

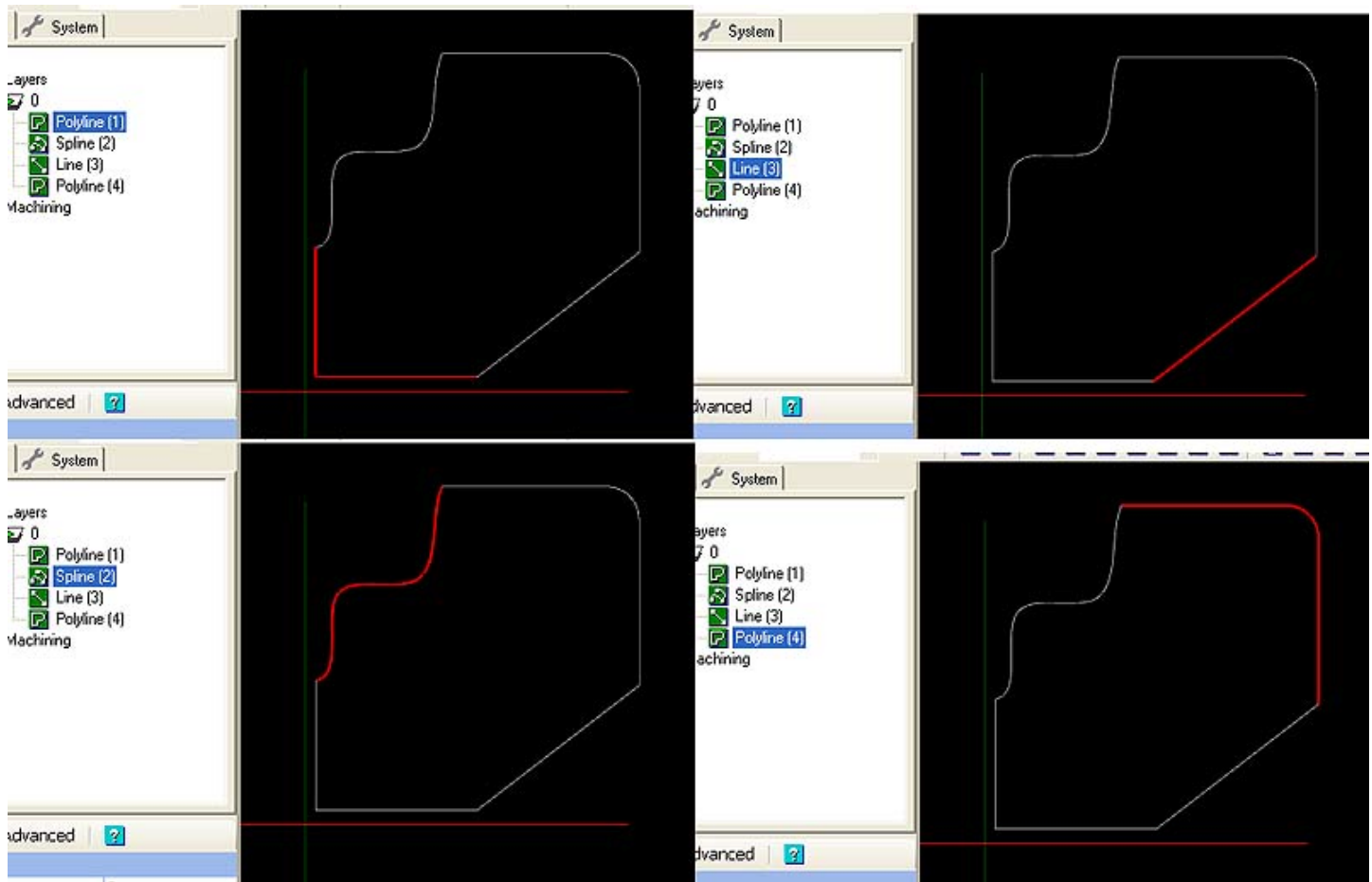
Preparation of drawings prior to application of a machining operation

The importing of **.dxf** files usually requires preparation of the drawing before you can apply machining operations.

