Open Design Specification for .dwg files

Version 5.4.1

Open Design Alliance

Open Design Specification for .dwg files

Table of Contents

1 Introduction

Originating in the late 1970s, drawing files created with microcomputer-based computer-aided design software were saved with the .dwg extension. In the early 1980s, Autodesk® released AutoCAD® which eventually became the most used CAD software in the world and which used Autodesk's undocumented and proprietary DWG™ file format (using the .dwg extension).

The Open Design Specification for .dwg files serves AutoCAD's undocumented and proprietary DWG file format. This specification includes DWG file format versions 13 up to and including version 2013. Further, the Open Design Specification for .dwg files serves the Teigha® software development platform of the Open Design Alliance.

While our Open Design Specification for .dwg files is able to read and write .dwg files with excellent AutoCAD compatibility, we continue to work to improve our understanding of all the data in a .dwg file. If you find information which will help us to understand any unknown values, please contact us at http://www.opendesign.com/contact.

2 BIT CODES AND DATA DEFINITIONS

NOTE: Unless otherwise stated, all data in this manual is in little-endian order, with the least significant byte first.

Much of the data in the DWG file format versions 13/14/2000/2004/2007/2010 must be read at the bit level. Various parts of the drawing use data in compressed forms, which are explained below. Here are the abbreviations used in this document for the various compressed forms:

B	:	bit (1 or 0)
BB	:	special 2 bit code (entmode in entities, for instance)
3B	:	bit triplet (1-3 bits) (R24)
BS	:	bitshort (16 bits)
BL	:	bitlong (32 bits)
BLL	:	bitlonglong (64 bits) (R24)
BD	:	bitdouble
2BD	:	2D point (2 bitdoubles)
3BD	:	3D point (3 bitdoubles)
RC	:	raw char (not compressed)
RS	:	raw short (not compressed)
RD	:	raw double (not compressed)
RL	:	raw long (not compressed)
2RD	:	2 raw doubles
3RD	:	3 raw doubles
MC	:	modular char
MS	:	modular short
H	:	handle reference (see the HANDLE REFERENCES section)
T	:	text (bitshort length, followed by the string).
TU	:	Unicode text (bitshort character length, followed by Unicode string, 2 bytes per character). Unicode text is read from the “string stream” within the object data, see the main Object description section for details.
TV	:	Variable text, T for 2004 and earlier files, TU for 2007+ files.
X	:	special form
U	:	unknown
SN	:	16 byte sentinel
BE	:	BitExtrusion
DD	:	BitDouble With Default
BT	:	BitThickness
3DD	:	3D point as 3 DD, needing 3 default values

CMC	:	CmColor value
TC	:	True Color: this is the same format as CMC in R2004+. OT : Object type

A “seeker” is an RL-type object which indicates either an absolute address in the file or an offset from some known address.

A “sentinel” is 16 bytes of data used for file recovery purposes.

Generally, the compressed forms are used to allow for compression of common data, usually values like 0.0 and 1.0 for doubles, 0 and 256 for shorts. The method for interpreting the code is to read the first two bits, which indicate either the size of the data to follow, or the actual value for the common values. Here are the compressed formats and some examples of how they appear in the file:

2.1 3B

This is a sequence of 1 to 3 bits. Keep reading bits until a zero bit is encountered or until the 3rd bit is read, whatever comes first. Each time a bit is read, shift the previously read bits to the left. The result is a number 0-7.

2.2 BITSHORT:

1st 2 bits	:	what it is
00	:	A short (2 bytes) follows, little-endian order (LSB first)
01	:	An unsigned char (1 byte) follows
10	:	0
11	:	256

The char size is used when positive shorts less than 256 are being stored. The short size is used when values <0 or >=256 are being stored. Obviously the special cases for 0 and 256 are used when those values are being stored.

Negative numbers use the short form, not the char form. That is, -1 is `00.11111111.11111111`, not `01.11111111`.

For instance, if we were known to be reading 5 shorts from the following stream of bits:

```
00000000010000000110110100001111110
```

It would be parsed like this:

```
00 00000001 00000001  (short 257)
10                    (0)
11                    (256)
```

```
01 00001111           (15)
10                    (0)
```

2.3 BITLONG:

1st 2 bits	:	what it is
00	:	A long (4 bytes) follows, little-endian order (LSB first)
01	:	An unsigned char (1 byte) follows
10	:	0
11	:	not used

The char size is used when positive longs less than 256 are being stored. The long size is used when values <0 or >=256 are being stored. Obviously the special case for 0 is used when storing 0.

Negative numbers use the short form, not the char form. That is, -1 is

```
00.11111111.11111111.11111111.11111111, not 01.11111111.
```

For instance, if we were known to be reading 5 longs from the following stream of bits:

```
0000000001000000010000000000000000100100001111 10
```

It would be parsed like this:

```
00 00000001 00000001 00000000 00000000 (long 257)
10                    (0)
01 00001111           (15)
10                    (0)
```

2.4 BITLONGLONG

The first 1-3 bits indicate the length *l* (see paragraph 2.1). Then *l* bytes follow, which represent the number (the least significant byte is first).

2.5 BITDOUBLE:

1st 2 bits	:	what it is
00	:	A double follows
01	:	1.0
10	:	0.0
11	:	not used

Doubles are eight byte IEEE standard floating point values.

2.6 MODULAR CHARS:

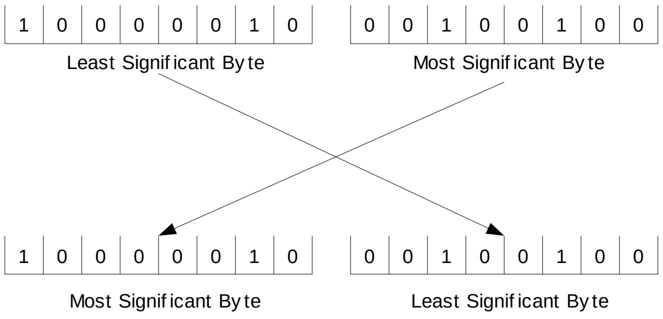
Modular characters are a method of storing compressed integer values. They are used in the object map to indicate both handle offsets and file location offsets. They consist of a stream of bytes, terminating when the high bit of the byte is 0.

In each byte, the high bit is a flag; when set, it indicates that another byte follows. The concept is not difficult to understand, but is a little difficult to explain. Let's look at an example.

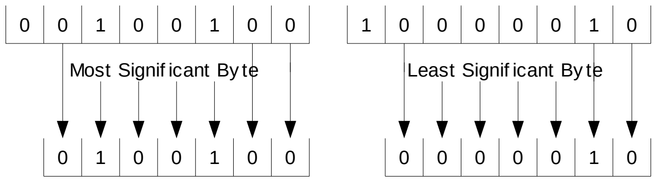
Assume the next two bytes in the file are:

```
10000010 00100100
```

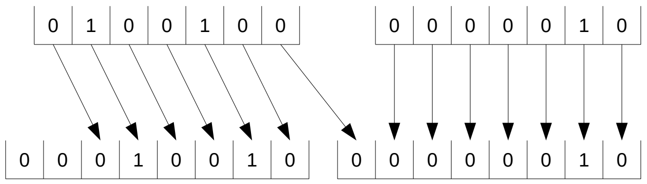
We read bytes until we reach a byte with a high bit of 0. Obviously the second byte meets that criterion. Since we are reading from least significant to most significant, let's reverse the order of the bytes so that they read MSB to LSB from left to right.



Now we drop the high order flag bits:



And then re-group the bits from right to left, padding on the left with 0's:

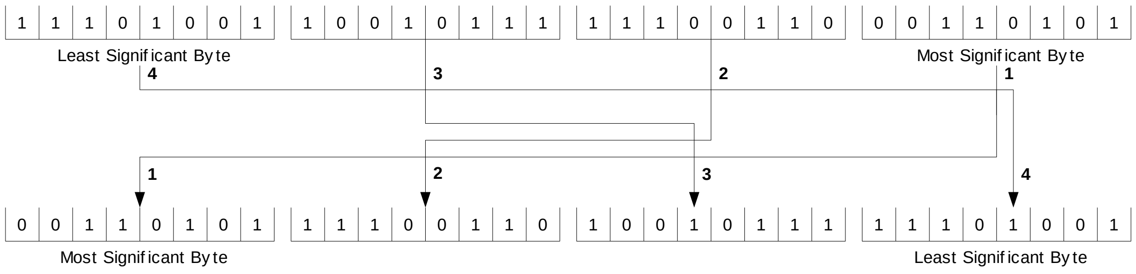


Result = 2 + 18*256 = 4610

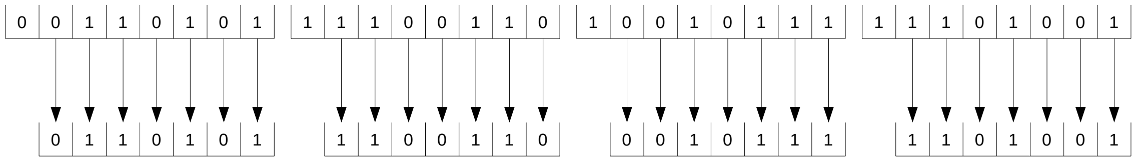
Here's another example using the basic form `F1101001 F0010111 F1100110 00110101`:

```
11101001 10010111 11100110 00110101
```

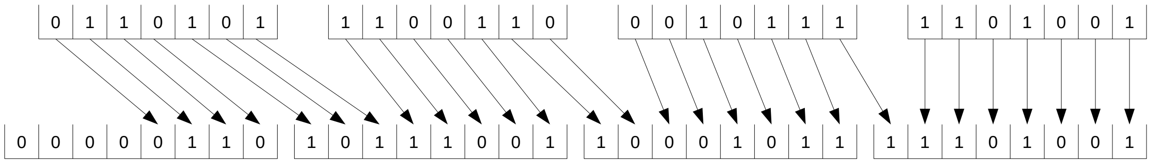
We read bytes until we reach a byte with a high bit of 0. Obviously the fourth byte meets that criterion. Since we are reading from least significant to most significant, let's reverse the order of the bytes so that they read MSB to LSB from left to right.



Now we drop the high order flag bits:



And then re-group the bits from right to left, padding on the left with 0's:

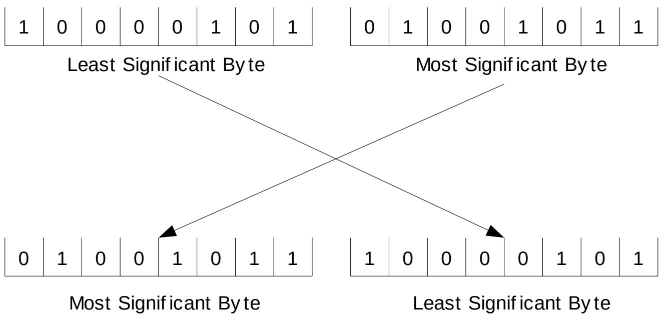


Result:233+139*256+185*256^2+6*256^3=112823273

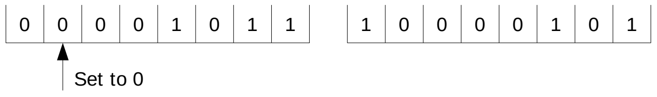
This process is further complicated by the fact that if the final byte (high bit 0) also has the 64 bit (0x40) set, this means to negate the number.

This is a negative number: `10000101 01001011`

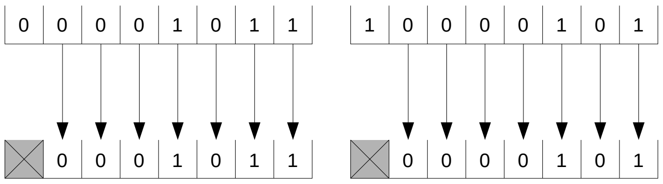
Since we are reading from least significant to most significant, let's reverse the order of the bytes so that they read MSB to LSB from left to right.



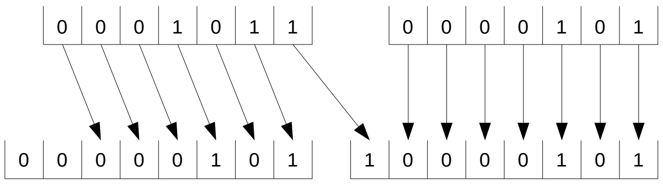
We then clear the bit that was used to represent the negative number, and note that the result must be negated:



Now we drop the high order flag bits:



And then re-group the bits from right to left, padding on the left with 0's:



Result: 133+5*256=1413, which we negate to get –1413

Modular chars are also used to store handle offsets in the object map. In this case there is no negation used; handles in the object map are always in increasing order.

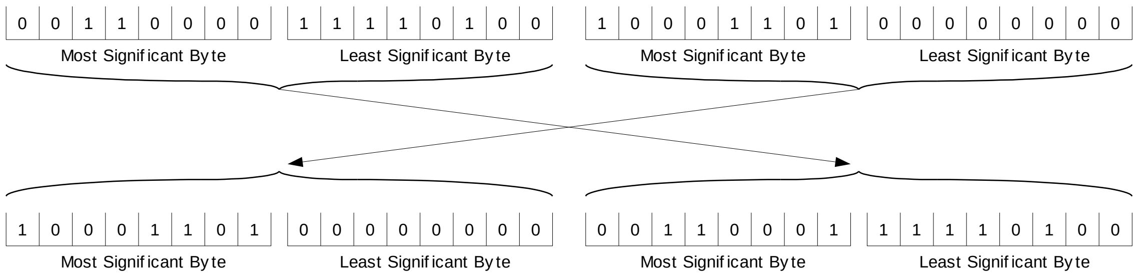
2.7 MODULAR SHORTS

Modular shorts work just like modular chars -- except that the base module is a short instead of a char.

