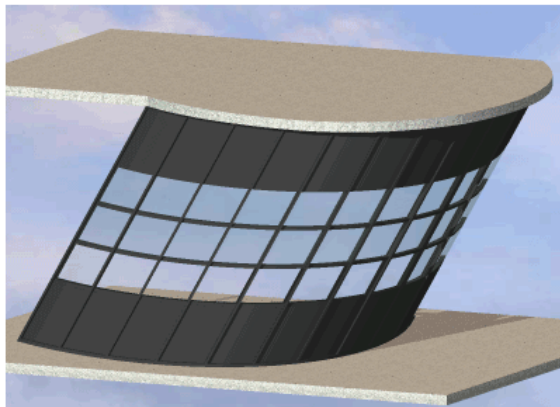
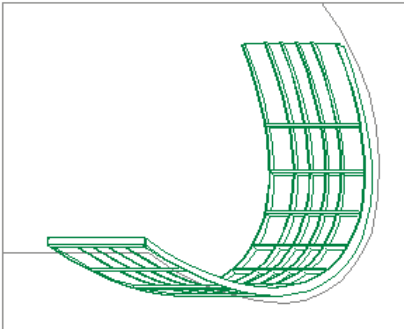
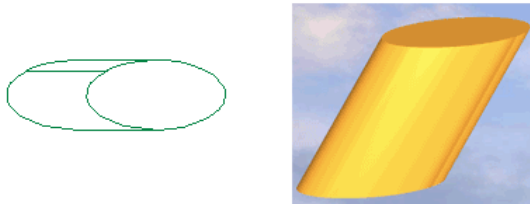
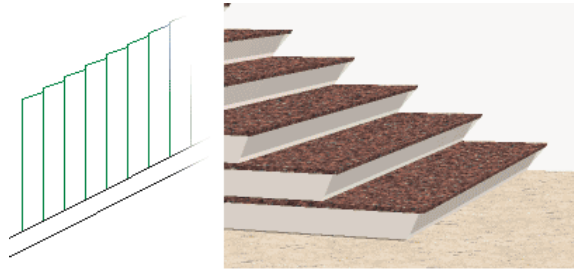


Skew It!

A Skewing Tool using XFORM by Karl Ottenstein

Have you ever wanted to skew an object to easily create a custom stair rail, a sloped barrel vault roof, or something similar to the illustrations below? Notice how the stairs are skewed several ways, with a back-angle to the risers and sides. The exhaust stack for a cruise liner is sheared vertically. A standard curved curtain wall has been skewed vertically and in plan to create a canted J-shape.



I'll show you how to do these kinds of things by using a simple GDL tool that uses XFORM to skew another library part. Then, for Voyagers, I'll explain the XFORM command in more detail.

Skewer

Skewer is a library part that is a GDL tool: it is only useful when combined with other library parts. The script is quite short and takes seven parameters: the name of a (loaded) library part to be skewed and six angle values to control the skewing. (Typically only one or two angles would be used.)

libpart	Abc	Skew what library part?	Railing
skewer_sp0	≡	Plan (x-y) Skew	
skxy	△	Skew x axis towards +y	0.00
skyx	△	Skew y axis towards +x	0.00
skewer_sp2	≡	Section (x-z, y-z) Skew	
skxz	△	Skew x axis towards +z	0.00
skyz	△	Skew y axis towards +z	0.00
skewer_sp1	≡	3D (z) Skew	
skzx	△	Skew z axis towards +x	0.00
skzy	△	Skew z axis towards +y	0.00

The 3D script does all of the work:

```

IF libpart <> "" THEN
XFORM 1,          tan(skyx), tan(skzx), 0,
              tan(skxy),      1, tan(skzy), 0,
              tan(skxz), tan(skyz),      1, 0
CALL libpart PARAMETERS
DEL 1
ENDIF
    
```

The XFORM command above performs a shear-type skewing on all six faces of the virtual cube enclosing your object. The CALL retains the original size of the saved library part. If instead you want that part to be resizeable, enter:

