Chapter 12

Circular arrays; clocks, date wheels, dials

Clock faces

Circular arrays are useful for creating clock faces, date wheels, and dials. The first example in this chapter shows you how to create a clock face.

This will be a small clock face with a 2 in. radius. The numbers around the clock will be 0.5 inches tall. To begin the clock face, create text for the digit 12, at a height of 0.5 inch. Then ***select*** the text, and then click the **Array** function.

There is nothing new to learn yet; you can use the same procedures for creating text and selecting it that were discussed in the previous chapter. However, after the **Array** function starts you will click the **Style** data field and pick the **Circular** array option.

This will provide the data entry fields seen in Figure 12-1. MillWrite set the **Center** point of the array to the center of the items you are making an array of, which means that unless you change the center point, the clock will be centered around the middle of that 12.

To begin, enter the number of items in this array, which will be 12 items if you do all the numbers on the clock face. That provides MillWrite with enough information to create an array, so it will draw 12 copies of that text in circular array. But because the radius of the array is zero, the text will be rotating on top of itself, creating the mess in Figure 12-2. So move the cursor bar to the **Radius** field and enter **2**.

When you enter a radius, MillWrite adds a few more fields and buttons, and then draws the 12 items on a 2 in. radius, centered around the **Center** point (Figure 12-3). Press the keyboard arrow keys to pan, if you cannot see the array.

The default center point of the array is the center of the item(s) the array is made of, which is why all the 12s are rotating around the original 12.

