                    3D Studio File Format (3ds).
                          Autodesk Ltd.

Document Revision 0.8 - December 1994. First Public Release.

If you have any additions or comments to this file please e-mail me.

A lot of the chunks are still undocumented if you know what they
do please email me. As I get more information of the file format
I will document it for everyone to see. I will post this regularly
to alt.3d and I can be contacted there if my email does not work.

Disclaimer.
This document describes the file format of the 3ds files of 3D studio
by Autodesk. By using the information contained within you agree not
to hold me liable if, from its use, you f^Hmuck something up. OK?

Oh and just to make it clear I DO NOT work for Autodesk if you have
any problems with their programs direct it to them not me!

Get to it!

Now with the joviality's aside all this info I have obtained with
lots of work hacking at 3ds files with a diskeditor and diff.
It has taken many months of hard work and piddling around with them
so I hope that it is appreciated.

Remember information wants to be free!

* Jim Pitts. - 18 December 1994

Contact me at jp5@ukc.ac.uk

1.

  The 3ds file format is made up of chunks. They describe what information
  is to follow and what it is made up of, its ID and the location of
  the next main block. If you don't understand a chuck you can quite simply
  skip it. The next chunk pointer is relative to the start of the current
  chunk and in bytes.

  * A Chunk.

  start end size name
  0   1   2   Chunk ID
  2   5   4   Next Chunk

  Chunks have a hierarchy imposed on them that is identified by its ID.
  A 3ds file has the Primary chunk ID 4D4Dh. This is always the first chunk
  of the file. With in the primary chunk are the main chunks.

  * Main Chunks

   id   Description

  3D3D   Start of object mesh data.
  B000   Start of keyframer data.

  The Next Chunk pointer after the ID block points to the next Main chunk.

  Directly after a Main chunk is another chunk. This could be any other
  type of chunk allowable within its main chunks scope.

  For the Mesh description (3D3D) they could be any multiples of.

  * Subchunks of 3D3D. - Mesh Block

   id   Description
   1100   unknown
   1200   Background Colour.
   1201   unknown
   1300   unknown

```
1400        unknown
1420        unknown
1450        unknown
1500        unknown
2100        Ambient Colour Block
2200        fog?
2201        fog?
2210        fog?
2300        unknown
3000        unknown
4000        Object Block
7001        unknown
AFFF        unknown
```

* Subchunks of 4000 - Object Description Block

- first item of Subchunk 4000 is an ASCIIZ string of the objects name.

Remember an object can be a mesh, a light or a camera.

```
 id         Description
4010        unknown
4012        shadow?
4100        Triangular Polygon Object
4600        Light
4700        Camera
```

* Subchunks of 4100 - Triangular Polygon Object

```
 id         Description
4110        Vertex List
4111        unknown
4120        Points List
4160        Translation Matrix
```

* 4110 - Vertex List

start	end	size	type	name
0	1	2	short int	Total vertices in object
2	5	4	float	X value
6	9	4	float	Y value
10	13	4	float	Z value
..	..	.	..	..
..	..	.	..	..

bytes 2 .. 13 are repeated [Total vertices in object] times for each vertex.

* 4111 - unknown

start	end	size	type	name
0	1	2	short int	Total vertices in object ?
2	3	2	short int	unknown
.	.	.	..	..
.	.	.	..	..

 bytes 2..3 are repeated for X times as described by
 short int at start of record.

* 4120 - Points List

start	end	size	type	name
0	1	2	short int	Total polygons in object - numpoly
2	3	2	short int	Point 1
4	5	2	short int	Point 2
6	7	2	short int	Point 3
.	.	.	..	..
.	.	.	..	..

  Repeats 'numpoly' times for each polygon.

  These points refer to the corresponding vertex of
  the triangular polygon from the vertex list.
  Points are organized in a clock-wise order.

