DXF Geometry Import Converter

The **DXF** geometry import converter is an intelligent and complex converter which will take any DXF file of 2d, 2-1/2d or 3d entities and transform them into optimized, grouped meshes of 3d polygons. It will convert many DXF files that most other converters fail to handle properly.

Whereas many DXF readers only read and convert **3D FACE**s and **3d PolyMeshes**, this converter will read in all R12 entities and convert them to 3d polygon equivalents. This includes processing all of the **PolyLine** attributes (bulges, line widths and thickness), **PolyMesh** types (**mesh**, **face** and **lines**) and 2d entities (**lines**, **arcs** and **circles**). In addition, the converter provides 5 powerful methods to "rip" apart the DXF file and convert it into separate objects. Also, if the DXF file uses blocks and instances then they will be converted into equivalent objects and instances within the converter's database - this will create a very small imported database since no data is duplicated unless necessary.
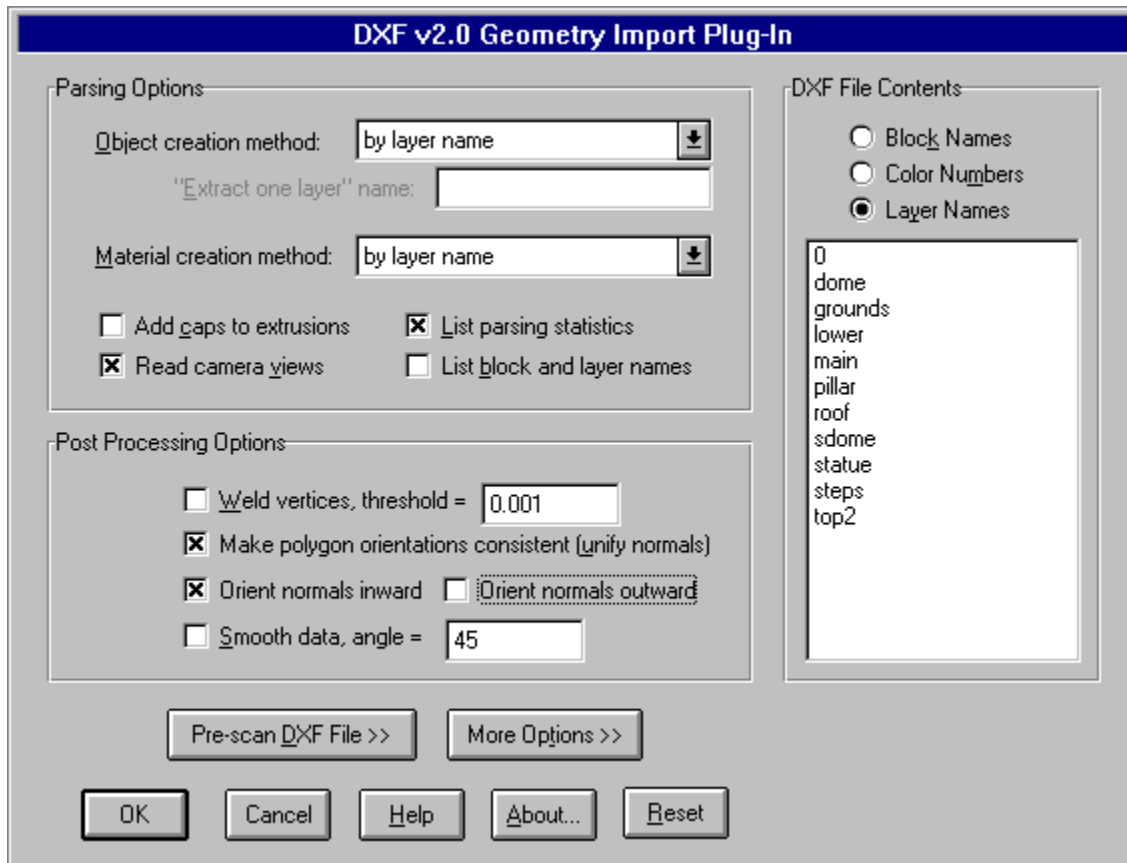
Press the **>>** button above, or select from one of the following sections:

Trademarks

3D Studio, DXF, DWG and AutoCAD are registered trademarks of AutoDesk Inc.