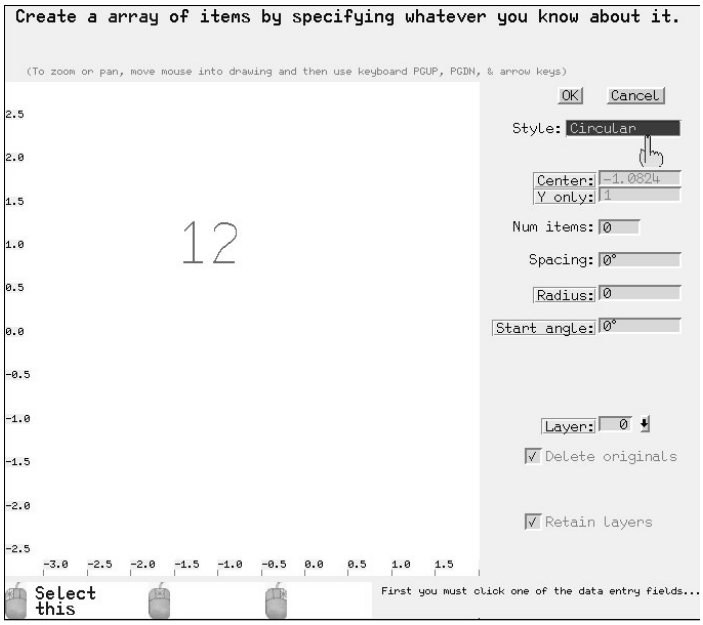

Figure 12-1

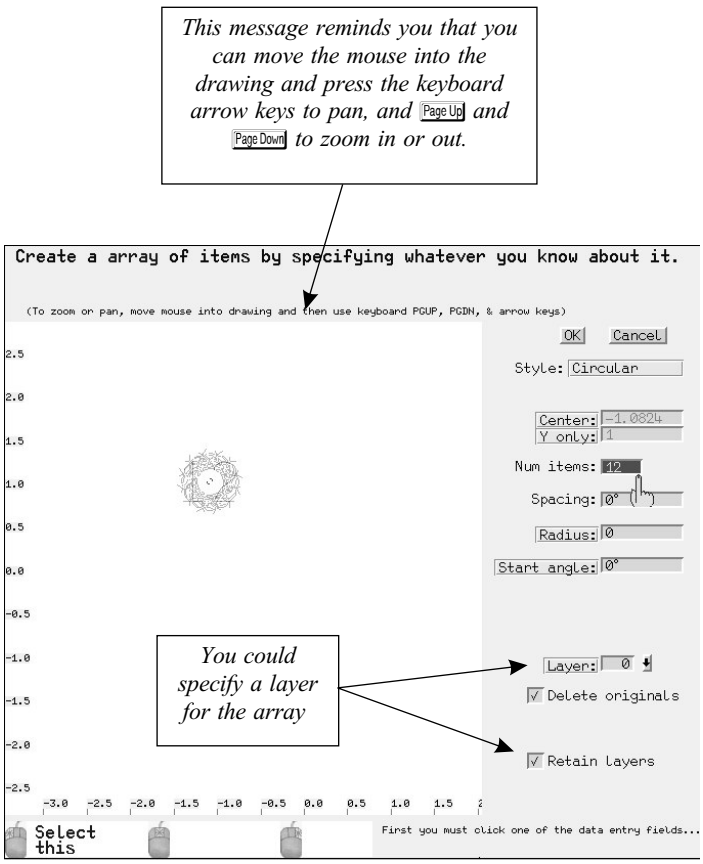

Figure 12-2

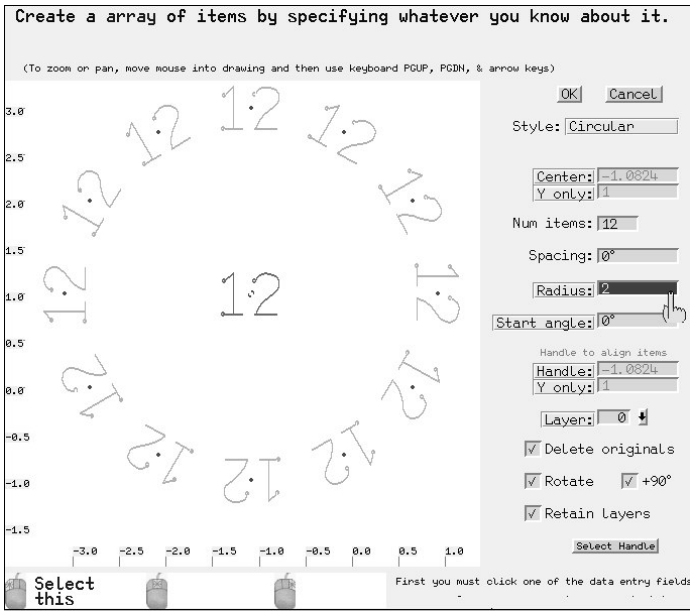

Figure 12-3

There is a field called **Spacing** for you to specify the number of degrees between each item in the array. If you leave the **Spacing** field blank or set to zero, MillWrite assumes you want the array to fill the entire 360°. Since this clock face has 12 items, each item will be separated by 30°. Therefore, you could enter 30° for the **Spacing** field, and this would have the same effect as leaving it empty. But if you decide later to change the number of items in the array (you might decide to engrave only half of the digits on the clock face), you would have to change the spacing field also. It's best to specify as little as possible and let MillWrite do as much of the work as possible.

The point of this is when you are creating an array that fills the entire 360°, it is best to set the **Spacing** field to **zero**.

If you enter a value in the **Spacing** field and later decide to remove it, move the cursor to it and either enter a zero or press the Delete key.

Rotation options in circular arrays

When you specify a radius for the array, MillWrite adds the handle field and to rotation check boxes. The two check boxes determine whether and how the items are rotated within the array. You can check one, both, or neither of the boxes, giving you a total of four different options. The arrays in Figures 12-3, 12-4, and 12-5 show three of the four possible options.

In the case of a clock face, you want both rotation options turned off, as seen in Figure 12-5. If you were making a date wheel, the options of Figure 12-3 are what you want. However, there is no need for you to understand or predict which of the two rotation check boxes should be on our off. Instead, just click the boxes until you get the result you want.

Arrays and Layers

This clock face was a simple example because there was only one item to make an array from. But you may want to make an array from several items, some of which are on different layers. You then have the option of retaining those different layers, or putting the entire array on one specific layer. If the **Retain Layers** box is checked, then the original layers will be retained. But if you specify a layer for the array, MillWrite will remove the check from that box, and every item in the array will be placed on the layer you specified.

One reason you might want to specify a layer for an array is when you are creating several arrays from the same item(s). By putting each array on a different layer, you can keep them separate from each other and from the original item(s).

The original 12 the array is created from can be deleted, so put a check in the **Delete Originals** box. This will spare you from deleting it later.

This completes the clock face so you can click the OK button and return to the drawing. The next step is to replace the 12s with the digits 1 to 11. You would use the same procedure that was explained in the previous chapter about creating a text array. In case you forgot, the procedure is to touch one of the 12s with the mouse, press the E key, and then type the digit(s) you want it to be. Figure 12-6 shows this partially completed.

