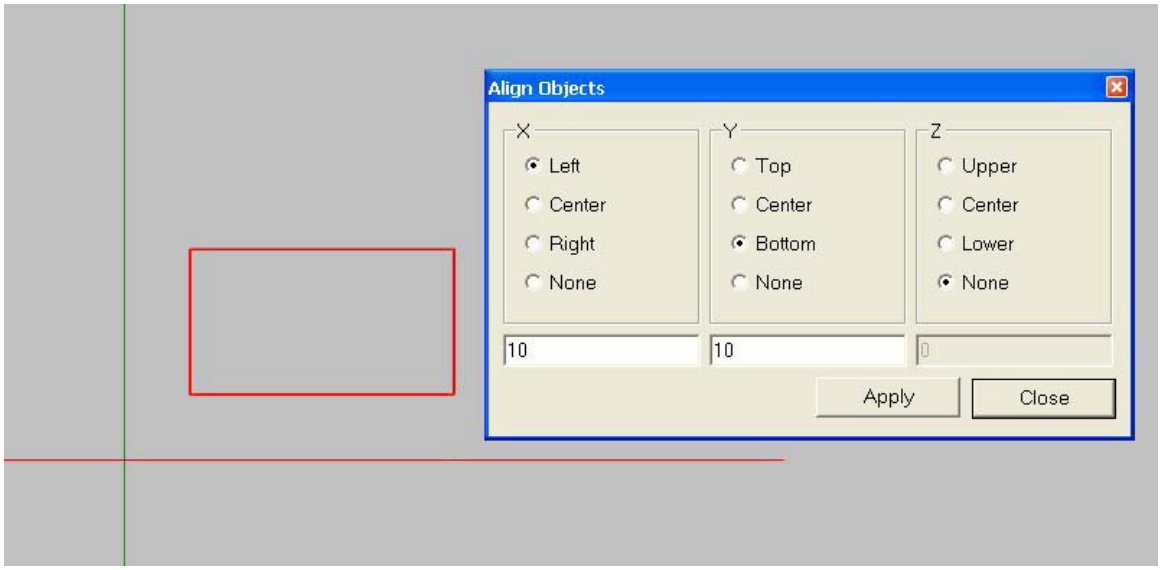
The rotation command can also be used to rotate about other axis. Pressing the **X**, **Y** or **Z** keys while rotating will select the axis of rotation. The angle of rotation is always set by moving the mouse around the center point in the plane of the drawing view, regardless of the axis setting.

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=0. 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.

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.

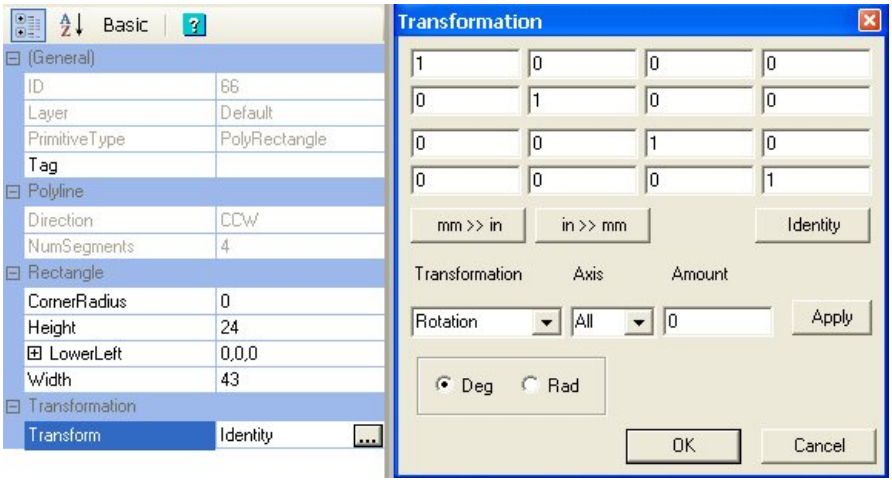
Center (Of Points) will align the 'average' point of all the control points contained in the selected shapes.

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 behaviour 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.

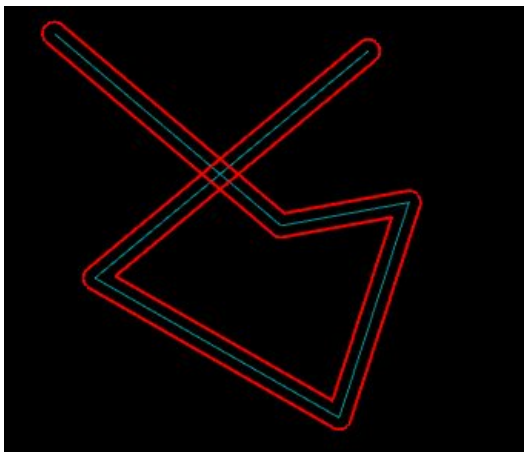
If a positive offset distance is supplied, the resulting offset polyline will be outside the selected shape. If a negative offset is given, the offset polyline should be inside the shape.

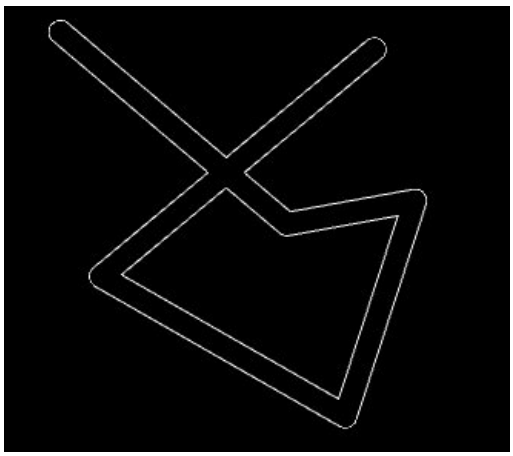
Open Offset

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.

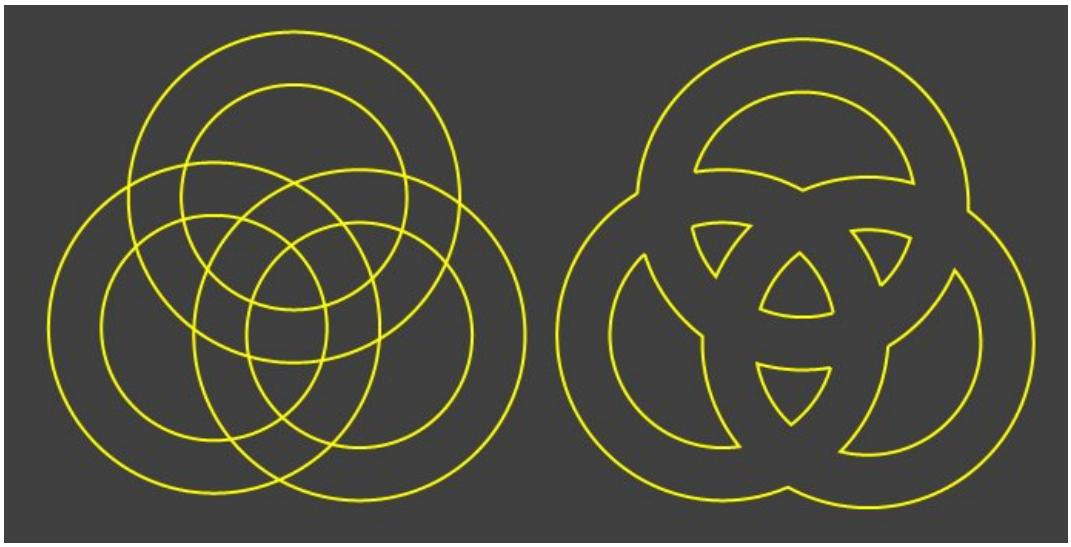
The image on the left shows the result of the ***Open Offset*** command used on an open polyline. Note that the closed polyline can loop back upon itself. On the right hand image, this open offset has been modified by using the ***Break at intersections*** operation, then the unwanted inner segments 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.

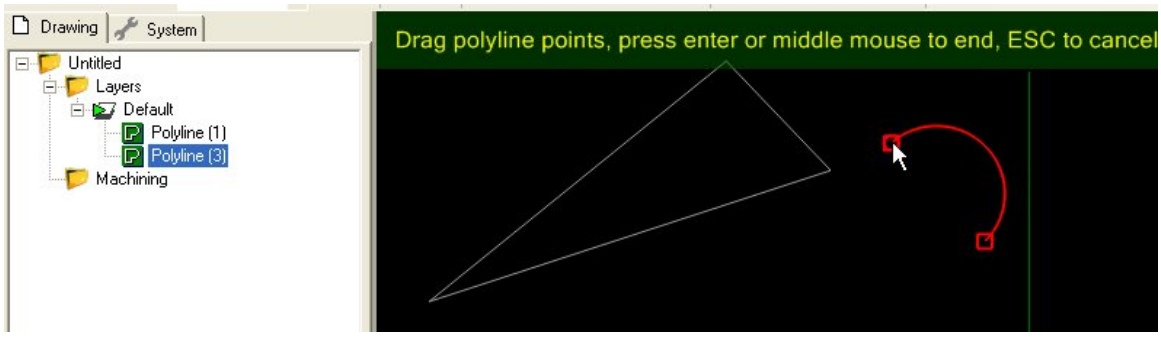
Edit Polyline

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

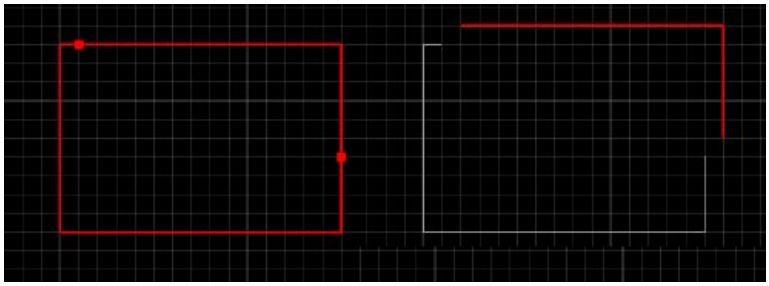
Reverses the order of the points within a polyline.

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

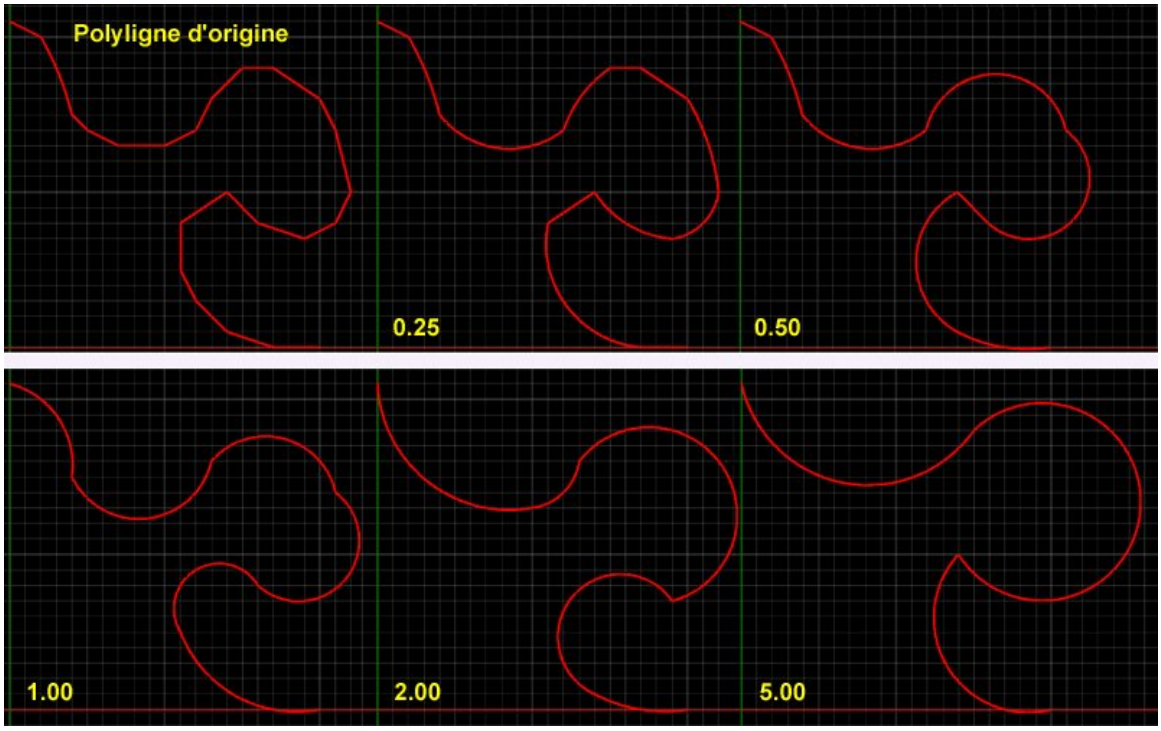
Changes the first polyline control point, (for closed polylines only).

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 vectorisation 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.

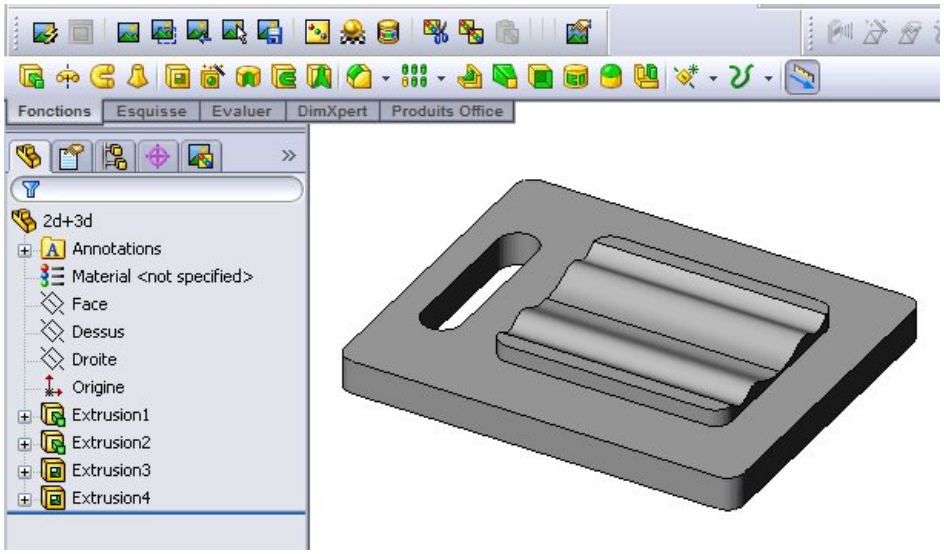
Plane Slice X, Y, Z

These functions obtain polylines from a 3D object by slicing the object along a given axis.