Dialog Box Options

Pre-Scan DXF File button

If this button is pressed then the file will be scanned and the color   numbers, layer names and block names referenced within the file will be displayed on the right   side of the dialog box. No data will be read into memory. This option is quite useful if you would like to know how the DXF file is structured.

Object Creation Method

This is a very important option which basically determines how the DXF converter is to turn the "mess" of 2d and 3d drawing entities contained in   the DXF file into nice and orderly 3d objects. You must decide which method is best to turn the file into 3d objects - this is best done by running the converter first with the **Pre-Scan DXF File** option described above which will list out all of the color numbers, block names and layer names used in the file.   From this information you should be able to decide which method is best (for example, if the entities are grouped by layer name in the file then choose the "**By Layer Name**" option).

Each entity in AutoCAD (ie: a line, PolyLine, 3D Face, a mesh) has a number of attributes associated

with it which include its "**layer**" name, its "**color**" number and which "**block**" it belongs to (entities are not always contained in blocks).

- The '**By Block Name**' option will create one new object for each block contained in the DXF file. If an entity is not associated with a block then it is added to a separate object named "**non-block entities**".

- The '**By Layer Name**' option will create one new object for all entities which belong to the same layer in the DXF file. If an entity is not associated   with a layer then it is added to a separate object named "**lone entities layer**".

- The '**Extract from One Specific Layer**' option is similar to the '**By Layer Name**' option except that only entities that belong to the layer specified in the **Extract One Layer name** type-in box will be imported. This option is useful is you only want to convert a specific layer.

- The '**By Color number**' option will create one new object for all entities which belong to the same color number in the DXF file. The objects will be named as '**color x**' where 'x' is the AutoCAD color number.

- The '**Create One Object By Entity**' option will create one object per entity contained in the DXF file. Normally you should **NOT** use this option since a DXF file typically contained tens of thousands of entities. However, if you know that the file contains only meshes, or only a few entities, then this option might be useful.

- The '**Create One Single Object**' option creates only ONE object which contains every entity found in the DXF file.

Material Creation Method

This is the second most important option which determines how the DXF converter is to be create and assign new materials (surfaces) to the new objects created.

- The '**By Block Name**' option creates one new material for each block name found within the DXF file. These new materials will then be assigned to the entities which belong to each block. If an entity does not belong to a block then it will be   assigned the "**default**" surface. The material name will be the same as the block name.

- The '**By Layer Name**' option creates one new material for each layer name found within the DXF file. These new materials will then be assigned to the entities which belong to each layer. If an entity does not belong to a layer then it will be assigned the "default" surface. The material name will be the same as the layer name.

- The '**By Color Number**' option creates one new material for each unique AutoCAD color number which was referenced in the DXF file. These new materials will then be assigned to the entities which use the referenced color numbers. The material name will be created using the syntax "**color xx**" where "xx" is the color number.

- The '**One Material Per Entity**' option creates and assigns one new material for each entity within the DXF file. The material name syntax will be "**entity xx**". Normally you should **NOT** use this option since a DXF file typically contained tens of thousands of entities. However, if you know that the file contains only meshes, or only a few entities, then this option might be useful.

- The '**Assign Default Material**' option assigns the "**default**" material (surface) to the entities in the DXF file. This option is useful is you will be creating your own materials and assigning

them to the imported objects yourself.

Add Caps To Extrusions

If this checkbox is enabled (check-marked) then polygon end caps will be added to 2d entities which have been extruded, such as arcs and circles. For example, an extruded circle will remain a hollow cylinder if this option is un-checked, else it will become a closed cylinder with each caps if this option is this option is check-marked. This option is disabled (un-checked) by default since some DXF files do not look good with this option enabled.

Read Camera Views

If this checkbox is enabled (check-marked) then the converter will parse the DXF **"VIEW"** table entries and convert them to camera definitions.