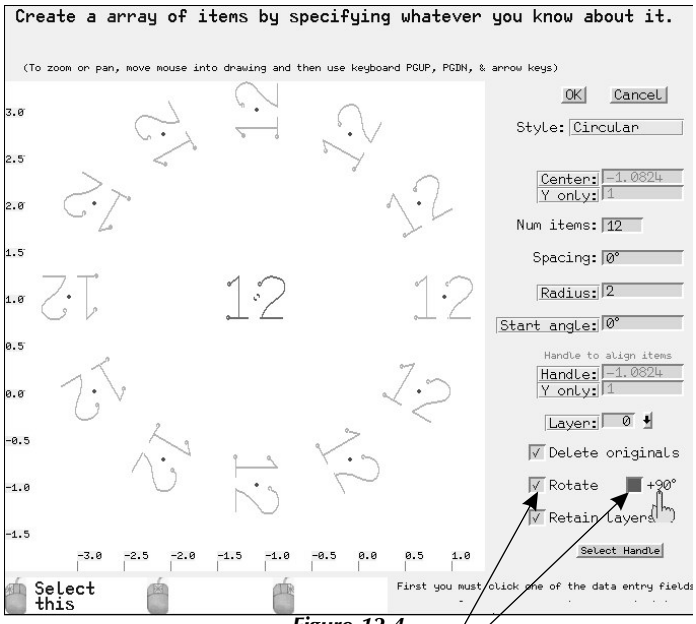

Figure 12-4

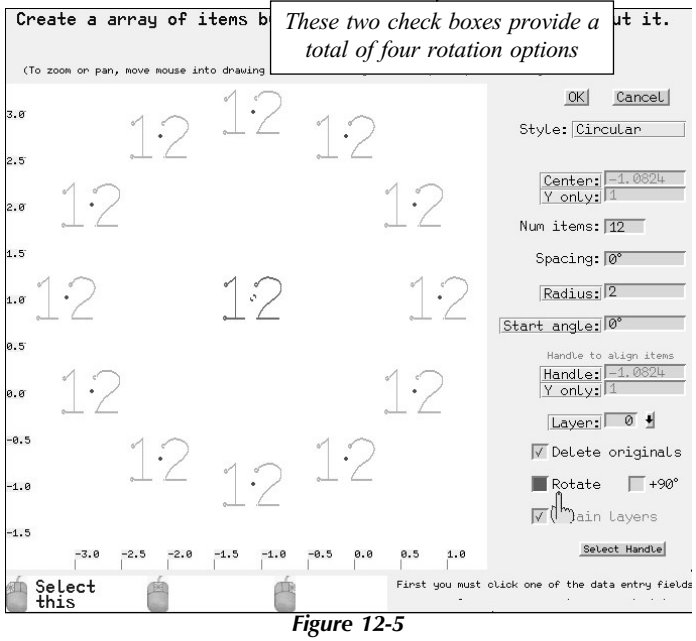

Figure 12-5

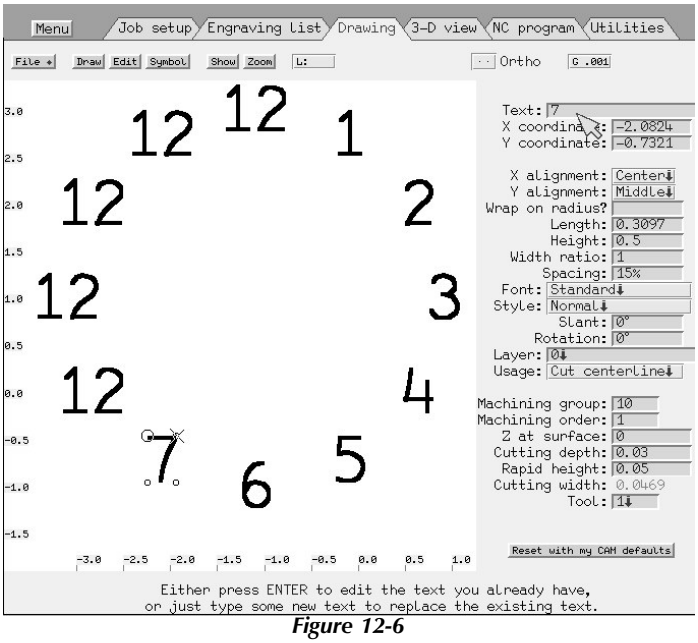

Figure 12-6

Dials

This example will create the dial in Figure 12-7 by making a circular array of two line segments. This is similar to making the clock face, but this example will show you some drawing and editing features that you haven't seen yet.

The dial has only two types of line segments; a short one and a long one. You only have to draw one long and one short line, and then use the array feature to fill a 360° array with these two line segments.

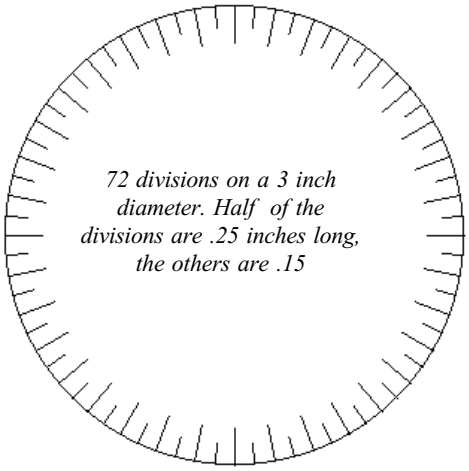

Figure 12-7

To begin, create a circle with a 1.5 inch radius. You can follow the same procedure that page 2 explains. Now to draw a vertical line that will become the .25 inch division line. But this time let's draw the line with the mouse rather than the **Draw** menu.

If the left or middle mouse button is not set to the **Draw Line** function, move the mouse to an empty part of the drawing and click the right mouse button. This will bring up the **New Item** menu (seen on page 1). Select the **Draw Line** function.

Now put the mouse at the center of the circle. You'll know when the mouse is at the center when you see the parameters for the circle appear on the right side of the screen, as seen in Figure 12-8.

Then click the left mouse button (or the middle button if you set the middle button to the **Draw Line** function), and move the mouse. A line will start to draw from the center of the circle. This line must be perfectly vertical, and it must touch the top of the circle, so press and hold the Ctrl key to restrict movement to only horizontal or only vertical. The line will become vertical, as seen in Figure 12-9. Next touch the mouse to the circle while holding down the Ctrl key. As seen in Figure 12-10, the line will jump to the top of the circle even if the mouse is not at the top of the circle. (This requires the **Snap To Node** function be turned ON. Press the F3 key if the vertical line does not snap to the circle.)

An explanation of what is happening is: you have **Snap To Node** on, and you have the Ctrl key down. The **Snap To Node** function causes MillWrite to snap the node of the line to the circle, and the Ctrl key restricts movement to only horizontal or only vertical, so the end result is the line snaps vertically upward to the circle.

