Chapter 4

Polylines

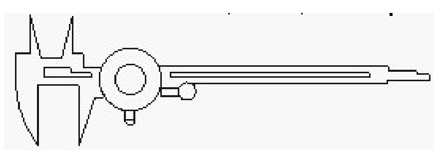
What are Polylines?

A polyline is a chain of lines, arcs, b-splines, Bezier curves, and some other types of geometry, all connected together, with or without a gap between them. Polylines can also be set to be ***symmetrical***.

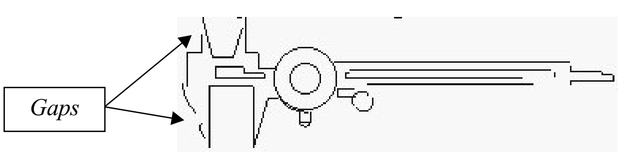
The nodes in the polyline can be set to either ***tangent*** or ***loose***, and between the nodes are the ***links*** of the polyline. A loose node allows the adjacent links to swivel with no regard to the position of the other. Some CAD systems refer to such nodes as "***cusps***". A tangent node forces the two links to be tangent at all times, even when you move the nodes. As you move the links or nodes, MillWrite maintains tangency if the nodes are tangent, or it merely stretches the adjacent links if the nodes are loose.

Why use polylines?

It is easy to create something by building it from individual lines, arcs, circles, etc:

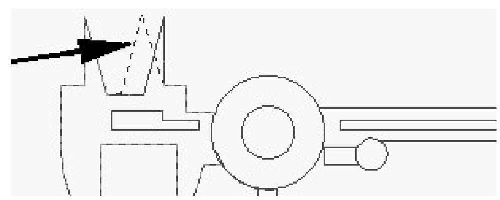


But when you decide later to alter the geometry, the individual pieces of geometry become disconnected, and arcs may no longer be tangent. You then have to spend time closing all the gaps and fixing the arcs.



But if you draw items with polylines, the lines, arcs, and splines remain connected, and arcs remain tangent (if you specified they be tangent).

For example in the image below, a line is being moved to the left, and the lines that are attached to its end nodes are also being moved automatically by MillWrite.



Tangent and Loose Nodes

If you set MillWrite to show all nodes, you can see all nodes of the polylines by touching it with the mouse. (As mentioned in Chapter 1, press the F5 key to turn on or off the showing of nodes.) As seen in Figure 4-1, the tangent nodes are identified by a small square, and the loose nodes are identified by a small circle. The start and end nodes are also circles, with the start node the largest circle.

You can set a node to be tangent or loose by putting the mouse over it and then pressing the ***T*** or ***L*** keys. You can also set the mouse button to have the function of toggling nodes to/from tangent and loose. This function is found in the "*Node Menu*" (the Node Menu is described in Chapter 6).

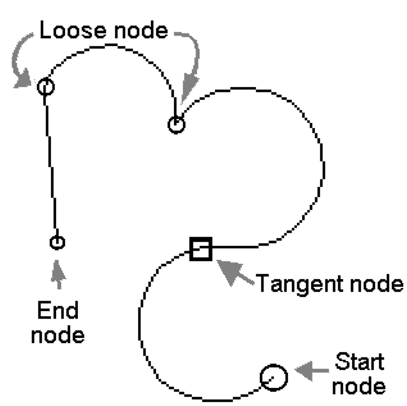


Figure 4-1

Start nodes and End nodes

All polylines have a node that is its ***Start*** node, and they all have an ***End*** node. If you put the starting node and ending node on top of each other, MillWrite will make a **closed** polyline. The start node will still exist, but there will not be any end node.

When a polyline is closed, you can set any of the nodes to be the start node. Here are two reasons why you would want to change the location of the start node.

1) When you want to engrave a polyline you set its CAM operation to cut **centerline**. MillWrite will create a tool path that starts at the start node. Therefore, if you want the tool to start at a specific node, set that node to be the start node.

2) If your polyline consists of tangent arcs, and if you want the arc at the start to be tangent with the arc at the end, you have to move the start node. The reason is that this version of MillWrite does not make arcs tangent at the start node. The start node can be tangent only if the entire polyline is a single spline.

For example, in the upper heart, the start node is at a location that you would want to be tangent. So you have to set some other node to be the start node, such as the node at the point of the heart, as seen in the bottom image. After setting that other node to be the start node, you can set the previous start node to be tangent.

You can set which node is the start node by picking the function from the **Node Menu** that lets you set the start node, and then you click the mouse on the node you want to be become the start node.

Polylines have a direction

The links of a polyline are in a certain sequence; ie, the polyline has a ***direction***. The direction is simply from the start node to the end node. You can reverse the direction of a polyline by putting the mouse on it and pressing the R key, which stands for *Reverse Direction*. You will be reminded of this keyboard command in the lower right corner of the screen when the mouse touches a polyline.

You can see the direction of a polyline when you touch it with the mouse. MillWrite draws a small arrow in the direction of the polyline. (You can see these arrows in the images of the hearts at the bottom of this page.)