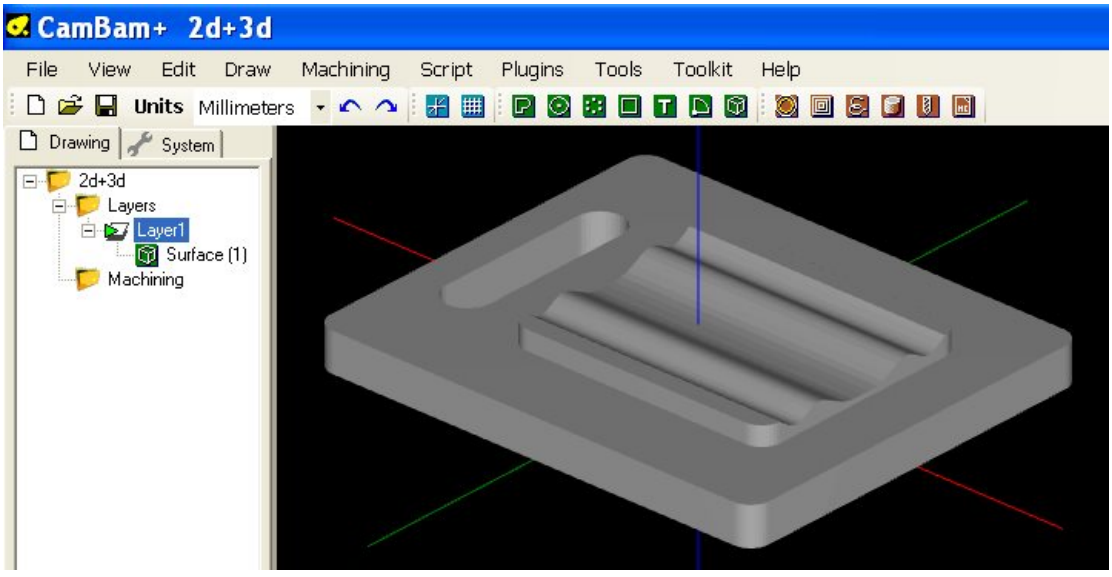
Plane slices provide a useful way of generating 2D machining operations from 3D models, without having to redraw the models in 2D. For many engineering or prismatic 3D shapes, 2D machining operations can provide much simpler, faster and more accurate operations than the 3D machining operations.

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 ®.

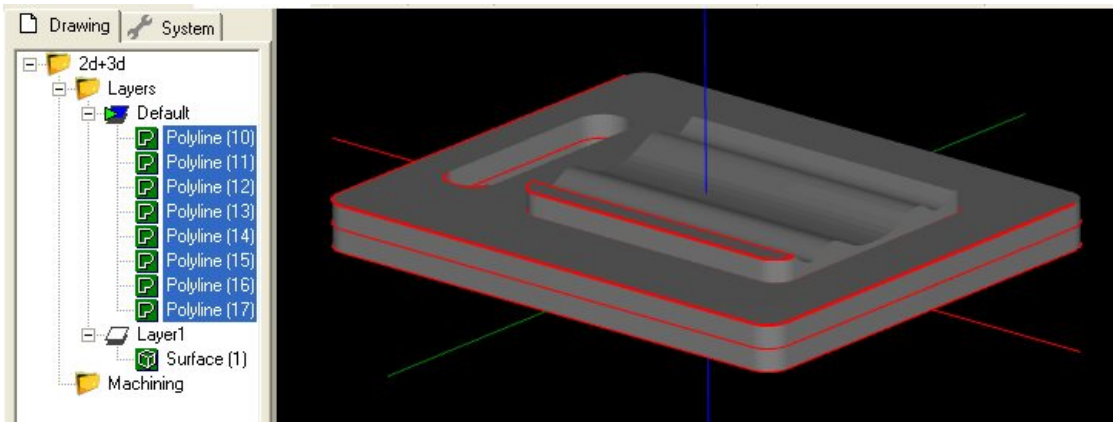


The model is then loaded into CamBam.



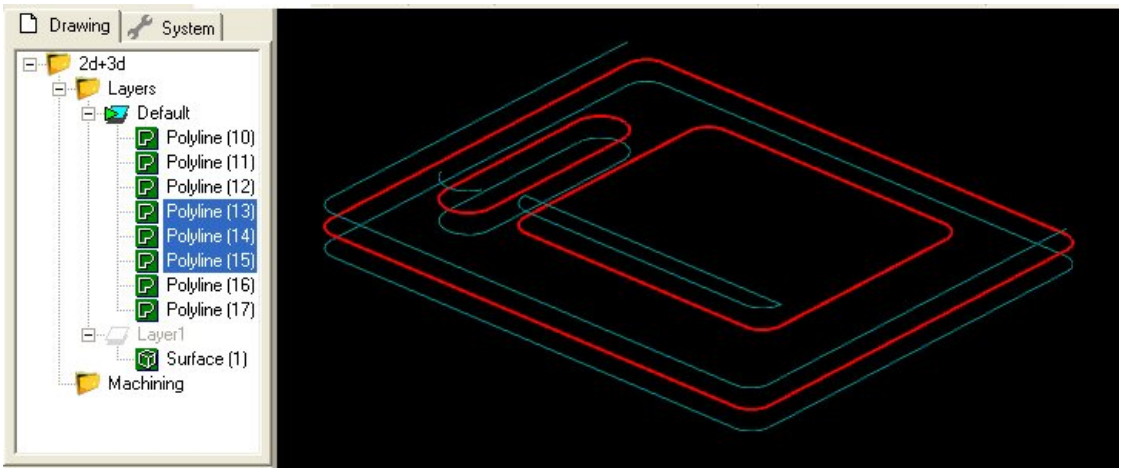
The wavy part of the model will be machined using 3D operations, but it would be more efficient to use 2D operations on the flat parts of the model.

Plane Slice Z is used to slice the model along planes, normal to the Z axis, at 5mm intervals.

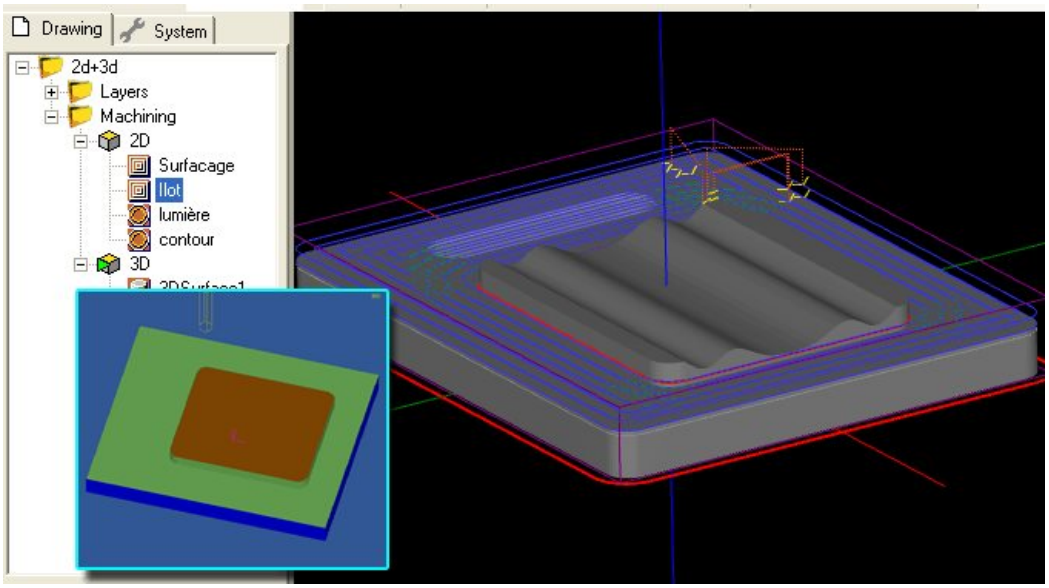


It is good practice to create a new layer to receive the plane slice polylines, to make the drawing more manageable and so that the polylines may be viewed and manipulated independently of the original 3D surface.

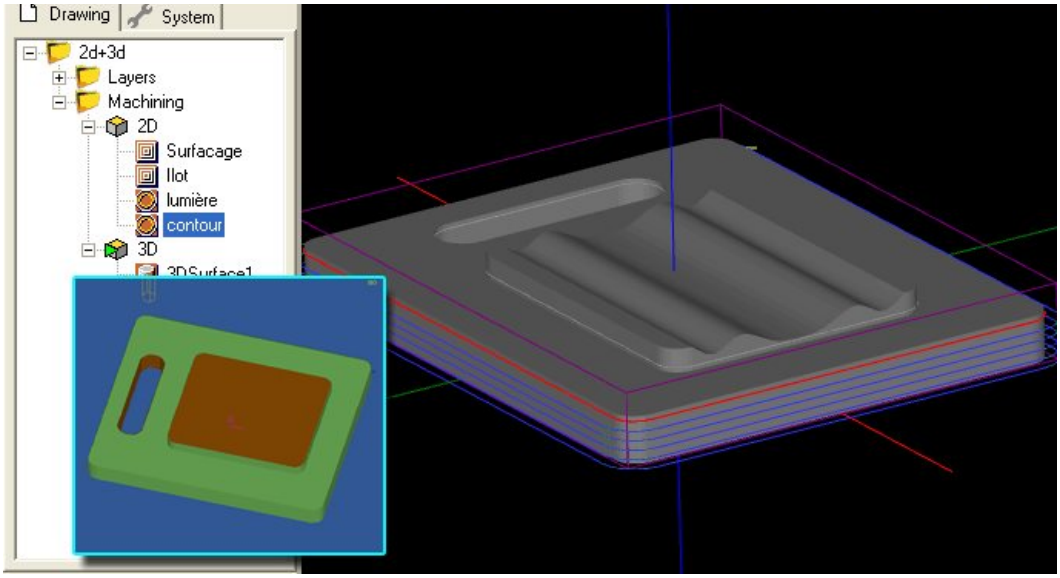
In this example, we are only interesting in a selection of the plane sliced polylines (shown highlighted below). The other polylines can be deleted. The layer containing the 3D surface has also been hidden for clarity.



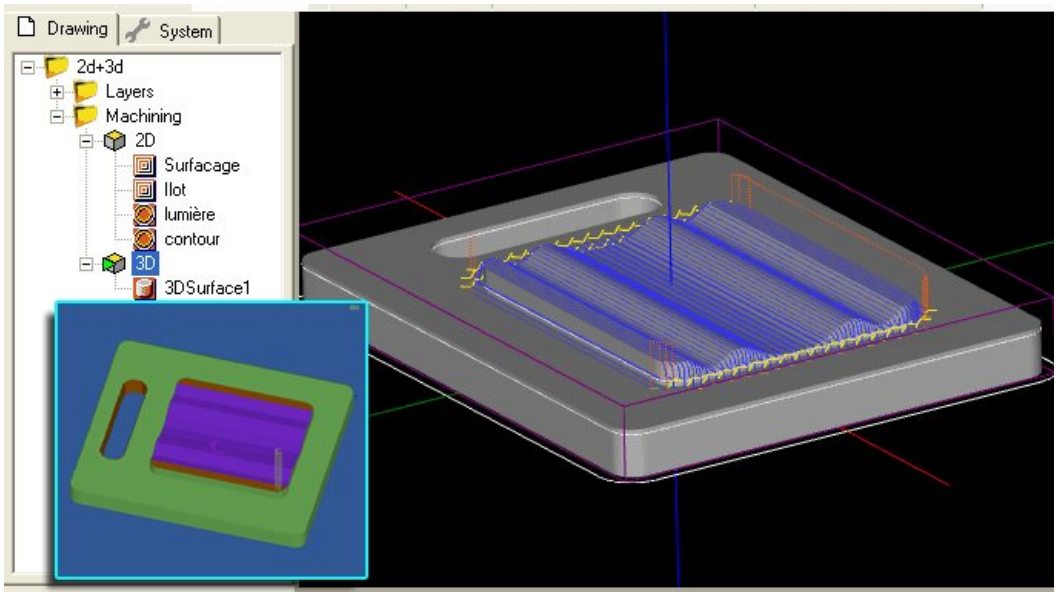
2D pocket operations are used to clear the flat areas of the mode. The inset image shows the results when simulated using CutViewer Mill.



2D profiles are then used for cutting slots and the outer shape.



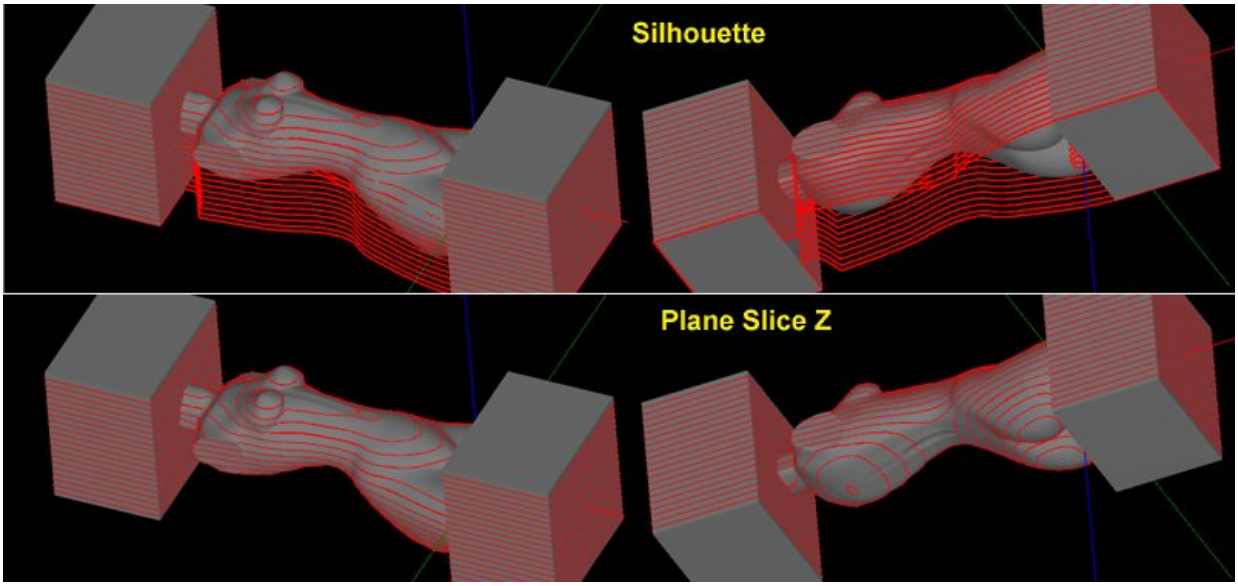
A 3D surface operation is then used for the wavy area of the shape. The inner plane slice polyline is used to restrict the 3D surface by specifying a boundary shape. See the 3D tutorial for more information on this operation.



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.

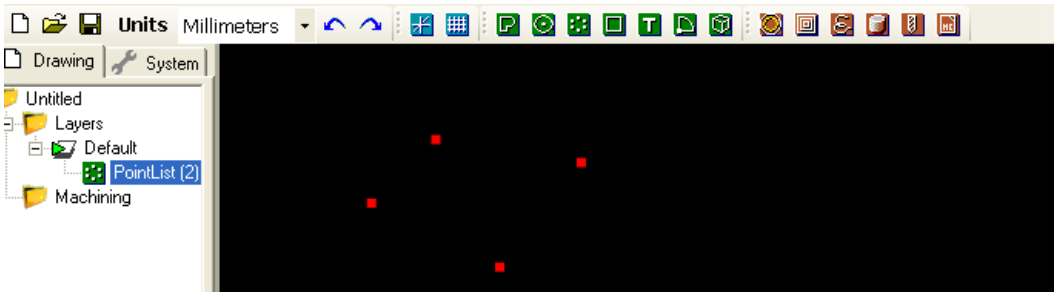
The routine prompts for a Projection Resolution. This is the distance along each drawing line at which the Z height is tested.