This can also be true for drawings made in CamBam, but they usually cause fewer problems.

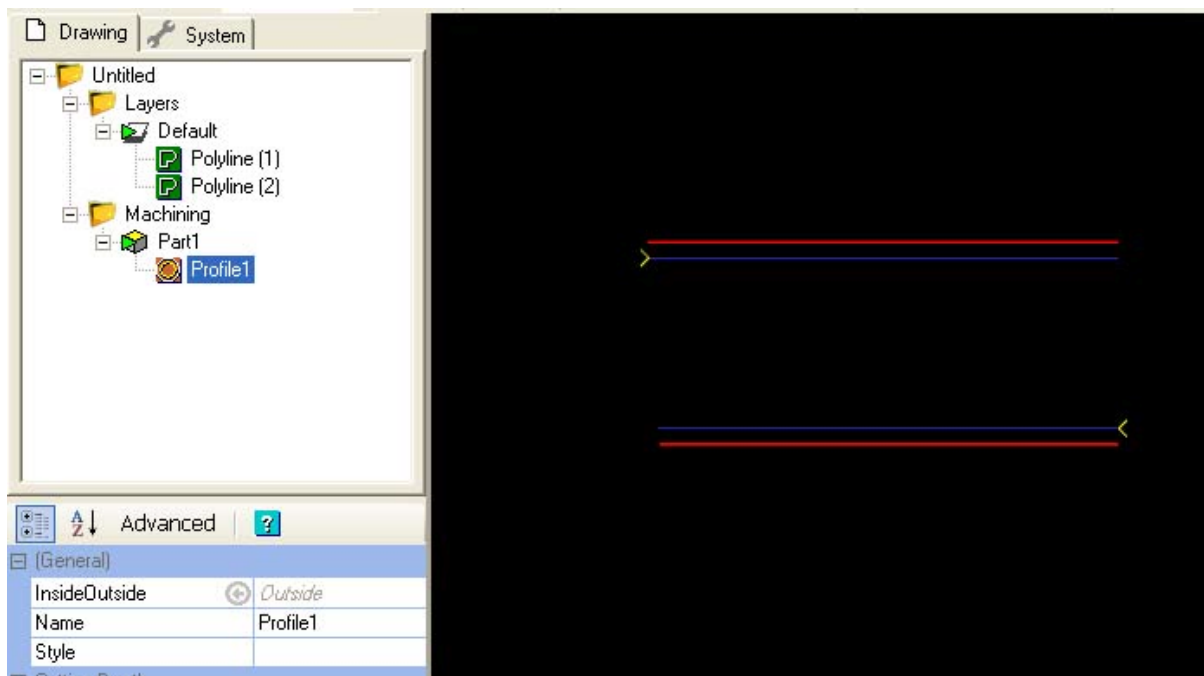
Prerequisites:

As can be seen in the images below, the fact that a shape appears closed does not mean that it is. Here, the shape is in four distinct parts and cannot be used for a profile or pocket machining operation until it has been joined into a single polyline.