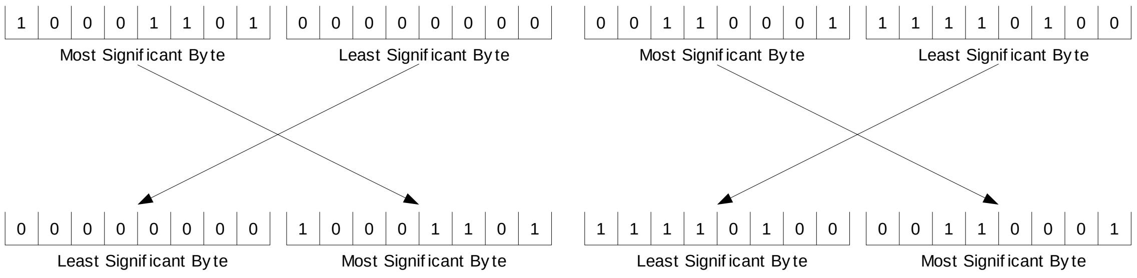
There are only two cases to worry about here (from a practical point of view), because, in the case of shorts, two modules make a long, and since these are used only to indicate object sizes, a maximum object size of 1 GB is probably correct.

```
00110001 11110100 10001101 00000000.
```

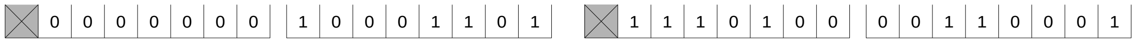
Reverse the order of the shorts:



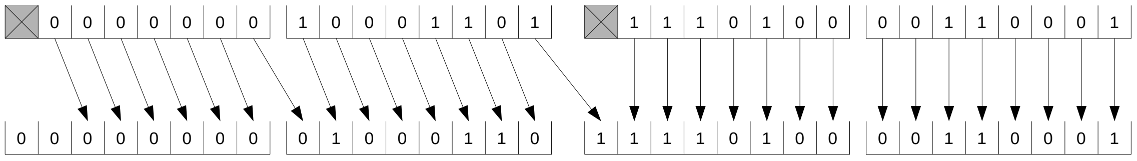
Reverse the order of the bytes in each short:



Drop the high order flag bit of each short:



And then re-group the bits from right to left, padding on the left with 0's:



Result: 62513+70*65536=4650033

2.8 Bit Extrusion

For R13-R14 this is 3BD. For R2000, this is a single bit, followed optionally by 3BD. If the single bit is 1, the extrusion value is assumed to be 0,0,1 and no explicit extrusion is stored. If the single bit is 0, then it will be followed by 3BD.

2.9 BitDouble With Default

This is a 2 bit opcode followed optionally by data, and it requires a default value. The different opcodes are described as follows:

00 No more data present, use the value of the default double.

01 4 bytes of data are present. The result is the default double, with the 4 data bytes patched in replacing the first 4 bytes of the default double (assuming little endian).

10 6 bytes of data are present. The result is the default double, with the first 2 data bytes patched in replacing bytes 5 and 6 of the default double, and the last 4 data bytes patched in replacing the first 4 bytes of the default double (assuming little endian).

11 A full RD follows.

2.10 BitThickness

For R13-R14, this is a BD. For R2000+, this is a single bit followed optionally by a BD. If the bit is one, the thickness value is assumed to be 0.0. If the bit is 0, then a BD that represents the thickness follows.

2.11 CmColor

```
R15 and earlier: BS color index


R2004+: There are two types of color definitions, below named as CMC and ENC:


CMC:
           BS : color index (always 0)
           BL : RGB value
           RC : Color Byte (& 1 => color name follows (TV),
                            &2 => book name follows (TV))


ENC: This color is used by entities: this definition may contain a DBCOLOR reference and optional
               transparency.


           BS : color number: flags + color index
                color flags: first byte of the bitshort.
                      0x8000: complex color (rgb).
                        Next value is a BS containing the RGB value (last 24 bits).
                      0x4000: has AcDbColor reference (0x8000 is also set in this case).
                        The handle to the color is written in the handle stream.
                      0x2000: color is followed by a transparency BL.
                        The first byte represents the transparency type:
                          0 = BYLAYER,
                          1 = BYBLOCK,
                          3 = the transparency value in the last byte.
               Color index: if no flags were set, the color is looked up by the color number (ACI
               color).
```

2.12 Object type

Until R2007, the object type was a bit short. In R2010 the object type changed:

A bit pair, followed by either 1 or 2 bytes, depending on the bit pair value:

Bit pair value	How to interpret following 1-2 bytes
0	Read the following byte
1	Read following byte and add 0x1f0.
2	Read the following two bytes (raw short)
3	The value 3 should never occur, but interpret the same as 2 nevertheless.

2.13 HANDLE REFERENCES:

All objects in R13+ .dwg files are referred to by object handles. These handles are stored in the file in the following form:

```
    |CODE (4 bits)|COUNTER (4 bits)|HANDLE or OFFSET|
```

In this document we write these as CODE.COUNTER.BYTE.BYTE..., such as 0101.0001.00001111 (the usual reference to LAYER 0 for drawings created under R13, which has handle F). In abbreviated form, we write 5.1.0F.

The CODE has different meanings depending on the handle. Certain object handles in AutoCAD have "ownership" relations with other objects. In these cases the code indicates the type of relation:

Code	Desciption
2	Soft ownership reference: the owner does not need the owned object. The owned object cannot exist by itself.
3	Hard ownership reference: the owner needs the owned object. The owned object cannot exist by itself.
4	Soft pointer reference: the referencing object does not need the referenced object and vice versa.
5	Hard pointer reference: the referencing object needs the referenced object, but both are owned by another object.

We will call these TYPEDOBJHANDLEs. Often their type is fixed to some value at all times; in other words, for instance in a certain position only a HARD_POINTER TYPEDOBJHANDLE would be allowed.

In other cases, the handle is stored as an offset from some other handle, and the code indicates how the offset is to be applied. These handles always represent a soft pointer reference. See the table below for the codes and their meaning:

Code:	**Action:**
`0x2, 0x3, 0x4, 0x5`	`none - just read offset and use it as the result`
`0x6`	`result is reference handle + 1  (length is 0 in this case)`
`0x8`	`result is reference handle – 1  (length is 0 in this case)`
`0xA`	`result is reference handle plus offset`
`0xC`	`result is reference handle minus offset`

We will call these OFFSETOBJHANDLEs. These handles are described with (CODE X), where X indicates the code if the offset is an ABSOLUTE reference (0x2 – 0x5).

COUNTER tells how many bytes of HANDLE follow.

EXAMPLE: An entity on a layer whose handle is 5E7 has the following handle reference near the end of the entity data (its code being 5):

```
       5   2    0   5    E   7
    01010010 00000101 11100111  (0101.0010.00000101.11100111)
```

2.14 CRCS:

2.14.1 8-bit CRC

The AutoCAD DWG file format uses a modification of a standard cyclic redundancy check as an error detecting mechanism. The CRC ends up being 2 bytes long due to a lookup in a table containing 256 16-bit values, and are not stored in any sort of bit code form. They also always appear on byte boundaries; they are not embedded within the stream of bits. Thus there may be extra unused bits at the end of an object. For instance, consider an object containing one bitshort, as follows:

```
01000000 11100000 01010101 01010101
```

This parses as:

```
01  bitshort with one character
00000011 the value of the bitshort (3)
100000 unused bits
01010101 01010101  the CRC
```

The modification that is made to the CRC is that a starting value different from 0 is used. Autodesk also uses a method whereby the result of the CRC is XORed with a "magic number". This method is used extensively in pre-R13 files, but seems only to be used in the header for R13 and beyond.

Here is the CRC function we use; it is simply a standard 8 bit CRC calculation:

```
int crctable[256]= {

0x0000,0xC0C1,0xC181,0x0140,0xC301,0x03C0,0x0280,0xC241,
0xC601,0x06C0,0x0780,0xC741,0x0500,0xC5C1,0xC481,0x0440,
0xCC01,0x0CC0,0x0D80,0xCD41,0x0F00,0xCFC1,0xCE81,0x0E40,
0x0A00,0xCAC1,0xCB81,0x0B40,0xC901,0x09C0,0x0880,0xC841,
0xD801,0x18C0,0x1980,0xD941,0x1B00,0xDBC1,0xDA81,0x1A40,
0x1E00,0xDEC1,0xDF81,0x1F40,0xDD01,0x1DC0,0x1C80,0xDC41,
0x1400,0xD4C1,0xD581,0x1540,0xD701,0x17C0,0x1680,0xD641,
0xD201,0x12C0,0x1380,0xD341,0x1100,0xD1C1,0xD081,0x1040,
0xF001,0x30C0,0x3180,0xF141,0x3300,0xF3C1,0xF281,0x3240,
0x3600,0xF6C1,0xF781,0x3740,0xF501,0x35C0,0x3480,0xF441,
0x3C00,0xFCC1,0xFD81,0x3D40,0xFF01,0x3FC0,0x3E80,0xFE41,
0xFA01,0x3AC0,0x3B80,0xFB41,0x3900,0xF9C1,0xF881,0x3840,
0x2800,0xE8C1,0xE981,0x2940,0xEB01,0x2BC0,0x2A80,0xEA41,
0xEE01,0x2EC0,0x2F80,0xEF41,0x2D00,0xEDC1,0xEC81,0x2C40,
0xE401,0x24C0,0x2580,0xE541,0x2700,0xE7C1,0xE681,0x2640,
0x2200,0xE2C1,0xE381,0x2340,0xE101,0x21C0,0x2080,0xE041,
0xA001,0x60C0,0x6180,0xA141,0x6300,0xA3C1,0xA281,0x6240,
0x6600,0xA6C1,0xA781,0x6740,0xA501,0x65C0,0x6480,0xA441,
0x6C00,0xACC1,0xAD81,0x6D40,0xAF01,0x6FC0,0x6E80,0xAE41,
0xAA01,0x6AC0,0x6B80,0xAB41,0x6900,0xA9C1,0xA881,0x6840,
0x7800,0xB8C1,0xB981,0x7940,0xBB01,0x7BC0,0x7A80,0xBA41,
0xBE01,0x7EC0,0x7F80,0xBF41,0x7D00,0xBDC1,0xBC81,0x7C40,
0xB401,0x74C0,0x7580,0xB541,0x7700,0xB7C1,0xB681,0x7640,
0x7200,0xB2C1,0xB381,0x7340,0xB101,0x71C0,0x7080,0xB041,
0x5000,0x90C1,0x9181,0x5140,0x9301,0x53C0,0x5280,0x9241,
0x9601,0x56C0,0x5780,0x9741,0x5500,0x95C1,0x9481,0x5440,
0x9C01,0x5CC0,0x5D80,0x9D41,0x5F00,0x9FC1,0x9E81,0x5E40,
0x5A00,0x9AC1,0x9B81,0x5B40,0x9901,0x59C0,0x5880,0x9841,
0x8801,0x48C0,0x4980,0x8941,0x4B00,0x8BC1,0x8A81,0x4A40,
0x4E00,0x8EC1,0x8F81,0x4F40,0x8D01,0x4DC0,0x4C80,0x8C41,
0x4400,0x84C1,0x8581,0x4540,0x8701,0x47C0,0x4680,0x8641,
0x8201,0x42C0,0x4380,0x8341,0x4100,0x81C1,0x8081,0x4040 };

short crc8(unsigned short dx,char *p,long n)
{
  register unsigned char al;

  while (n-- > 0) {
    al = (unsigned char)((*p) ^ ((char)(dx & 0xFF)));
    dx = (dx>>8) & 0xFF;
    dx = dx ^ crctable[al & 0xFF];
    p++;
  }
  return(dx);
}
```

This function takes as its input an initial CRC value, a pointer to the data to be CRC'd, and the number of bytes of data. The return value is the new CRC. This function can be used to accumulate a CRC by running the first set of bytes with an initial value of 0 (or the "starting value" for this type of object), and subsequent calls with the initial value equal to the last returned CRC.