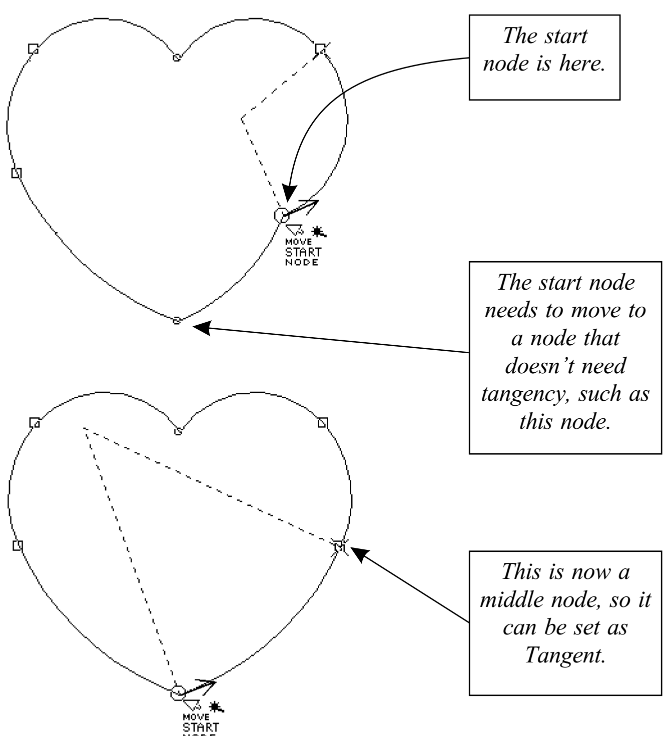
Why would you care about a polyline's direction?

One reason you may care about the direction of a polyline is when you are engraving it. MillWrite starts the tool at the **start** node and follows the links in a certain direction until it gets to the **end** node.

But the main reason you may care about the direction is when the polyline has tangent arcs. When you move a node that has tangent arcs on both sides, MillWrite leaves the arcs that **precede** the node alone, but the arcs that **follow** are altered to maintain tangency. The significance of this is best explained with Bart Simpson.

As seen in Figure 4-2, Bart's mouth is a polyline of tangent arcs. The small squares identify the nodes as tangent. The polyline starts at Bart's nose. The start of a polyline is identified by a large circle.

The mouse is touching one of the middle nodes. If you were to move that node, the arcs that **follow** it will change their radius in order to remain tangent. But the arcs that **precede** that node will remain exactly as they are. Figure 4-3 shows what happens when the node is dragged upwards. Figure 4-4 shows



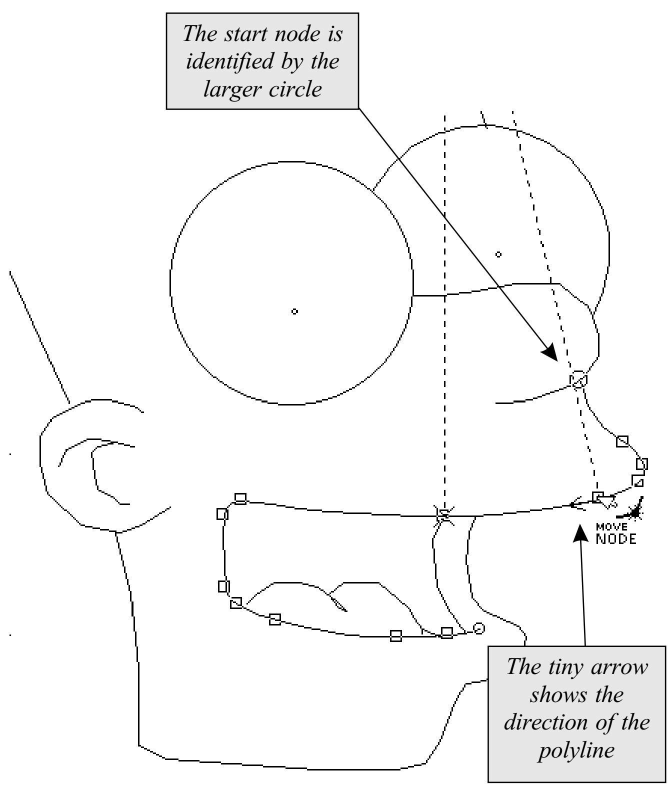


Figure 4-2

what happens when that same node is pushed downwards. Bart's lip stays exactly as it is in both cases, but his mouth changes shape as the node moves.

If you wanted to change the shape of his mouth then this is exactly what you wanted to happen. But if you were trying to change the shape of his lip, and if you wanted his mouth to remain exactly as it was, then you would have to reverse the direction of the polyline.

Figure 4-5 shows the the same polyline except now the direction has been reversed. Figures 4-6 and 4-7 show that same node being pushed up and down. However, this time Bart's mouth remains exactly as it is and only his lip changes shape.