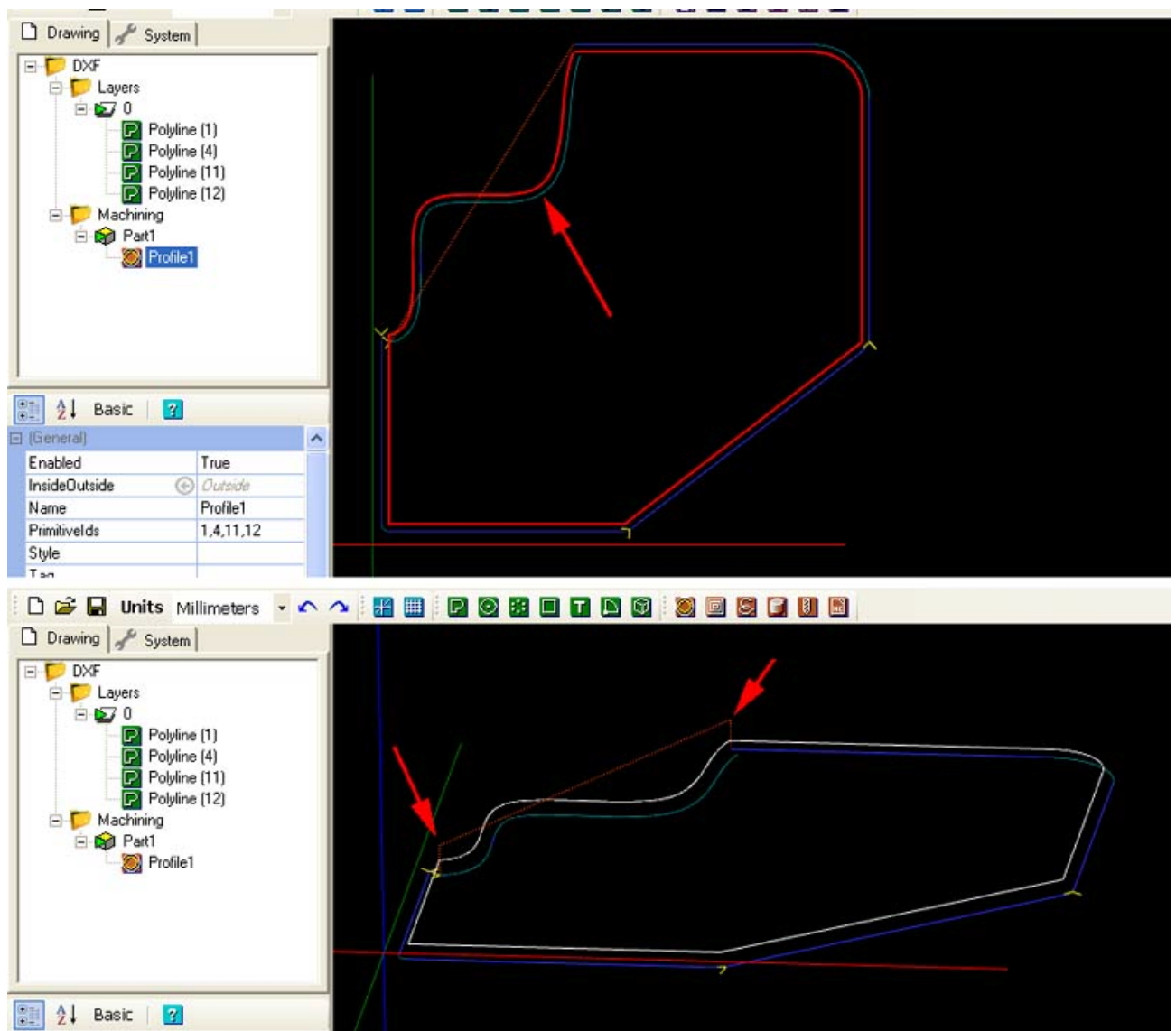
Consider what can happen if you apply a machining operation to an apparently continuous contour which is actually in several parts. Two problems can arise:

First: CamBam will interpret the 'inside' or 'outside' for an open line depending upon the direction in which the line was originally drawn. For a closed shape, that's no problem, the inside is – well - the inside! For open lines, you cannot easily determine which is inside and which is outside after the drawing is done. See the example below.



