
Subject: Re: DXF and other CAD file formats

Posted by [Dick Jackson](#) on Mon, 03 Dec 2001 18:21:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

```
> "Georg Wiora" <georg.wiora@DaimlerChrysler.com> wrote in message
> news:1104_1007043323@rtulmrb142...
>> Hi everybody,
>>
>> I have problems using the polyline feature in an IDLffDXF object. I
> connect 4 points with a line, using the CONNECTIVITY array. When I add
this
> object, I allways get the following message:
>> % IDLFFDXF::PUTENTITY: DXF error: DXF Polyline: explicit connectivity
> ignored
>> For a little test program see the code below.
>>
>> dxf = OBJ_NEW('IDLffDXF')
>>
>> ; 4 points
>> points = DBLARR(3,5)
>> points(*,0) = [0,0,0]
>> points(*,1) = [1,0,0]
>> points(*,2) = [1,1,0]
>> points(*,3) = [1,1,1]
>> points(*,4) = [0,1,1]
>> ; Connectivity
>> connect = [0,1,2]
>> ; Create pointers
>> ppoints = PTR_NEW(points,/NO_COPY)
>> pconnect = PTR_NEW(connect,/NO_COPY)
>>
>> ; Create a polyline object
>> pline = {IDL_DXF_POLYLINE}
>> pline.Vertices=ppoints
>> pline.Connectivity=pconnect
```

"Rick Towler" <rtowler@u.washington.edu> wrote in message

news:9ugdov\$hd6\$1@nntp6.u.washington.edu...

```
>
> Although I haven't tried writing DXF files, my guess is that your
> connectivity array is wrong. I know that if I were trying to create an
IDL
> (opengl) polyline, my connectivity array would have more elements than
that
> (in your case it would be [4,0,1,2,3]). I didn't see any documentation
> regarding this in the IDLffDXF docs but there was an example similar to
> yours. Maybe you should start there?
```

Rick's right, there is this clue in IDLffDXF::GetEntity help:

CONNECTIVITY is the array used to connect these points into polygons (see the POLYGONS keyword for IDLgrPolygon::Init). If this array is not present, the connectivity is implicit in (U, V) space defined by the values in MESH_DIMS; the vertices represent a quad mesh of dimensions (MESH_DIMS[0], MESH_DIMS[1]).

and this leads to this clue in IDLgrPolygon::Init:

A polygon description is an integer or longword array of the form: [n, i0, i1, ..., in-1], where n is the number of vertices that define the polygon, and i0..in-1 are indices into the X, Y, and Z arguments that represent the polygon vertices.

This explains how to describe one polygon, but doesn't tell the whole story, that you can put any number of polygons in one description array. It's just treated as a 1D array of numbers, e.g.:

[3, 0, 1, 2, 4, 2, 1, 3, 4] describes two polygons:
| \ | / | \ \ / /
| indices | indices
vertices # vertices

So we get a triangle joining vertices 0, 1 and 2, and a quadrilateral joining vertices 2, 1, 3 and 4.

If you have a set of polygons of same type (same number of sides), you might find it easier to work with a 2D array, e.g:

[[3, 0, 1, 2],
[3, 2, 1, 3],
[3, 3, 4, 0]] describes three triangles
| \ | /
| indices of each triangle
vertices (will all be same)

This is a technique that comes up in a few places when working with IDL polygons. Hope this helps.

Cheers,

--

-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392