2.14.2 32-bit CRC

From R18 onwards a 32-bit CRC is used. The algorithm is similar to the 8-bit version, but uses a CRC lookup table containing 256 32-bit values.

```
OdUInt32 crc32Table[] =
{
  0x00000000, 0x77073096, 0xee0e612c, 0x990951ba,
  0x076dc419, 0x706af48f, 0xe963a535, 0x9e6495a3,
  0x0edb8832, 0x79dcb8a4, 0xe0d5e91e, 0x97d2d988,
  0x09b64c2b, 0x7eb17cbd, 0xe7b82d07, 0x90bf1d91,
  0x1db71064, 0x6ab020f2, 0xf3b97148, 0x84be41de,
  0x1adad47d, 0x6ddde4eb, 0xf4d4b551, 0x83d385c7,
  0x136c9856, 0x646ba8c0, 0xfd62f97a, 0x8a65c9ec,
  0x14015c4f, 0x63066cd9, 0xfa0f3d63, 0x8d080df5,
  0x3b6e20c8, 0x4c69105e, 0xd56041e4, 0xa2677172,
  0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b,
  0x35b5a8fa, 0x42b2986c, 0xdbbbc9d6, 0xacbcf940,
  0x32d86ce3, 0x45df5c75, 0xdcd60dcf, 0xabd13d59,
  0x26d930ac, 0x51de003a, 0xc8d75180, 0xbfd06116,
  0x21b4f4b5, 0x56b3c423, 0xcfba9599, 0xb8bda50f,
  0x2802b89e, 0x5f058808, 0xc60cd9b2, 0xb10be924,
  0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d,
  0x76dc4190, 0x01db7106, 0x98d220bc, 0xefd5102a,
  0x71b18589, 0x06b6b51f, 0x9fbfe4a5, 0xe8b8d433,
  0x7807c9a2, 0x0f00f934, 0x9609a88e, 0xe10e9818,
  0x7f6a0dbb, 0x086d3d2d, 0x91646c97, 0xe6635c01,
  0x6b6b51f4, 0x1c6c6162, 0x856530d8, 0xf262004e,
  0x6c0695ed, 0x1b01a57b, 0x8208f4c1, 0xf50fc457,
  0x65b0d9c6, 0x12b7e950, 0x8bbeb8ea, 0xfcb9887c,
  0x62dd1ddf, 0x15da2d49, 0x8cd37cf3, 0xfbd44c65,
  0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2,
  0x4adfa541, 0x3dd895d7, 0xa4d1c46d, 0xd3d6f4fb,
  0x4369e96a, 0x346ed9fc, 0xad678846, 0xda60b8d0,
  0x44042d73, 0x33031de5, 0xaa0a4c5f, 0xdd0d7cc9,
  0x5005713c, 0x270241aa, 0xbe0b1010, 0xc90c2086,
  0x5768b525, 0x206f85b3, 0xb966d409, 0xce61e49f,
  0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4,
  0x59b33d17, 0x2eb40d81, 0xb7bd5c3b, 0xc0ba6cad,
  0xedb88320, 0x9abfb3b6, 0x03b6e20c, 0x74b1d29a,
  0xead54739, 0x9dd277af, 0x04db2615, 0x73dc1683,
  0xe3630b12, 0x94643b84, 0x0d6d6a3e, 0x7a6a5aa8,
  0xe40ecf0b, 0x9309ff9d, 0x0a00ae27, 0x7d079eb1,
  0xf00f9344, 0x8708a3d2, 0x1e01f268, 0x6906c2fe,
  0xf762575d, 0x806567cb, 0x196c3671, 0x6e6b06e7,
  0xfed41b76, 0x89d32be0, 0x10da7a5a, 0x67dd4acc,
  0xf9b9df6f, 0x8ebeeff9, 0x17b7be43, 0x60b08ed5,
  0xd6d6a3e8, 0xa1d1937e, 0x38d8c2c4, 0x4fdff252,
  0xd1bb67f1, 0xa6bc5767, 0x3fb506dd, 0x48b2364b,
  0xd80d2bda, 0xaf0a1b4c, 0x36034af6, 0x41047a60,
  0xdf60efc3, 0xa867df55, 0x316e8eef, 0x4669be79,
  0xcb61b38c, 0xbc66831a, 0x256fd2a0, 0x5268e236,
  0xcc0c7795, 0xbb0b4703, 0x220216b9, 0x5505262f,
  0xc5ba3bbe, 0xb2bd0b28, 0x2bb45a92, 0x5cb36a04,
  0xc2d7ffa7, 0xb5d0cf31, 0x2cd99e8b, 0x5bdeae1d,
  0x9b64c2b0, 0xec63f226, 0x756aa39c, 0x026d930a,
  0x9c0906a9, 0xeb0e363f, 0x72076785, 0x05005713,
  0x95bf4a82, 0xe2b87a14, 0x7bb12bae, 0x0cb61b38,
  0x92d28e9b, 0xe5d5be0d, 0x7cdcefb7, 0x0bdbdf21,
  0x86d3d2d4, 0xf1d4e242, 0x68ddb3f8, 0x1fda836e,
  0x81be16cd, 0xf6b9265b, 0x6fb077e1, 0x18b74777,
  0x88085ae6, 0xff0f6a70, 0x66063bca, 0x11010b5c,
  0x8f659eff, 0xf862ae69, 0x616bffd3, 0x166ccf45,
  0xa00ae278, 0xd70dd2ee, 0x4e048354, 0x3903b3c2,
  0xa7672661, 0xd06016f7, 0x4969474d, 0x3e6e77db,
  0xaed16a4a, 0xd9d65adc, 0x40df0b66, 0x37d83bf0,
  0xa9bcae53, 0xdebb9ec5, 0x47b2cf7f, 0x30b5ffe9,
  0xbdbdf21c, 0xcabac28a, 0x53b39330, 0x24b4a3a6,
  0xbad03605, 0xcdd70693, 0x54de5729, 0x23d967bf,
  0xb3667a2e, 0xc4614ab8, 0x5d681b02, 0x2a6f2b94,
  0xb40bbe37, 0xc30c8ea1, 0x5a05df1b, 0x2d02ef8d
};

  OdUInt32 crc(OdUInt8 *p, OdUInt32 n, OdUInt32 seed)
  {
    OdUint32 invertedCrc = ~seed
    while (n--) {
      OdUInt8 byte = *p++;
      invertedCrc = (invertedCrc >> 8) ^ crc32Table[(invertedCrc ^ byte) & 0xff];
    }
    return ~invertedCrc;
  }
```

3 R13-R15 DWG FILE FORMAT ORGANIZATION

3.1 FILE STRUCTURE

The structure of the DWG file format changed between R13 C2 and R13 C3. Notations regarding C3 below indicate the differences.

The general arrangement of data in an R13/R14/R15 file is as follows:

```
HEADER
  FILE HEADER
  DWG HEADER VARIABLES
  CRC

CLASS DEFINITIONS
TEMPLATE (R13 only, optional)
PADDING (R13C3 AND LATER, 200 bytes, minutes the template section above if present)
IMAGE DATA (PRE-R13C3)
OBJECT DATA
  All entities, table entries, dictionary entries, etc. go in this
  section.
OBJECT MAP
OBJECT FREE SPACE (optional)
TEMPLATE (R14-R15, optional)
SECOND HEADER
IMAGE DATA (R13C3 AND LATER)
```

3.2 FILE HEADER

3.2.1 VERSION ID:

The first 6 bytes are:

Bytes (ascii encoded)	Version
AC1012	R13
AC1014	R14
AC1015	R2000
AC1018	R2004
AC1021	R2007
AC1024	R2010
AC1027	R2013
AC1032	R2018

The next 7 starting at offset 0x06 are to be six bytes of 0 (in R14, 5 0's and the ACADMAINTVER variable) and a byte of 1. We have occasionally seen other values here but their meaning (and importance) is unclear.

3.2.2 IMAGE SEEKER:

At 0x0D is a seeker (4 byte long absolute address) for the beginning sentinel of the image data.

3.2.3 OBJECT FREE SPACE

TODO.

3.2.4 TEMPLATE

This section is optional, see chapter 22.

3.2.5 DWGCODEPAGE:

Bytes at 0x13 and 0x14 are a raw short indicating the value of the code page for this drawing file.

3.2.6 SECTION-LOCATOR RECORDS:

At 0x15 is a long that tells how many sets of recno/seeker/length records follow. Each record has the following format:

```
Record number (raw byte) | Seeker (raw long) | Size (raw long)
```

The records are as follows:

```
          0  :  Header variables (covers beginning and ending sentinels).

          1  :  Class section.

          2  :  Object map.

          3  :  (C3 and later.)  A special table (no sentinels). See unknown section (R13 C3 and
                later). The presence of the 4th record (3) indicates that the C3 file format
                applies.  Just look at the long at 21; if it's 4 or greater, it's the C3-and-later
                format.

          4  :  In R13-R15, points to a location where there may be data stored.  Currently we
                have seen only the MEASUREMENT variable stored here. See chapter 22. This section
                is optional.

Remarks:

                We have seen files with up to 6 sets in this section; the meaning of the sixth one
                is unknown.  The Open Design Toolkit emits files with the first 5 sets only.

         RS  :  CRC for BOF to this point.  Use 0 for the initial value, and depending on the
                number of sets of section-locators, XOR the result with one of the following:

          3  :  0xA598
          4  :  0x8101
          5  :  0x3CC4
          6  :  0x8461
```

The following 16 byte sentinel appears after the CRC:

```
0x95,0xA0,0x4E,0x28,0x99,0x82,0x1A,0xE5,0x5E,0x41,0xE0,0x5F,0x9D,0x3A,0x4D,0x00
```

4 R2004 DWG FILE FORMAT ORGANIZATION

4.1 R2004 File Header

Address	Length	Description
0x00	6	"AC1018" version string
0x06	5	5 bytes of 0x00
0x0B	1	Maintenance release version
0x0C	1	Byte 0x00, 0x01, or 0x03
0x0D	4	Preview address (long), points to the image page + page header size (0x20).
0x11	1	Application version (Acad version that writes the file)
0x12	1	Application maintenance release version (Acad maintenance version that writes the file)
0x13	2	Codepage
0x15	3	3 0x00 bytes
0x18	4	Security flags, default value is 0 (long) 0x0001 = encrypt data (used for all data sections except AcDb:Preview and AcDb:SummaryInfo) 0x0002 = encrypt properties (used for sections AcDb:Preview and AcDb:SummaryInfo) 0x0010 = sign data 0x0020 = add timestamp
0x1C	4	Unknown long (ODA writes 0)
0x20	4	Summary info Address, points to summary info page + page header size (0x20)

0x24	4	VBA Project Address (Optional, write 0 if not present)
0x28	4	0x00000080
0x2C	0x54	0x00 bytes
0x80	0x6C	Encrypted Data (see below)

The encrypted data at 0x80 can be decrypted by exclusive or'ing the 0x6c bytes of data from the file with the following magic number sequence:

29 23 BE 84 E1 6C D6 AE 52 90 49 F1 F1 BB E9 EB

B3 A6 DB 3C 87 0C 3E 99 24 5E 0D 1C 06 B7 47 DE

B3 12 4D C8 43 BB 8B A6 1F 03 5A 7D 09 38 25 1F

5D D4 CB FC 96 F5 45 3B 13 0D 89 0A 1C DB AE 32

20 9A 50 EE 40 78 36 FD 12 49 32 F6 9E 7D 49 DC

AD 4F 14 F2 44 40 66 D0 6B C4 30 B7

This magic sequence can be generated by the following code, which generates the sequence and stores it in the data vector:

```
OdUInt8* p = data.asArrayPtr();
OdUInt32 sz = 0x6c;
int randseed = 1;
while (sz--)
{
  randseed *= 0x343fd;
  randseed += 0x269ec3;
  *p++ = (OdUInt8)(randseed >> 0x10);
}
```

Once decrypted, this sequence of bytes consists of the following data (we will call this data the 2004 File Header Data throughout the remainder of this document). The file header data is repeated at the end of the file (this is the second header data).

Address (from start of 0x6c byte sequence)	Length	Description

0x00	12	“AcFssFcAJMB” file ID string
0x0C	4	0x00 (long)
0x10	4	0x6c (long)
0x14	4	0x04 (long)
0x18	4	Root tree node gap
0x1C	4	Lowermost left tree node gap
0x20	4	Lowermost right tree node gap
0x24	4	Unknown long (ODA writes 1)
0x28	4	Last section page Id
0x2C	8	Last section page end address
0x34	8	Second header data address pointing to the repeated header data at the end of the file
0x3C	4	Gap amount
0x40	4	Section page amount
0x44	4	0x20 (long)
0x48	4	0x80 (long)
0x4C	4	0x40 (long)
0x50	4	Section Page Map Id
0x54	8	Section Page Map address (add 0x100 to this value)
0x5C	4	Section Map Id
0x60	4	Section page array size
0x64	4	Gap array size
0x68	4	CRC32 (long). See paragraph 2.14.2 for the 32-bit CRC calculation, the seed is zero. Note that the CRC calculation is done including the 4 CRC bytes that are initially zero! So the CRC calculation takes into account all of the 0x6c bytes of the data in this table.

The next 0x14 bytes will be copied from the magic number sequence, starting at 0x100 – 0x14. These 0x14 bytes are present in the file header at the beginning of the file, but not at the copy at the end of the file.

The remaining data in the file is broken up into sections. There are 2 types of sections, System Sections and Data Sections. A data section consists of 1 or more section pages, a system section consists of just 1 section page. System sections contain maps to navigate through the data sections and pages. System and data section pages have different page headers.

4.2 Section page checksum

The following function (pseudocode) is used to calculate system and data page checksums as stored in the page header (note that system and data page headers are different).

```
OdUInt32 checksum(OdUInt32 seed, OdUInt8* data, OdUInt32 size)
{
  OdUInt32 sum1 = seed & 0xffff;
  OdUInt32 sum2 = seed >> 0x10;
  while (size != 0)
  {
    OdUInt32 chunkSize = min(0x15b0, size);
    size -= chunkSize;
    for (int i = 0; i < chunkSize; i++)
    {
      sum1 += *data++;
      sum2 += sum1;
    }
    sum1 %= 0xFFF1;
    sum2 %= 0xFFF1;
  }
  return (sum2 << 0x10) | (sum1 & 0xffff);
}
```

4.3 System section page

A System Section page starts with of the following 0x14 bytes of header data:

Address (from start of section)	Length	Description
0x00	4	Section page type: Section page map: 0x41630e3b

		Section map: 0x4163003b
0x04	4	Decompressed size of the data that follows
0x08	4	Compressed size of the data that follows (CompDataSize)
0x0C	4	Compression type (0x02)
0x10	4	Section page checksum

Immediately following this data, there will be CompDataSize bytes of compressed data, which is the actual data for the section. See the Compression section later in this document for details on the compression algorithm used. After the compressed data there is second header chunk, but fields decompressed size, compressed size and checksum are set to zero.

The section page checksum is calculated in two stages. First the checksum (using the data page checksum function) is calculated from the header data, using a seed of 0. The header data's checksum being 0 at this stage, but all other fields should be filled. In the second stage the final checksum is calculated from the compressed data, using the first checksum as the seed.

Each section page must start on a 0x20 byte boundary of the raw data stream. The empty bytes between the start of this section and then end of the previous section are filled with as many bytes as needed from the magic number sequence.

System Sections includeSection page map and Section map. These 2 sections serve as a table of contents for the remaining sections of the file and their pages. Once these 2 sections have been processed, all other sections in the file can be accessed randomly.

4.4 2004 Section page map

The uncompressed (global) section page map contains the following data:

Offset	Length	Description
0x00	4	Section page number, starts at 1, page numbers are unique per file.
0x04	4	Section size

This repeats, with one number and size for each section page in the file, until the end of the section page map. Note that this map also contains a reference to the section map, which is a system section. All other pages are data section pages. The address of each section can be calculated as 0x100 for the first section, and for each subsequent section the address is the previous section address plus the previous section size.

If the section number is negative, this represents a gap in the sections (unused data). For a negative section number, the following data will be present after the section size:

Offset	Length	Description
0x00	4	Parent
0x04	4	Left
0x08	4	Right
0x0C	4	0x00

Taken together, these units of file section information form a vector (1 indexed) of all sections in the file, and this vector will be referred to as the SectionPageMap throughout the remainder of this document.

Section pages are numbered consecutively. The system section pages are the last pages, with a gap of 1 between the page numbers for the data sections and system sections.

4.5 2004 Data section map

The data section map is a map for locating all data sections (i.e. system sections are not present in this map).

The uncompressed Section Info section contains the following data:

Offset	Length	Description
0x00	4	Number of section descriptions (NumDescriptions)
0x04	4	0x02 (long)
0x08	4	0x00007400 (long)
0x0C	4	0x00 (long)
0x10	4	Unknown (long), ODA writes NumDescriptions here.