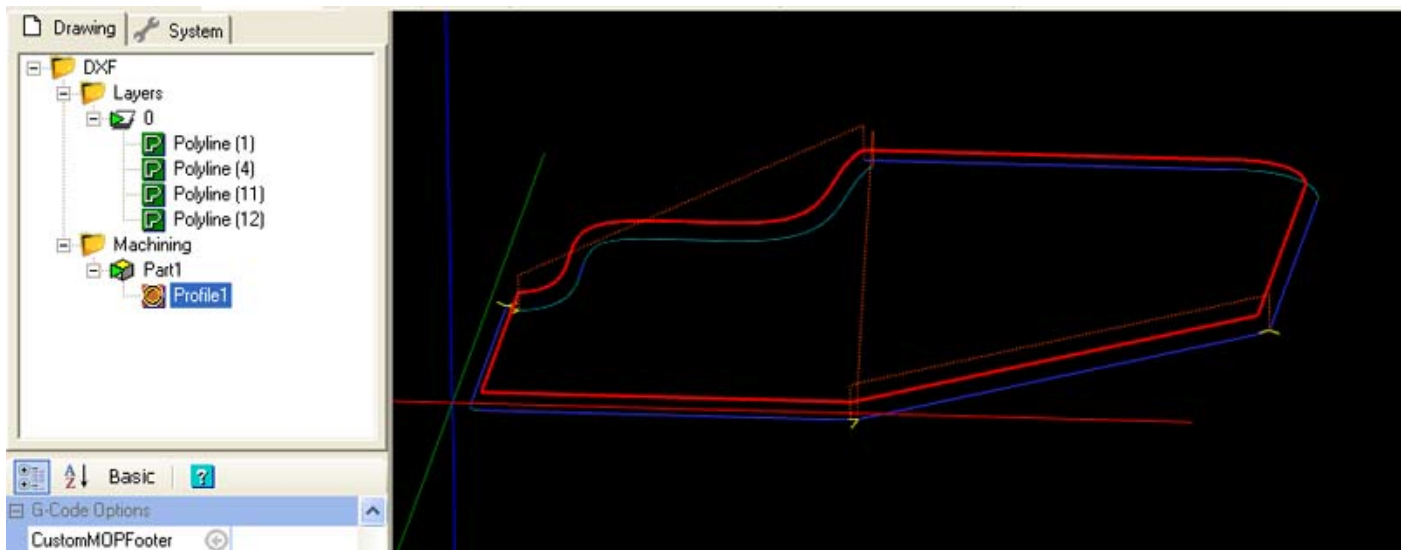
Here, the top line has been drawn from left to right, and the bottom line from right to left. You can see that the starting point of machining and the machined 'sides' are not the same.

Second: The next example shows what a MOp (Machining Operation) could yield with the 'open' shape if it were not converted into a single polyline prior to creating the MOp.