 * 4160 - Translation Matrix

  This structure describes a matrix for the object.
  It is stored as a 3 X 4 matrix because it is assumed that
  the right most column is 0,0,0,1

```
start end  size  type          name
 0     3    4    float         matrix 1,1
 4     7    4    float         matrix 1,2
 8    11    4    float         matrix 1,3
12    15    4    float         matrix 2,1
16    19    4    float         matrix 2,2
20    23    4    float         matrix 2,3
24    27    4    float         matrix 3,1
28    31    4    float         matrix 3,2
32    35    4    float         matrix 3,3
36    39    4    float         matrix 4,1
40    43    4    float         matrix 4,2
44    47    4    float         matrix 4,3
```


 * 4600 - Light


```
start end  size  type          name
 0     3    4    float         Light pos X
 4     7    4    float         Light pos Y
 8    11    4    float         Light pos Z
```

  after this structure check for more chunks.

```
    id              Description    ( full description later )
   0010             RGB colour
   0011             24 bit Colour
   4610             Spot light
   4620             Light is off   (Boolean)
```


 * 4610   - Spot Light


```
start end  size  type          name
 0     3    4    float         Target pos X
 4     7    4    float         Target pos Y
 8    11    4    float         Target pos Z
12    15    4    float         Hotspot
16    19    4    float         Falloff
```


 * 0010 - RGB colour

```
start end  size  type          name
 0     3    4    float         Red
 4     7    4    float         Green
 8    11    4    float         Blue
```


 * 0011 - RGB colour - 24 bit

```
start end  size  type          name
 0     0    1    byte          Red
 1     1    1    byte          Green
 2     2    1    byte          Blue
```


 * 4700 - Camera

```
   Describes the details of a camera in the scene.

   start end  size  type          name
    0    3     4     float         Camera pos X
    4    7     4     float         Camera pos Y
    8    11    4     float         Camera pos Z
    12   15    4     float         Target pos X
    16   19    4     float         Target pos Y
    20   23    4     float         Target pos Z
    24   27    4     float         Camera Bank
    28   31    4     float         Camera Lens
```

* 7001 - unknown chunk

nothing known about this chunk except for its Subchunks.
This chunk also exists as a Subchunk in chunk B000 (keyframer info).

```
  id          Description
 7011         unknown
 7020         unknown
```

* B000   - Keyframer Main chunk.

Subchunks are

```
  id          Description
 B00A         unknown
 7001         unknown
 B008         Frames
 B009         unknown
 B002         Start object description
```

* B008   - Frame information

   simple structure describing frame info.

```
 start end  size  type          name
    0    3     4    integer      start frame
    4    7     4    integer      end frame
```

* B002   - Start of Object info

   Subchunks

```
  id          Description
 B010         Name & Hierarchy
 B011*        Name Dummy object
 B013         unknown
 B014*        unknown
 B015         unknown
 B020         Objects pivot point?
 B021         unknown
 B022         unknown
```

      (* only on dummy objects)

* B010   - Name & hierarchy descriptor

```
 start end  size  type          name
    0    ?    ?    ASCIIZ        Object name
    ?    ?    ?    short int     unknown
    ?    ?    ?    short int     unknown
    ?    ?    ?    short int     Hierarchy of object
```
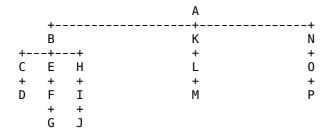
   The object hierarchy is a bit complex but works like this.
   Each object in the scene is given a number to identify its
   order in the tree. Also each object is ordered in the 3ds
   file as it would appear in the tree.
   The root object is given the number -1 (FFFF).
   As the file is read a counter of the object number
   is kept.

Is the counter increments the object are children of the previous
objects. But when the pattern is broken by a number that will be
less than the current counter the hierarchy returns to that level.

for example.

```
        object    hierarchy
        name

          A        -1
          B         0
          C         1                This example is taken
          D         2                from 50pman.3ds.
          E         1
          F         4                I would really recommend
          G         5                having a look at one of
          H         1                the example with the
          I         7                hierarchy numbers to help
          J         8                work it out.
          K         0
          L        10
          M        11                (if you can describe it
          N         0                 any better please let
          O        13                 me know. )
          P        14
```

```
                            A
          +-------------------+---------------+
          B                   K               N
     +---+---+                +               +
     C   E   H                L               O
     +   +   +                +               +
     D   F   I                M               P
         +   +
         G   J
```

Still not done with this chunk yet!

If the object name is $$$DUMMY then it is a dummy object
and therefore you should expect a few extra chunks.

* B011   - Dummy objects name.

    Names a dummy object. ASCIIZ string.

* B020 - Pivot Point?

The objects pivot point. Not quite sure what the first
five floats do yet (ideas?).

start	end	size	type		name
0	3	4	float		unknown
4	7	4	float		unknown
8	11	4	float		unknown
12	15	4	float		unknown
16	19	4	27 4	float	Pivot Y
28	32	4	float		Pivot Z