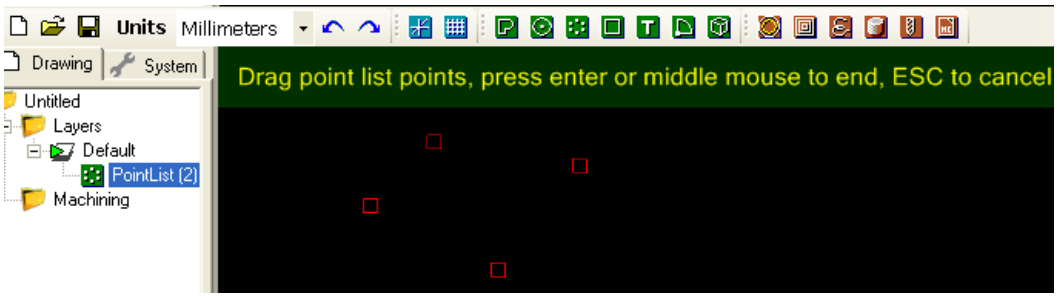
If a point along the line is outside any surfaces, the minimum depth of all selected surfaces is used.

Editing Point Lists

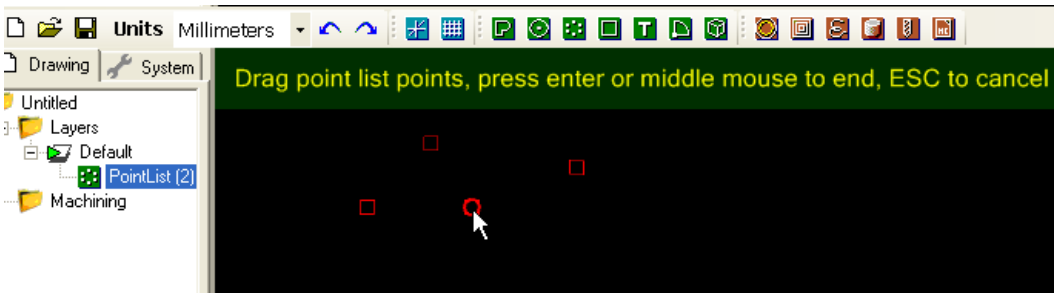
Move or add points

You can edit a point list object by double clicking any of its points in the drawing view to activate the edit mode.

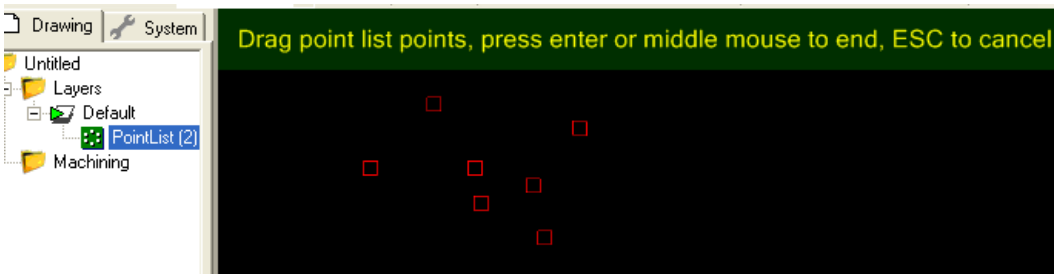




To move a point, click and drag the square point icons.



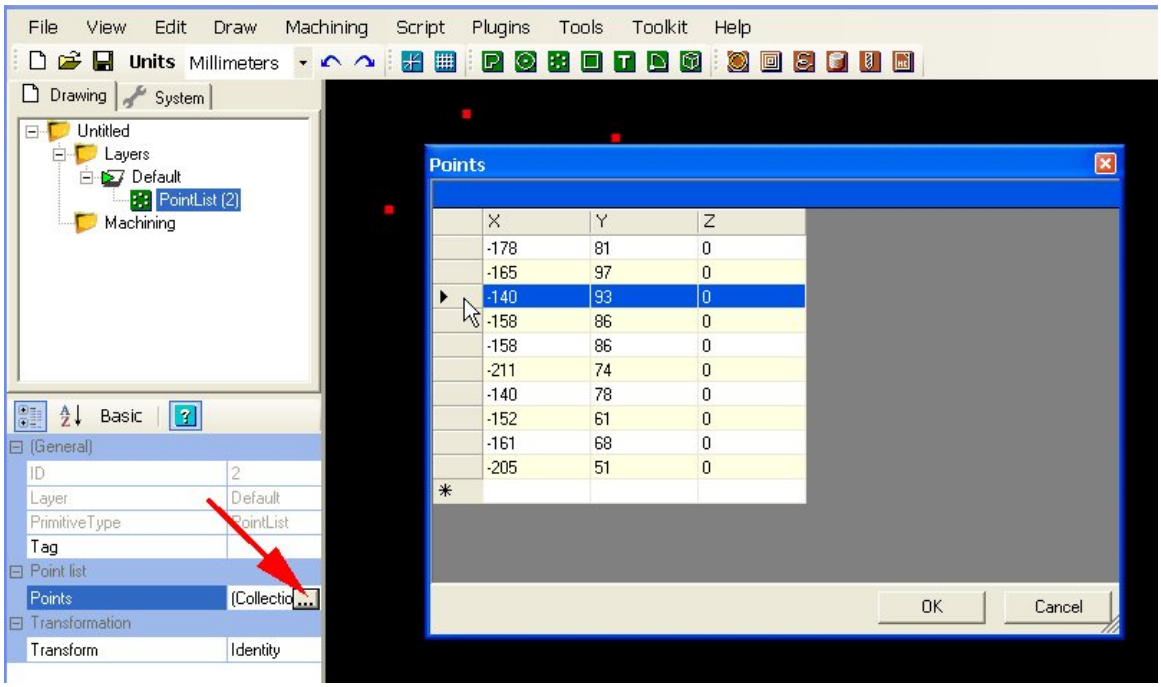
Click on empty areas to add new points to the point list.



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.

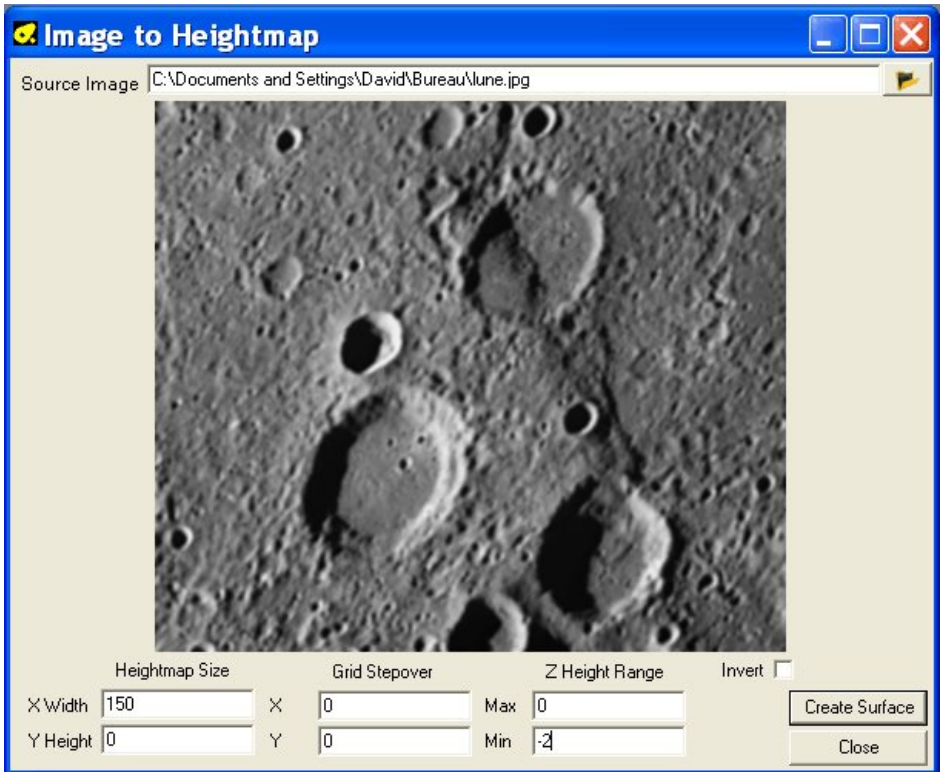
From Mesh File

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.

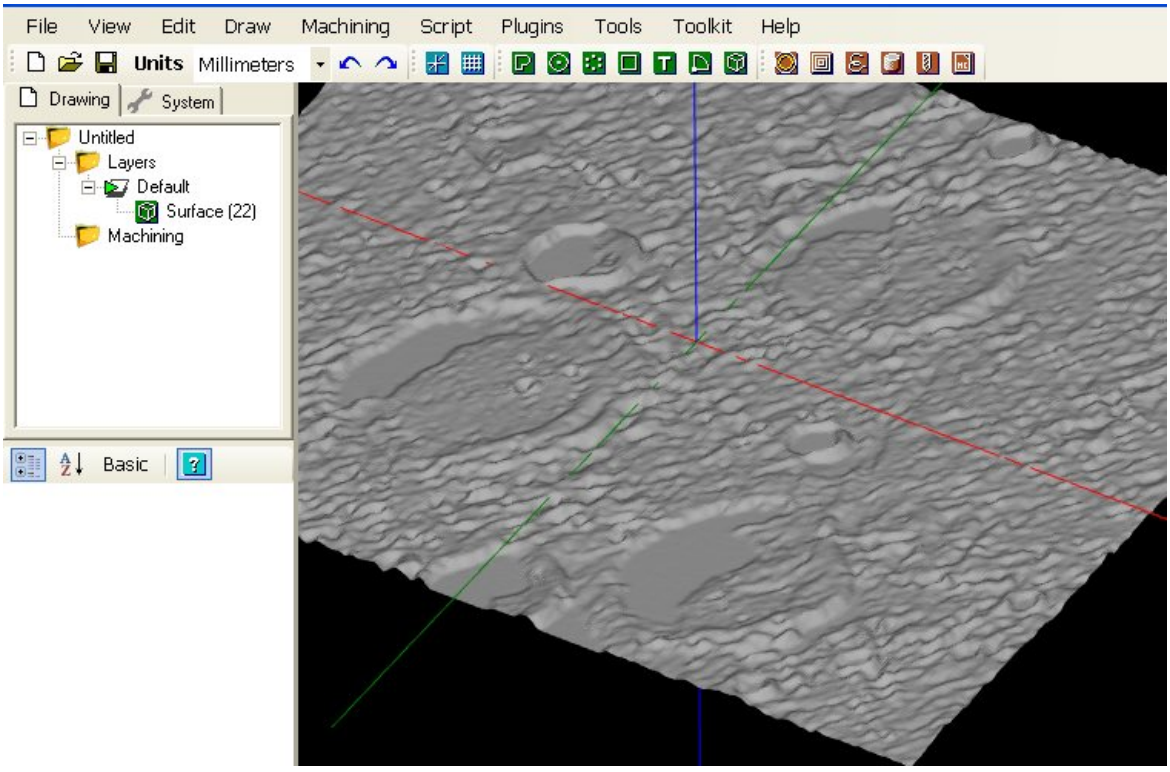
Heightmap Size: Set the X and Y dimensions (in current drawing units) of the surface to be generated. If the X or Y dimension is left at 0, this dimension will be automatically calculated so as to preserve the aspect ratio of the original image.

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.

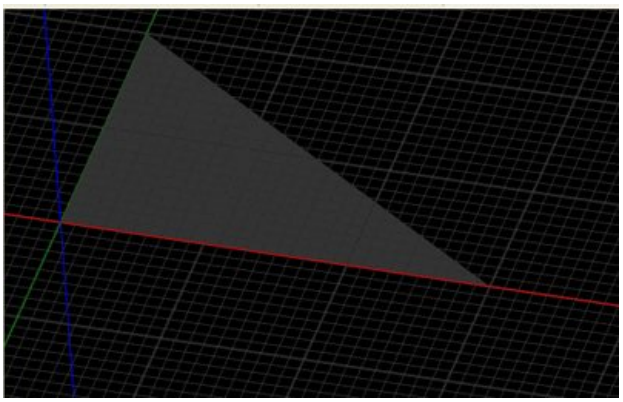
Each line consists of nine coordinates, separated by a space, corresponding to the coordinates X, Y and Z of three vertices defining a triangle.

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.

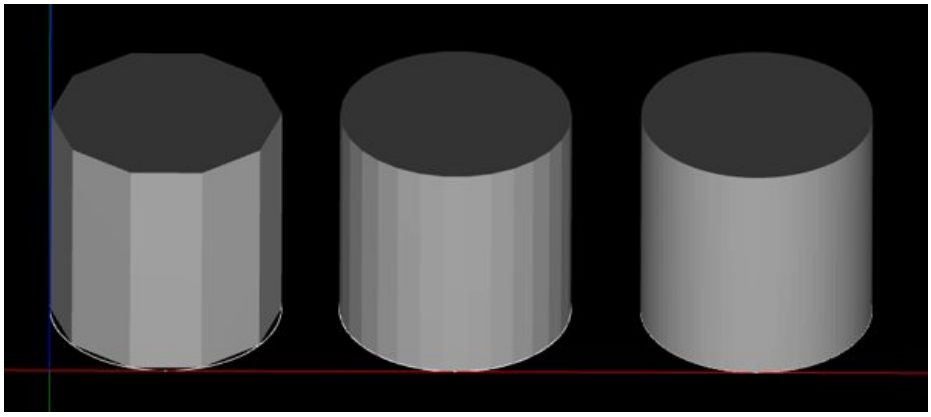
To create extrusions along other axis, the shape must first be extruded in Z, then rotate the extruded surface object

to the required orientation.