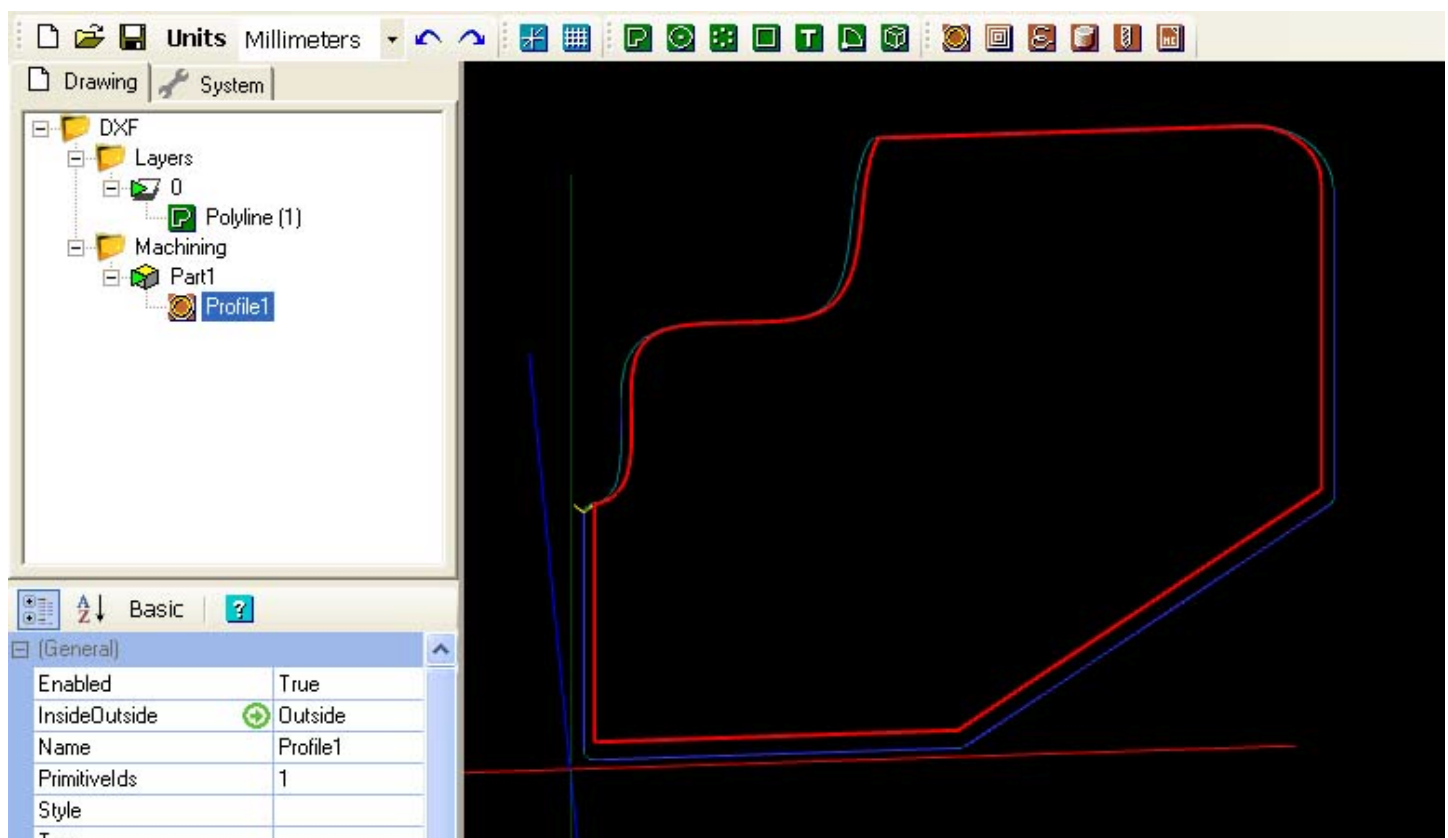


The tool path is inside the line at the top, outside the line at the bottom, and there is also a rise in Z at the bottom.

If you use "*optimization = None*" in the MOP, it's worse, because there is now a rise in Z at the end of each line.



However, after joining all the lines into a single polyline, there are no problems. (The distortion of 'upper' tool paths is because the view has been tilted in perspective – it is not from directly above, so the upper tool paths appear to be too close to the profile. They are not.)