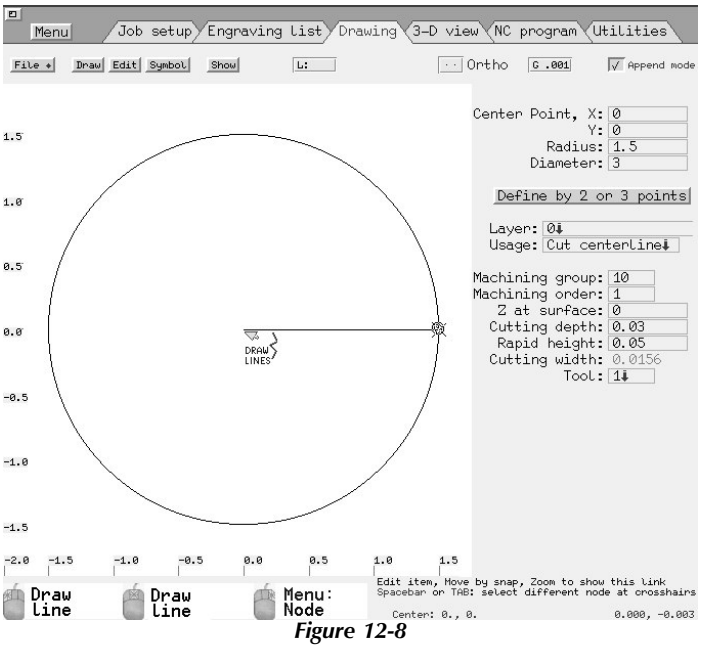

Figure 12-8

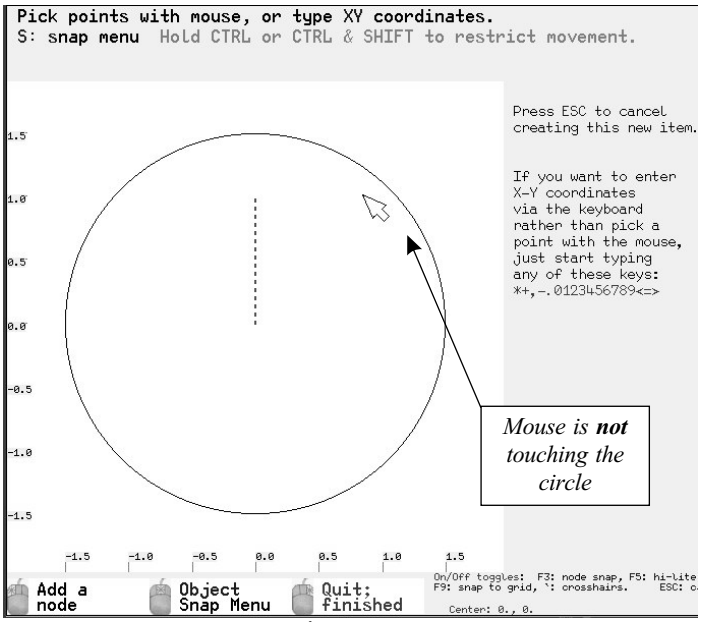

Figure 12-9

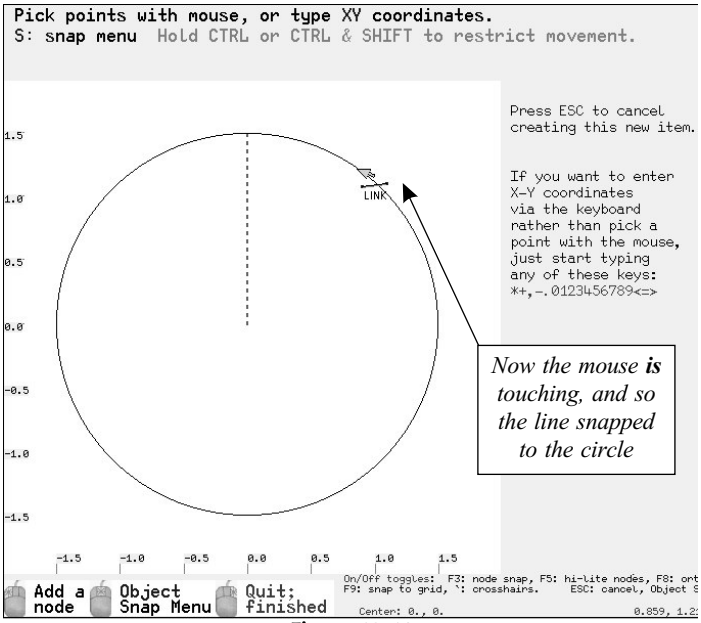

Figure 12-10

Click the left mouse button when you see the line snap to the circle. This vertical line will become the long line segment in the dial, but this line is much too long, so now let's set the line to the correct length of .25 inches. This time let's use a method you haven't seen before.

Touch the line with the mouse as seen in Figure 12-11. The parameters for the line will appear on the right side of the screen. By default, the **Lock On Headers** box will be checked, which causes the tool and other information to show rather than the node coordinates and length of the line. So you'll need to click either the button **Show Link #1**, or **First Link**. This will bring up the node coordinates and other specifications of the line itself.