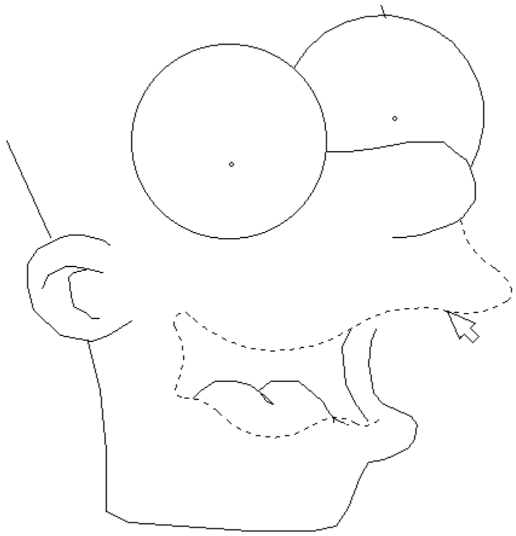


Figure 4-3

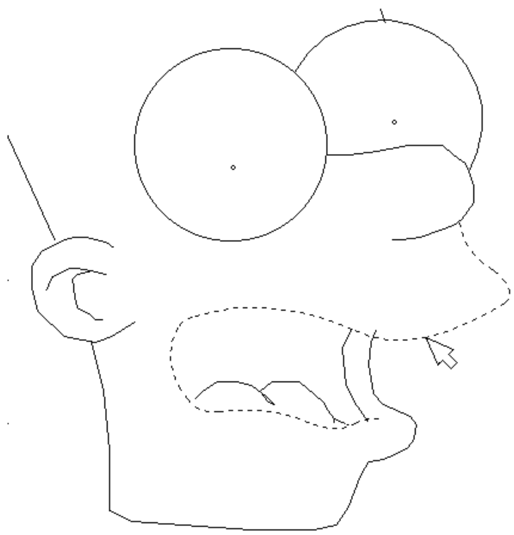


Figure 4-4

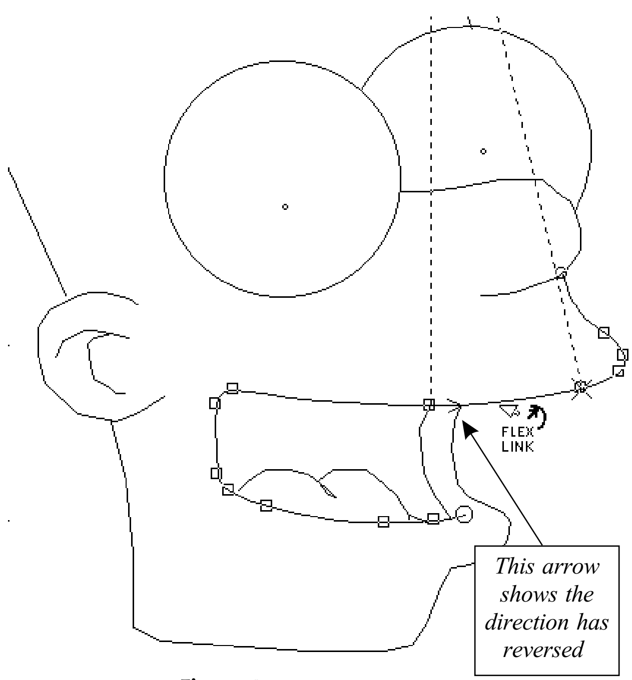


Figure 4-5

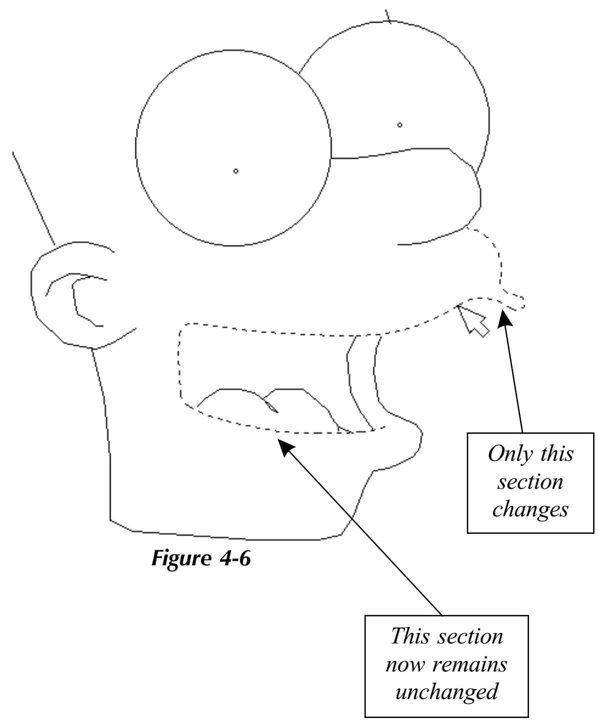


Figure 4-6

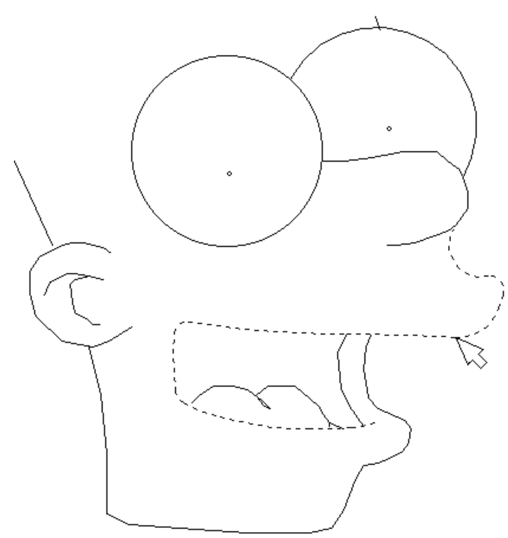
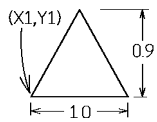