Next, the following data is repeated NumDescriptions times:

Offset	Length	Description
0x00	8	Size of section (OdUInt64)

0x08	4	Page count (PageCount). Note that there can be more pages than PageCount, as PageCount is just the number of pages written to file. If a page contains zeroes only, that page is not written to file. These “zero pages” can be detected by checking if the page’s start offset is bigger than it should be based on the sum of previously read pages decompressed size (including zero pages). After reading all pages, if the total decompressed size of the pages is not equal to the section’s size, add more zero pages to the section until this condition is met.
0x0C	4	Max Decompressed Size of a section page of this type (normally 0x7400)
0x10	4	Unknown (long)
0x14	4	Compressed (1 = no, 2 = yes, normally 2)
0x18	4	Section Id (starts at 0). The first section (empty section) is numbered 0, consecutive sections are numbered descending from (the number of sections – 1) down to 1.
0x1C	4	Encrypted (0 = no, 1 = yes, 2 = unknown)
0x20	64	Section Name (string)

Following this, the following (local) section page map data will be present, repeated PageCount times:

Offset	Length	Description
0x00	4	Page number (index into SectionPageMap), starts at 1
0x04	4	Data size for this page (compressed size).
0x08	8	Start offset for this page (OdUInt64). If this start offset is smaller than the sum of the decompressed size of all previous pages, then this page is to be preceded by zero pages until this condition is met.

Maximum section page size appears to be 0x7400 bytes in the normal case. If a logical section of the file (the database objects, for example) exceeds this size, then it is broken up into pages of size 0x7400. In this case, the PageCount value above will contain the number of 0x7400 byte pages, and the data from the pages can be appended together in order and treated as a single logical section.

Section Types seen so far in 2004 files include (sections are present in the section map in this order):

Section Name	Description	Compressed?	Page size
	Empty section	Yes	0x7400

AcDb:Security	Contains information regarding password and data encryption. This section is optional.	no	0x7400
AcDb:FileDepList	Contains file dependencies (e.g. IMAGE files, or fonts used by STYLE).	no	0x80
AcDb:VBAProject	Contains VBA Project data for this drawing (optional section)	no	Data size + 0x80 + padding size
AcDb:AppInfo	Contains information about the application that wrote the .dwg file (encrypted = 2).	no	0x80
AcDb:Preview	Bitmap preview for this drawing.	no	0x400
AcDb:SummaryInfo	Contains fields like Title, Subject, Author.	no	0x100
AcDb:RevHistory	Revision history	yes	0x7400
AcDb:AcDbObjects	Database objects	yes	0x7400
AcDb:ObjFreeSpace		yes	0x7400
AcDb:Template	Template (Contains the MEASUREMENT system variable only.)	yes	0x7400
AcDb:Handles	Handle list with offsets into the AcDb:AcDbObjects section	yes	0x7400
AcDb:Classes	Custom classes section	yes	0x7400
AcDb:AuxHeader		yes	0x7400
AcDb:Header	Contains drawing header variables	yes	0x7400
AcDb:Signature	Not written by ODA		

The section order in the stream is different than the order in the section map. The order in the stream is as follows:

Section:
File header
Empty section
AcDb:SummaryInfo

AcDb:Preview
AcDb:VBAProject
AcDb:AppInfo
AcDb:FileDepList
AcDb:RevHistory
AcDb:Security
AcDb:AcDbObjects
AcDb:ObjFreeSpace
AcDb:Template
AcDb:Handles
AcDb:Classes
AcDb:AuxHeader
AcDb:Header
Section map
Section page map

4.6 Encrypted Data Section Page Headers

Data section pages in the file start with a 32 byte encrypted section header. This encrypted header can be decrypted using the following algorithm (assume that the raw section page header data is stored in the hdrData array):

```
OdUInt32 secMask = 0x4164536b ^ offset;
OdUInt32* pHdr = (OdUInt32*)hdrData.asArrayPtr();
For (int j = 0; j < 8; j++)
  *pHdr++ ^= secMask;
```

The decrypted section page header data consists of the following:

Offset	Length	Description
0x00	4	Section page type, since it's always a data section: 0x4163043b
0x04	4	Section number
0x08	4	Data size (compressed)
0x0C	4	Page Size (decompressed)
0x10	4	Start Offset (in the decompressed buffer)
0x14	4	Page header Checksum (section page checksum calculated from unencoded header bytes, with the data checksum as seed)
0x18	4	Data Checksum (section page checksum calculated from compressed data bytes, with seed 0)
0x1C	4	Unknown (ODA writes a 0)

Each section page must start on a 0x20 byte boundary of the raw data stream. The empty bytes between the start of this section and then end of the previous section are filled with as many bytes as needed from the magic number sequence.-

4.7 Compression

The DWG file format version 2004 compression is a variation on the LZ77 compression algorithm. LZ77 is a sliding window algorithm that stores references (offset + length) to previous data. Note that length might be greater than the offset, which is an important feature of this algorithm. The different opcodes are explained below. Compression is a bit more difficult to implement than decompression. The bottleneck with compression is finding a match. The simplest approach would be a brute force approach, the ODA uses hashing for speed, sacrificing some compression.

A compressed section starts with a Literal Length (see below), which indicates the length of the first sequence of uncompressed or literal data.

Following the first literal run, there will be a set of compression opcodes that define 3 values:

compressedBytes – Number of "compressed" bytes that are to be copied to this location from a previous location in the uncompressed data stream.

compOffset – Offset backwards from the current location in the decompressed data stream, where the "compressed" bytes should be copied from.

litCount – Number of uncompressed or literal bytes to be copied from the input stream, following the addition of the compressed bytes.

After copying the specified compressed data and literal bytes, there will be another set of compression opcodes that should be processed in a similar manner.

Each set of compression opcodes starts with a single byte, call it opcode1, which can have the following values:

0x00 – 0x0F : Not used, because this would be mistaken for a Literal Length in some situations.

0x10:

- compressedBytes is read as the next Long Compression Offset (see format below), with 9 added.
- compOffset is read as the next Two Byte Offset (see format below), with 0x3FFF added.
- If the litCount obtained from the Two Byte Offset is 0, then litCount is read as the next Literal Length (see format below). Otherwise use the litCount value from the Two Byte Offset (0-3).

0x11 : Terminates the input stream.

0x12– 0x1F :

- compressedBytes = (opcode1 & 0x0F) + 2
- compOffset is read as the next Two Byte Offset (see format below), with 0x3FFF added.
- If the litCount obtained from the Two Byte Offset is 0, then litCount is read as the next Literal Length (see format below). Otherwise use the litCount value from the Two Byte Offset (0-3).

0x20 :

- compressedBytes is read as the next Long Compression Offset (see format below) + 0x21.
- compOffset is read as the next Two Byte Offset (see format below).
- If the litCount obtained from the Two Byte Offset is 0, then litCount is read as the next Literal Length (see format below). Otherwise use the litCount value from the Two Byte Offset (0-3).

0x21 – 0x3F :

- compressedBytes = opcode1 – 0x1E.
- compOffset is read as the next Two Byte Offset (see format below).

- If the litCount obtained from the Two Byte Offset is 0, then litCount is read as the next Literal Length (see format below). Otherwise use the litCount value from the Two Byte Offset (0-3).

0x40 – 0xFF :

- compressedBytes = ((opcode1 & 0xF0) >> 4) – 1
- Read the next byte (call it opcode2):
- compOffset = (opcode2 << 2) | ((opcode1 & 0x0C) >> 2)
- The value of litCount is set based on the value of (opcode1 & 0x03):
  - 0x00 : litCount is read as the next Literal Length (see format below)
  - 0x01 : litCount = 1
  - 0x02 : litCount = 2
  - 0x03 : litCount = 3

Literal Length

0x01 – 0x0E : 4 – 0x12 (add 3 to the actual value)

0xF0 : any bit set in the high nibble indicates that the literal length is 0, and this byte is actually the next compression opcode (opcode1).

0x00 : Set the running total to 0x0F, and read the next byte. From this point on, a 0x00 byte adds 0xFF to the running total, and a non-zero byte adds that value to the running total and terminates the process. Add 3 to the final result.

Examples:
0x05 : 0x08
0x00 0x02 : 0x14 (0x0F + 2 + 3)

Two Byte Offset

firstByte = readByte()
offset = (firstByte >> 2) | (readByte() << 6))
litCount = (firstByte & 0x03)

Long Compression Offset

0x01 – 0xFF : Use this value as is.

0x00 : Set the running total to 0xFF, and read the next byte. For each 0x00 byte read, add 0xFF to the running total. When a non-zero byte is encountered, add this value to the running total, and terminate the process.

Examples:
0xDD : 0xDD
0x00 0x00 0x34 : 0x232 (0xFF + 0xFF + 0x34)

5 R2007 DWG FILE FORMAT ORGANIZATION

5.1 Sections and pages overview

Like the R18 format the R21 format has sections and pages. There are system sections and data sections. The system sections contain information about where the data sections and their pages are in the stream. A system section only has a single page, while a data section can have multiple pages. The page map contains information about where each data page is in the file stream. The section map has information about which pages belong to which section. The file header, which is at the beginning of the file, just after the meta data, contains the stream locations of the page map and section map.

The following table shows the section and page order in the stream:

Section/page	Size	Description
Meta data	0x80	Meta data (version info etc)
File header	0x400	File header, contains page/section map addresses, sizes, CRC's etc.
Page map 1	0x400 or more	The data page map
Page map 2	0x400 or more	A copy of the data page map
AcDb:SummaryInfo		
AcDb:Preview		
AcDb:VBAProject		
AcDb:AppInfo		
AcDb:FileDepList		
AcDb:RevHistory		
AcDb:Security		
AcDb:AcDbObjects		

AcDb:ObjFreeSpace		
AcDb:Template		
AcDb:Handles		
AcDb:Classes		
AcDb:AuxHeader		
AcDb:Header		
Section map 1		The section map.
Section map 2		A copy of the section map.

5.2 R2007 Meta Data

Address	Length	Description
0x00	6	"AC1021" version string
0x06	5	5 bytes of 0x00
0x0B	1	Maintenance release version
0x0C	1	Byte 0x00, 0x01, or 0x03
0x0D	4	Preview address (long)
0x11	1	Dwg version (Acad version that writes the file)
0x12	1	Maintenance release version (Acad maintenance version that writes the file)
0x13	2	Codepage
0x15	3	Unknown (ODA writes zeroes)
0x18	4	SecurityType (long), see R2004 meta data, the definition is the same, paragraph 4.1.
0x1C	4	Unknown long

0x20	4	Summary info Address in stream
0x24	4	VBA Project Addr (0 if not present)
0x28	4	0x00000080
0x2C	4	Application Info Addr

At offset 0x80 there is a 0x400 byte section. The last 0x28 bytes of this section consists of check data, containing 5 Int64 values representing CRC's and related numbers (starting from 0x3D8 until the end). The first 0x3D8 bytes should be decoded using Reed-Solomon (255, 239) decoding, with a factor of 3. The format of this decoded data is:

Address	Length	Description
0x00	8	CRC
0x08	8	Unknown key
0x10	8	Compressed Data CRC
0x18	4	ComprLen
0x1C	4	Length2
0x20	ComprLen	Compressed Data

Note that if ComprLen is negative, then Data is not compressed (and data length is ComprLen). If ComprLen is positive, the ComprLen bytes of data are compressed, and should be decompressed using the OdDwgR21Compressor::decompress() function, where the decompressed size is a fixed 0x110. The decompressed data is in the following format:

Address	Length	Description
0x00	8	Header size (normally 0x70)
0x08	8	File size
0x10	8	PagesMapCrcCompressed
0x18	8	PagesMapCorrectionFactor
0x20	8	PagesMapCrcSeed

0x28	8	Pages map2offset (relative to data page map 1, add 0x480 to get stream position)
0x30	8	Pages map2Id
0x38	8	PagesMapOffset (relative to data page map 1, add 0x480 to get stream position)
0x40	8	PagesMapId
0x48	8	Header2offset (relative to page map 1 address, add 0x480 to get stream position)
0x50	8	PagesMapSizeCompressed
0x58	8	PagesMapSizeUncompressed
0x60	8	PagesAmount
0x68	8	PagesMaxId
0x70	8	Unknown (normally 0x20)
0x78	8	Unknown (normally 0x40)
0x80	8	PagesMapCrcUncompressed
0x88	8	Unknown (normally 0xf800)
0x90	8	Unknown (normally 4)
0x98	8	Unknown (normally 1)
0xA0	8	SectionsAmount (number of sections + 1)
0xA8	8	SectionsMapCrcUncompressed
0xB0	8	SectionsMapSizeCompressed
0xB8	8	SectionsMap2Id
0xC0	8	SectionsMapId
0xC8	8	SectionsMapSizeUncompressed
0xD0	8	SectionsMapCrcCompressed
0xD8	8	SectionsMapCorrectionFactor

0xE0	8	SectionsMapCrcSeed
0xE8	8	StreamVersion (normally 0x60100)
0xF0	8	CrcSeed
0xF8	8	CrcSeedEncoded
0x100	8	RandomSeed
0x108	8	Header CRC64

This section will be referred to as the **File Header** throughout the remainder of this document.

The page map is stored in a single system section page. The page size of this system section page depends on how much data is stored in it. One page should be able to fit ((dataSectionPageCount + 5) * 16) bytes. PagesMapOffset indicates the starting address of the Page Map section of the file, PagesMapSizeCompressed is the compressed size of this section, PagesMapSizeUncompressed is the uncompressed size, PagesMapCorrectionFactor is the correction factor used, and PagesMapCrcCompressed and PagesMapCrcUncompressed are the compressed and uncomressed CRC values, respectively. The data at PagesMapOffset is in the following format (to be referred to as “System Page” format throughout the remainder of this document) should be decoded and optionally decompressed using the OdDwgR21FileController::loadSysPage function. The resulting pages map data consists of a sequence of pairs, where each pair consists of an Int64 **SIZE** value, and an Int64 **ID** value. This sequence creates a set of pages where each. These values create a pages map using the following algorithm:

```
OdInt64 offset = 0;

while (!pStream->isEof())
{
    size = OdPlatformStreamer::rdInt64(*pStream);
    id = OdPlatformStreamer::rdInt64(*pStream);
    ind = id > 0 ? id : -id;

    m_pages[ind].m_id = id;
    m_pages[ind].m_size = size;
    m_pages[ind].m_offset = offset;
    offset += size;
}
```

The **File Header** value PagesMaxId indicates the largest index that will be used for the m_pages array.