As seen in Figure 12-12, one of the fields is **Length Of Line**. You can change the length of the line by entering a value in this field. Since the length of a line is measured from the start of the line to the end of the line, if you change at the line length the start point will remain at the same position and the end point will change. But in this case that is the opposite of what we want. Enter a value in this field to understand what this means, and then set it back to 1.5 inches. We want the end of the line to remain where it is. The solution is to ***reverse*** the direction of the line. So rather than change the line length, move the mouse back into the drawing and touch the line with the mouse, as seen in Figure 12-11. Then press the R key to reverse its direction (the message at bottom right reminds you about this command).

If you had known this ahead of time you would have pressed the R key before bothering to click in the parameters area. This is an example of when the direction of a line is important, and when you need to reverse the direction. You can determine the direction of the line by the direction arrow. Also, the tiny circles that identify the nodes of the line are larger at the start node.

After you reverse the direction of that line, click the button for the **First Link**, and then set the line length to 0.25 inches. MillWrite will shorten the line segment as seen in Figure 12-12. This completes one of the lines in the dial.

Now to create the 0.15 inch line. To show you how the **rotate** function works, let's create the 0.15 inch line by duplicating the 0.25 inch line, and then rotating it 5°. Then we can change its length to 0.15 inches.

The first thing to do is to ***select*** the line. The screen will change as seen in Figure 12-13. Pick the **Duplicate** function. As seen in Figure 12-14, MillWrite will prompt you to specify the ***starting*** point for the copy. After that you'll specify the ***ending*** point of the copy. In this case we just want to duplicate the line and leave it in the ***same position***, and that requires the starting point and the ending point to be the same point. It doesn't matter where the point is; rather, you just pick any point ***twice***.

In Figure 12-14 the mouse is picking the arc center twice, but you could also pick the arc start node or one of the nodes on the line itself. However, you should ***not*** pick a point in the drawing that does not have a node on it because it is very difficult to pick the same point a second time. The reason is that it's difficult to keep the mouse in the exact same position as you click the mouse buttons. The lesson to learn is that when you want to pick the same point twice, click a ***node.***

After you've duplicated that line segment, click the **Rotate** button, as seen in Figure 12-15.

MillWrite will prompt you to specify the manner in which you want to rotate the item (Figure 12-16). Pick the first choice,