Figure 4-7

Creating polylines with the keyboard

When you are at the drawing page, you can create a polyline of lines simply by moving the mouse to a point in the drawing where it is ***not*** touching any geometry, and then typing X and Y coordinates. You do ***not*** have to select any menu items or click any mouse buttons to do this. Rather, as soon as you type a digit (ie, 0 through 9) or an * or a minus sign, MillWrite assumes you want to create a polyline.

As an example, assume you want to create the triangle in the image to the right. You know the coordinate for the lower left corner of the triangle, and you know the height and length of the triangle.



Let's start by entering the coordinate for the lower left corner. Just start typing the X and Y coordinates, ie, type 1,1. You do not need to specify the letters X or Y because the comma will separate the X from the Y. (The format you use for entering coordinates and distances are described in more detail in Chapter 6.) The screen will change as soon as you type the first digit. Those three keystrokes specify the first coordinate. The screen will look like that of Figure 4-8.

Notice the upper left corner is showing you the coordinate of a ***reference*** point. (The reference point is set to X0.0,Y0.0 because you've never used it before.) The reference point is the small circle, and the **X** marks the location that you are specifying. The line between the reference point and the **X** is to show you the distance and angle you are specifying from the reference point. Of course, in this case we don't need the reference point. The reference point is useful mainly when you specify distances and/or angles. But if you wanted to change the reference point, click the left mouse button at the point you want it to be.

After you press the Enter key, MillWrite will put a small circle at X1,Y1, and that point will become the new reference point. Also, underneath the coordinate you just specified, MillWrite will show you that you just created one node and zero links (Figure 4-9).

Now to create the top node of the triangle. You could calculate its XY coordinate (ie, X1.5, Y1.9), and then type **1.5,1.9**, but MillWrite lets you enter distances and directions from the reference point, so let's do that instead.

You need to specify a distance of 0.5 inches in the positive X direction and 0.9 inches in the positive Y direction. To specify distances, type * in front of the distance. Therefore, to specify a distance of X.5,Y.9 you would type ***.5*.9**

A comma will not hurt anything, but you do not need to specify a comma in this case because the * separates the values.

Figure 4-9 shows the screen after you've type those distances. They are distances from the reference point, and the reference point is the node you previously created. The **X** shows the location you are now specifying. Press the Enter key and notice that the reference point changes to the node you just created.

Now to create the node at the lower right corner of the triangle. The reference point is now the top node of the triangle, and the lower right corner is at a distance of 0.5 in the positive X direction, and 0.9 in the negative Y direction from that reference point. So you would type ***.5*-.9**, as seen in Figure 4-10.