Next, the Section Map should be loaded. The offset of the section map data is the m_offset value of the page with index SectionsMapId in the Page Map of the file. The File Header values SectionsMapSizeCompressed, SectionsMapSizeUncompressed, SectionsMapCrcCompressed, SectionsMapCrcUncompressed, and SectionsMapCorrectionFactor make of the remainder of the arguments to pass to the OdDwgR21FileController::loadSysPage function (see paragraph 5.3) for decoding and decompression of the Section Map data. The decoded and decompressed Section Map data consists of the following attributes for each section in the file:

Address	Length	Description
0x00	8	Data size
0x08	8	Max size
0x10	8	Encryption
0x18	8	HashCode
0x20	8	SectionNameLength
0x28	8	Unknown
0x30	8	Encoding
0x38	8	NumPages. This is the number of pages present in the file for the section, but this does not include pages that contain zeroes only. A page that contains zeroes only is not written to file. If a page's data offset is smaller than the sum of the decompressed size of all previous pages, then it is to be preceded by a zero page with a size that is equal to the difference between these two numbers.
0x40	SectionNameLength x 2 [+ 2]	Unicode Section Name (2 bytes per character, followed by 2 zero bytes if name length > 0)
Repeat NumPages times:		
	8	Page data offset. If a page's data offset is smaller than the sum of the decompressed size of all previous pages, then it is to be preceded by a zero page with a size that is equal to the difference between these two numbers.
	8	Page Size
	8	Page ID
	8	Page Uncompressed Size
	8	Page Compressed Size

	8	Page Checksum
	8	Page CRC

This data repeats until the decoded & decompressed Section Map data is exhausted, giving a set of Sections, where each section can contain data for an arbitrary number of pages. The data from all pages together forms a section in the file. Each page may be optionally RS encoded, compressed, or encrypted. The OdR21PagedStream class implements RS decoding, decompression, and decryption of the page data within a section (see OdDwgR21FileSection::read() for sample code to set up an OdR21PagedStream object).

The section map may contain the following sections (in this order, the order in the file stream is different):

Section Name	Description	Property	Value
AcDb:Security	Contains information regarding password and data encryption. This section is optional.	hashcode	0x4a0204ea
		pagesize	0xf800
		encryption	0
		encoding	1
AcDb:FileDepList	Contains file dependencies (e.g. IMAGE files, or fonts used by STYLE).	hashcode	0x6c4205ca
		pagesize	If no entries, 0x100, otherwise 0x80 * (countEntries + (countEntries >> 1))
		encryption	2
		encoding	1
AcDb:VBAProject	Contains VBA Project data for this drawing (optional section)	hashcode	0x586e0544
		pagesize	VBA data size + 0x80 rounded to the next 0x20.
		encryption	2
		encoding	1
AcDb:AppInfo	Contains information about the application	hashcode	0x3fa0043e

	that wrote the .dwg file (encrypted = 2).	pagesize	0x300
		encryption	0
		encoding	1
AcDb:Preview	Bitmap preview for this drawing.	hashcode	0x40aa0473
		pagesize	Default 0x400, if image is written, preview size rounded to the next 0x20 bytes.
		encryption	1 if properties are encrypted, 0 otherwise
		encoding	1
AcDb:SummaryInfo	Contains fields like Title, Subject, Author.	hashcode	0x717a060f
		pagesize	0x80
		encryption	1 if properties are encrypted, 0 otherwise
		encoding	1
AcDb:RevHistory	Revision history	hashcode	0x60a205b3
		pagesize	0x1000
		encryption	0
		encoding	4
		compressed	true
AcDb:AcDbObjects	Database objects	hashcode	0x674c05a9
		pagesize	0xf800
		encryption	1 if data is encrypted, 0 otherwise

		encoding	4
		compressed	true
AcDb:ObjFreeSpace		hashcode	0x77e2061f
		pagesize	0xf800
		encryption	0
		encoding	4
		compressed	true
AcDb:Template	Template (Contains the MEASUREMENT system variable only.)	hashcode	0x4a1404ce
		pagesize	0x400
		encryption	0
		encoding	4
		compressed	true
AcDb:Handles	Handle list with offsets into the AcDb:AcDbObjects section	hashcode	0x3f6e0450
		pagesize	0xf800
		encryption	1 if data is encrypted, 0 otherwise
		encoding	4
		compressed	true
AcDb:Classes	Custom classes section	hashcode	0x3f54045f
		pagesize	0xf800
		encryption	1 if data is encrypted, 0 otherwise
		encoding	4
		compressed	true

AcDb:AuxHeader		hashcode	0x54f0050a
		pagesize	0x800
		encryption	0
		encoding	4
		compressed	true
AcDb:Header	Contains drawing header variables	hashcode	0x32b803d9
		pagesize	0x800
		encryption	1 if data is encrypted, 0 otherwise
		encoding	4
		compressed	True
AcDb:Signature	Not written by ODA		

By default data/properties are not encrypted. Encryption still needs to be described.

5.2.1 File header creation

Creating the R21 file header is very complex:

Compute and set all the file header fields. In this process also compute CRC's and generate check data, derived from a CRC seed value (paragraph 5.2.1.1).

Write the file header data to a buffer and calculate/write the 64-bit CRC (paragraph 5.2.1.2).

Compress the file header data and calculate the 64-bit CRC (paragraph 5.2.1.3).

Create a checking sequence and calculate a CRC over this sequence data (paragraph 5.2.1.4).

Create a buffer in preparation of Reed-Solomon encoding (Pre-Reed-Solomon encoded data). This contains checking sequence, compressed CRC, compressed size, compressed data and random data (as padding) (paragraph 5.2.1.5).

Encode the data using Reed-Solomon (for error correction).

Write the encoded data, followed by the check data from the first step.

5.2.1.1 Calculating the file header CRC's and check data

The file header data consists of regular data fields and CRC values and check data to verify the data's correctness. All fields pertaining to the file header's correctness are discussed in more detail in the following paragraphs. Note that the order of CRC calculation is important, so the order of the following paragraphs should be used.

5.2.1.1.1 RandomSeed

Is filled with the CRC random encoding's seed (see paragraph 5.11).

5.2.1.1.2 CrcSeed

The ODA always initializes this with value 0.

5.2.1.1.3 SectionsMapCrcSeed

Is filled with crcSeed initially. Then it's encoded using the CRC random encoding as described in paragraph 5.11.

5.2.1.1.4 PagesMapCrcSeed

Is filled with crcSeed initially. Then it's encoded using the CRC random encoding as described in paragraph 5.11.

5.2.1.1.5 Check data

The check data for the file header page is present at the end of the header page at location 0x3d8. It contains data generated based on the CrcSeed and the current state of the CRC random encoder. The check data contains the following UInt64 fields (computed in this order):

Random value 1 (third value in stream)

Random value 2 (fourth value in stream)

Encoded CRC Seed (fifth value in stream)

Normal 64-bit CRC (first value in stream)

Mirrored 64-bit CRC (second value in stream)

Random value 1 is set to the CRC random encoder's next random value.

Random value 2 is set to the CRC random encoder's next random value.

The Encoded CRC seed is gotten by letting the CRC random encoder encode the CRC seed.

The normal 64-bit CRC value is calculated as follows. A buffer of 8 UInt64 values is created and initialized with zeroes. The values are encoded using the Encode function below:

```
UInt64 Encode(UInt64 value, UInt64 control) {
    Int32 shift = (Int32)(control & 0x1f);
    if (shift != 0) {
        value = (value << shift) | (value >> (64 - shift));
    }
    return value;
}
```

The buffer is initialized by encoding several values. Later this buffer becomes the input to a normal 64-bit CRC calculation:

```
UInt64 CalculateNormalCrc() {
    UInt64[] buffer = new UInt64[8];
    buffer[0] = Encode(random1, random2);
    buffer[1] = Encode(buffer[0], buffer[0]);
    buffer[2] = Encode(random2, buffer[1]);
    buffer[3] = Encode(buff[2], buffer[2]);
    buffer[4] = Encode(random1, buffer[3]);
    buffer[5] = Encode(buffer[4], buffer[4]);
    buffer[6] = Encode(buffer[5], buffer[5]);
    buffer[7] = Encode(buffer[6], buffer[6]);

    // Convert each UInt64 in the buffer from big-endian to little-endian if
    // the machine is big-endian.
    ...

    UInt64 normalCrc = CalculateNormalCrc64(buffer, 64, ~random2);
    return normalCrc;
}
```

Similarly the mirrored CRC value is calculated:

```
UInt64 CalculateMirroredCrc() {
    UInt64[] buffer = new UInt64[8];
    buffer[0] = Encode(random1, random2);
    buffer[1] = Encode(normalCrc, buffer[0]);
    buffer[2] = Encode(random2, buffer[1]);
    buffer[3] = Encode(normalCrc, buffer[2]);
    buffer[4] = Encode(random1, buffer[3]);
    buffer[5] = Encode(normalCrc, buffer[4]);
    buffer[6] = Encode(random2, buffer[5]);
    buffer[7] = Encode(buffer[6], buffer[6]);

    // Convert each UInt64 in the buffer from big-endian to little-endian if
    // the machine is big-endian.
    ...

    UInt64 mirroredCrc = CalculateMirroredCrc64(buffer, 64, ~random1);
    return mirroredCrc;
}
```

5.2.1.1.6 CrcSeedEncoded

Encoded value of CrcSeed, using the CRC random encoding as described in paragraph 5.11.

5.2.1.2 Calculate file header data 64-bit CRC (decompressed)

The last field in the file header is a normal 64-bit CRC (see paragraph 5.12) which is the CRC calculated from the file header data, including the 64-bit CRC with value zero. The CRC seed value is 0, and then updated with method UpdateSeed2 before calling UpdateCrc (see again paragraph 5.12). The initial CRC value of 0 is replaced with the calculated value.

5.2.1.3 Compress and calculate 64-bit CRC (compressed)

The file header data is compressed. If the compressed data is not shorter than the uncompressed data, then the uncompressed data itself is used. Another normal 64-bit CRC value is calculated from the resulting data (see paragraph 5.12).

5.2.1.4 Create checking sequence and 64-bit CRC

Another checking sequence of 2 UInt64 values is created, very similar to the check data in paragraph 5.2.1.1.5. The first value is filled with the next value from the random encoder (see paragraph 5.11). The second value is calculated using the check data’s Encode function, with the first sequence value passed as first (value) and second (control) parameter. The sequence bytes are then converted to little endian format. The last step is calculating a normal 64-bit CRC value (see paragraph 5.12). The CRC seed value is 0, updated by method UpdateSeed1.

5.2.1.5 Create a buffer in preparation of Reed-Solomon encoding

In preparation of the next step, which is Reed-Solomon (RS) encoding, a buffer is created which is going to be encoded. The size of this buffer is 3 x 239 bytes (239 is the RS data size for a block (k) used for system pages, see paragraph 5.13). First a block is created, of which the size is a multiple of 8 bytes:

Position	Size	Description
0	8	Checking sequence CRC (paragraph 5.2.1.4)
8	8	Checking sequence first UInt64 value (paragraph 5.2.1.4)
16	8	Compressed data CRC (paragraph 5.2.1.3)
24	8	Compressed data size. In case the compressed data size is larger than the uncompressed data size, then the negated uncompressed data size is written.
32	n	Compressed data in case the size is smaller than the uncompressed data size. Otherwise the uncompressed data.
32 + n	m	Padding so the block size is a multiple of 8 bytes. The padding bytes are gotten from the CRC random encoding, see paragraph 5.11.

This block is repeated as many times as possible within the buffer. The remaining bytes are filled using random padding data from the CRC random encoding (see paragraph 5.11).

5.2.1.6 Encode the data using Reed-Solomon

In this step the header data is encoded using the Reed-Solomon (RS) encoding for interleaved system pages (see paragraph 5.13). The encoded size is 3 x 255 bytes. The remaining bytes of the page (of total size 0x400) are filled using random padding data from the CRC random encoding (see paragraph 5.11).

5.2.1.7 Add check data at the end of the page

The last 0x20 bytes of the page should be overwritten using the check data, calculated in paragraph 5.2.1.1.5. The page size remains 0x400 bytes.

5.2.1.8 Write the file header to the file stream

The file header is written to position 0x80 and to the end of the file stream.

5.3 System section page

The system section page is used by the data section map and the section page map.

Inputs for writing a system section page are:

- The data.
- The 64-bit CRC seed.

- The page size (minimum 0x400). The page size is determined from the decompressed data size as described in paragraph 5.3.1.

Outputs are:

- Compressed and Reed-Solomon (RS) encoded data.
- Derived properties of the (compressed/encoded) data: compressed 64-bit CRC, decompressed 64-bit CRC, data repeat count (or data factor). These derived properties are written in the file header (see paragraph 5.2).

First the 64-bit CRC of the decompressed data is calculated, using the mirrored 64-bit CRC calculation (see paragraph 5.12). This uses theUpdateSeed1 method to update the CRC seed before entering the CRC computation.

Next step is compression. If the compressed data isn't shorter than the original data, then the original data is used instead of the compressed data.

Of the resulting data (either compressed or not), another 64-bit CRC is computed (similarly to described above).

The resulting data is padded with zeroes so the length is a multiple of the CRC block size (8).

Now the resulting data is repeated as many times as possible within the page, RS encoded (see paragraph 5.13) and padded. The maximum RS block count (integer) is the page size divided by the RS codeword size (255). The maximum RS pre-encoded size is the maximum RS block count times the k-value of the RS system page encoding (239). So the data repeat count is the maximum RS pre-encoded size divided by the resulting (padded) data length. Next a buffer is created, with the resulting (padded) data repeated (data repeat count times). This buffer is encoded using RS encoding for system pages, interleaved. Note that the actual RS block count is less than or equal to the maximum RS block count calculated above. The encoded size is the RS block count times 255. The final step is to add padding using random data from the random encoding to fill the remainder of the page, see paragraph 5.11.