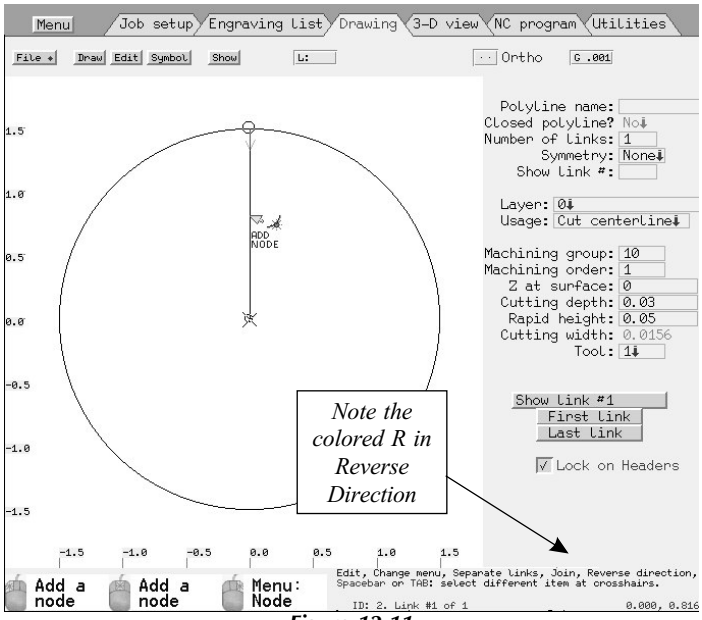

Figure 12-11

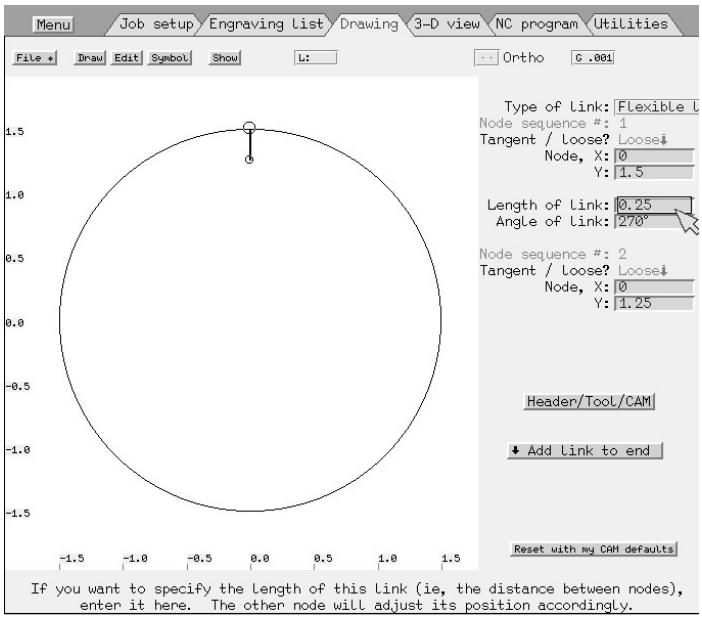

Figure 12-12

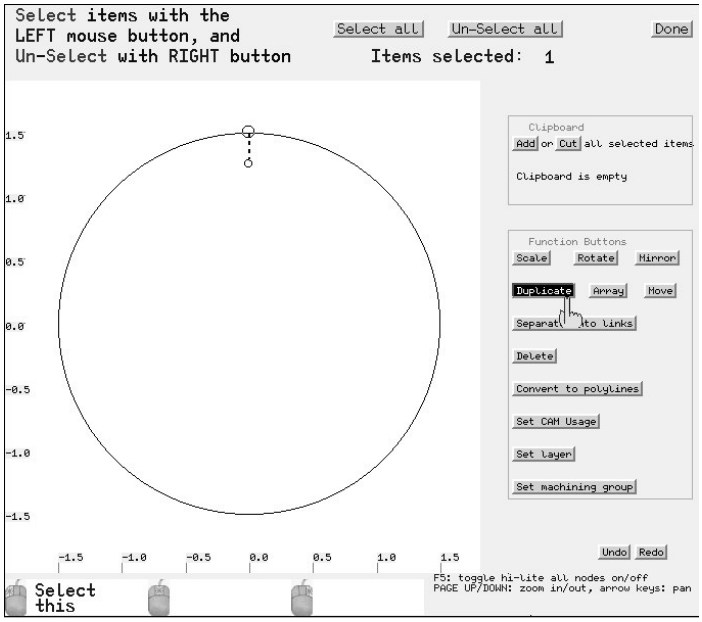

Figure 12-13

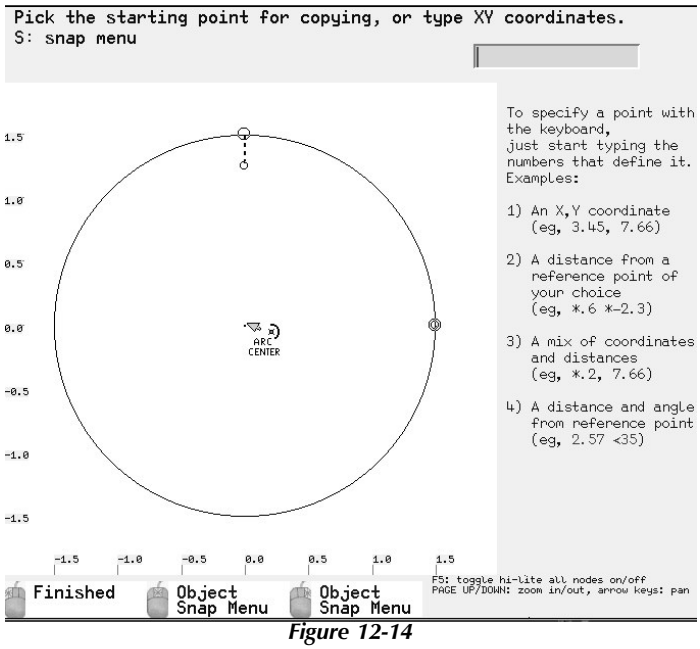

Figure 12-14

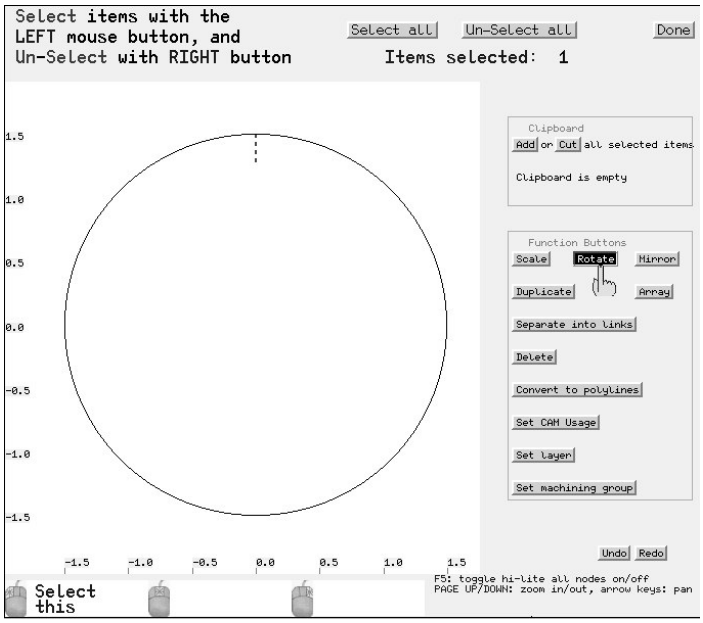

Figure 12-15

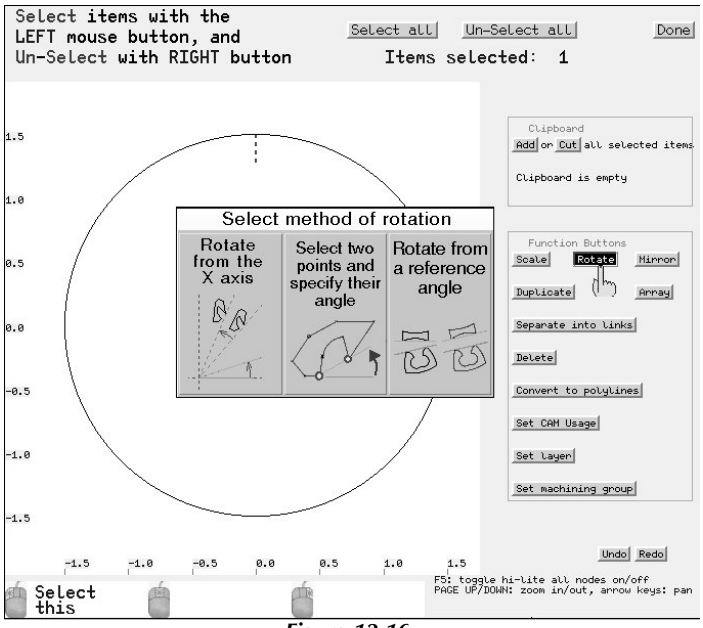

Figure 12-16

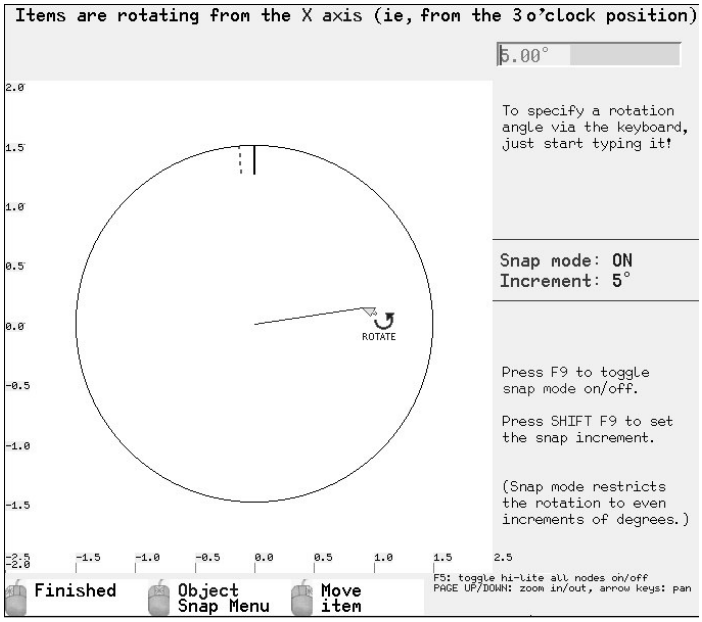

Figure 12-17

which is **From The X Axis**. This is the most common method of rotating objects.

MillWrite will prompt you to pick the center point for the rotation. Since this line must rotate around the circle, the center of rotation will be the circle's center. Therefore, move the mouse to the center of the circle, and when you see the mouse icon change to show the words **ARC CENTER**, click the left mouse button.

After you identified the center of rotation, the screen will change to that of Figure 12-17. As you move the mouse the line will rotate, but the original line will seem to have vanished. In order to bring it back you must force the screen to redraw, such as by a pressing a keyboard arrow key to pan, or Page Up or Page Down to zoom in or out.

The default for MillWrite will be to turn the **rotation snap** on so that rotation occurs in increments of 5°. As the prompt on the right side of the screen will remind you, the F9 key will turn the rotation snap on and off. Also, you can press Shift F9 to set the increments of the rotation snap. For example, if you wanted to rotate in increments of 15°, press Shift F9. MillWrite will then let you type a new value for the increment.

While you are rotating, you could just let go of the mouse and type the rotation value that you want. You do not have to use the mouse to set the rotation. So to rotate this line by 5°, you could just type 5 and then press the Enter key.

After rotating the line by 5°, you can exit the selection mode and return to the drawing. Now you have to shorten this line segment. Since this is a duplicate of the other line, it will have the same direction. In other words, the ***start node*** of this line will be on the circumference of the circle, so when you change its length, the start node will remain touching the circle and the ***end