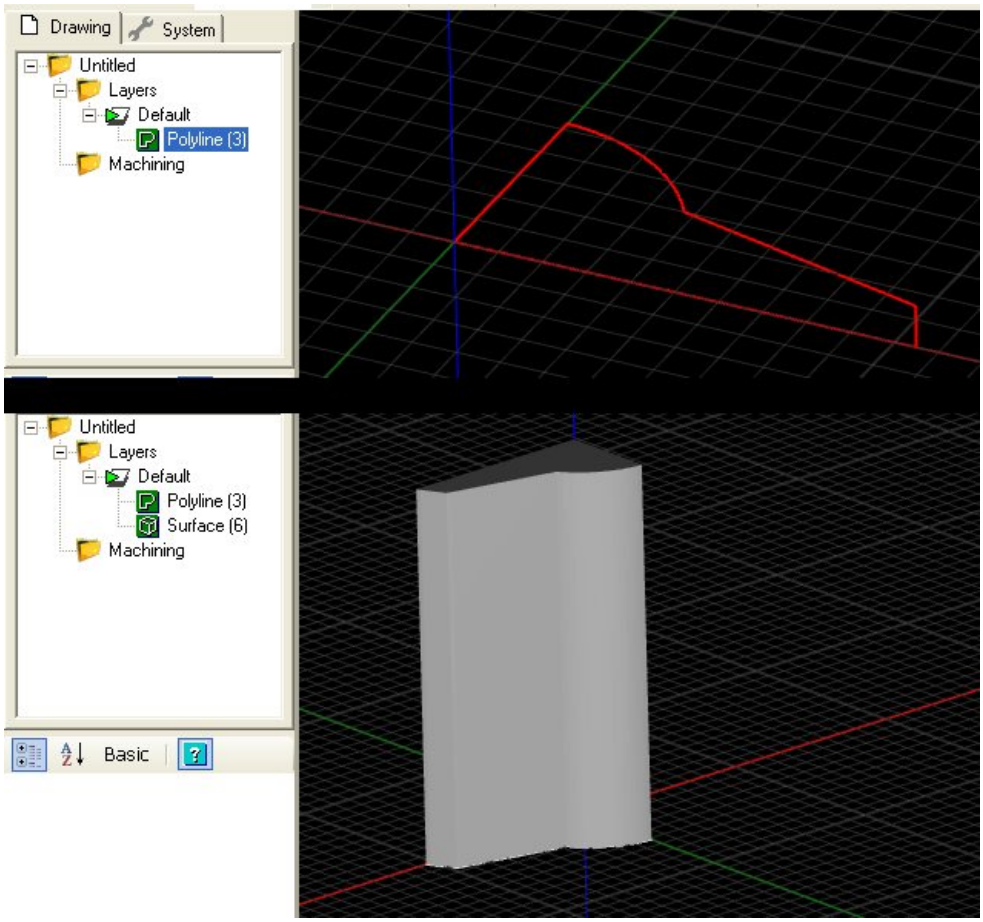
The ***Extrude*** operation will first prompt for an ***Extrusion Height***. This will be the Z height of the extruded surface. A positive height will extend toward the positive Z axis (ie toward the viewer when drawing is in the normal orientation with the XY plane parallel to the screen). A negative 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.



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.

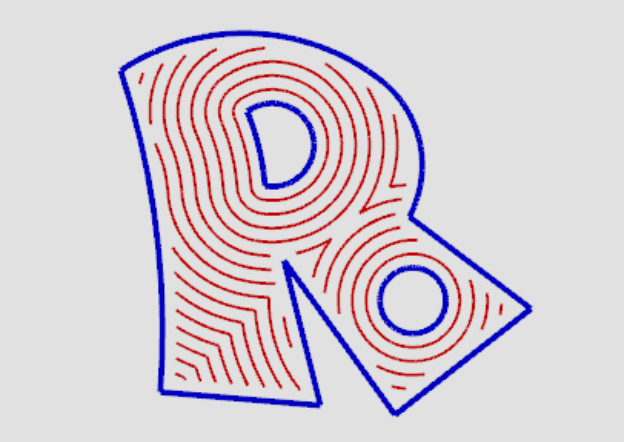
Region fillers take the following parameters:

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.

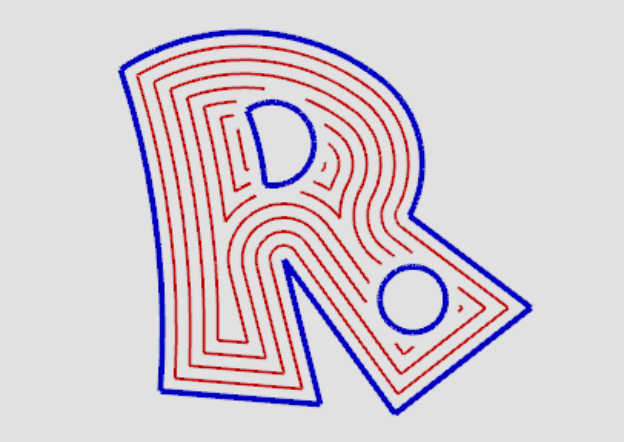
Filling Patterns

Inside Offsets



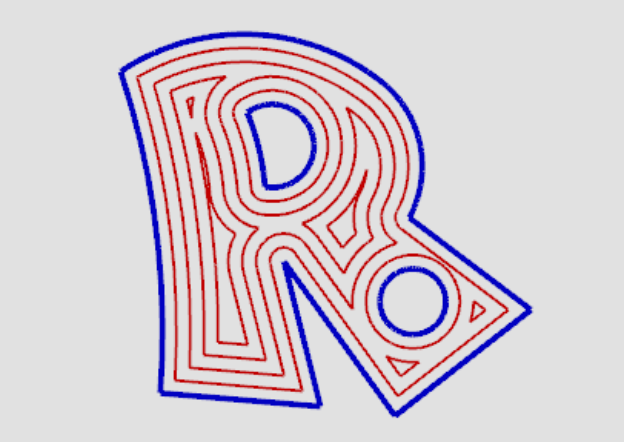
Progressive offsets from holes outwards.

Outside Offsets



Progressive offsets from outside shape inwards.

Inside + Outside Offsets

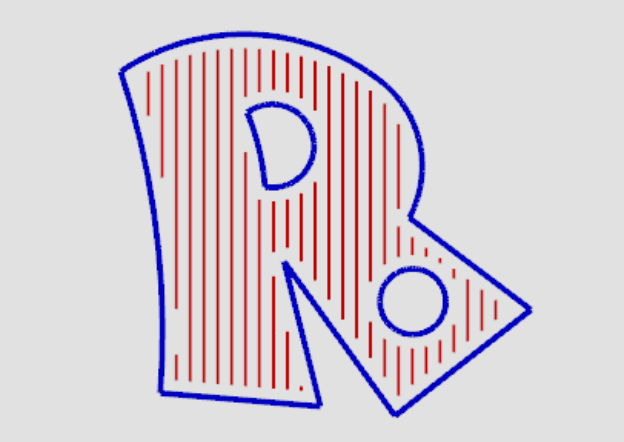


Progressive offsets from outside shape inwards unioned with offsets from holes outwards.

Horizontal Hatch

Horizontal line fill style.

Vertical Hatch



Vertical line fill style.

Tutorials

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.

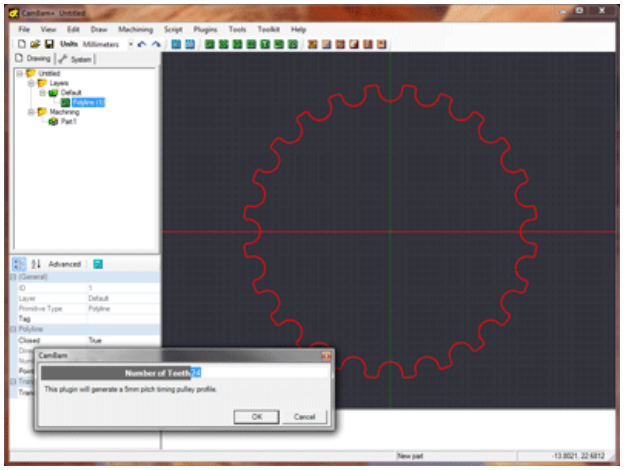
Download the files used in this tutorial

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.

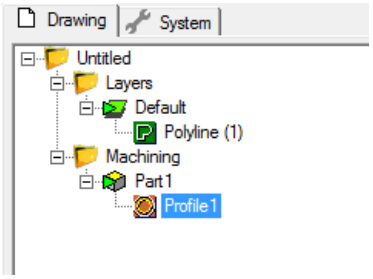
The plugin will prompt for the number of teeth for a 5mm pitch pulley, then insert a new curve with the center of the pulley about the origin.

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.

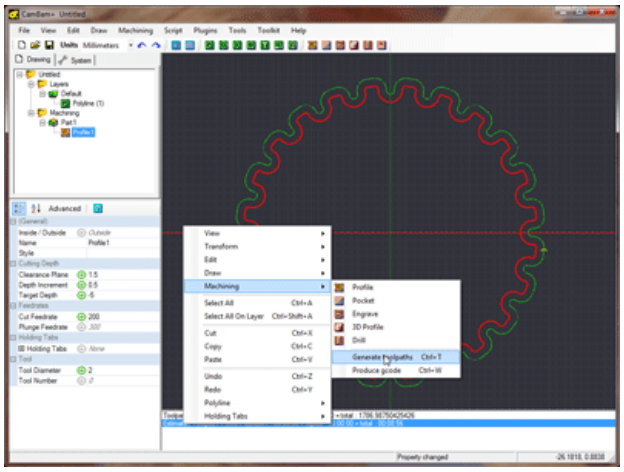


Change the Profile machine operation's properties to the following:

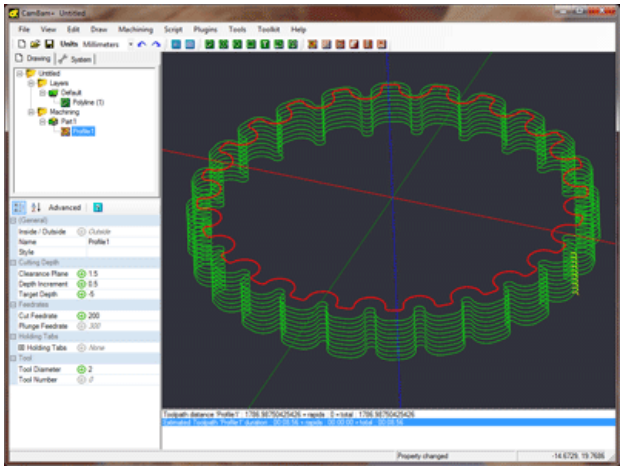
Tool Diameter	2
Depth Increment	0.5
Target Depth	-5
Cut Feedrate	200
Plunge Feedrate	100
Clearance Plane	1.5

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 ***Machine - 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.



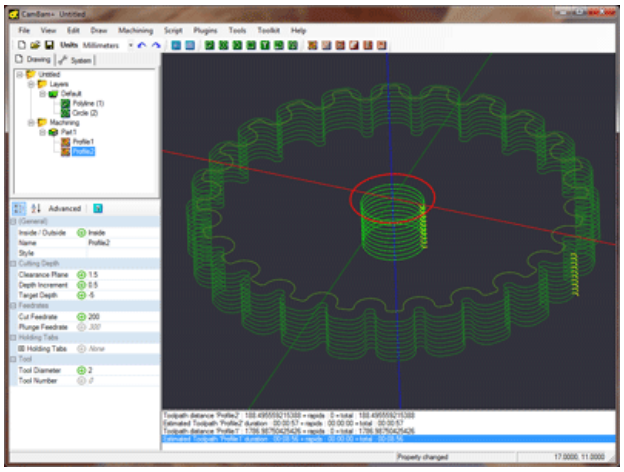
Step 3 - Creating the inner hole

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 ***Machine - 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.

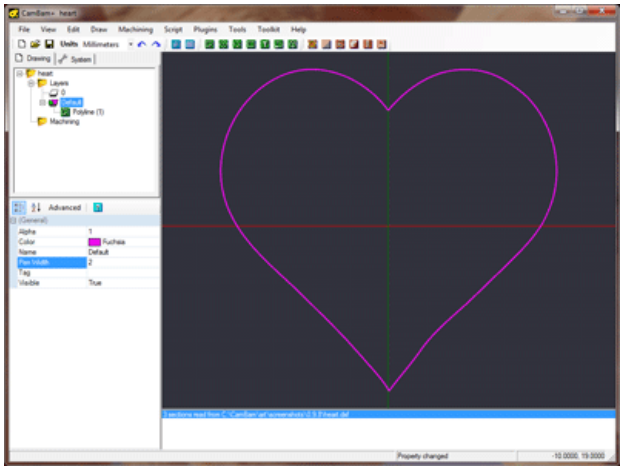
Download the files used in this tutorial

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!

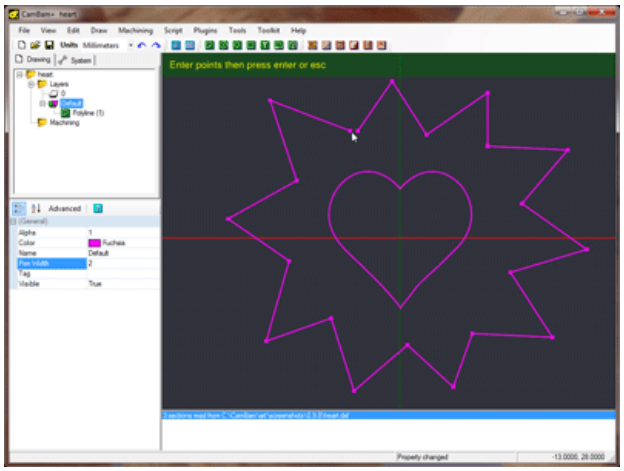
This shape is a nice and clean, closed polyline. If your DXF files contain many small segments or uses non polyline objects you should tidy the drawing before creating any machining operations.

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

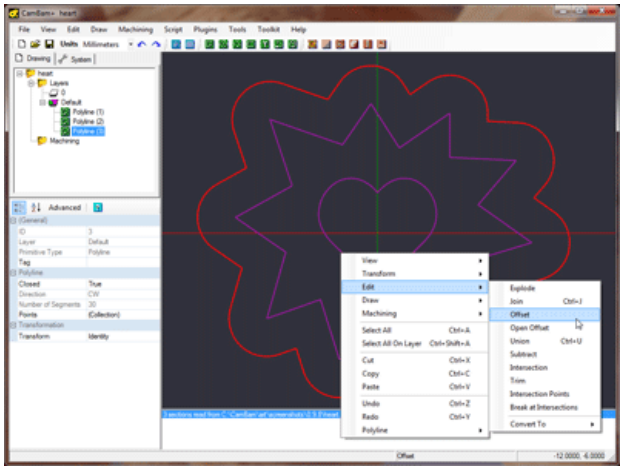
Use the polyline drawing tool  to draw a random shape around the heart. This will form the outer boundaries of an island pocket. For the last point, press the **C** key to close the shape, or click on the first polyline point (the cursor should snap to it), then press **ENTER** or click the middle mouse 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.

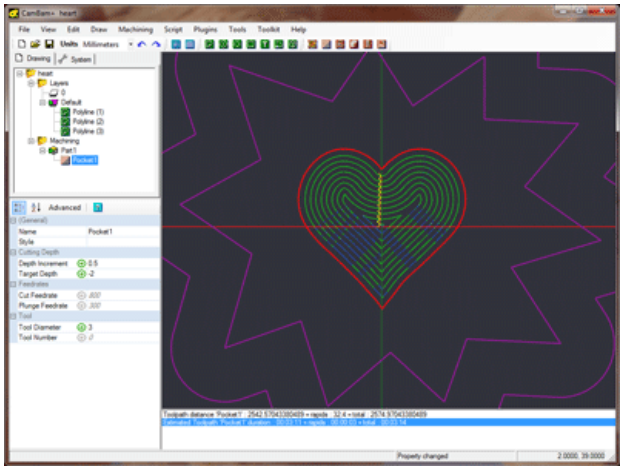
The important thing to remember is that the ***Target Depth*** should be lower than ***Stock Surface***. If stock surface is at zero then the **target depth should be negative**.