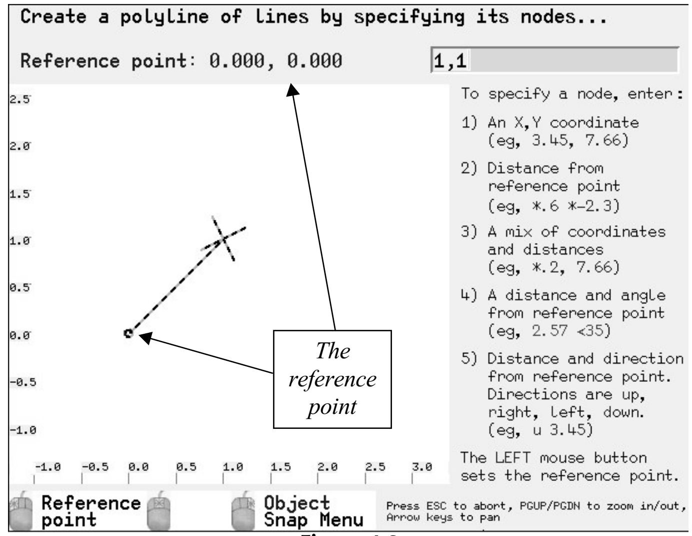


Figure 4-8

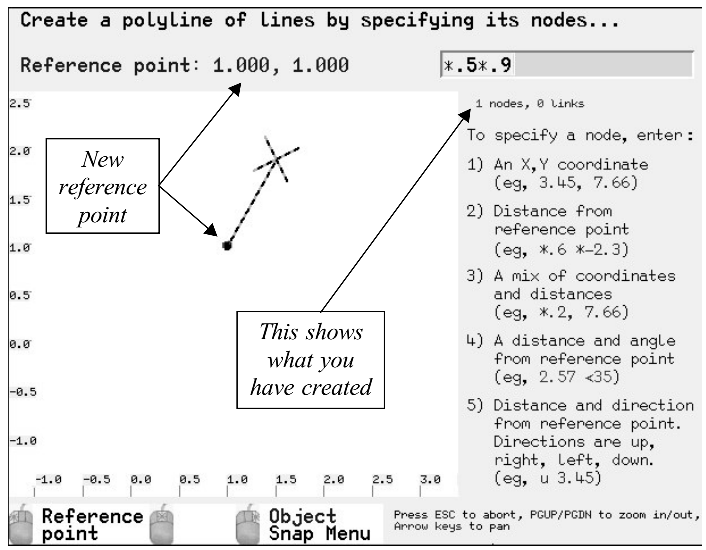


Figure 4-9

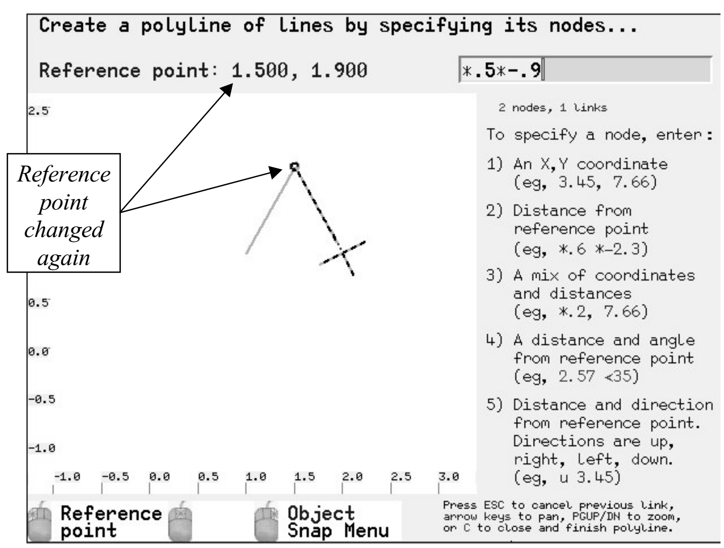


Figure 4-10

Note: If you were to ***change*** the reference point to the lower ***left*** corner of the triangle by clicking the left mouse button on that node, then since the lower right corner is at a distance of 1 inch in the positive X direction and zero in the Y direction, you would type ***1*0** or just ***1** to create the lower right node.

This triangle is so simple that it doesn't matter which reference point you pick, but you may encounter situations in which you need to change the reference point in order to make it easier for you to create the next node.

When you press the Enter key the node you just created will become the reference point, as seen in Figure 4-11.

The final point of the triangle is the same as the starting node of the triangle. You could type **1,1** again, or you could type the distance from the reference point, which in this case is ***-1*0**, or you can type the letter **C** to tell MillWrite to close the polyline. If you want the polyline to form a closed loop, it is better to press the letter **C** because MillWrite will set the parameters of the polyline to **closed**.

Although typing the values for the nodes creates a polyline of straight lines, you can later change any of the lines into arcs or splines.

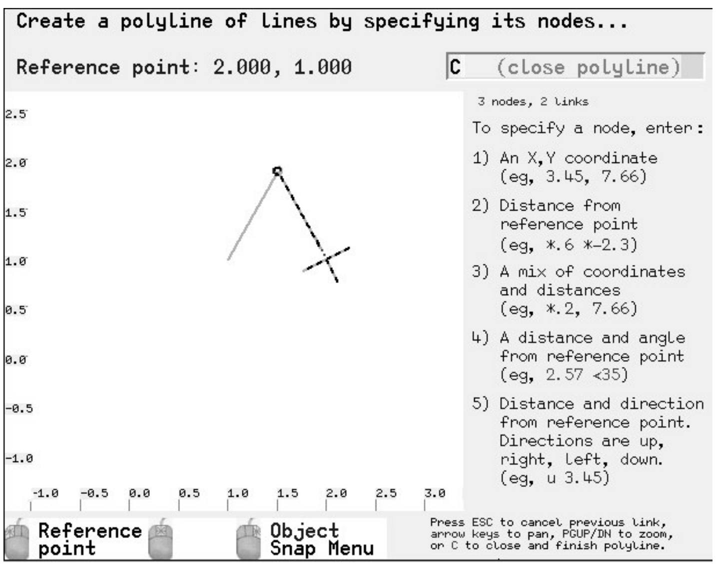


Figure 4-11

The keys you press to create this triangle are:

1,1 Enter
.5.9 Enter
.5-.9 Enter
C Enter

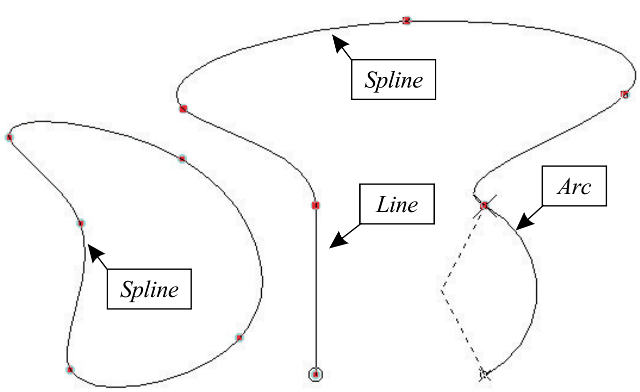


Figure 4-12

B-Splines

B-splines can be independent polylines, or they can be part of a larger polyline.