node*** will move, which is what we need in this case. So touch the mouse to this line and then click in the parameters area. If the **Lock On Headers** box is checked, you'll have to click the **First Link** button in order to get to the data field that holds the length of the line. Enter 0.15 inches in that field and press Enter.

You just completed both lines, so now you are ready to make an array of them.

To make an array of those two lines you must first ***select*** them both. After selecting them, click the **Array** button, and fill in the fields as seen in Figure 12-18.

Set the center of the array to X0, Y0. Specify 36 items in the array. However, do ***not*** specify a radius for this array.

Common circular arrays do not need a radius

If you specify a radius, MillWrite will add the **Handle** field and two rotation check boxes. This adds complexity, but you don't need this complexity for this particular dial. This dial is a common circular array in which the items rotate around the center of the array. The clock face, by comparison, was more unusual because the items did ***not*** rotate; rather, they were placed at array points in the same orientation that the original item was in.

The default for MillWrite is to create the most common type of circular array, such as this dial; ie, the default is to rotate the items as they are placed in the array. Therefore, when setting up the clock face, you ***had*** to specify a radius in order to force MillWrite to put the rotation boxes on the screen so that you could turn off the rotation of the items.

For this dial, all you have to do is specify the center point of the array and the number of items. MillWrite then creates the array, as seen in Figure 12-19. The array is complete, so you can press the OK button.

Note: normallly you **delete** the **original** items for common arrays, otherwise the originals will be on top of the array.

Complex arrays need a radius

When you specify a radius for the array you get access to the **Handle** field. If you recall from Chapter 1 in which the analogy of a suction cup was used, by specifying the handle location you specify how MillWrite picks up each item and lays it into the array.