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

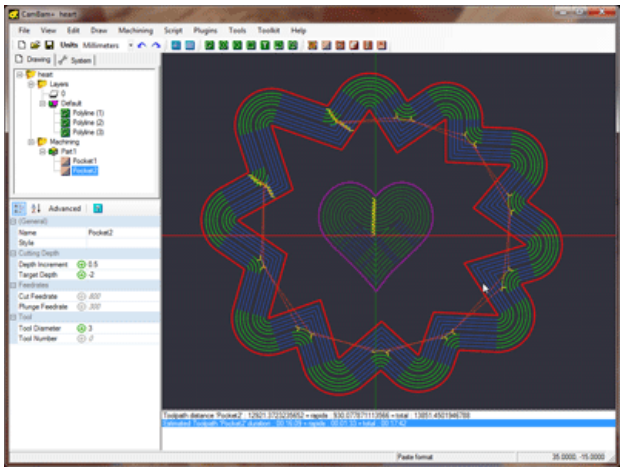
Island pockets can be created automatically by selecting inner and outer polylines then inserting a pocket as usual. Shapes within other shapes will be detected and toolpaths will be excluded from these inner 'hole' shapes.

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.

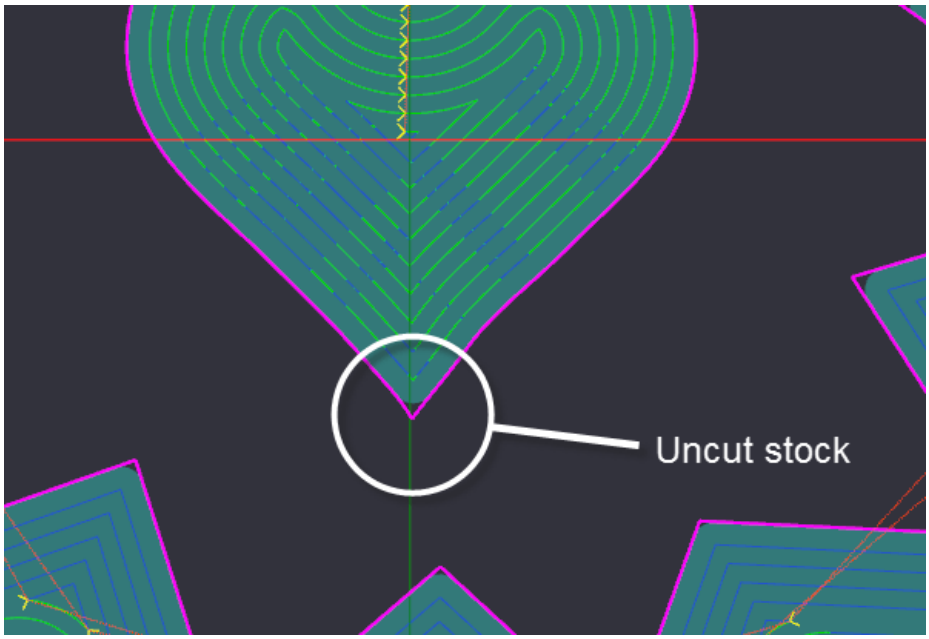


Step 5 - Show Cut Widths

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.



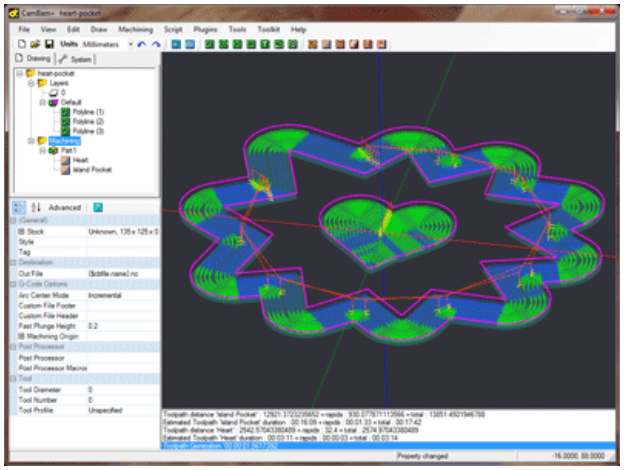
Step 6 - Machine operation renaming.

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.

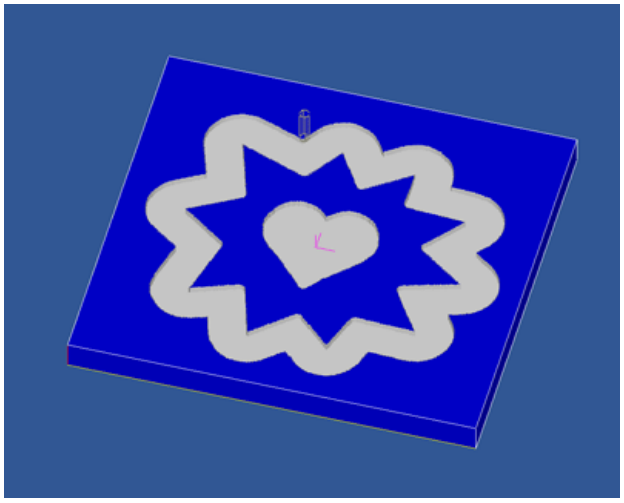
Machine operations can be given a more meaningful name, to help with readability and debugging. To rename a machine operation, select it in the drawing tree and press **F2**, or click the name a second time. Avoid using special characters in the name such as parenthesis as could cause problems due to 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.

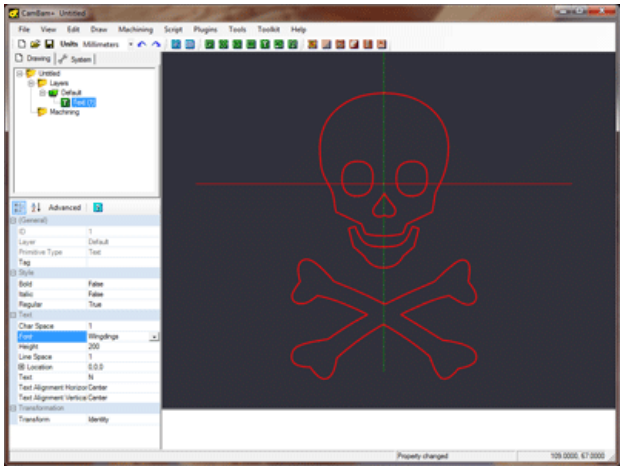
Download the files used in this tutorial

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.

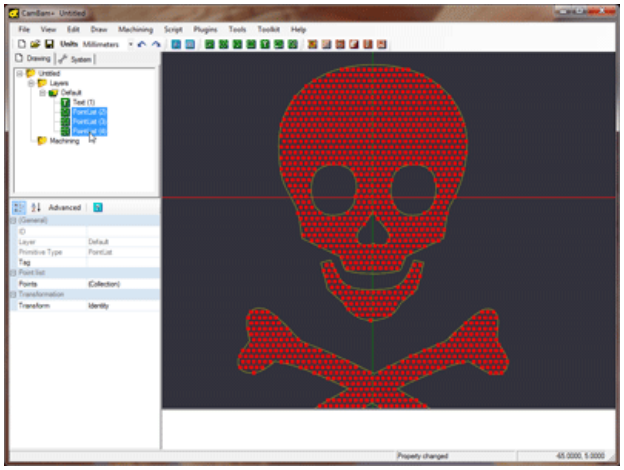
Set the text ***Height*** to something large like 200 (I am working in mm here), and change the ***Font*** property to *Wingdings*.



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).



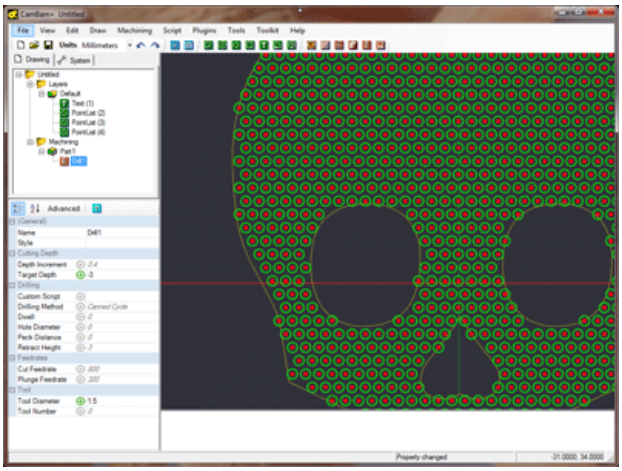
Step 3 - Insert a drilling machine operation

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.

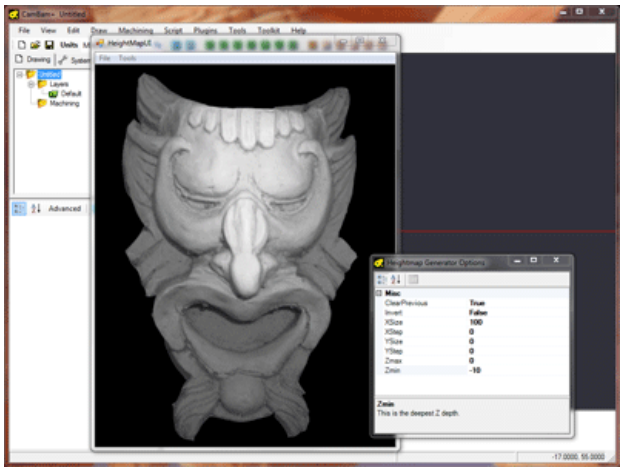
Step 1 - Open the Heightmap plugin

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

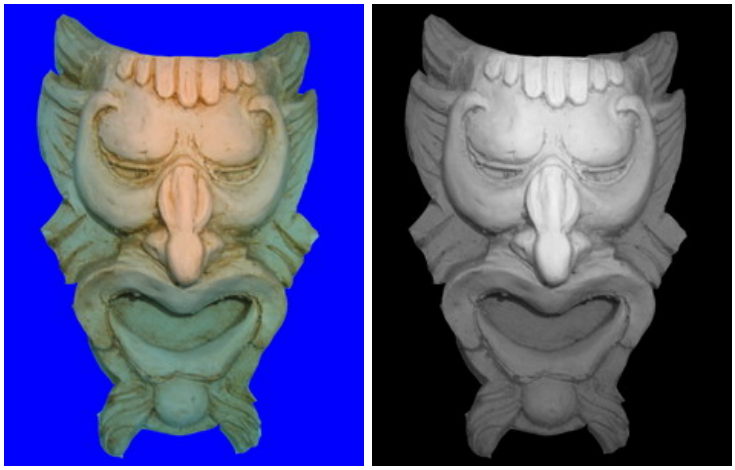
The source code for the Heightmap generator can also be found in a zip file in this folder.



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.



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.

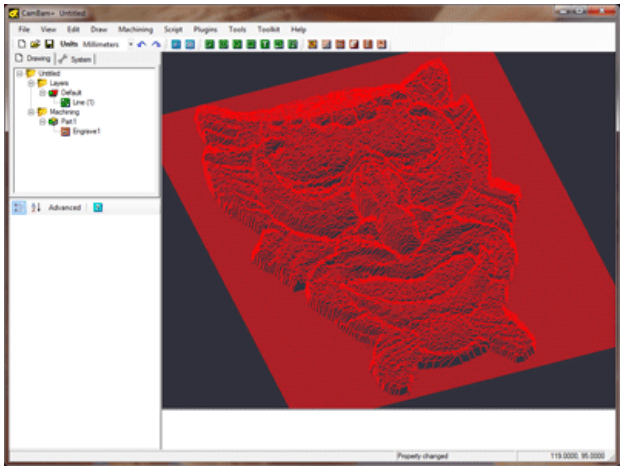
Step 4 - Generate Heightmap

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 here...

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.

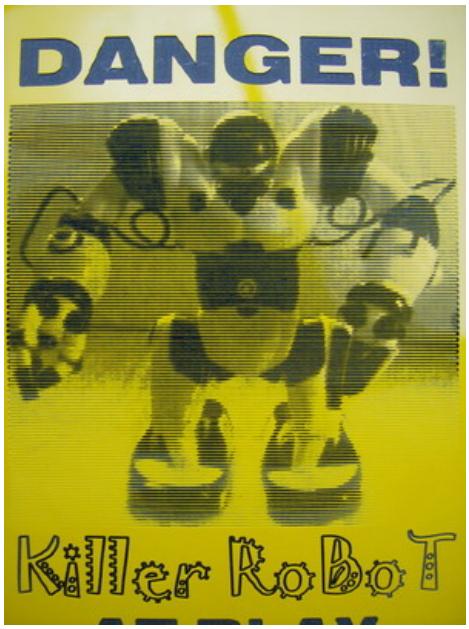
To convert the heightmap into gcode for your machine, right click the ***Machining*** group in the CamBam tree view then select the ***Create GCode File*** menu option.

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. I will try to find some nice examples.

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

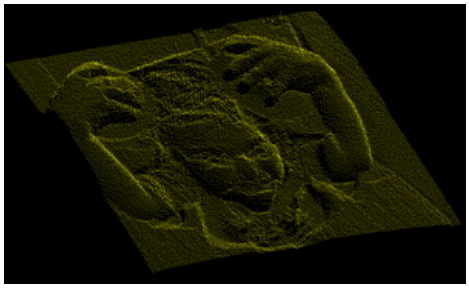
Generate a heightmap polyline as per usual and select the line if it is not already.

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.

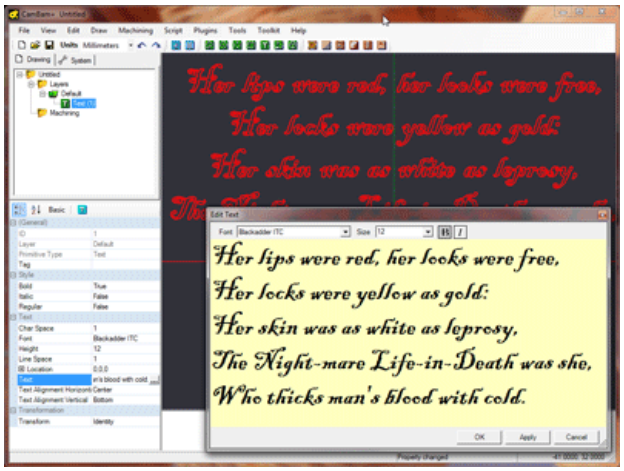
Inserting Text

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.

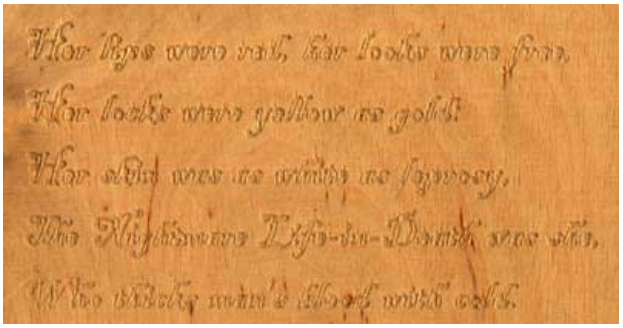
Creating Engraving GCode

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.