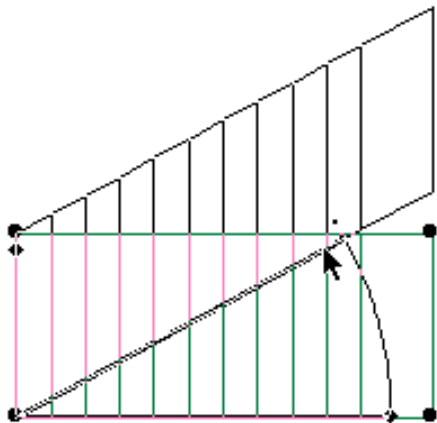
```
CALL libpart PARAMETERS A=A, B=B, ZZYZX=ZZYZX
```

In the illustration with the stairs, the parameters for that Skewer object were set as follows:

Skew what library part?	slab-stairs
Plan (x-y) Skew	
Skew x axis towards +y	20.00°
Skew y axis towards +x	0.00°
Section (x-z, y-z) Skew	
Skew x axis towards +z	0.00°
Skew y axis towards +z	0.00°
3D (z) Skew	
Skew z axis towards +x	-25.00°
Skew z axis towards +y	10.00°

where "slab stairs" is the name of the saved library part for that straight run of stairs. The angles skew the stairs in 3 directions.

The angles for skewing along the X or Y axes can be entered dynamically via editable 2D hotspots.



The 2D script is as follows:

```
PROJECT2 3, 270, 2

!Lower left and right object boundary hotspots
HOTSPOT2 0, 0
HOTSPOT2 A, tan(skxy)*A

!Upper left and right object boundary hotspots
HOTSPOT2 tan(skxy)*B, B
HOTSPOT2 A+tan(skxy)*B, B+tan(skxy)*A

! Editable hotspots for skx

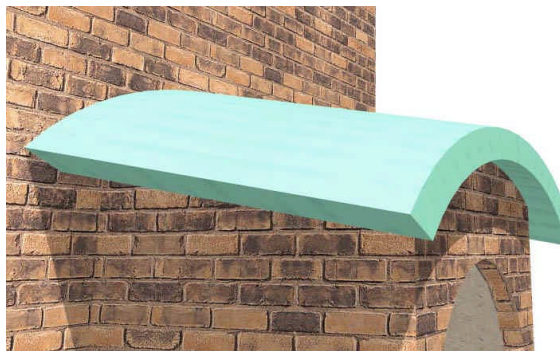
hsid=1
HOTSPOT2 0, 0, hsid, skxy, 6
HOTSPOT2 A*0.9*cos(skxy), A*0.9*sin(skxy),
hsid, skxy, 5
HOTSPOT2 A, 0, hsid, skxy, 4

! Editable hotspots for sky

hsid=hsid+1
HOTSPOT2 0, 0, hsid, skyx, 6+512
HOTSPOT2 B*0.9*sin(skyx), B*0.9*cos(skyx),
hsid, skyx, 5
HOTSPOT2 0, B, hsid, skyx, 4
```

Skew What?

Since Skewer can only skew library parts (that it can CALL), if you have some geometry in your model that you want to skew, simply save it as a library part and enter its name into Skewer and tweak the shear angles. Naturally, this can be just about anything. For example, you can use this to get a canted wall with windows that are in the plane of the wall. Or, use the roof tool to create a barrel roof, then save it as an object, and skew it as shown here:



Shear vs Rotational Skewing

The XFORM above performs shear-type skewing: points along one axis are linearly pushed along another axis as illustrated in the hotspot-editing image of the plan view of the stairs: the X-width of the stairs there has not changed.

XFORM can also perform rotational-type skewing in which the object collapses onto itself like a fan, with points following the arc of rotation. This is interesting theoretically, but I don't see a practical use for it at the moment so won't describe it further.

How does it Work? (For Voyagers)

XFORM performs what is called a coordinate transformation. The command:

```
XFORM    x1,  y1,  z1,  dx,
         x2,  y2,  z2,  dy,
         x3,  y3,  z3,  dz
```

is capable of doing the equivalent of a ROT, a MUL, an ADD and a skew all in one step! Let's work our way through each transformation, finishing with rotation, which is the most complex.