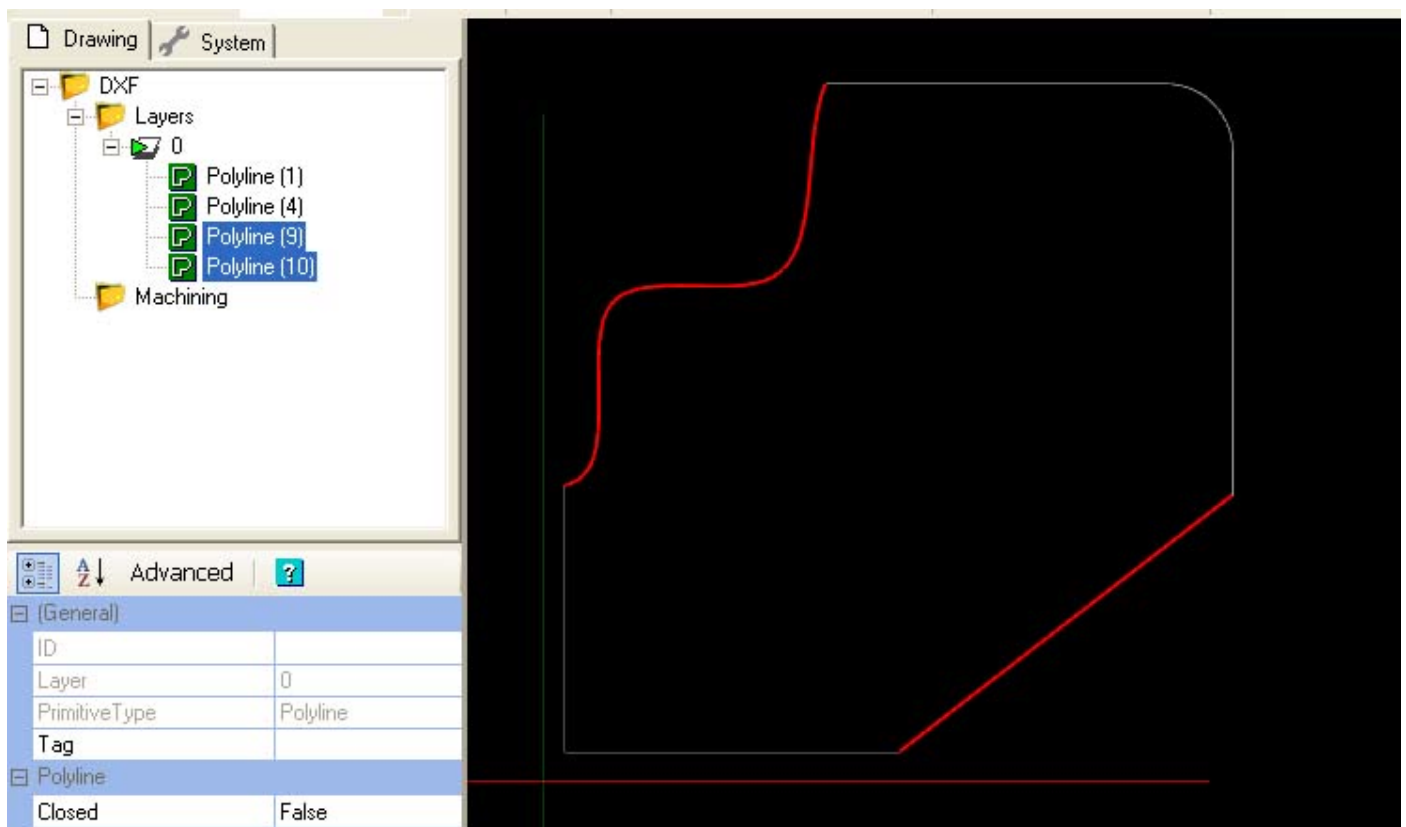
To accomplish this may require several steps:

The first operation will be to convert all the profile lines into polylines.

To do this, select all lines (Ctrl A) and use the **Edit/Convert to/Polyline** menu (Ctrl P). (The fact that some of the objects are already polylines does not matter.)