List Parsing Statistics

If this checkbox is enabled (check-marked) then the converter will print out information about the parsing process after the DXF file has been parsed and read into memory.
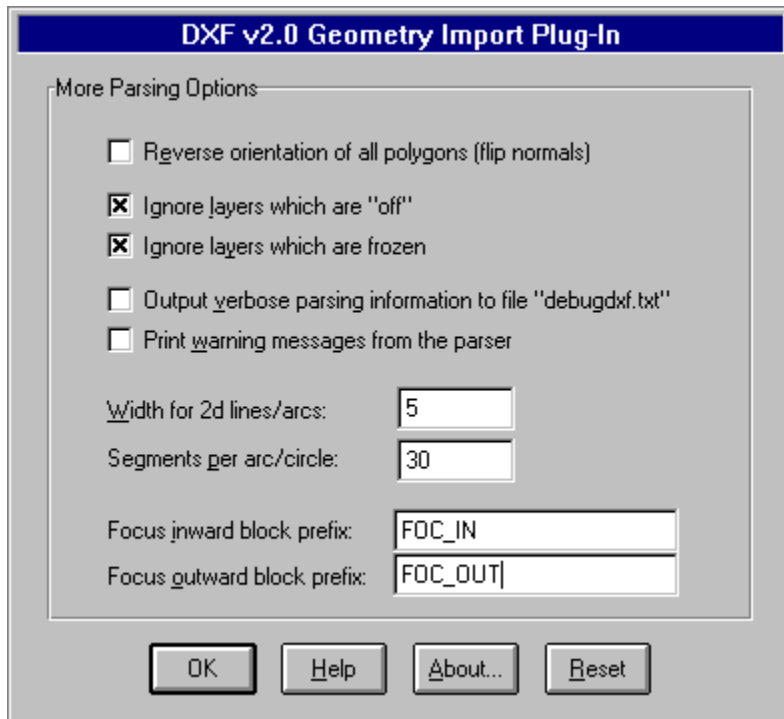
List Block and Layer Names

If this checkbox is enabled (check-marked) then the import converter will print out the names of the blocks and layers found within the DXF file after it has been parsed and loaded.

Dialog Box #2 Options

See also:

Reverse Orientation of Polygons (Flip Normals)

If this checkbox is enabled (check-marked) then the orientation of all polygons will be reversed. You might want to enable this option if you are finding that a 3d renderer is displaying only the back-facing polygons rather than the front-facing polygons.

Ignore Layers Which are Off

If this checkbox is enabled (check-marked) then entities which reside on layers which are "**off**" will be ignored.

Ignore Layers Which are Frozen

If this checkbox is enabled (check-marked) then entities which reside on "**frozen**" layers will be ignored.

Output verbose parsing information to file debugdxf.txt

If this checkbox is enabled (check-marked) then verbose parsing information will be output to the ASCII file '**debugdxf.txt**' located within the current working directory. This option is disabled by default.

Print Warning Messages from the Parser

If this checkbox is enabled (check-marked) then the parser will print out warning messages encountered during the parsing process. This option is disabled by default.

Width for 2d lines/arcs

This value controls an **interesting** aspect of the DXF converter. If this option is set greater than 0.0, and a 2d line has no width or thickness, then the converter will artificially widen the 2d line into a rectangular 3d polygon which has a width equal to the value set by this option and a length equal to original line length. **This option is useful because it allows 2d lines (which implicitly have no area) to be rendered in a 3d renderer which only accepts 3d polygons (with an area).**

This value should be set non-zero if you want 2d line segments to be turned into polygons which will be visible within a 3d rendering. If you set the value to 0.0 then the 2d line segment will be thrown away. The magnitude of this parameter should be set fairly small compared to the overall size of your scene's 3d size (ie: 0.01 for a 10x10 scene). For example, a DXF file was loaded in which contained a street map; the overall size of the "world" was about 500,000 x 500,000 units - in this case the **Width for 2d lines/arcs** parameter was set to 5 so that the streets (which were polylines) could be rendered in 3d.

Segments per Arc/Circle

This parameter determines the number of straight line segments that an arc or circle should be broken into, per 360 degrees. For example, a circle uses an angle of 360 degrees so it would be broken into 30 line segments, whereas an arc of angle 90 degrees would be broken into 8 line segments.