In Figures 12-20 and 12-21 are two arrays to show how the handle affects a circular array. The item that the array was created from is in the center of the array. The dotted circle shows the diameter of the array. There are six items in each array, and each array location is identified by a small circle. The **+90°** box was also checked for both arrays. The only difference between the two arrays is their handles.

Figure 12-22 shows the same item in an 8-element array just to show you how different the array can be by changing handle locations and rotation options.

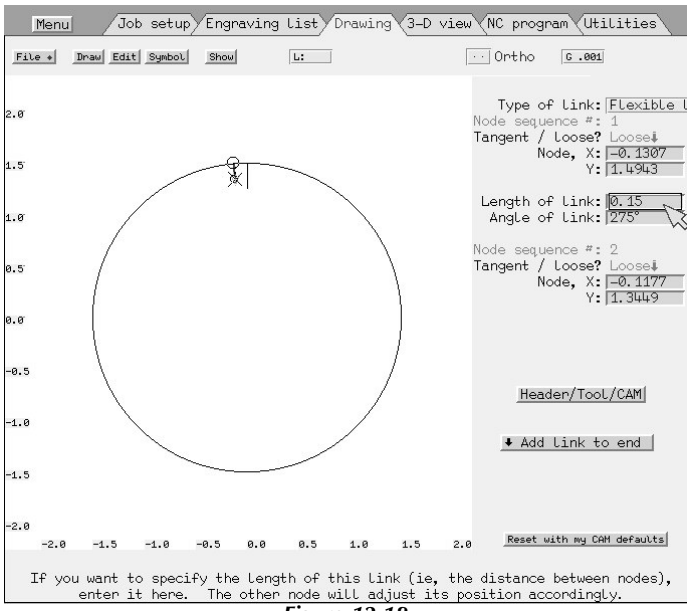

Figure 12-18

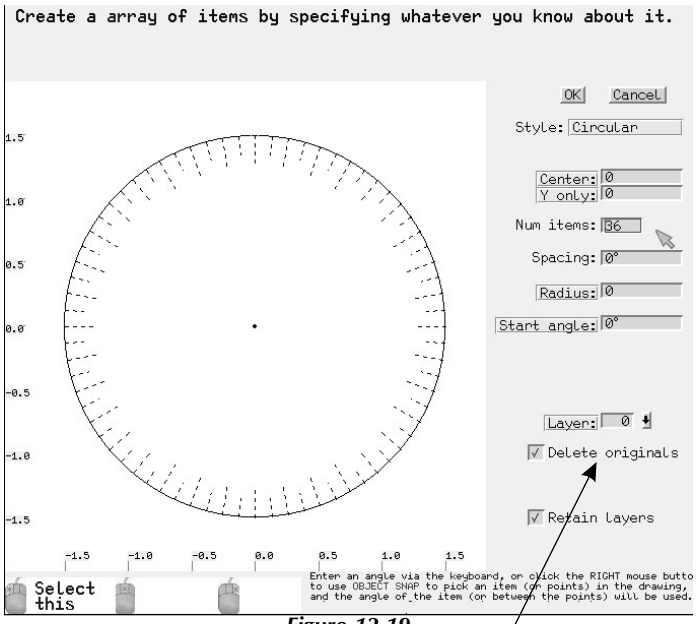

Figure 12-19

Note: For common arrays that do not have a radius, you normally delete the originals to avoid having items on top of each other.

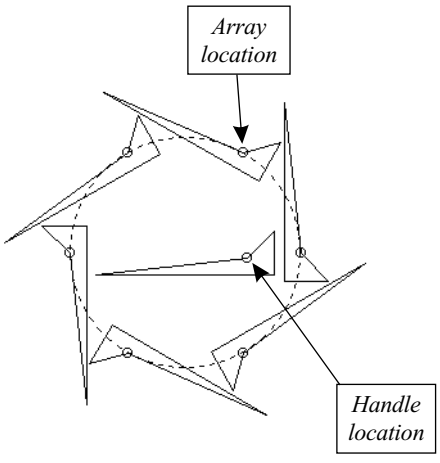

Figure 12-20

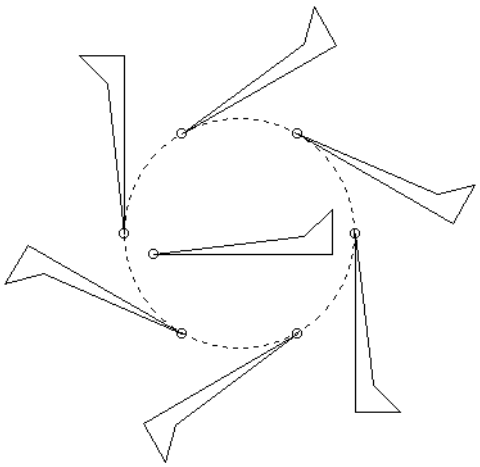

Figure 12-21

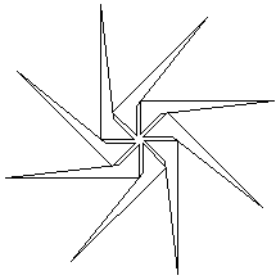

Figure 12-22