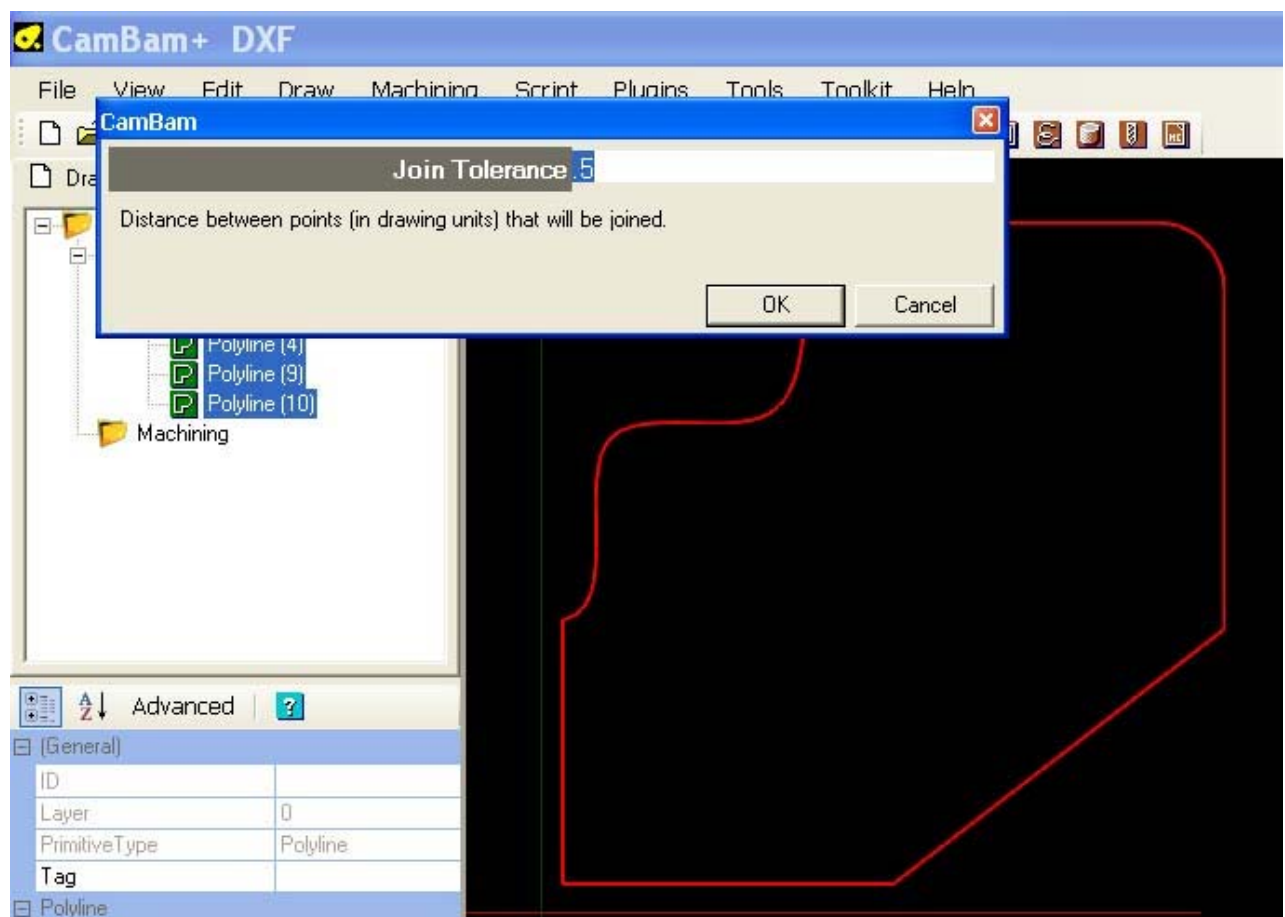
After conversion you'll see two things:

- 1) All objects that were originally selected are now polylines.
- 2) Objects that have been converted into polylines are selected, and those which were already polylines become deselected.