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

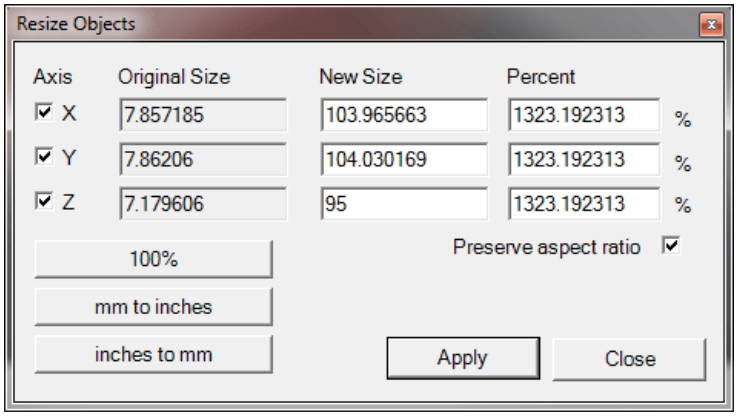
CamBam currently can read .3DS file, .STL and .RAW 3D mesh files. These can be loaded using the ***File - Open*** menu option or by dragging files onto the CamBam drawing window.

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.

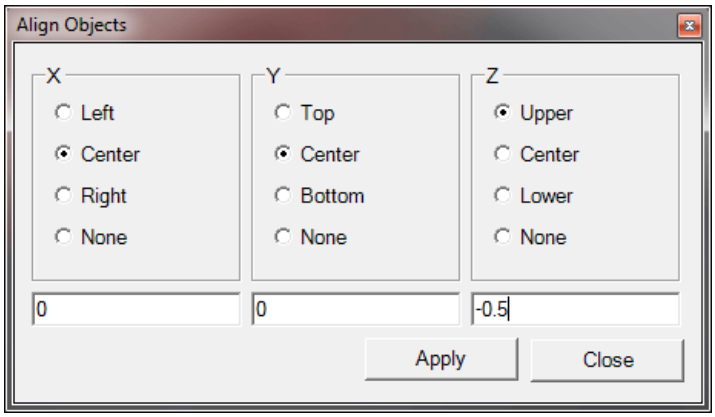
Another alternative is to use free-hand rotation. This is done by selecting objects, then holding the **SHIFT** key while using the view rotation key+mouse combination (ie **ALT** + left mouse drag or Center mouse + Left mouse drag, depending on your configuration settings).

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***.

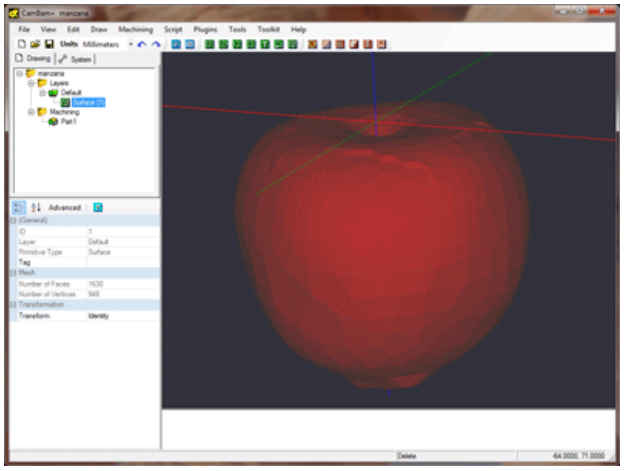
For example, to position an object so that it's lower left corner is at the drawing origin and the highest Z point is just below the stock surface (if using Z=0), use the following alignment values:

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.

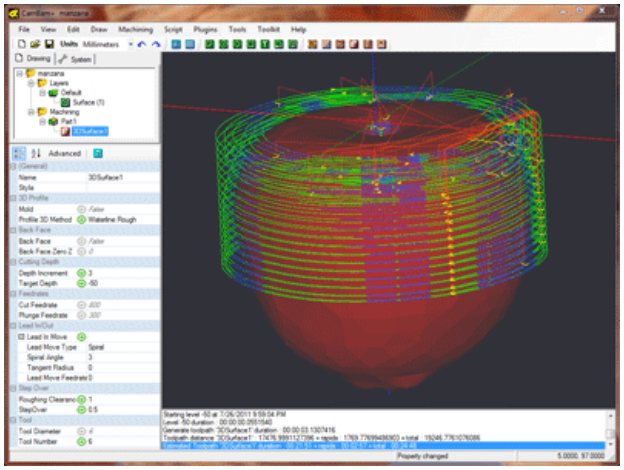
Advanced Properties

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.

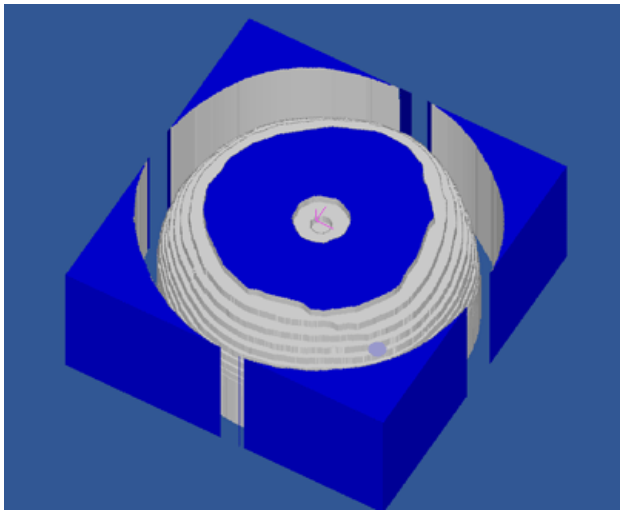
General Settings

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

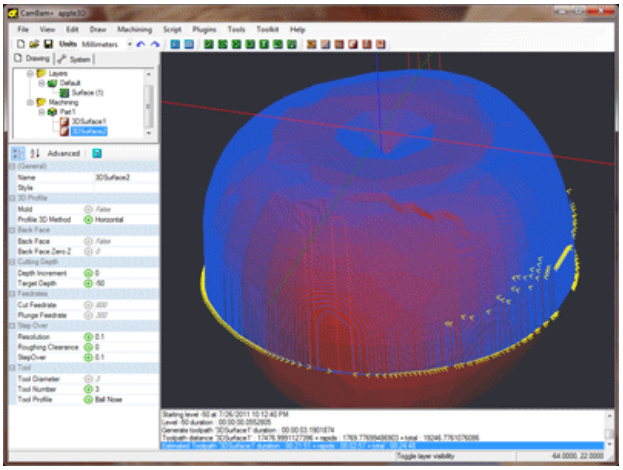
Once the bulk of the material has been cleared by roughing, a scanline finishing pass can be applied. Select the 3D surface and insert a second ***3D Profile*** operation.

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.

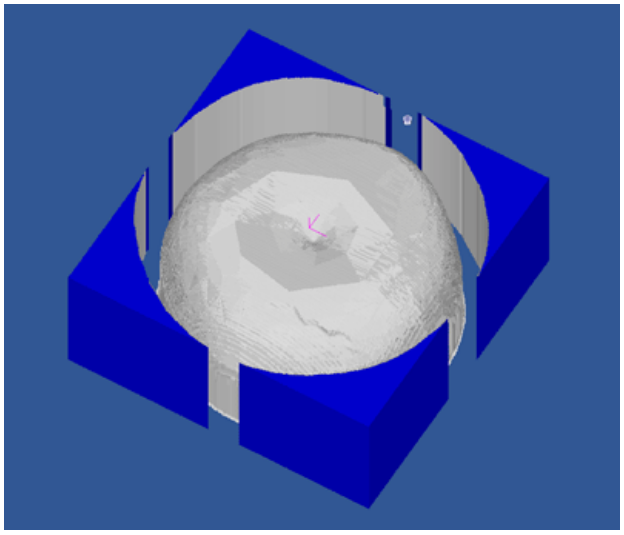
Basic Properties

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.

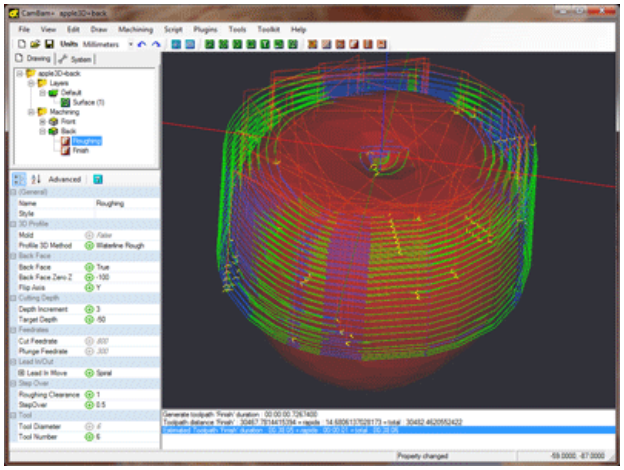
If the top of the stock is referenced as the machine Z=0, ***Back Face Zero Z*** would be the deepest Z coordinate of the model. When the model is flipped, this point will then be at or ideally just below the stock surface (Z=0) plane.

Basic Properties

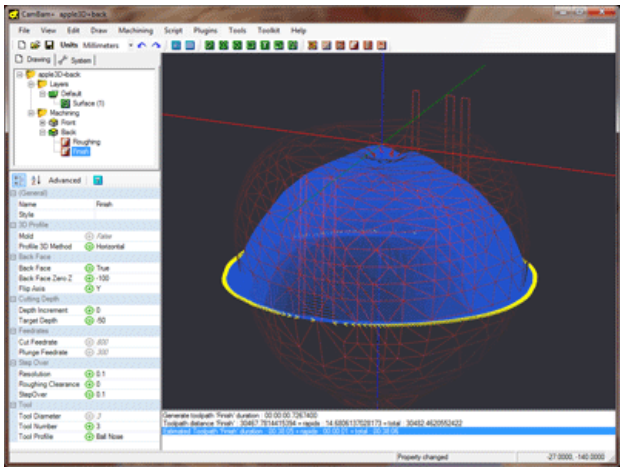
Hint: If a ***Part*** has already been created containing the roughing and finishing passes for the front face, creating the back face can be simplified by copying and pasting the ***Part*** used by the front face and then altering the properties specific to back face machining.

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)

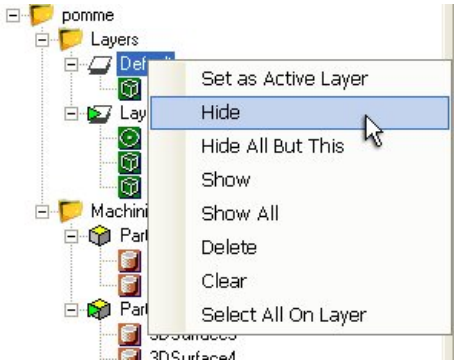
3D Holding Tabs.

There is currently no automatic 3D holding tabs functionality, but this feature is planned for a future release.

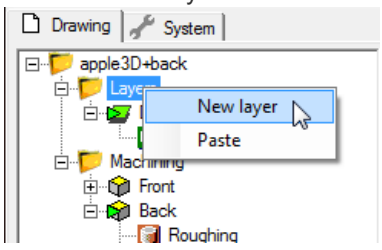
Here is a method to manually create 3D holding tabs or sprues using cylinder meshes.

Extrude a circle

Hide the drawing layer containing the 3D mesh.



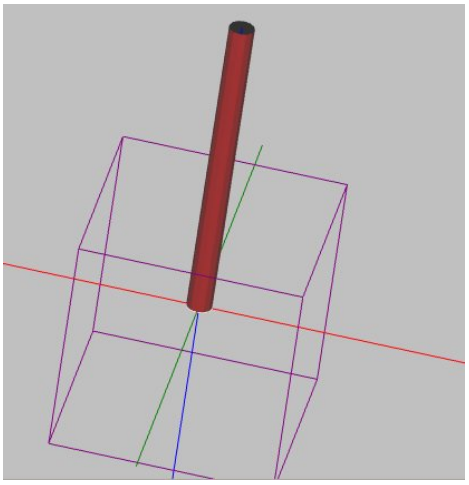
Create a new layer to contain the holding cylinders ('tabs').



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. Enter the number of extrusion steps or facets around the extruded mesh to create.

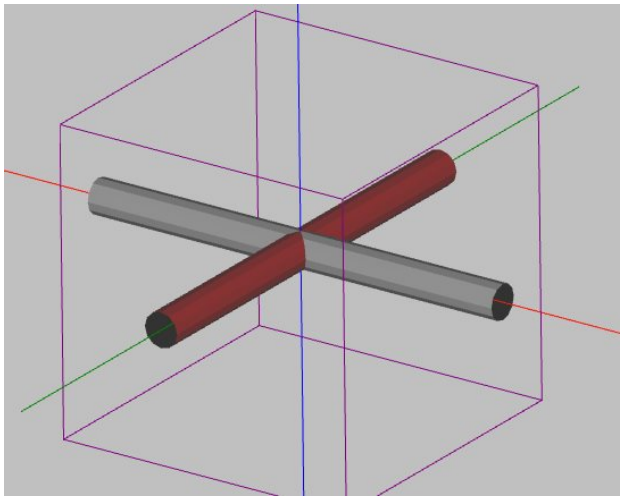
Rotating the drawing view should show a 3D cylinder extending in the positive Z direction.



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.

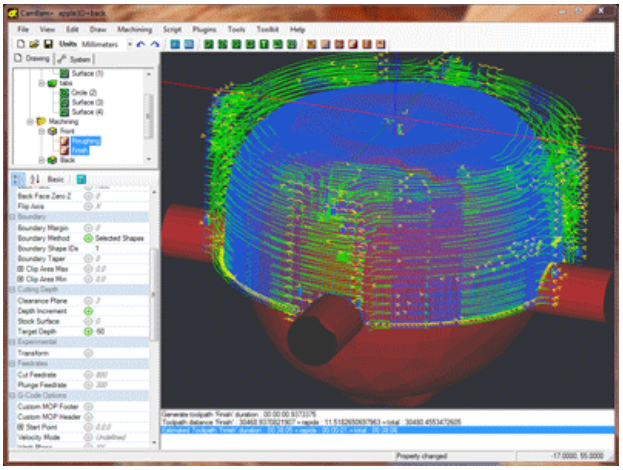


Adjust the machining boundary

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.

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

Note: - The way plugin menu handlers are registered has changed in version 0.9.4. For more information, refer to the What's New? section.

Configuration

Tools - Options

Arc Display Degrees **New [0.9.8f]**	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 **New [0.9.8N]**	The tolerance used when automatic arc fitting is applied in some drawing operations. Zero will use an automatically calculated value.
Auto-Apply Transformations **New [0.9.8]**	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.
Auto Arc Fitting **New [0.9.8N]**	Whether to apply arc fitting in certain drawing operations (such as plane slicing).
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 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.
File Backups **New [0.9.8]**	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.
GCode Editor **New [0.9.8]**	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 **New [0.9.8k]**	If *True*, flatten all layers to a single (unioned) layer.
Gerber - Subtract Layers **New [0.9.8k]**	If *True*, '*Clear*' layers will be subtracted from previous layers.
Gerber - Union Layers **New [0.9.8k]**	If *True*, all shapes on each layer will be unioned together.
Gerber - Union Traces **New [0.9.8k]**	If *True*, each trace will be unioned together. If *False* the traces will be left as line and arc sections.
GLSL Shader Version **New [0.9.8N]**	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 **New [0.9.8i]**	*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 **New [0.9.8k]**	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.
Offset Backtrack Check **New [0.9.8f]**	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.
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* **New [0.9.8]** - 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.
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.
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.
System Path **New [0.9.8f]**	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: *{$common}* - Common application data folder (%ALLUSERPROFILE%). In Windows XP this is typically: \Documents and Settings\All Users\Application Data\CamBam plus 0.9.8\ And in Windows 7 \ProgramData\CamBam plus 0.9.8\ *{$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 **New [0.9.8N]**	If *True*, selected 3D meshes will be displayed using the selection color.
View 3D Wireframe **New [0.9.8]**	If *True*, 3D meshes will be displayed in wireframe mode.
View Background Color	The color of the drawing background.