Focus Inward Block Prefix

If the name of a DXF BLOCK begins with this prefix string (by your convention of block naming) then the converter will orient all of the polygon's normals outward towards the block's 3d base-point (the 'origin' of the block). This option basically allows you finer control over the automatic orientation of polygon normals than with the **Unify Normals** option. Take for example the case whether you have created a 3d object of a telephone in AutoCAD and grouped it as a block named "**foc_in_telephone**". Whenever this block is inserted into the diagram the converter will re-orient all of the polygon's normals so that they face inwards towards the blocks base point. This option will be applied to an object after the optional **Unify Normals** command has been applied to the object.
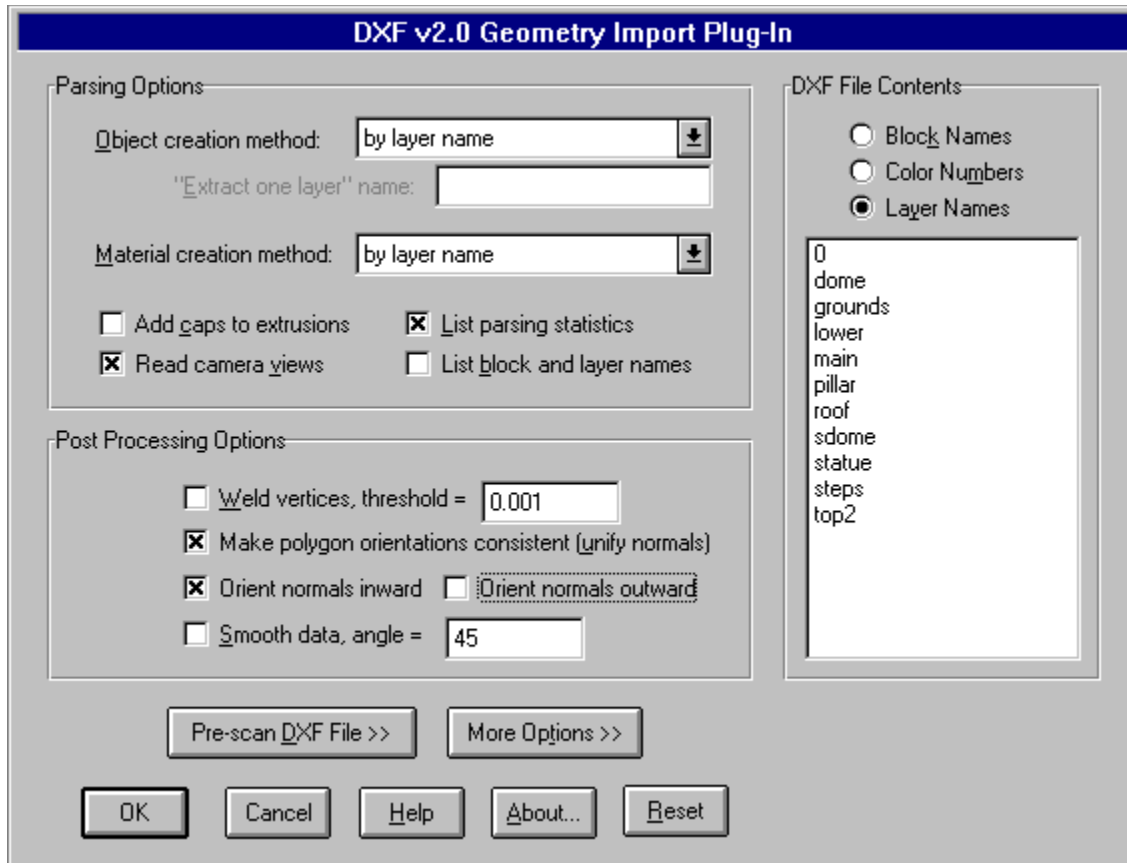
Focus Outward Block Prefix

This is the same as the **Focus Inward Block Prefix** option except that the polygon normals are oriented outwards towards the base point rather than oriented inwards.

Welding, Unification & Smoothing Dialog Box Options

See also:
Important information on welding, unifying and smoothing
Dialog Box Options
Dialog Box # 2 Options

The following options are applicable once the DXF file has been read into memory but before the resulting polygonal data has been   stored within the converter's internal database. See the Important information section for background information on welding, normal unification and normal smoothing.