5.3.1 System section page size calculation

The data stored in a system section is first padded until its size is a multiple of the CRC block size (8). This is called the aligned size. The Reed-Solomon encoded aligned data should fit the system section at least two times. The minimimum page size is 0x400 bytes.

The system section page size can be calculated from the uncompressed data size in bytes as shown in the following pseudo code (function GetSystemPageSize):

```
const Int32 CrcBlockSize = 8;
const Int32 PageAlignSize = 0x20;
const Int32 ReedSolomonDataBlockSize = 239;
const Int32 ReedSolomonCodewordSize = 255;
```

```
public static UInt64 GetAlignedPageSize(UInt64 pageSize) {
    UInt64 result = (UInt64)((Int64)(pageSize + PageAlignSize - 1) & (Int64)(-PageAlignSize));
    return result;
}

public static UInt64 GetSystemPageSize(UInt64 dataSize) {
    UInt64 alignedSize = (UInt64)((Int64)(dataSize + CrcBlockSize - 1) & (Int64)(-CrcBlockSize));
    // The page should fit the data at least 2 times.
    UInt64 filePageSize = ((alignedSize * 2) + ReedSolomonDataBlockSize - 1) /
        ReedSolomonDataBlockSize * ReedSolomonCodewordSize;
    if (filePageSize < 0x400) {
        filePageSize = 0x400;
    } else {
        filePageSize = GetAlignedPageSize(filePageSize);
    }
    return filePageSize;
}
```

5.4 Data section page

Data sections are used for all sections except the data section map and the section page map. The section's data is partitioned into pages, each of Max size length, except for the last page which may be of size less than Max size. The following steps are taken when writing data page.

First a 32-bit data checksum of the page's data is calculated. The pseudo code for this calculation is presented in paragraph 5.4.1.

Next the page data is optionally compressed (depending on the section). If the compressed data isn't shorter than the original data, then this page's data is not compressed.

If the file is encrypted, the page is encrypted (to be described).

The page's 64-bit CRC is calculated (mirrored CRC, see paragraph 5.12). The page CRC seed is the file's CRC seed updated using UpdateSeed1 (see again paragraph 5.12).

Pad the data with zero bytes so the size becomes a multiple of the CRC block size (0x8).

The data is Reed-Solomon encoded (see paragraph 5.13). Depending on the section encoding, the data is either interleaved (value 4) or not (value 1).

The page start position should be aligned on a 0x20 byte boundary (if all is well nothing has to be done at this point to achieve this). The data is written and padded with zero bytes so the stream position is again at a 0x20 byte boundary.

Finally the current page ID is incremented.

5.4.1 Data section page checksum

The function below shows how to calculate the 32-bit data page checksum:

```
UInt32 GetCheckSum(UInt64 seed, byte[] data, UInt32 start, UInt32 length) {
    seed = (seed + length) * 0x343fd + 0x269ec3;
    UInt32 sum1 = (UInt32) (seed & 0xffff);
    UInt32 sum2 = (UInt32) ((seed >> 0x10) & 0xffff);

    fixed (byte* dataStartPtr = data) {
        byte* dataPtr = dataStartPtr + start;
        while (length != 0) {
            UInt32 bigChunkLength = System.Math.Min(0x15b0, length);
            length -= bigChunkLength;

            // Process small chunks of 8 bytes each.
            UInt32 smallChunkCount = bigChunkLength >> 3;
            while (smallChunkCount-- > 0) {
                UpdateSums2Bytes(dataPtr + 6, sum1, sum2);
                UpdateSums2Bytes(dataPtr + 4, sum1, sum2);
                UpdateSums2Bytes(dataPtr + 2, sum1, sum2);
                UpdateSums2Bytes(dataPtr + 0, sum1, sum2);
                dataPtr += 8;
            }

            // Processing remaining 0..7 bytes.
            UInt32 smallChunkRemaining = bigChunkLength & 7;
            if (smallChunkRemaining > 0) {
                switch (smallChunkRemaining) {
                    case 1:
                        UpdateSums1Byte(dataPtr + 0, sum1, sum2);
                        break;
                    case 2:
                        UpdateSums2Bytes(dataPtr + 0, sum1, sum2);
                        break;
                    case 3:
                        UpdateSums2Bytes(dataPtr + 0, sum1, sum2);
                        UpdateSums1Byte(dataPtr + 2, sum1, sum2);
                        break;
                    case 4:
                        UpdateSums2Bytes(dataPtr + 2, sum1, sum2);
                        UpdateSums2Bytes(dataPtr + 0, sum1, sum2);
                        break;
                    case 5:
                        UpdateSums2Bytes(dataPtr + 2, sum1, sum2);
                        UpdateSums2Bytes(dataPtr + 0, sum1, sum2);
                        UpdateSums1Byte(dataPtr + 4, sum1, sum2);
                        break;
                    case 6:
                        UpdateSums2Bytes(dataPtr + 2, sum1, sum2);
                        UpdateSums2Bytes(dataPtr + 0, sum1, sum2);
                        UpdateSums2Bytes(dataPtr + 4, sum1, sum2);
                        break;
                    case 7:
                        UpdateSums2Bytes(dataPtr + 2, sum1, sum2);
                        UpdateSums2Bytes(dataPtr + 0, sum1, sum2);
                        UpdateSums2Bytes(dataPtr + 4, sum1, sum2);
                        UpdateSums1Byte(dataPtr + 6, sum1, sum2);
                        break;
                }
                dataPtr += smallChunkRemaining;
            }

            sum1 %= 0xfff1;
            sum2 %= 0xfff1;
        }
    }

    return (sum2 << 0x10) | (sum1 & 0xffff);
}

private static unsafe void UpdateSums1Byte(byte* p, UInt32& sum1, UInt32& sum2) {
    sum1 += *p;
    sum2 += sum1;
}

private static unsafe void UpdateSums2Bytes(byte* p, UInt32& sum1, UInt32& sum2) {
    UpdateSums1Byte(p, sum1, sum2);
    UpdateSums1Byte(p + 1, sum1, sum2);
}
```

5.5 AcDb:Security Section

The AcDb:Security section is optional in the file—it is present if the file was saved with a password. The data in this section is in the same format as in the R2004 format, 2 unknown 32-bit integers, a 32-bit integer with value 0xABCDABCD, etc.

5.6 AcDb:AuxHeader Section

This section is in the same format as in R2004. See details in chapter 27.

5.7 AcDb:Handles Section

This section is in the same format as in R2004.

5.8 AcDb:Classes Section

This section contains the defined classes for the drawing. It contains a new string stream for unicode string—see the Objects Section for a description of how to extract the string stream from an object.

```
SN  :  0x8D 0xA1 0xC4 0xB8 0xC4 0xA9 0xF8 0xC5 0xC0 0xDC 0xF4 0x5F 0xE7 0xCF 0xB6 0x8A.

RL  :  size of class data area in bytes

RL  :  total size in bits

BL  :  Maxiumum class number

 B  :  bool value

    :  Class Data (format described below)

 X  :  String stream data

 B  :  bool value (true if string stream data is present).
```

Class data (repeating):

```
BS  :  classnum

BS  :  proxy flags:

       Erase allowed = 1,
       transform allowed = 2,
       color change allowed = 4,
       layer change allowed = 8,
       line type change allowed = 16,
       line type scale change allowed = 32,
       visibility change allowed = 64,
       cloning allowed = 128,
       Lineweight change allowed = 256,
       Plot Style Name change allowed = 512,
       Disables proxy warning dialog = 1024,
       is R13 format proxy= 32768

TU  :  appname

TU  :  cplusplusclassname

TU  :  classdxfname

 B  :  wasazombie

BS  :  itemclassid -- 0x1F2 for classes which produce entities, 0x1F3 for classes which
       produce objects.
```

```
        BL  :  Number of objects created of this type in the current DB (DXF 91).
        BL  :  Dwg Version
        BL  :  Maintenance release version.
        BL  :  Unknown
        BL  :  Unknown (normally 0L)
```

We read sets of these until we exhaust the data.

5.9 AcDb:Header Section

This section contains the "DWG Header Variables" data in a similar format as R15 files (see details in the DWG HEADER VARIABLES section of this document), except that string data is separated out into a string stream. See the Objects Section for details about string stream location within an object. Also, the handles are separated out into a separate stream at the end of the header, in the same manner as is done for Objects.

5.10 Decompression

The compression uses another variant of the LZ77 algorithm, different from the one used in R18. Like the R18 compression, the compressed stream (source buffer) contains opcodes, offsets and lengths of byte chunks to be copied from either compressed or decompressed buffer.

An opcode consists of a single byte. The first byte contains the first opcode. If the first opcode's high nibble equals a 2, then:

- the source buffer pointer is advanced 2 bytes, and a *length* is read from the next byte, bitwise and-ed with 0x07
- the pointer is advanced another byte (3 bytes in total).

Next the decompression enters a loop. A byte chunk from the compressed stream is followed by one or more byte chunks from the decompressed stream. The last chunk may be a compressed chunk.

5.10.1 Copying a compressed chunk

If the *length* was zero, it is read from the source buffer. The following pseudo function reads the length:

```
UInt32 ReadLiteralLength(byte[] buffer) {
    UInt32 length = opCode + 8;
    if (length == 0x17) {
        UInt32 n = buffer[sourceIndex++];
        length += n;
        if (n == 0xff) {
            do {
                n = buffer[sourceIndex++];
                n |= (Uint32)(buffer[sourceIndex++] << 8);
                length += n;
            } while (n == 0xffff);
        }
    }
}
```

Next *length* bytes are copied from the source buffer and the source buffer pointer is advanced to one after the copied bytes. The order of bytes in source and target buffer are different. The copying happens in chunks of 32 bytes, and the remainder is copied using a specific copy function for each number of bytes (so 31 separate copy functions). For copying 1-32 bytes, a combination of sub byte blocks is made, according to the following table (the smallest block is 1 byte):

Byte count	Byte count [source array index]
1	1 [0]
2	1 [1], 1[0]
3	1 [2], 1[1], 1[0]
4	1 [0], 1 [1], 1 [2], 1 [3]
5	1 [4], 4 [0]
6	1 [5], 4 [1], 1 [0]
7	2 [5], 4 [1], 1 [0]
8	4 [0], 4[4]
9	1 [8], 8 [0]
10	1 [9], 8 [1], 1 [0]
11	2 [9], 8 [1], 1 [0]
12	4 [8], 8 [0]
13	1 [12], 4 [8], 8 [0]
14	1 [13], 4 [9], 8 [1], 1[0]

15	2 [13], 4 [9], 8 [1], 1[0]
16	8 [8], 8 [0]
17	8 [9], 1 [8], 8 [0]
18	1 [17], 16 [1], 1 [0]
19	3 [16], 16 [0]
20	4 [16], 16 [0]
21	1 [20], 4 [16], 16 [0]
22	2 [20], 4 [16], 16 [0]
23	3 [20], 4 [16], 16 [0]
24	8 [16], 16 [0]
25	8 [17], 1 [16], 16 [0]
26	1 [25], 8 [17], 1 [16], 16 [0]
27	2 [25], 8 [17], 1 [16], 16 [0]
28	4 [24], 8 [16], 16 [0]
29	1 [28], 4 [24], 8 [16], 16 [0]
30	2 [28], 4 [24], 8 [16], 16 [0]
31	1 [30], 4 [26], 8 [18], 16 [2], 2 [0]
32	16 [16], 16 [0]

To copy any number of bytes, first blocks of 32 bytes are copied. The remainder (1 – 31) is copied using one of the other 31 byte block copy functions as outlined in the table above.

5.10.2 Copying decompressed chunks

After copying a compressed chunk, one or more decompressed chunks are copied (unless the compressed chunk was the last chunk). First an opcode byte is read. Depending on the opcode the source buffer offset, length and next opcode are read.

```
private void ReadInstructions(
    byte[] srcBuf,
    UInt32 srcIndex,
    ref byte opCode,
    out UInt32 sourceOffset,
    out UInt32 length
) {
    switch ((opCode >> 4)) {
        case 0:
            length = (opCode & 0xf) + 0x13;
            sourceOffset = srcBuf[srcIndex++];
            opCode = srcBuf[srcIndex++];
            length = ((opCode >> 3) & 0x10) + length;
            sourceOffset = ((opCode & 0x78) << 5) + 1 + sourceOffset;
            break;

        case 1:
            length = (opCode & 0xf) + 3;
            sourceOffset = srcBuf[srcIndex++];
            opCode = srcBuf[srcIndex++];
            sourceOffset = ((opCode & 0xf8) << 5) + 1 + sourceOffset;
            break;

        case 2:
            sourceOffset = srcBuf[srcIndex++];
            sourceOffset = ((srcBuf[srcIndex++] << 8) & 0xff00) | sourceOffset;
            length = opCode & 7;
            if ((opCode & 8) == 0) {
                opCode = srcBuf[srcIndex++];
                length = (opCode & 0xf8) + length;
            } else {
                sourceOffset++;
                length = (srcBuf[srcIndex++] << 3) + length;
                opCode = srcBuf[srcIndex++];
                length = (((opCode & 0xf8) << 8) + length) + 0x100;
            }
            break;

        default:
            length = opCode >> 4;
            sourceOffset = opCode & 15;
            opCode = srcBuf[srcIndex++];
            sourceOffset = (((opCode & 0xf8) << 1) + sourceOffset) + 1;
            break;
    }
}
```

Below is the pseudocode for the decompressed chunk copy loop:

```
private UInt32 CopyDecompressedChunks(
    byte[] srcBuf,
    UInt32 srcIndex,
    UInt32 compressedEndIndexPlusOne,
    byte[] dstBuf,
    UInt32 outputIndex
) {
    UInt32 length = 0;
    byte opCode = srcBuf[inputIndex++];
    inputIndex++;
    UInt32 sourceOffset;
    ReadInstructions(srcBuf, srcIndex, ref opCode, out sourceOffset, out length);
    while (true) {
        CopyBytes(dstBuf, outputIndex, length, sourceOffset);
        outputIndex += length;
        length = opCode & 7;
        if ((length != 0) || (inputIndex >= compressedEndIndexPlusOne)) {
            break;
        }
        opCode = srcBuf[inputIndex];
        inputIndex++;
        if ((opCode >> 4) == 0) {
            break;
        }
        if ((opCode >> 4) == 15) {
            opCode &= 15;
        }
        ReadInstructions(srcBuf, srcIndex, ref opCode, out sourceOffset, out length);
    }
    return outputIndex;
}
```

5.11 CRC random encoding

Some CRC values are encoded by taking 10 bits from an UInt16 and adding bits from a pseudo random encoding table to form a UInt64 result. The decoding does the opposite, it takes an encoded UInt64, and extracts the 10 data bits from it and returns the result in an UInt16. The pseudo random encoding table holds 0x270 UInt32 values and is generated from a single UInt64 seed value. An index points to an entry into this table. As values are encoded, this index loops through this table. When the counter reaches 0x270 it is reset to 0 again.