View Focal Length Scale **New [0.9.8N]**	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 **New [0.9.8N]**	*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 **New [0.9.8L]**	If *True*, prevents gcode creating if errors are suspected in 3D waterline toolpaths.
Worker Threads	Number of simultaneous worker threads to use.

Appendix

New for 0.9.8

Note! As of version 0.9.8f the post processor definitions folder (post), scripts, and samples are now located in the system folder. If you have customised post processors, these will need to be copied from the current folder (usually 'Program Files\CamBam plus 0.9.8\post', to the system folder.
The default system folder is located at:
(Windows XP)
\Documents and Settings\All Users\Application Data\CamBam plus 0.9.8
(Windows 7)
\ProgramData\CamBam plus 0.9.8
Or from within CamBam, select the ***Tools - Browse system folder*** menu item.

New 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.

Machining properties and CAM styles

Machining templates from previous CamBam versions have been renamed ***CAM Styles*** in this version and their behaviour has changed considerably. As with templates, styles are a way of grouping machining parameters into reusable objects to help simplify common machining tasks. Refer to the following section for more detailed information.

CAMStyle documentation...

The property grid has also been changed to show a select minimum number of common properties, as well as any properties that have been changed from default. To toggle between this simplified view and the full list of available properties, use the ***Advanced*** and ***Basic*** buttons at the top of the property grid.

A new system tree display has been added to the user interface. This is selected by clicking the ***System*** tab at the top of the drawing tree.
The system tree is used to manage configuration settings and common library files available to all CamBam drawings.

New [0.9.8f]
A new ***Paste Format*** command has been added to the CAM Style and machining operation context menus, with a short cut key combination of Shift+Ctrl+V. This will copy all the properties of the machining operation or style that has been copied to the clipboard (using Ctrl+C or the Copy menu command), to the selected style or machine operation. The current name and source primitive ID list are preserved. This provides a similar functionality to 'Copy MOP to template' and 'Apply template to MOP' of version 0.9.7 as well as a method of populating style libraries. If used to copy properties from a MOP to a style, it may also be useful to use the new ***Reset to defaults*** MOP context menu command.

Tool libraries

The ***System*** tab also contains a ***Tools*** section where libraries of tools can be defined.

Each tool in the library has a unique Index property. This index is used in the ***Tool Number*** selection drop down in Machining operations, Parts and the top level Machining folder.

Many properties can be stored against the tool definition, but currently the most important ones are ***Tool Diameter***, and ***Tool Profile***. A machining operation will use the tool number to look up the tool diameter and profile information from the library when needed. If an explicit value is supplied for the tool diameter or profile in the machining operation, this will be used in preference to the value stored in the library.

Multiple tool libraries can be defined. These can also be thought of as tool palettes. Tools can be copied and pasted between libraries using the clipboard.

The Machining folder and Part objects also have a ***Tool Library*** property. If these are left blank, the default tool library will be used (Default-{$Units}), otherwise the specified library will be used when looking up tool numbers. The ToolLibrary property will only be displayed under the Advanced property section.

New [0.9.8f]
A new ***Tool Name Format*** property has been added to the tool library. This is a pattern used to generate descriptive names for tools from their properties. The following macros can be used: {$diameter}, {$flutes} and {$profile}.
If the tool name format property has change, the tool library context menu provides a new command - ***Rename all tools***, which will generate new names for each of the tools in the library based on the new pattern.

Other system libraries

Material and ***Machine*** definition libraries are also provided under the system tab. These libraries are largely place holders for the time being and their use will be expanded upon in coming releases.

As of 0.9.8f, the post processors can also be maintained from the system tab.

The system tab also contains a ***Configuration*** folder. This provides an alternative method of accessing the general configuration settings available from the ***Tools - Options*** menu.

New Stock object

A stock object can now be defined at the Machining or Part level. Currently only rectangular block stock objects are supported.

Stock Size is used to set the X, Y and Z dimensions of the stock block.

Stock Offset is used to define the position of the lower left corner of the stock. 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 defines the Z coordinate of the top surface of the stock.

The stock object can be used to automatically calculate 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.

The stock object defined at the part level will take precedence over the stock object defined at the top, machining level. In this way it is possible to define different stock objects for each part if needed. Stock is undefined if it's X Y and Z sizes are all set to zero.

Note: For people using the ***CutViewer Mill*** simulator enabled post processor (Mach3-CV.cbpp), it is now no longer necessary to enter the stock dimensions into ***Post Processor Macros*** machining property. The stock dimensions will now be determined from the stock object and written to the gcode output automatically.

Basic nesting support added.

To make multiple copies of a part, select the part in the drawing tree then expand the nesting property to define a simple nesting pattern using the following parameters.

Nest Method: Change this to *Grid* or *IsoGrid*, 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*.

New [0.9.8f]
Grid Order Controls the direction of the grid layout. For example RightUp will make copies to the right of the original, then move up to the next row.

New [0.9.8f]
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.

New [0.9.8f]
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.

New [0.9.8f]
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.

Support for manually rotating each nest copy are planned soon, as well as APIs for third party nesting applications, which can employ sophisticated algorithms to optimise stock usage.

The file **snap-together-box.cb** in the CamBam samples folder illustrates the use of a nested part.

Holding tab changes

A new ***Tab Style*** property has been added. This can be used to select between *Square* and *Triangle* cross section shaped tabs. Triangle tabs are a new addition and should hopefully reduce the tool marks associated with square holding tab retracts.

Size Threshold is another new property. If a shapes perimeter is smaller than this value, no holding tabs will be added.

Holding tabs can now be added to Text and Region objects.

Changes to Pockets

The method used to generate pocket toolpaths can now be set using a new pocket machining operation parameter called ***Region Fill Style***. This property offers various offset fills as well as horizontal and vertical hatching. The fill style options are the same as those illustrated in the CAD fill region documentation.

The default pocketing method is ***Inside + Outside Offsets***, which generates successive inward offsets from the outer boundary of the selected shapes, as well as offsets radiating outwards from any islands within the pocket.

These toolpaths will union together to form a rather efficient toolpath. This is the same toolpath that is generated by using a profile with a large cut width for a pocket, as described in previous CamBam releases.

The **CamBam.cb** file in the CamBam samples folder contains some examples of the various pocketing methods.

Corner overcut option.

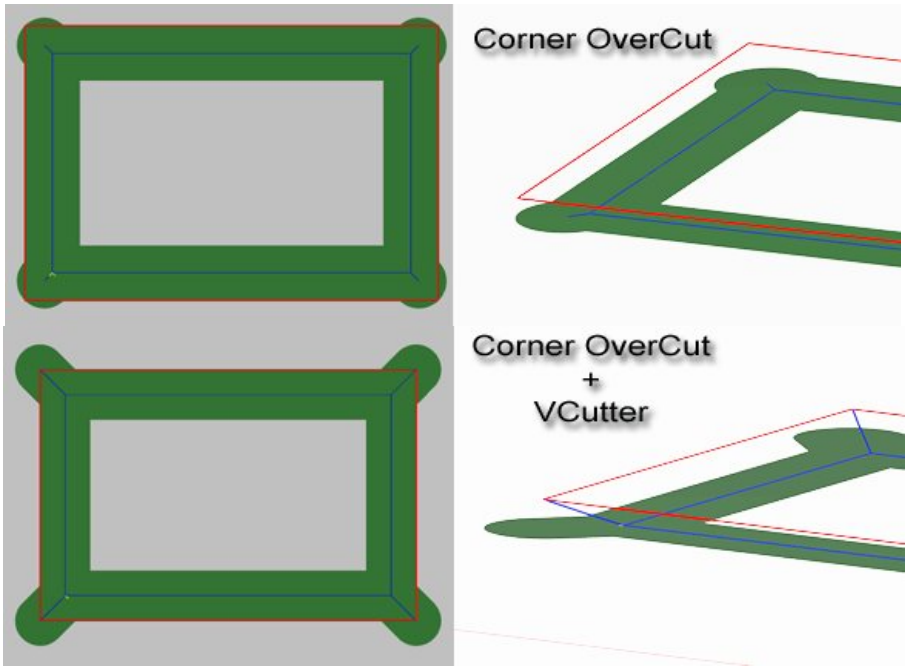
Inside corners, where a tool diameter typically will not fit, can cause problems if they define a shape where another shape needs to be inserted into it, as with a slotted joint or inlay.

A new parameter: ***Corner Overcut***, set to *True* or *False* can be defined, which will add an overcut in towards these inside corners.

If the ***Tool Profile*** parameter for a machining operation is set to *V-Cutter*, a ramped cut is inserted up to the stock surface, all the way into the inside corner. This can result in nice clean square corners when an appropriate V cutter is used.

The file **snap-together-box.cb** in the CamBam samples folder illustrates the use of corner overcuts to clear joints that will be slotted into each other.

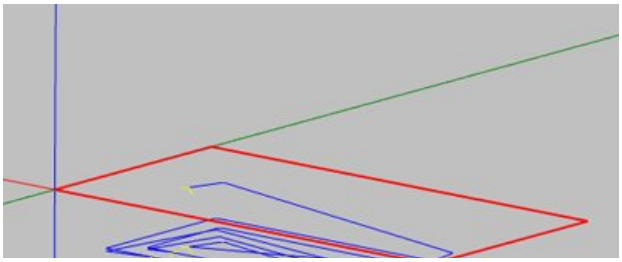
The **CamBam.cb** sample file contains an example of a V corner profile in the last part.

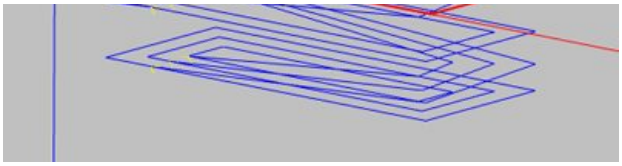


New spiral lead in behaviour and lead out option.

A ***Lead In Move*** with the ***Lead In Type*** set to *Spiral* is a useful way to specify a toolpath that gently enters the stock at a specified angle. In this release, if the spiral angle is set to zero, the ***Depth Increment*** is used to calculate a ramp angle, such that the toolpath will continuously spiral one depth increment in each pass around the object. In this situation, the lead in move replaces the original toolpath, so that the part will be cut in one continuous spiral feed.

An additional (non-spiral) toolpath is added at the final cut depth.





Lead In Type = *Spiral* + ***Spiral Angle*** = *0*

A new ***Lead Out Move*** property has now also been added. Currently only tangential lead outs are supported.

The file **continuous-spiral-feed.cb** in the CamBam samples folder illustrates the use of both continuous spiral feeds and lead out moves.

Changes to 3D profiles.

In previous releases, the 3D volume to be machined was controlled by two 3D points, VolumeMin and VolumeMax. However this was both confusing and inconsistent with other machining operations.

In this release, the 3D operation will machine between the ***Stock Surface*** property, down to the ***Target Depth*** value. This makes the 3D profile more consistent with other machining operations. Note that existing 3D profile files may not work correctly until a correct target depth (or stock thickness), is defined.

A clipping area can still be specified using two new 2D points : ***Clip Area Max*** and ***Clip Area Min***, which only restrict the X and Y area to be machined. These points are only valid when the ***Boundary Method*** is set to *Bounding Box*.

The file **skull-big-foam.cb** in the CamBam samples folder demonstrates the new 3D profile methods. It is also another good example of using multiple parts in one drawing.

> **Note:** with the new CAM style way of working, if a horizontal or vertical finishing 3D operation is defined, it is likely to inherit a non zero ***Depth Increment*** value from the default style. To prevent this, enter a DepthIncrement=0 explicitly in the machining operation, or base the machining operation on a style created for 3D finishing, containing DepthIncrement=0.

Changes to drilling

A new ***Retract Height*** property has been added. For canned cycle drilling methods, the drill cycle will start at this height, and return to it for peck drilling. If retract height is set to 'auto' the clearance plane will be used.

The G98 gcode is now used at the beginning of a block of canned drill cycles, when the Z position is at the clearance plane. This allows the processor to rapid back to the clearance plane move to the next hole position, then rapid down to the retract height.

A ***Depth Increment*** property has been added. This is now used to calculate the pitch for spiral drilling methods. If depth increment is 0, the old method of using the ratio of the plunge and cutting feedrates will be used.

Target Depth is now specified as an absolute Z coordinate, to make it consistent with all other machining operations. In previous versions target depth was a relative offset from the stock surface.

New ***Drill Lead Out*** and ***Lead Out Length*** properties. For spiral drilling, if DrillLeadOut is True, an extra move will be added at the bottom of the spiral to move towards or away from the hole center. If LeadOutLength is positive, the move will be toward the center. If negative the move will be away from the center. If LeadOutLength=0, then a move to the circle center is assumed.

With the new depth increment and leadout properties, it should now be straight forward to define thread cutting operations using the appropriate tool.

Auto ***Hole Diameter*** support. If hole diameter is set to 'auto' and the drilling operation is based on a circle, the circle diameters will be used.

New polyline edit mode

Double clicking a polyline, or selecting ***Edit*** from the polyline context menu, will enter a polyline edit mode, where the control points can be dragged to required positions.

Polyline arcs cannot yet be edited interactively, but this addition is planned.

New text edit mode

When entering new text, or double clicking existing text objects, a new text entry user interface is displayed where the font, size and style can also be defined.

New Open Offset method

Open offsets can be used to generate a ***Closed*** offset polyline from an open polyline. Select an open polyline and then select ***Edit - Open Offset*** from the right click context menu.

An open offset can be useful to draw a 'slot' shape from a simple line or arc.

The line on which an open offset is based can back track, overlap and be quite convoluted. This makes it useful for drawing 'tracks' used for printed circuit boards.

Improvements to toolpath generation speed and reliability

Much of the geometry engine has been rewritten in this release to significantly reduce toolpath generation times and provide a more robust method when dealing with shapes with flaws, overlaps and duplicate segments.

Gerber importer

Basic Gerber file support has been added but is still in development.

The files **opto_input.pcb.output_back.gbr** and **opto_input.pcb.output_back.cb** files in the CamBam samples folder demonstrate the Gerber importer and resulting isolation milling file.

Toolpath view filter

To filter the displayed toolpaths, use the ***View - Toolpath view filter*** menu option, or **`CTRL+F`**.

To display an individual toolpath, check the ***Toolpath index*** filter, then enter the index of the toolpath to view.
The toolpaths are ordered in the sequence that they will be cut.
Scrolling the mouse wheel in the Toolpath index entry box will scroll through the toolpaths in their cutting order.
Pressing the **`Home`** key will reset the index to 0.

To display all the toolpaths at a specific depth, check the ***Z depth index*** filter.
This filter is useful when diagnosing complicated layered toolpaths such as 3D waterlines.