When a spline is the only geometry in a polyline, you can set the last node of the spline on top of the first node, creating a closed loop. To make the loop smooth, you can set that common node to be tangent. MillWrite refers to as that common node as the **Start Node** because the spline starts at that node.

For example, in Figure 4-12, the spline on the left has six nodes, and the last node on top of the first node. The starting node was set to be tangent to create a continuous, smooth loop.

When a spline is part of a polyline, one or both ends of the splines may have a line or arc attached to it. MillWrite allows you to specify the node between them as tangent or loose. If you set the node between a spline and a line or arc to be tangent, only the spline will be affected. Specifically, as you move the nodes, the spline will change its shape at the node to maintain tangency with the line or arc, but the line or arc will never change its shape to maintain tangency with the spline.

For example, in Figure 4-12, the spline on the right is part of a polyline that contains other geometry besides the spline. The polyline starts with a vertical line, and then comes a spline that is tangent to it, and it ends in an arc (the dashed lines shows the center point and radius of the arc). The spline is tangent to the arc also. These two splines can be found in the job file “Sample Geometry Types” that are included with MillWrite.

When to use splines?

Spline have advantages and disadvantages. Their main advantage is that it is much easier to create complex, smooth curves with splines compared to using a series of tangent arcs.

One of the minor disadvantages of splines is that they create more lines of NC code compared to arcs. They also require more computer time to process, so a drawing that has hundreds of splines will be slightly slower to draw and pocket than if you used tangent arcs.

Tangent and loose nodes in a spline

If you set one of the middle nodes in a spline to ***loose***, the effect is the same as if you created two, independent splines that are connected together with a loose node. The loose node creates what some CAD programs refer to as a “cusp”.

The opposite is also true; ie, if you create a polyline that has two splines in it that are joined at a ***loose*** node, and if you change that node between them to ***tangent***, you have two splines tangent to each other. MillWrite will then combine them into one larger spline.

To summarize this, a spline is cannot have a loose node within it. A loose node breaks a spline into two, independent splines. A spline requires all its nodes to be tangent. Furthermore, MillWrite will combine two tangent splines into one larger spline.

There is no limit on the number of nodes you can have in a polyline, other than the limits of your computer’s memory. However, MillWrite has a limit of 200 ***consecutive*** tangent nodes in any one spline. If you find that you need more than

200 consecutive tangent nodes, let us know or we will assume nobody does.

As an example of how this limitation might affect you, if you create a polyline with 5000 nodes, it is OK to have 200 consecutive tangent spline nodes if at the end of the 200 nodes you have either a loose node, a line, or an arc. Then you could have 200 more consecutive tangent spline nodes if you wanted.

Bezier curves

You can convert polylines or sectons of polylines to Bezier curves. However, MillWrite provides only partial support for Bezier curves. For example, MillWrite does not make Bezier curves tangent to other geometry. The addition of Bezier curves is mainly to let you see what they are and experiment with them to find out if anybody wants them. I personally find them annoying because the control points are not on the curve, which makes altering the shape of the curve impossible if you zoom in on the curve. But since people have different tastes, if you want Bezier curves, just let us know.

Symmetrical Polylines

You can set a polyline to be **symmetrical** and MillWrite will maintain its symmetry as you move or alter the links of the polyline. For example, the tree in Figure 4-13 has been set to be symmetrical. If you move any of the nodes or branches on that tree, MillWrite will move the opposite nodes to maintain symmetry.

When you touch a symmetrical polyline with the mouse, a symmetry line appears with a node at each end. When you see that symmetry line appear, you know that that polyline has been set for symmetry.

You can move the entire polyline by grabbing that symmetry line and moving it with the mouse. The symmetry line has a **start node** which is identified by a large circle. If you move the *other* node, which is identified with a small circle, you can rotate the polyline.

The tree in Figure 4-13 forms a **closed** polyline because the start node is on top of the end node, but symmetrical polylines do not have to be closed.

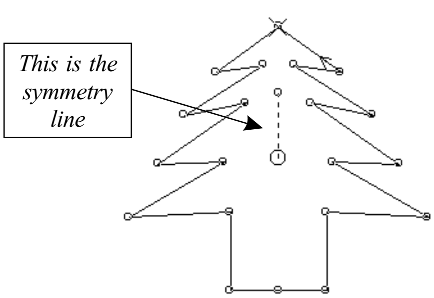


Figure 4-13

How to make a Symmetrical Polyline

There are two different ways to make symmetrical polylines. One is to draw one half of the polyline and let MillWrite mirror the other half for you. The other method is to draw both halves of the polyline and let MillWrite make them identical to each other.