Weld Vertices, Threshold =

If this checkbox is enabled (check-marked) then the vertex welding operation will be applied to all polygon vertices once the DXF file has been read into memory and the objects created. Vertex welding collapses adjacent vertices which are within a distance less than or equal to the threshold value specified on the dialog box This welding operation should be performed if the **Smooth Data** option (see below) is enabled. Note that welding of vertices can only occur within a single object and not between different objects. Also, the welding operation can become quite slow for objects with large number of vertices since every vertex must be compared against every other vertex (an n-squared problem). This option is disabled by default.

Threshold Value Type-In

If the distance between two vertices is less than or equal to this number, and the **Weld Vertices** checkbox is check-marked (enabled) then the two vertices will be collapsed (welded) into one.

Unify Normals

If this checkbox is enabled (check-marked)   then the DXF import converter will walk across the polygon mesh and reorient each polygon so that it has the same orientation as its neighboring polygon(s). This "**unification**" process if often necessary because 3d models created with AutoCAD typically do not have any consistent polygon orientation (ie: some polygons are oriented clockwise while other are oriented counter-clockwise). Consistent polygon orientation is a prerequisite for the '**Smooth Data**' option (described below).

Notes:

1. This function will only work if neighboring polygons share   the same vertices; if this is not the case then enable the '**Weld Vertices**' option described above.

2. The unification of normals can only occur within a single object and not between different objects.

3. This option is enabled by default and should be quite fast for most objects.

4. This option should NOT be enabled if any of the DXF entities are **AME solid models** in which case you should unify the objects on a one-by-one basis (such as with the NuGraf software or within 3D Studio).

Orient Normals Outward

In some cases the '**Unify Normals**' option described above will flip all of the geometric and vertex normals so that they point inwards towards the center of each object. If this occurs then this option should be enabled (checkmarked) so that the majority of the normals point outwards.

Orient Normals Inward

This option can be enabled to cause the majority of the polygon normals to point inward toward the centroid of the object. Normally you would want to enable the '**Orient Normals Outward**' option instead.

Smooth Data, Angle =

If this checkbox is enabled (check-marked) then new vertex normals will be computed for a polygon if it does not have any vertex normals already. The smoothing criterion is based on the angle between adjacent polygons; common smoothed vertex normals will be computed if the angle between their geometric surfaces normals is less than the **Smoothing Angle** type-in value described below.

If this checkbox is disabled (un-checkmarked) then no new vertex normals will be created.

NOTE: for this option to work properly the input data must have its vertices welded together so that adjacent polygons share the same vertices (the **Weld Vertices** checkbox must be enabled (check-marked)). You may optionally have to enable the **Unify Normals** option so that the polygon normals are oriented properly in order for the smoothed vertex normals to be computed. This option is enabled by default.

Smoothing Angle Type-in

This type-in value sets the cut-off smoothing angle (in degrees) used by the vertex normal computation algorithm. If the angle between two adjacent polygons is greater than this number then no smoothing will occur (a   crease will be evident between the two polygons), else smoothing will occur where the polygons meet. The default angle is 45 degrees; higher values will make the surface smoother.

Import Information on Welding/Unifying/Smoothing

If the imported DXF data is to be sent to a 3D renderer for rendering then it is quite important to understand the concepts of **welding**, **unifying** and **smoothing**.

Except for DXF meshes, AutoCAD entities are imported and converted into 3-point or 4-point planar polygons. Since the polygons are created on a one-by-one basis no polygons share the same vertex coordinates as any other polygons, even though they may appear to share the same vertices. In order for the normal unification and smoothing code to work all polygons must be welded together so that adjacent polygons share the same physical vertices (the result is similar to an AutoCAD mesh where all of the vertices are shared between polygons). This welding is achieved by enabling the **Weld Vertices** option and setting the **Weld Vertices Threshold** value to an appropriate value. **If you know that the input data only uses polygon mesh entities then this welding operation is not required.**

If the distance from one vertex to another is less than or equal to the **Weld Vertices Threshold** parameter then the two vertices will be collapsed into one, else the two vertices will be kept unique. After you import a DXF file you may find that some or all polygon faces did not get smoothed properly; this is an indication that the threshold parameter was set too small since the polygons did not get welded together. In this case examine the imported model and try to determine the maximum distance between adjacent un-welded vertices and try re-importing the model again with the new threshold parameter value.