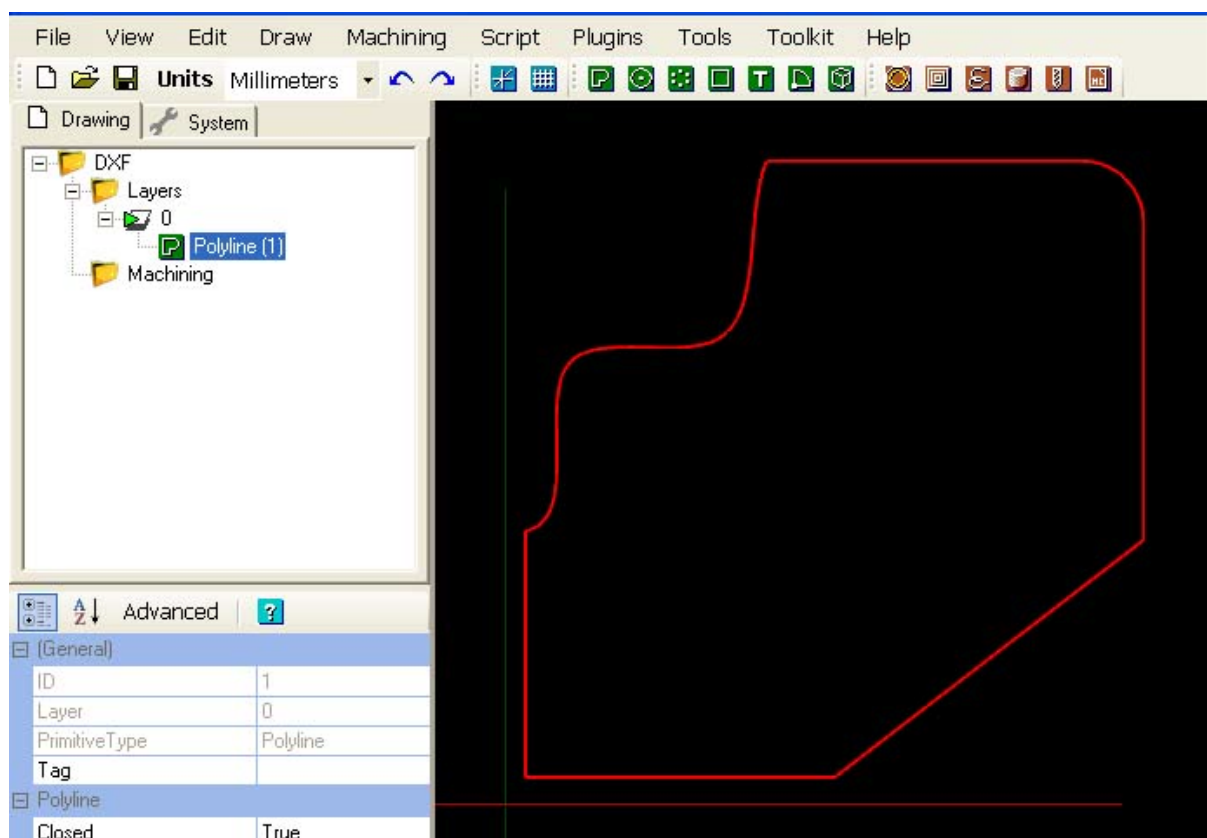


To obtain a shape that consists of a single polyline, you should now select all the polylines you wish to join together, and then use the command **Edit / Join** (Ctrl J).

CamBam will request a 'join tolerance' value, which is the maximum distance apart between their ends beyond which the lines will not be joined. In this case I chose 0.5mm.



The result is that the entire shape now consists of one polyline.



Difficult Cases

It may happen that, despite all these manipulations, it is impossible to join the lines to make an object composed of a single line. This occurs mainly with imported **.dxf** files (especially from SolidWorks).

The problem is because there may be lines that overlap, or that there are duplicates, one atop the other. Sometimes, there are hundreds of small lines of a few tenths of a millimeter long. This is not easily visible when looking at the drawing, but it can be seen if you select the objects one-by-one in the drawing tree. Some are tiny. To join the useful lines, you should remove all duplicate lines and overlapping segments, but it's a long and tedious job to do this by hand. Fortunately, CamBam has a function to perform this action.


This feature is called "**Remove Overlaps**" and it is located in the menu **Edit / Polyline**. This function does not change existing lines, but creates one or more polylines representing the "joined" composite contour of the selected lines. The problem often is to find the useful polylines that were created in the midst of all those which already existed. To do this, you should un-complicate things by creating a new layer into which the resulting new lines will be deposited. Make that layer is active before proceeding.

Any new object created in CamBam is always created on the current 'active layer'. It's the same thing with **Remove Overlaps**; new lines will be created on the active layer.


To do this, create an empty layer (which will automatically become active after its creation), or activate an existing empty layer, (Layer context menu, Set as active layer). This way, any new lines that are created can be easily accessible and editable, and the source lines will be easy to remove after you're satisfied by the results (for example, by deleting the entire original lines layer).

The active layer is always represented with a green arrow icon.



As can be seen in this video,  the 'problem' shape in the example is made up of many overlapping lines, and a join operation will not fix that. It may reduce the number of lines, but it can also fail to create a single polyline in the desired shape. You can use the **Remove Overlaps** function to solve the problem.

Initially, create or activate an empty layer where the new lines will be generated, then select all the lines in the drawing tree and apply the **Remove Overlaps** function.

CAMBAM will request a cleaning tolerance value. I have good results with values between 0.1 and 0.01 (in mm). Feel free to try other values if you do not get what you want; you can always use UNDO and try again. If you get what you want, then you can delete the unnecessary lines. Here, I delete the entire original layer with everything that is on it. 

Of course, this entire example is an artificially created problem. Here, recoveries and other problems are easily visible, and could be manually fixed. In a real file it could be a different story. In some cases, it may be necessary to repeat the join operation on the resulting polylines several times, and to remove some unnecessary line segments created by Remove Overlaps.

(David – dh42, ed. LS March 3, 2011)