From the encoding table a padding data table is calculated. This padding data is used to add padding bytes until the proper byte alignment is achieved in the main file stream. The byte order of the padding bytes in memory has to be little endian, so the encoding table is also stored in little endian format. Whenever retrieving data as a UInt32 from the encoding table, the original endianness of the bytes must restored (depending on the machine the 4 bytes are thus reversed or not).

The following pseudocode shows how to generate the pseudo random encoding table and the padding table:

```
public void Init(UInt64 seed) {
    encodingTable = new UInt32[0x270];
    this.seed = seed;
    index = 0;

    encodingTable[0] = ((UInt32) seed * 0x343fd) + 0x269ec3;
    encodingTable[1] = ((UInt32) (seed >> 32) * 0x343fd) + 0x269ec3;
    UInt32 value = encodingTable[1];
    encodingTable[0] = PlatformUtil.ToLittleEndian(encodingTable[0]);
    encodingTable[1] = PlatformUtil.ToLittleEndian(encodingTable[1]);
    for (UInt32 i = 2; i < 0x270; i++) {
        value = (((value >> 0x1e) ^ value) * 0x6c078965) + i;
        encodingTable[i] = PlatformUtil.ToLittleEndian(value);
    }

    InitPadding();
}

private void InitPadding() {
    padding = new UInt32[0x80];
    for (int i = 0; i < 0x80; i++) {
        UpdateIndex();
        padding[i] = encodingTable[index];
        index++;
    }
}

private void UpdateIndex() {
    if (index >= 0x270) {
        index = 0;
    }
}
```

The encoding method takes a UInt16 argument, and encodes the 10 least significant bits into a UInt64 (spread evenly), and uses values from the encoding table to fill the remaining 54 bits. Below is the pseudocode for encoding:

```
public UInt64 Encode(UInt32 value) {
    UInt64 random = GetNextUInt64();
    UInt32 lo = (UInt32)(random & 0x0df7df7df);
    UInt32 hi = (UInt32)((random >> 32) & 0x0f7df7df7);
    if ((value & 0x200) != 0) {
        lo |= 0x20;
    }
    if ((value & 0x100) != 0) {
        lo |= 0x800;
    }
    if ((value & 0x80) != 0) {
        lo |= 0x20000;
    }
    if ((value & 0x40) != 0) {
        lo |= 0x800000;
    }
    if ((value & 0x20) != 0) {
        lo |= 0x20000000;
    }

    if ((value & 0x10) != 0) {
        hi |= 0x08;
    }
    if ((value & 0x8) != 0) {
        hi |= 0x200;
    }
    if ((value & 0x4) != 0) {
        hi |= 0x8000;
    }
    if ((value & 0x2) != 0) {
        hi |= 0x200000;
    }
    if ((value & 0x1) != 0) {
        hi |= 0x8000000;
    }
    return lo | ((UInt64)hi << 32);
}

public UInt64 GetNextUInt64() {
    index += 2;
    UpdateIndex();

    UInt32 low = PlatformUtil.FromLittleEndian(encodingTable[index]);
    UInt32 hi = PlatformUtil.FromLittleEndian(encodingTable[index + 1]);
```

```
        UInt64 result = low | (hi << 32);
        return result;
    }
```

The decoding does the opposite: it takes an encoded UInt64 and extracts the 10 data bits from it:

```
    public UInt32 Decode(UInt64 value) {
        UInt32 result = 0;

        UInt32 hi = (UInt32)(value >> 32);
        if ((hi & 0x8000000) != 0) {
            result |= 0x01;
        }
        if ((hi & 0x200000) != 0) {
            result |= 0x02;
        }
        if ((hi & 0x8000) != 0) {
            result |= 0x04;
        }
        if ((hi & 0x200) != 0) {
            result |= 0x08;
        }
        if ((hi & 0x8) != 0) {
            result |= 0x10;
        }

        UInt32 lo = (UInt32)value;
        if ((lo & 0x20000000) != 0) {
            result |= 0x20;
        }
        if ((lo & 0x800000) != 0) {
            result |= 0x40;
        }
        if ((lo & 0x20000) != 0) {
            result |= 0x80;
        }
        if ((lo & 0x800) != 0) {
            result |= 0x100;
        }
        if ((lo & 0x20) != 0) {
            result |= 0x200;
        }

        return result;
```

```
}
```

5.12 64-bit CRC calculation

DWG file format version 2007 uses 64-bit CRC values in the file header. The calculation uses a CRC lookup table with 256 64-bit values. There are two flavors of 64-bit CRC's:

normal (see ECMA-182: http://www.ecma-international.org/publications/standards/Ecma-182.htm),

mirrored, where the actual CRC computation is shifting bits the other way around.

Also the CRC tables used are different for these two versions. The way the CRC is computed for an array of bytes is the same for both CRC flavors. Only the way the CRC for a single byte is calculated is different (function CalculateCrcFor1Byte). For byte counts 1-8 the CRC calculation is ordered as follows:

Byte count	Sequence of blocks byte counts [source offset for each block]
1	1 [0]
2	1 [0], 1 [1]
3	2 [0], 1 [2]
4	2 [2], 2 [0]
5	4 [0], 1 [4]
6	4 [0], 2 [4]
7	4 [0], 3 [4]
8	4 [4], 4 [0]

For byte counts greater than 8, first blocks of 8 bytes each are processed. The remainder is processed according to the table above.

At the end of the CRC calculation the CRC value is inverted (bitwise not) for the normal CRC (not the mirrored CRC).

In addition to the CRC calculation itself there are two CRC initialization functions, called before calculating the CRC. Which one is used depends on the context.

```
UInt3264 UpdateSeed1(UInt3264 seed, UInt32 dataLength) {
    seed = (seed + dataLength) * 0x343fdUL + 0x269ec3UL;
    seed |= seed * (0x343fdUL << 32) + (0x269ec3UL << 32);
    seed = ~seed;
    return seed;
}

UInt3264 UpdateSeed2(UInt3264 seed, UInt32 dataLength) {
    seed = (seed + dataLength) * 0x343fdUL + 0x269ec3UL;
    seed = seed * ((1UL << 32) + 0x343fdUL) + (dataLength + 0x269ec3UL);
    seed = ~seed;
    return seed;
}
```

5.12.1 Normal CRC

The CRC for a single byte is calculated as follows:

```
UInt8 CalculateCrcFor1Byte(UInt8 data, UInt64 crc) {
    return crcTable[(data ^ (crc >> 56)) & 0xff] ^ (crc << 8);
}
```

The CRC table is initialized with the following values:

```
0x0000000000000000, 0x42f0e1eba9ea3693, 0x85e1c3d753d46d26, 0xc711223cfa3e5bb5,
0x493366450e42ecdf, 0x0bc387aea7a8da4c, 0xccd2a5925d9681f9, 0x8e224479f47cb76a,
0x9266cc8a1c85d9be, 0xd0962d61b56fef2d, 0x17870f5d4f51b498, 0x5577eeb6e6bb820b,
0xdb55aacf12c73561, 0x99a54b24bb2d03f2, 0x5eb4691841135847, 0x1c4488f3e8f96ed4,
0x663d78ff90e185ef, 0x24cd9914390bb37c, 0xe3dcbb28c335e8c9, 0xa12c5ac36adfde5a,
0x2f0e1eba9ea36930, 0x6dfeff5137495fa3, 0xaaefdd6dcd770416, 0xe81f3c86649d3285,
0xf45bb4758c645c51, 0xb6ab559e258e6ac2, 0x71ba77a2dfb03177, 0x334a9649765a07e4,
0xbd68d2308226b08e, 0xff9833db2bcc861d, 0x388911e7d1f2dda8, 0x7a79f00c7818eb3b,
0xcc7af1ff21c30bde, 0x8e8a101488293d4d, 0x499b3228721766f8, 0x0b6bd3c3dbfd506b,
0x854997ba2f81e701, 0xc7b97651866bd192, 0x00a8546d7c558a27, 0x4258b586d5bfbcb4,
0x5e1c3d753d46d260, 0x1cecdc9e94ace4f3, 0xdbfdfea26e92bf46, 0x990d1f49c77889d5,
0x172f5b3033043ebf, 0x55dfbadb9aee082c, 0x92ce98e760d05399, 0xd03e790cc93a650a,
0xaa478900b1228e31, 0xe8b768eb18c8b8a2, 0x2fa64ad7e2f6e317, 0x6d56ab3c4b1cd584,
0xe374ef45bf6062ee, 0xa1840eae168a547d, 0x66952c92ecb40fc8, 0x2465cd79455e395b,
0x3821458aada7578f, 0x7ad1a461044d611c, 0xbdc0865dfe733aa9, 0xff3067b657990c3a,
0x711223cfa3e5bb50, 0x33e2c2240a0f8dc3, 0xf4f3e018f031d676, 0xb60301f359dbe0e5,
0xda050215ea6c212f, 0x98f5e3fe438617bc, 0x5fe4c1c2b9b84c09, 0x1d14202910527a9a,
0x93366450e42ecdf0, 0xd1c685bb4dc4fb63, 0x16d7a787b7faa0d6, 0x5427466c1e109645,
0x4863ce9ff6e9f891, 0x0a932f745f03ce02, 0xcd820d48a53d95b7, 0x8f72eca30cd7a324,
0x0150a8daf8ab144e, 0x43a04931514122dd, 0x84b16b0dab7f7968, 0xc6418ae602954ffb,
0xbc387aea7a8da4c0, 0xfec89b01d3679253, 0x39d9b93d2959c9e6, 0x7b2958d680b3ff75,
0xf50b1caf74cf481f, 0xb7fbfd44dd257e8c, 0x70eadf78271b2539, 0x321a3e938ef113aa,
0x2e5eb66066087d7e, 0x6cae578bcfe24bed, 0xabbf75b735dc1058, 0xe94f945c9c3626cb,
0x676dd025684a91a1, 0x259d31cec1a0a732, 0xe28c13f23b9efc87, 0xa07cf2199274ca14,
0x167ff3eacbaf2af1, 0x548f120162451c62, 0x939e303d987b47d7, 0xd16ed1d631917144,
0x5f4c95afc5edc62e, 0x1dbc74446c07f0bd, 0xdaad56789639ab08, 0x985db7933fd39d9b,
0x84193f60d72af34f, 0xc6e9de8b7ec0c5dc, 0x01f8fcb784fe9e69, 0x43081d5c2d14a8fa,
0xcd2a5925d9681f90, 0x8fdab8ce70822903, 0x48cb9af28abc72b6, 0x0a3b7b1923564425,
0x70428b155b4eaf1e, 0x32b26afef2a4998d, 0xf5a348c2089ac238, 0xb753a929a170f4ab,
0x3971ed50550c43c1, 0x7b810cbbfce67552, 0xbc902e8706d82ee7, 0xfe60cf6caf321874,
0xe224479f47cb76a0, 0xa0d4a674ee214033, 0x67c58448141f1b86, 0x253565a3bdf52d15,
0xab1721da49899a7f, 0xe9e7c031e063acec, 0x2ef6e20d1a5df759, 0x6c0603e6b3b7c1ca,
0xf6fae5c07d3274cd, 0xb40a042bd4d8425e, 0x731b26172ee619eb, 0x31ebc7fc870c2f78,
0xbfc9838573709812, 0xfd39626eda9aae81, 0x3a28405220a4f534, 0x78d8a1b9894ec3a7,
0x649c294a61b7ad73, 0x266cc8a1c85d9be0, 0xe17dea9d3263c055, 0xa38d0b769b89f6c6,
0x2daf4f0f6ff541ac, 0x6f5faee4c61f773f, 0xa84e8cd83c212c8a, 0xeabe6d3395cb1a19,
0x90c79d3fedd3f122, 0xd2377cd44439c7b1, 0x15265ee8be079c04, 0x57d6bf0317edaa97,
0xd9f4fb7ae3911dfd, 0x9b041a914a7b2b6e, 0x5c1538adb04570db, 0x1ee5d94619af4648,
0x02a151b5f156289c, 0x4051b05e58bc1e0f, 0x87409262a28245ba, 0xc5b073890b687329,
0x4b9237f0ff14c443, 0x0962d61b56fef2d0, 0xce73f427acc0a965, 0x8c8315cc052a9ff6,
0x3a80143f5cf17f13, 0x7870f5d4f51b4980, 0xbf61d7e80f251235, 0xfd913603a6cf24a6,
0x73b3727a52b393cc, 0x31439391fb59a55f, 0xf652b1ad0167feea, 0xb4a25046a88dc879,
0xa8e6d8b54074a6ad, 0xea16395ee99e903e, 0x2d071b6213a0cb8b, 0x6ff7fa89ba4afd18,
0xe1d5bef04e364a72, 0xa3255f1be7dc7ce1, 0x64347d271de22754, 0x26c49cccb40811c7,
0x5cbd6cc0cc10fafc, 0x1e4d8d2b65facc6f, 0xd95caf179fc497da, 0x9bac4efc362ea149,
0x158e0a85c2521623, 0x577eeb6e6bb820b0, 0x906fc95291867b05, 0xd29f28b9386c4d96,
0xcedba04ad0952342, 0x8c2b41a1797f15d1, 0x4b3a639d83414e64, 0x09ca82762aab78f7,
0x87e8c60fded7cf9d, 0xc51827e4773df90e, 0x020905d88d03a2bb, 0x40f9e43324e99428,
0x2cffe7d5975e55e2, 0x6e0f063e3eb46371, 0xa91e2402c48a38c4, 0xebeec5e96d600e57,
0x65cc8190991cb93d, 0x273c607b30f68fae, 0xe02d4247cac8d41b, 0xa2dda3ac6322e288,
0xbe992b5f8bdb8c5c, 0xfc69cab42231bacf, 0x3b78e888d80fe17a, 0x7988096371e5d7e9,
0xf7aa4d1a85996083, 0xb55aacf12c735610, 0x724b8ecdd64d0da5, 0x30bb6f267fa73b36,
0x4ac29f2a07bfd00d, 0x08327ec1ae55e69e, 0xcf235cfd546bbd2b, 0x8dd3bd16fd818bb8,
0x03f1f96f09fd3cd2, 0x41011884a0170a41, 0x86103ab85a2951f4, 0xc4e0db53f3c36767,
0xd8a453a01b3a09b3, 0x9a54b24bb2d03f20, 0x5d45907748ee6495, 0x1fb5719ce1045206,
0x919735e51578e56c, 0xd367d40ebc92d3ff, 0x1476f63246ac884a, 0x568617d9ef46bed9,
0xe085162ab69d5e3c, 0xa275f7c11f7768af, 0x6564d5fde549331a, 0x279434164ca30589,
0xa9b6706fb8dfb2e3, 0xeb46918411358470, 0x2c57b3b8eb0bdfc5, 0x6ea7525342e1e956,
0x72e3daa0aa188782, 0x30133b4b03f2b111, 0xf7021977f9cceaa4, 0xb5f2f89c5026dc37,
0x3bd0bce5a45a6b5d, 0x79205d0e0db05dce, 0xbe317f32f78e067b, 0xfcc19ed95e6430e8,
0x86b86ed5267cdbd3, 0xc4488f3e8f96ed40, 0x0359ad0275a8b6f5, 0x41a94ce9dc428066,
0xcf8b0890283e370c, 0x8d7be97b81d4019f, 0x4a6acb477bea5a2a, 0x089a2aacd2006cb9,
0x14dea25f3af9026d, 0x562e43b4931334fe, 0x913f6188692d6f4b, 0xd3cf8063c0c759d8,
0x5dedc41a34bbeeb2, 0x1f1d25f19d51d821, 0xd80c07cd676f8394, 0x9afce626ce85b507
```