The next (optional) step after having the vertices welded is to unify the polygon normals. A "**polygon normal**", in the simplest terms, is a line sticking out perpendicular from the "**outside face**" of a polygon. The orientation of this polygon normal is important to some rendering programs since they will throw away polygons which have their normals facing away from the camera (termed "**backface culling**") in order to speed up the rendering process. Unfortunately, many DXF files contain polygons which have random polygon normal orientations. The converter can attempt to "unify" or re-orient these normals by having the **Unify Normals** option enabled. By doing so the converter will walk across the polygon mesh and reorient each polygon so that all adjoining polygons share the same orientation.

The last step is to smooth out the data. "**Smoothing**" is the process of creating artificial normals for each polygon vertex which will be used by a rendering program to make a faceted model appear to be completely smooth. These vertex normals are used, for example, to make a sphere appear to be smooth when in fact the sphere is made up of flat polygons. The import converter can attempt to create smoothed vertex normals by having the **Smooth Data** option enabled and by setting the **Smooth Data Angle** type-in value to an appropriate angle (defaults to 45 degrees). See the appropriate options in the [Welding, Unification and Smoothing Options](#) section for an explanation of these options. NOTE: a polygon's vertices must be welded together and its polygon normal unified before the smoothing operation can be applied to it.

Option Notes

- Entities residing on layers which are "**frozen**" or which are "**off**" will be ignored if the **Ignore Layers Which are Frozen** and/or **Ignore Layers Which are Off** converter options are enabled (checkmarked).

- If you do not want 2d lines to be imported as 3d polygons then set the **Width for 2d lines/arcs** option to 0. Take, for example, the case of the '**nozzle3d.dwg**' diagram of AutoCAD fame; if this option is set to non-zero then the 2d drafting diagram within this 3d diagram will be imported (as very thin polygons in place of the line segments) in addition to the true 3d mesh geometry of the nozzle - you may not want the 2d diagram to be imported so just set this option to 0.

- If you are accustomed to exploding AutoCAD AME meshes prior exporting a DXF file then this step is no longer necessary.

Highlights of the DXF Converter

The following describe the major features and highlights of the DXF converter:

- Conformance to the AutoCAD R12 DXF specification (use AutoDesk's "**dxfix13**" program to convert R13 DXF files to R12).

- Reads both ASCII and binary versions of the DXF format.

- The '**Pre-Scan DXF File**' option allows the DXF file to be pre-scanned for color numbers, layer names and block names.

- Retains blocks and instances within the imported database (no object replication).

- Creates cameras from the **VIEW** entries found in the DXF file. The cameras are compliant with the unpublished AutoCAD **'VIEW'** algorithm for perspective cameras.

- Reads in almost every popular DXF entity and converts them into optimized 3d polygon meshes by a variety of means.

- Allows 2d entities with no width or thickness to be intelligently converted into 3d polygons of very narrow width. This allows, for example, 2d drafting diagrams to be imported and rendered in 3-dimensions. It can also be used, for example, to import GIS data that consists of DEM terrain and simple 3d polylines (very powerful!). This feature is quite useful since 3D AutoCAD diagrams often use 2d lines and 2d arcs to add fine detail to a scene; without this feature such detail would be non-existent within the 3d imported scene.

- The converter is extremely memory efficient, thanks to many users sending rather large (> 10Mb) data files.

Please note that the DXF file format can be considered more of a "guideline" document rather than a strict 2d/3d data exchange format; the result of this is that many software programs write out DXF files in a non-standard method, contrary to the AutoCAD DXF file specification. For example, some 3D modeling programs write out only the **ENTITIES** section of the DXF file format with no **HEADER** or **TABLE** information - this is strictly illegal. In addition, some   DXF files **INSERT** blocks which have not been previously defined and other DXF   files write out entity information with bad parameters. This converter should be able to handle most bad files; however, if you find a DXF file which cannot be read by this converter then please forward it to us and we will attempt to correct the problem.