This example will show how to draw one half of the polyline and let MillWrite complete it. Assume you want to create the symmetrical polyline seen in Figure 4-14. The easiest way is to draw one half of that polyline, as seen in Figure 4-15.

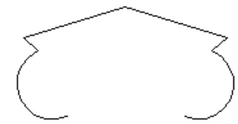


Figure 4-14

After you draw half of the polyline, you let MillWrite know that you want to convert it to a symmetrical polyline. You begin by moving the mouse over the polyline. This causes its parameters to appear along the right side of the screen, as seen in Figure 4-15. Then move the mouse to the parameters area, click on the **Symmetry** data field, and set the field from **None** to **Dual**.

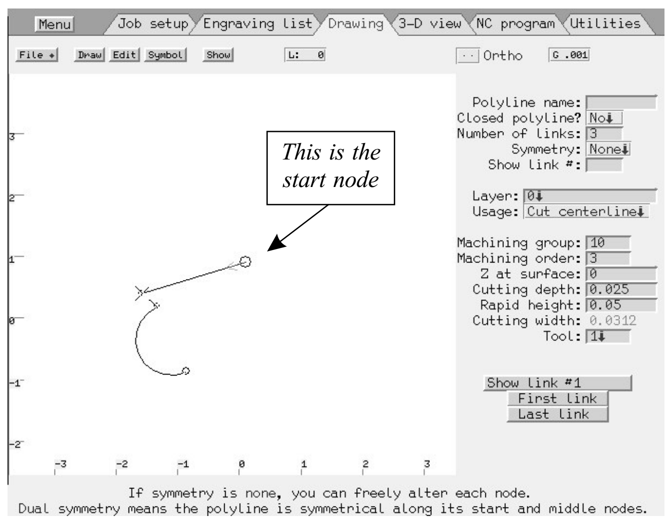


Figure 4-15

Note: You could also convert it to a symmetrical polyline by using the **Change/Convert** menu. To bring up this menu, click the **right** mouse button while the mouse is over the polyline, and in the upper right corner of the node menu you will see an option for a secondary menu to change items.

After specifying that the polyline be symmetrical, MillWrite will put up the menu seen in Figure 4-16. This menu lets you tell MillWrite whether you drew both halves of the polyline, or if you want MillWrite to mirror the other half for you.

You would select the first option if you drew both halves of the polyline. In this case MillWrite will not mirror anything; rather, it will make both halves identical. It is important to note that if you draw both halves of the polyline, you do not have to draw both paths to be identical because MillWrite will make them identical when you select this option. All you have to do is ensure that both halves have the same number of lines and arcs.

An extreme example is shown in Figure 4-17. The right half of the tree and the left half have same number of lines, but the

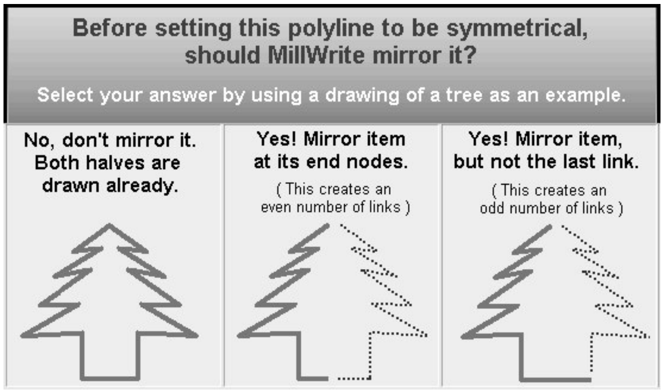


Figure 4-16

left half does not look like the right half. If you were to request that the polyline be made symmetrical, MillWrite will assume the first half is the correct half, and will force the second half to be symmetrical to it. MillWrite determines which half of the polyline is the **first** half and which half is the **second** half by looking at the **direction** of the polyline.

In this example, the start node is at the top, and the direction of this polyline is clockwise. Therefore, the first half of the polyline is on the **right** side and the second half is on the **left** side. When MillWrite makes this polyline symmetrical, it looks for the halfway point in the polyline and alters all the lines in the second half to make them symmetrical with the first half. The end result is seen in Figure 4-18.

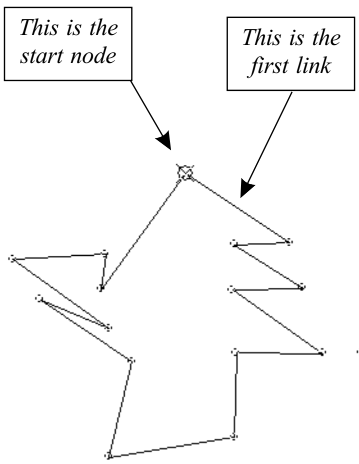


Figure 4-17

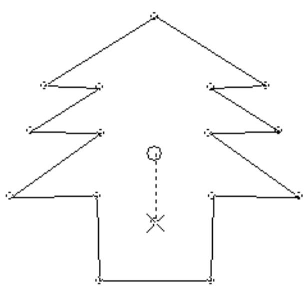


Figure 4-18

Even or odd number of links?