In one of its simplest forms:

```
XFORM    1,  0,  0,  dx,
         0,  1,  0,  dy,
         0,  0,  1,  dz
```

the XFORM is equivalent to:

```
ADD      dx, dy, dz
```

The first three columns of the matrix represent the shape of the X, Y and Z coordinate axes relative to one another. The axes are lines drawn from (0,0,0) to the point represented by the columns (as typed above), read downwards. So, the X axis is drawn to (1, 0, 0), Y to (0, 1, 0) and Z to (0, 0, 1). In words, this means that the axes are "normal" – being 90 degrees to one another and the same size. The value 1 is actually a scale (MUL) factor for the corresponding axis, not an absolute value, and so indicates that no scaling is performed by this particular XFORM. If dx, dy and dz above are all 0, then the XFORM really does absolutely nothing other than occupy a spot on the stack. It would be the same as ROT 0, 0, 0 : MUL 1, 1, 1 : ADD 0, 0, 0.

The equivalent to a MUL is:

```
XFORM  sx,  0,  0,  0,
        0,  sy,  0,  0,
        0,  0,  sz,  0
```

Here, the X, Y and Z axes are potentially *scaled* by the values of sx, sy and sz. This command produces the same result as:

```
MUL  sz, sy, sz
```

And, of course the two can be combined. The equivalent of an ADD and MUL together is:

```
XFORM  sx,  0,  0,  dx,
        0,  sy,  0,  dy,
        0,  0,  sz,  dz
```

How about skewing? To skew, we want to change our axes so that they are not all at 90 degrees to one another (orthogonal). For example, skewing just the X axis up (in the Z direction in the X-Z plane) in order to convert a straight railing into a sloped one for a stair rail merely involves changing the value of the Z coordinate of the X-axis vector:

```
XFORM  1,  0,  0,  0,
        0,  1,  0,  0,
        xz, 0,  1,  0
```

The Y and Z axis are normal, but the X axis is 'drawn' from (0,0,0) to the point (1,0,xz). The way to visualize this is that the X axis is *hinged* at the origin and *lifted* upwards. The effect of the XFORM is to skew any points proportionally with that new, non-orthogonal axis.

Of course, an absolute value such as xz is of limited use in precision modeling. If we want an angle, as used in Skewer, we just need to make a quick reference to the Cookbook Maths Primer ("Right Angle Triangles") to see that if we want to lift the X axis by *angl* degrees, the corresponding Z value would be *tan(angl)*:

```
XFORM  1,  0,  0,  0,
        0,  1,  0,  0,
        tan(angl), 0,  1,  0
```

As you saw in the 3D script for Skewer, any axis can be moved anywhere in space, thus permitting skewing (and rotation) in all directions. Whew!

Finally, we have the ability to rotate using XFORM. If you think about it, rotation is merely moving all three axes in a way that keeps them all orthogonal. Of course, there's no reason to ever use XFORM to rotate, as ROT makes this task so simple. But, to complete this discussion, let's just look at what is involved ... and what in fact happens behind the scenes by ArchiCAD itself when you do perform a ROT.

First, we take another visit to the Cookbook Maths Primer ("Circles") to refresh our memories

on the trigonometry involved in determining the modified coordinates given a rotation of a particular angle. An XFORM that does the same as a ROTY *angl* (and nothing more) is thus:

```
XFORM  cos(angl),  0,  sin(angl),  0,
        0,  1,  0,  0,
        -sin(angl), 0,  cos(angl),  0
```

Observe that the Y axis is unchanged – it still points to (0, 1, 0) – which is correct since we are rotating about this axis. The X axis no longer points to (1, 0, 0) but instead to (cos(angl), 0, -sin(angl)) and the Z axis no longer points to (0, 0, 1) but to (sin(angl), 0, cos(angl)). Since we are rotating the X-Z axes in their own plane, it is the X and Z coordinates that have changed. These axes are still perpendicular to Y (the Y coordinate is still 0) and to each other.

Digging Deeper

Working with a modified coordinate system is also seen in the VERT and COOR command combination. The basis for coordinate transformation is in Linear Algebra. For more in-depth information, see the encyclopedic *Computer Graphics: Principles and Practice* by Foley, van Dam, Feiner and Hughes (Addison-Wesley, 1995, 1200 pages ISBN 0201848406) or most any other computer graphics text.

Acknowledgements

Thanks to Oleg Schmidt for suggesting the practical use of XFORM for skewing and for reviewing this article and to the 1999 Graphisoft News for suggesting the use for skewing stairs. Thanks to David Nicholson-Cole for his helpful comments on this article.

Karl Ottenstein, Ph.D.

Archispectives LLC

Sandpoint, Idaho USA

March 17, 2004