For a detailed description of how this converter performs its various   conversion tasks please see the Supported Entities and their Conversions section.

Notes About DWG Conversion

Many people are lead to believe that the AutoDesk DWG file format contains more data than an equivalent DXF file. This is purportedly not true. Both files contain the exact same data except that the DWG file is written in a proprietary format which Autodesk continues to change between releases of AutoCAD. Thus, importing from a DXF file should provide the same results as if the data were imported from a DWG file (the DWG format is not supported by this converter).

Overview of DXF Files

DXF (Drawing Interchange Format) is an alternative method to the DWG file format used by AutoCAD to import and export its native file information. While less accurate and more bulky than DWG, the DXF format is recognized by a large number of third party software applications and hence has become a more-or-less standard for interchanging 2d and 3d data.

A DXF file is either written in a pure ASCII format or a mixture of binary and ASCII. A typical DXF file is segmented into four sections: the **Header** information, the **Tables** section, the **Blocks** and the **Entities**. The **Header** contains comments about the file (999 group) and variables relating to the drawing; the **Tables** section lists the layer names contained in the DXF file as well as the 3d camera views; the **Blocks** section contains a collection of grouped drawing entities which can be inserted one or more times (INSERT'ed or INSTANCE'd) in the final drawing - the use of blocks saves considerable space in a DXF file since only one copy of drawing data has to be specified; the **Entities** section contains all of the drawing commands used to specify the drawing as well as references to blocks which are to be inserted into the diagram (for example, the blocks section could define a pillar and the entities section could insert this block 200 times in a regular grid).

Drawing primitives in AutoCAD are called **'entities'**. Many such entities exist but the most popular are as follows: line, circle, arc, trace, solid, insert, 2d polyline, 3d polyline, 3d face, 3d PolyFace meshes and 3d Polyline meshes. Most of these entities are two dimensional (flat) although the 3d face and 3d mesh entities allow 3d data to be explicitly defined.

AutoCAD allows 2d entities to be extruded along a third axis by way of the optional **'thickness'** parameter associated with some entities. This allows, for example, a 2d line to be extruded into a 3d face, or a 2d circle to be extruded into a cylinder.

In addition, AutoCAD allows lines with no width (infinitely narrow lines) to be widened at their start and end points. These widened lines can then be extruded to form volumes if their **'thickness'** parameter is set greater than 0.

Along with each entity is specified a number of optional entity properties. These are:

- The color number which corresponds to the AutoCAD display color or to the BYLAYER option which causes the entity to take on whatever color has been assigned to the current layer;

- The layer name which specifies on which layer this entity resides; and

- The **'thickness'** parameter for lines, arcs, circles and 2d polylines.

Supported Entities and their Conversions

In order for the converter to accept an entity into its internal database the entity must be 3-dimensional. This excludes the direct importation of DXF arcs, circles, lines and polylines since they have no area. However, this DXF import converter performs a great deal of manipulation of these 2d entities so that they may be read in as 3d polygons.

The conversion process is described as follows:

- **arc** - If an arc has 'width' then it will be turned into a series of polygons that approximate an arc of the desired width (all arcs of 0-width can be artificially widened so that they become 3d polygons by setting the **Width for 2d lines/arcs** option non-zero). If the arc has thickness then it will be extruded in the 'Z' direction of the ECS (entity's coordinate system); in addition, if the **Add Caps To Extrusions** option is enabled then the top and bottom faces of the extruded arc will be capped with polygons. If the angle of the arc is 360 degrees then the arc will be replaced with a circle.

- **circle** - If a circle has no thickness then it will be converted into a flat 3d 'disc' which approximates a circle with a series of polygonal faces. If the circle has 'thickness' then it will be extruded into a hollow cylinder with no ends caps unless the **Add Caps To Extrusions** option is enabled.