The ***Cut Toolpath Color*** option can be used to hide or color toolpaths that were cut before the currently selected toolpaths.

The ***Toolpath Color*** option is used to highlight the currently selected toolpath.

Lathe operations

An experimental lathe machining operation plugin has been included in this release.

> This is an early, experimental feature and resulting gcode should be used with caution!

Other changes and new features:

- Version 0.9.8 has introduced a number of 'breaking' changes from previous versions. CamBam (.cb) files now contain an internal version number. CamBam will attempt to automatically convert older .cb files to the current version but these files should be manually verified. There is currently no easy way to convert 0.9.8 created files to older versions.
- Some properties may have been moved or renamed. In particular, the view controlling options have been moved from the ***Machining*** folder, to properties listed under the first object listed in the drawing tree. Many new view options are now also accessible from the top and context menus.
- Many transformation operations (Moving, Rotating, Array copies etc) will now automatically alter the shapes rather than just changing the Transform matrix. This should reduce the need for using the Apply Transformations menu option.
- A new ***Transform - Align*** method has been added which greatly simplifies to positioning of 2D and 3D objects.
- ***GCodeOriginOffset*** has been renamed to ***Machining Origin*** and the ***GCodeOrigin*** option has been removed. Note, the ***Machining Origin*** can now be defined at the machining operation, part or top level machining folder levels.
- The new System folder also contains libraries to define tool libraries, machining definitions and materials. This work is still in development and will be expanded upon in coming releases.
- Improvements to post-processor performance should result in much fast gcode generation.
- A new ***Tag*** property has been added to drawing objects, layers, parts and machining operations. This is a general purpose text property and can be used to enter notes. This property may also be used to store parameters used by tools or plugins which automatically generate objects.
- Engraving operations now support the ***Depth Increment*** parameter and will also display cut widths.
- A drawing constraint mode has been added. Hold the shift key when drawing polylines or other shapes, to constrain the point selection along set angles from the previously selected point.
- GDI display mode has been rewritten. It should now be possible to use CamBam in GDI mode for all operations. This is useful for people experiencing problems with OpenGl with some display drivers.
- Machining operations (and parts) can now be copied using the clipboard by cut and paste. If multiple CamBam instances are open, mops and parts can be copied from one drawing to another. As with the previous releases, the drawing objects used by the machining operation can be reselected by dragging selected shapes from the drawing tree onto the machining operation icon.
- A new ***Wireframe*** view option is available from the context menu to toggle the display of 3D meshes in wireframe mode.
- When the evaluation period has expired, the number of gcode lines allowed to be produced has increased from 300 to 500 lines.

Revision P changes

- Bug fix: polyline offset with shapes having fillets near the offset radius.
- Bug fix: nesting icon not being displayed correctly.
- Bug fix: Apply transformation to stick font and arc fitting polylines with back tracks in general.
- Bug fix: Post Processor, adding G1 move to XY start when above clearance.
- Improved open polyline + mixed + spiral lead in handling (in 0.9.8 toolpath optimiser).
- Bug fix: Spline ApplyTransformations not updating fit points.
- Bug fix: Profile CutWidth optimisation when MaxCrossover ~= Stepover.
- Bug fix: Open polyline problems for some large drawings.
- Bug fix: Remove arcs losing end line segments on open polylines.
- Bug fix: Prevent error when setting same named properties from different object types.
- Bug fix: Various fixes (extrema, draw point list...) for shapes that cannot apply transformations (such as rotated rectangles).
- Bug fix: Point list divide geometry.

Revision N changes

- New shader based OpenGL Display Mode, use *OpenGL_Legacy* if errors encountered.
- Holding SHIFT when CamBam starts will force GDI display mode.
- New display option : ***View Projection Mode*** : *Perspective* / *Orthographic*.

- Ball nose compensation added for 3D waterline finish operations.
- New ***Scanline Gradient Threshold*** option in 3D mop, to suppress toolpaths steeper than given gradient (in degrees with 90 degrees vertical)
- New post processor macros {$mop.first.x} {$mop.first.y} {$mop.first.z} {$part.name}
- View rotation will now rotate about center of selected objects.
- ***Use Surface Selection Color*** config option added.
- Right click on drawing tree will not clear current selections.
- ***Arc Center Mode*** added to post processors, plus new incremental options.
- Obsolete Distance Mode and Arc Output All Axis properties removed from drawing machining options.
- Can now set default post processor by right click menu in system tree.
- ***Fast Plunge Height*** = -1 (default), now automatically calculates a default height. Use 0 to turn off fast plunges.
- Line numbering ability in post processor.
- Changes to G18 (XZ) arc handling : Warning! post processor ***Invert Arcs*** should now be set for (most) lathe post processors.
- Holding tabs: new ***Tab Methods*** : *Automatic (Inner/Outer)* to avoid tabs on inside/outside region shapes.
- Holding tabs: new ***Tab Style*** : *Skip*. Similar to a square tab but uses a rapid move across top. Intended for plasma tabs.
- Holding tabs: Bug fixes when applying tabs to toolpaths. Triangle tabs now compensate for tool diameter.
- DXF bug fixes : mirror extrusions on polyline, region ouput, multi-line text writing.
- Python improvements - module support, search paths, treat tab characters as 4 spaces, better error messaging.
- Beginning of flat tangent lead ins are no longer assumed safe to fast plunge down to.
- 3D waterline methods now add roughing clearance vertically (as well as horizontally).
- Improved 3D scanline resolution checks for large stepovers.
- New {$tool.name} and {$tool.veeangle} macros in post postprocessor.
- {$veeangle} macro added to tool library name format expression.
- Edit - Polyline - Remove Arcs operation.
- Polyline apply transformations will now remove arcs if different axis scaling or skewing detected.
- Gerber import now also recognises (*.gbl,*.gtl,*.gbo).
- Tool.ToolChange now overrides post processor Tool Change property. {$tool.toolchange} macro replaced with {$tool.comment}.
- 3D Surface boundary bug fix and added support for region boundaries.
- Toopath to geometry now generates holding tab toolpaths.
- NCFile.SourceFile property is now relative to drawing path.
- Plane slice bug fixes : Plane slice X or Y axis with vertical walls and Z axis with multiple surfaces selected.
- Drilling operations now use ***Start Point*** when optimising.
- Added {$arc.start} {$arc.end} and {$arc.sweep} angle macros to the post processor.

Revision M changes

- Bug fix: Drilling negative depth increment was causing crash.
- Bug fix: Z and F registers are now cleared after a G80 as this prevented a clearance move at program end.
- Post processor {$set()} macro now recognises A,B,C,F,P,Q,R registers and can also set values to NaN (ie undefiend).
- Gerber improvements including Union and Subtract bug fixes.
- Python scripting support adding using IronPython.

Revision L changes

- Added {$index} and {$length} to tool library Tool Name Format macros.
- 3D surface, 'Auto' target depth and stock surface now use model height/depth if no stock defined.
- Degenerate Tolerance property removed from 3D surface machine operation.
- Edit - Open Offset changed to more reliable method in Gerber importing.
- Bug fix: Modal canned cycles now working in post processor.
- Added system config ***Waterline Safety Check*** setting, which prevents gcode creation if waterline errors

detected.
- Bug fix: Post processor, Arc Output = Convert to Lines now working for helical arcs.
- Added new post processor, Arc Output option ***Helix Convert To Lines***.
- Improvements to scanline 3D toolpath optimisation. Corrected Milling Direction behaviour.
- Cut widths now shown for tangent lead in / out.
- 3D scanline methods now only alternate direction when Milling Direction = Mixed.
- Bug fix: Region filler over cutting margin when initial shape too small for any fill offsets.
- Edit - Move, now uses SHIFT key constraint to move along snap angles.
- Bug fix: Fixed back plotting error when gcode values contained leading + sign.
- Bug fix: Backtrack removal code was causing occasional 3D waterline glitches.
- Bug fix: Post processor bug when mixing literal code and macros in move commands.
- New post processpr macros (see post processor documentation for specifics).
- Bug fix: Gerber loading errors when gerber contained blank lines.
- Bug fix: Precision error when joining was causing some unioned segments to be lost.
- Speed improvements to union code. Complex Gerber loading times should be improved considerably.
- Bug fix: Tool library 'Rename All' was potentially causing duplicate tool names.

Revision K changes

- Language translation support. See www.cambam.info/ref/ref.lang for details.
- Reformatted all property and value display names to be easier to read.
- Edit - Subtract now works with multiple shapes.
- Improvements to Gerber file handling including more aperture support.
- Added Roughing clearance support to spiral drill operations.
- Bug fix: 3D surface boundary shape improvements.
- Bug fix: Improved memory handling for fine stepover 3D scanline operations.
- Bug fix: 'Cusp' toolpath bugs fixed.
- Bug fix: Prevent {$common} folder appearing when error handler called.
- Bug fix: Now always assumes XY workplane if no workplane specified in the post processor.
- Transformations on Drill operations now working.
- Bug fix: Boundary bugs on 3D Mold operation.
- Bug fix: Boundary rounding bug on scanline 3D.
- Bug fix: Nesting bug on custom drill scripts.
- Improvements to holding tab toolpaths on spiral lead moves.
- Added ***Spiral Flat Base*** to drill operation to help with thread milling.
- Show cut widths for spiral drill now shows spiral outline rather than filled circle.
- Changes to post processor to aid laser support.
- Engraving operation now uses tool vee angle and depth increment to determine max crossover distance.

Revision J changes

- New PostBuildCommand and PostBuildCommandArguments properties for post processors.
- Point list editor improvements including cut/paste to/from spreadsheet.
- Rotation transformation now prompts for start angle.
- Bug fixed: Was not using leadin feedrate for tangent moves
- Bug fixed: Leadin moves for square holding tabs fixed.
- Bug fixed: Spiral leadout fixed.
- Bug fixed: Various toolpath bugs including polylines with sharp cusps fixed.
- Bug fixed: 'Object reference not set to an instance of an object' error in experimental optimiser.
- Bug fixed: Errors when creating mops with no Part object present.
- Bug fixed: {$tool.length} post processor macro value now read from tool library.
- Bug fixed: Backplotting drill gcode, convert toolpath to geometry fixed.

Revision I changes

- New Machining.InnerTabScale and OuterTabScale values to adjust lengths of holding tabs.

- Holding tab now displays a cross if it fails to insert any toolpath holding tabs.
- Added HoldingTabDragToolpathRefresh config option to prevent slow toolpath recalculations after tab drags.
- Changed 3D Profile boundary behaviour to machine up to boundary and not expand polyline boundaries.
- Bug fixed: Experimental optimiser with depth first on profiles with holes was doing level first.
- Bug fixed: "Unable to create a file that already exists" error when FileBackup=0.
- Bug fixed: Occasional "Key not in dictionary" errors in toolpath optimiser.
- Bug fixed: Toolpath to geometry from NCFile op was creating polylines with NaN values.
- Bug fixed: Polylines with only 2 segments causing offset errors when OffsetBacktrackCheck=True.
- Bug fixed: Entering 'c' in polyline drawing mode was starting circle draw.
- Style tool number select now shows tools from tool library set in active part or machining properties.
- Added a Help - Samples menu option.
- Tools in libraries can now be reindexed by selecting an existing tool index.
- Added new RapidDownToClearance option to post processor. Setting False will prevent this rapid.
- Bug fixes: A number of GDI display mode, drawing and editing problems were fixed.

Revision H changes

- Bug fix: fixed stepover bug causing leadings at each stepover fixed.
- Toolpath filter now works with engraving mop.
- Warning message when refreshing libraries with unsaved changes.
- Warning when post processor missing.
- Bug fix when drawing circles with center not at Z=0

Revision G changes

- Bug fix - Tool details weren't being looked up from tool library for styles with tool numbers.
- Transform bug fixes for align, move, arc fit and splines.
- Toolpath optimisation bug fixes and improvements.
- Engraving 'Auto' TargetDepth changed to StockSurface minus 1 x DepthIncrement.

Revision F changes

- Post processors moved to system folder and maintainable from System tab within CamBam.
- New post processor properties.
    - ***Notes*** - general purpose text notes
    - ***Lathe X Mode*** - For lathe use, sets whether X is Radius or Diameter.
    - ***Lathe Tool Radius Offset*** - If False, the toolpath as at the center of the tool radius is output
      If True, an appropriate tool radius offset is applied.
    - ***X Mode Diameter*** - For lathe use, code to use to set X diameter mode (eg G7 for EMC2).
    - ***X Mode Radius*** - For lathe use, code to use to set X radius mode (eg G8 for EMC2).
    - ***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*** - This has been removed from the .cb file Machining section into the post processor. If set to ConvertToLines, small line moves are used rather than arc commands.
    - ***Arc To Lines Tolerance*** - If ArcOutput=ConvertToLines is used, this value controls the maximum allowed error when converting arcs to lines. Smaller tolerances will result in smoother curves but larger files.
- Config file moved to system folder.
- More nesting options:
  GridOrder, GridDirectionAlternate, FromPointList. See nesting section for details.
- Machining operation and CAM style properties now show an icon to indicate Default, Auto or Value status. Clicking the icon will invoke the context menu to change the status.
- Changing the gcode filename (Machining.Outfile) now uses a macro {$cbfile.name} to avoid copied/renamed cambam files using the previous gcode file name.
- Library maintenance in system tab, can now copy,paste,rename, drag and drop items.
- Tool library now has an automatic name macro to generate tool names from properties using various macros

For example '{$diameter}mm {$flutes} flute {$profile}'.
- New resize gui with scale by size, percent, preserve aspect ratio.
- Performance fixes for holding tabs on text objects.
- Depth first cut order toolpath optimisation improvements. This applies to the Experimental optimisation mode which should be the default.
- New 'Browse gcode folder' Machining context menu function.
- New 'Edit gcode' Machining context menu function, an external editor can be specified in System Configuration, GCodeEditor property.
- New 'Set machining origin' Machining and Part context menu function.
- New 'Set start point' Machining operation context menu function.
- Added Undo/Redo buttons to toolbar and display current undo step on status bar
- New 'Tools - Browse system folder' menu function.
- Heightmap plugin bug fix causing endless loop (Many thanks to Brian Paquette)
- Point lists can now be edited by double clicking.
- Double clicking NCFile machining operations in tree now opens gcode editor.
- DepthRelativeTo property has now been removed. All depths are now considered absolute Z values.
- New hot keys
    - P - draw polyline
    - C - draw circle
    - D - draw point list (D for dots?)
    - R - draw rectangle
    - T - draw text
    - A - draw arc
    - M - measure
    - CTRL+R - rotated selected entities (previously was convert to region)
    - CTRL+E - resize selected entities
    - CTRL+M - move selected entities
    - CTRL+W - write gcode file
    - SHIFT+CTRL+V - past format
    - F1 - Help Contents

Revision E changes

- Fixed bug with non-integer text sizes.
- Added back up file support - creates filename.b# files where # is a number and number of backups set in config option FileBackups).
- Errors relating to library folder refresh.
- Restrict style drop down to searched libraries only.
- Added Right+Left mouse drag view rotate option.
- Some fixes to Gerber importer.
- Added drawing snap to stock corners.
- Bugs fixed relating to spline dragging.
- Bugs fixed relating to rectangle rotating.