The second and third options of the menu of Figure 4-16 let you specify that you want MillWrite to complete the polyline by mirroring the half you drew. The difference between these two options is that the second option will create a symmetrical polyline with an even number of links because it makes a mirrored image of the half that you drew. The third option mirrors everything you drew **except** for the last link.

In this example, we would select the second option because we want the entire polyline mirrored. After selecting that option, the screen will change as seen in Figure 4-19. The top of the screen will change to allow you to specify the **angle** of the symmetry line; whether you want to **reverse** the polyline; and whether you want to **close** it. In Figure 4-19, MillWrite has arbitrarily selected the symmetry line to be vertical.

In many cases you will not be able to figure out which options to select. Rather, it is easier to experiment with the different options and look at the result. Figures 4-20 through 4-23 show four possible results. The dashed lines in these figures show the portion that MillWrite will create for you.

To understand what the **Reverse** box does, compare Figures 4-20 and 4-21. In Figure 4-20 the polyline has been mirrored at the end of the arc. The reason is that the arc is the last link in this polyline, and MillWrite mirrors the polyline around its last node. In Figure 4-21 MillWrite was told to **reverse** the polyline. This causes the arc to become the **first** link in the polyline, so MillWrite mirrored the polyline at the end of the line instead.

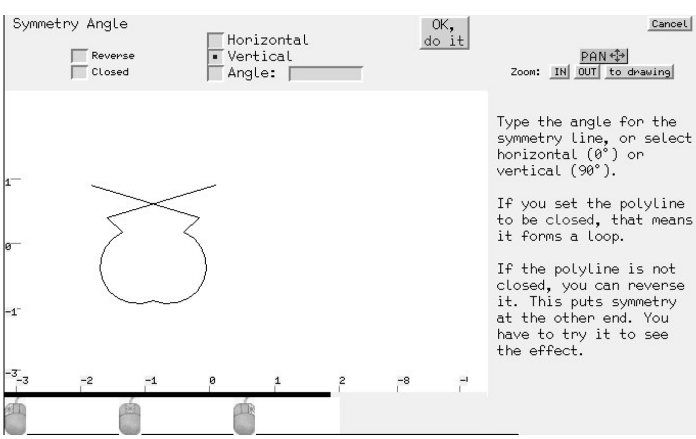


Figure 4-19

Both of these are set for Vertical symmetry

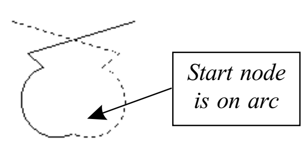


Figure 4-20

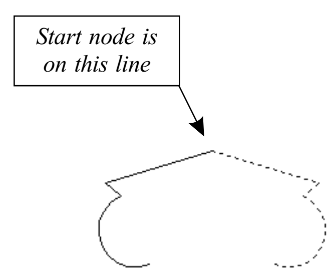


Figure 4-21

Figures 4-22 and 4-23 show examples of horizontal symmetry. The difference between them is that the **Reverse** box was checked for Figure 4-23.

The same item with horizontal symmetry

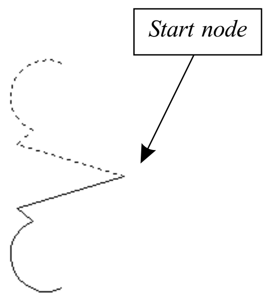


Figure 4-22

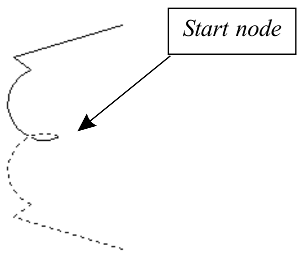


Figure 4-23

Figure 4-24 shows a symmetry angle of 75 degrees, and the **reverse** box has been checked. The dotted lines show how the 75° angle is measured. An angle of 90° would make the line vertical.

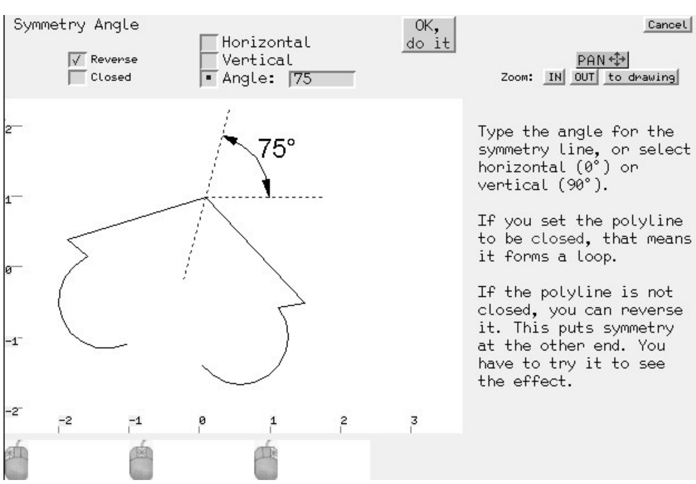


Figure 4-24