- **line** - If a line has 'width' then it will be widened into a 3d polygon (of varying start and end widths). If the line has 'thickness' then it will be extruded in the 'Z' direction into a single 3d polygon or a series of 3d polygons defining a volume. If a line has no width or thickness then it will be rejected by the converter unless the **Width for 2d lines/arcs** option is set non-zero in which cause the line will be artificially widened into a very thin polygon; this latter option allows a 2d flat drafting diagram (for example) to be turned into very thin polygons that can be subsequently rendered in a 3d rendering program.

- **2d and 3d polylines**. If the polyline has a non-zero 'bulge' factor in any of its segments then those line segments will be replaced with segmented arcs. If any line segments have a non-zero 'width' factor then those line segment(s) will be replaced with 'fat line' segments represented as planar 3d polygons. If any line segments have a non-zero 'thickness' factor then a 2d line segment will be extruded upwards in the 'Z' direction to form a 3d polygon, whereas a previously pre-widened 3d polygon (a line with a non-zero 'width' factor) will be extruded upwards in the 'Z' direction to form an enclosed volume defined by a number of polygons. If a polyline is not CLOSED, and it does not have any width or thickness then the converter will not import it into the 3d database (note, however, that if the **Width for 2d lines/arcs** option is set non-zero then such 2d lines will be widened into very thin 3d polygons).

- **polyface mesh**. This is a series of vertices and their associated face indexes that compactly specify a 3d object. This entity is very similar to the **'IndexedPolygon'** primitive used internally within the converter's database and as such this entity converts with little change.

- **polygon MxN mesh**. This is a uniform grid of 3d polygons. The converter handles the optional closure of the mesh in the M, N or M&N directions. The entity is converted directly to the converter's internal **'IndexedPolygon'** primitive.

- **3D face** - 3d faces are collected and stored together as 3-point or 4-point polygons in objects using the converter's internal **'IndexedPolygon'** primitive.

- **trace** - traces are 3 or 4 sided line segments with non-zero widths which get converted to equivalent closed 3d polygons.

- **solid** - solids get converted to 3 or 4 sided polygons. Do not confuse the 2d simplistic 'solid' with AutoCAD R13's 3D Solid primitives.

- **insert** - Blocks will be inserted with scale/rotate/translate transformations and optionally the converter can instance a block several times in a MxN grid. Note that instancing a block will not duplicate the block geometry data within the converter's internal database - rather, instances will be created of the imported block just as with AutoCAD.

Special Treatment of Entities

The **2d polyline** is given "extra special" treatment by the DXF converter.

- First, all bulges in the polyline are converted to arc segments.

- Next, the converter orients all of the resulting line segments of the polyline so   that they are connected in a head-to-tail fashion (a requirement of a 3d polygon).

- Finally, the converter removes any redundant vertex coordinates (such as the last one in the list) so that the vertices form a valid 3d polygon.

- If the polyline has thickness (but no width) then the polyline is extruded into a 3d volume and directly stored in the converter's internal   database as a highly efficient and verified mesh primitive.

- If the polyline has width then it will be stored as a series of less-efficient individual polygon faces.

Since a polyline with extrusion is commonly used to create 3d objects within an AutoCAD drawing, these processing steps ensure that the resulting imported data will result in good 3d renderings of such data.

Limitations of the DXF Converter

- This converter does not support the following entities: **Text**, **Point**, **Attdef**, **Dimension**.

- In addition, no AutoCAD Release 13 primitives are supported, including the ELLIPSE primitive and all of the ACIS 3D Solids. The recommended method for importing DXF R13 files into a R12 compliant DXF file reader is to convert the R13 file to R12 using AutoDesk's "dxfix13" program available from their ftp site "ftp.autodesk.com" in the "pub/autocad/patches/dxfix13/dxfix13.exe" directory.

- The converter does not turn non-planar polygons (which are sometimes common in DXF files) into planar polygons. This may cause problems in some 3d rendering systems.

- The converter does not accommodate the various spline fitting modes (B-Spline or Bezier) associated with the polygon mesh entity.

General Notes

- Objects created with the "**donut**" command are imported properly as arcs with non-zero widths.

- All entities are ultimately converted to 3d polygons and compactly stored within the converter's internal database using the **'IndexedPolygon'** primitive. This primitive allows vast numbers of polygons to be stored in a compact amount of memory.