5.12.2 Mirrored CRC

The CRC for a single byte is calculated as follows:

```
UInt8 CalculateCrcFor1Byte(UInt8 data, UInt64 crc) {
    return crcTable[(crc ^ data) & 0xff] ^ (crc >> 8);
}
```

The CRC table is initialized with the following values:

```
0x0000000000000000, 0x7ad870c830358979, 0xf5b0e190606b12f2, 0x8f689158505e9b8b,
0xc038e5739841b68f, 0xbae095bba8743ff6, 0x358804e3f82aa47d, 0x4f50742bc81f2d04,
0xab28ecb46814fe75, 0xd1f09c7c5821770c, 0x5e980d24087fec87, 0x24407dec384a65fe,
0x6b1009c7f05548fa, 0x11c8790fc060c183, 0x9ea0e857903e5a08, 0xe478989fa00bd371,
0x7d08ff3b88be6f81, 0x07d08ff3b88be6f8, 0x88b81eabe8d57d73, 0xf2606e63d8e0f40a,
0xbd301a4810ffd90e, 0xc7e86a8020ca5077, 0x4880fbd87094cbfc, 0x32588b1040a14285,
0xd620138fe0aa91f4, 0xacf86347d09f188d, 0x2390f21f80c18306, 0x594882d7b0f40a7f,
0x1618f6fc78eb277b, 0x6cc0863448deae02, 0xe3a8176c18803589, 0x997067a428b5bcf0,
0xfa11fe77117cdf02, 0x80c98ebf2149567b, 0x0fa11fe77117cdf0, 0x75796f2f41224489,
0x3a291b04893d698d, 0x40f16bccb908e0f4, 0xcf99fa94e9567b7f, 0xb5418a5cd963f206,
0x513912c379682177, 0x2be1620b495da80e, 0xa489f35319033385, 0xde51839b2936bafc,
0x9101f7b0e12997f8, 0xebd98778d11c1e81, 0x64b116208142850a, 0x1e6966e8b1770c73,
0x8719014c99c2b083, 0xfdc17184a9f739fa, 0x72a9e0dcf9a9a271, 0x08719014c99c2b08,
0x4721e43f0183060c, 0x3df994f731b68f75, 0xb29105af61e814fe, 0xc849756751dd9d87,
0x2c31edf8f1d64ef6, 0x56e99d30c1e3c78f, 0xd9810c6891bd5c04, 0xa3597ca0a188d57d,
0xec09088b6997f879, 0x96d1784359a27100, 0x19b9e91b09fcea8b, 0x636199d339c963f2,
0xdf7adabd7a6e2d6f, 0xa5a2aa754a5ba416, 0x2aca3b2d1a053f9d, 0x50124be52a30b6e4,
0x1f423fcee22f9be0, 0x659a4f06d21a1299, 0xeaf2de5e82448912, 0x902aae96b271006b,
0x74523609127ad31a, 0x0e8a46c1224f5a63, 0x81e2d7997211c1e8, 0xfb3aa75142244891,
0xb46ad37a8a3b6595, 0xceb2a3b2ba0eecec, 0x41da32eaea507767, 0x3b024222da65fe1e,
0xa2722586f2d042ee, 0xd8aa554ec2e5cb97, 0x57c2c41692bb501c, 0x2d1ab4dea28ed965,
0x624ac0f56a91f461, 0x1892b03d5aa47d18, 0x97fa21650afae693, 0xed2251ad3acf6fea,
0x095ac9329ac4bc9b, 0x7382b9faaaf135e2, 0xfcea28a2faafae69, 0x8632586aca9a2710,
0xc9622c4102850a14, 0xb3ba5c8932b0836d, 0x3cd2cdd162ee18e6, 0x460abd1952db919f,
0x256b24ca6b12f26d, 0x5fb354025b277b14, 0xd0dbc55a0b79e09f, 0xaa03b5923b4c69e6,
0xe553c1b9f35344e2, 0x9f8bb171c366cd9b, 0x10e3202993385610, 0x6a3b50e1a30ddf69,
0x8e43c87e03060c18, 0xf49bb8b633338561, 0x7bf329ee636d1eea, 0x012b592653589793,
0x4e7b2d0d9b47ba97, 0x34a35dc5ab7233ee, 0xbbcbcc9dfb2ca865, 0xc113bc55cb19211c,
0x5863dbf1e3ac9dec, 0x22bbab39d3991495, 0xadd33a6183c78f1e, 0xd70b4aa9b3f20667,
0x985b3e827bed2b63, 0xe2834e4a4bd8a21a, 0x6debdf121b863991, 0x1733afda2bb3b0e8,
0xf34b37458bb86399, 0x8993478dbb8deae0, 0x06fbd6d5ebd3716b, 0x7c23a61ddbe6f812,
0x3373d23613f9d516, 0x49aba2fe23cc5c6f, 0xc6c333a67392c7e4, 0xbc1b436e43a74e9d,
0x95ac9329ac4bc9b5, 0xef74e3e19c7e40cc, 0x601c72b9cc20db47, 0x1ac40271fc15523e,
0x5594765a340a7f3a, 0x2f4c0692043ff643, 0xa02497ca54616dc8, 0xdafce7026454e4b1,
0x3e847f9dc45f37c0, 0x445c0f55f46abeb9, 0xcb349e0da4342532, 0xb1eceec59401ac4b,
0xfebc9aee5c1e814f, 0x8464ea266c2b0836, 0x0b0c7b7e3c7593bd, 0x71d40bb60c401ac4,
0xe8a46c1224f5a634, 0x927c1cda14c02f4d, 0x1d148d82449eb4c6, 0x67ccfd4a74ab3dbf,
0x289c8961bcb410bb, 0x5244f9a98c8199c2, 0xdd2c68f1dcdf0249, 0xa7f41839ecea8b30,
0x438c80a64ce15841, 0x3954f06e7cd4d138, 0xb63c61362c8a4ab3, 0xcce411fe1cbfc3ca,
0x83b465d5d4a0eece, 0xf96c151de49567b7, 0x76048445b4cbfc3c, 0x0cdcf48d84fe7545,
0x6fbd6d5ebd3716b7, 0x15651d968d029fce, 0x9a0d8ccedd5c0445, 0xe0d5fc06ed698d3c,
0xaf85882d2576a038, 0xd55df8e515432941, 0x5a3569bd451db2ca, 0x20ed197575283bb3,
0xc49581ead523e8c2, 0xbe4df122e51661bb, 0x3125607ab548fa30, 0x4bfd10b2857d7349,
0x04ad64994d625e4d, 0x7e7514517d57d734, 0xf11d85092d094cbf, 0x8bc5f5c11d3cc5c6,
0x12b5926535897936, 0x686de2ad05bcf04f, 0xe70573f555e26bc4, 0x9ddd033d65d7e2bd,
0xd28d7716adc8cfb9, 0xa85507de9dfd46c0, 0x273d9686cda3dd4b, 0x5de5e64efd965432,
0xb99d7ed15d9d8743, 0xc3450e196da80e3a, 0x4c2d9f413df695b1, 0x36f5ef890dc31cc8,
0x79a59ba2c5dc31cc, 0x037deb6af5e9b8b5, 0x8c157a32a5b7233e, 0xf6cd0afa9582aa47,
0x4ad64994d625e4da, 0x300e395ce6106da3, 0xbf66a804b64ef628, 0xc5bed8cc867b7f51,
0x8aeeace74e645255, 0xf036dc2f7e51db2c, 0x7f5e4d772e0f40a7, 0x05863dbf1e3ac9de,
0xe1fea520be311aaf, 0x9b26d5e88e0493d6, 0x144e44b0de5a085d, 0x6e963478ee6f8124,
0x21c640532670ac20, 0x5b1e309b16452559, 0xd476a1c3461bbed2, 0xaeaed10b762e37ab,
0x37deb6af5e9b8b5b, 0x4d06c6676eae0222, 0xc26e573f3ef099a9, 0xb8b627f70ec510d0,
0xf7e653dcc6da3dd4, 0x8d3e2314f6efb4ad, 0x0256b24ca6b12f26, 0x788ec2849684a65f,
0x9cf65a1b368f752e, 0xe62e2ad306bafc57, 0x6946bb8b56e467dc, 0x139ecb4366d1eea5,
0x5ccebf68aecec3a1, 0x2616cfa09efb4ad8, 0xa97e5ef8cea5d153, 0xd3a62e30fe90582a,
0xb0c7b7e3c7593bd8, 0xca1fc72bf76cb2a1, 0x45775673a732292a, 0x3faf26bb9707a053,
0x70ff52905f188d57, 0x0a2722586f2d042e, 0x854fb3003f739fa5, 0xff97c3c80f4616dc,
0x1bef5b57af4dc5ad, 0x61372b9f9f784cd4, 0xee5fbac7cf26d75f, 0x9487ca0fff135e26,
0xdbd7be24370c7322, 0xa10fceec0739fa5b, 0x2e675fb4576761d0, 0x54bf2f7c6752e8a9,
0xcdcf48d84fe75459, 0xb71738107fd2dd20, 0x387fa9482f8c46ab, 0x42a7d9801fb9cfd2,
0x0df7adabd7a6e2d6, 0x772fdd63e7936baf, 0xf8474c3bb7cdf024, 0x829f3cf387f8795d,
0x66e7a46c27f3aa2c, 0x1c3fd4a417c62355, 0x935745fc4798b8de, 0xe98f353477ad31a7,
0xa6df411fbfb21ca3, 0xdc0731d78f8795da, 0x536fa08fdfd90e51, 0x29b7d047efec8728
```

5.13 Reed-Solomon encoding

R21 uses Reed-Solomon (RS) encoding to add error correction. Error correction codes are typically used in telecommunication to correct errors during transmittion or on media to correct e.g. errors caused by a scratch on a CD. RS coding takes considerably study to master, and books on the subject require at least some mathematical base knowledge on academic level. For this reason it's recommended to use an existing RS implementation, rather than to build one from scratch. When choosing to learn about the subject, a good book on the subject is "Error Control Coding, Second Edition", by Shu Lin and Daniel J. Costello, Jr. This book is taught over two semesters, to give an idea of the depth of the subject. RS coding is treated in Chapter 7 out of 22, to have a full understanding of the subject chapters 1-7 should be read.

An open source RS implementation is available from http://www.eccpage.com/, item "Reed-Solomon (RS) codes", by Simon Rockliff, 1989. This implementation uses Berlekamp-Masssey for decoding. Note that there are many ways to encode and decode, the implementation above is just one example. Though only 404 lines of code, the math involved is very sophisticated.

DWG file format version R21 uses two configurations of RS coding:

- Data pages: use a (n, k) of (255, 251), the primitive polynomial coefficients being (1, 0, 1, 1, 1, 0, 0, 0). This configuration can correct (255 – 251) / 2 = 2 error bytes per block of 255 bytes. For each 251 data bytes (k), 4 parity bytes are added to form a 255 byte (code word) block.

- System pages: use a (n, k) of (255, 239), the primitive polynomial coefficients being (1, 0, 0, 1, 0, 1, 1, 0). This configuration can correct (255 – 239) / 2 = 8 error bytes per block of 255 bytes. For each 239 data bytes (k), 16 parity bytes are added to form a 255 byte (code word) block.

In the RS implementation by Simon Rockliff the primitive polynomial coefficients are stored in variable pp. From these coefficients the lookup tables for Galois field math and the generator polynomial coefficients are created.

The encoded bytes may be interleaved depending on the context. Some data/system pages are interleaved, some are not.

5.13.1 Non-interleaved

All original data blocks are followed by the parity byte blocks (i.e. the first parity block follows the last data block).

When the last block is not entirely filled, then random bytes are added from the random encoding (see paragraph 5.11) to fill the block to have size k.

5.13.2 Interleaved

When more than 1 block of data is encoded, the encoded block data is interleaved. E.g. when there are 3 blocks to be encoded, then the data bytes and parity bytes of the first block are written to positions 3 x i (where i is an integer >= 0). The encoded bytes of the second block are written to positions 3 x i + 1 andof the third block to positions 3 x i + 2.

When the last block is not entirely filled, then random bytes are added from the random encoding (see paragraph 5.11) to fill the block to have size k.

6 R2010 DWG FILE FORMAT ORGANIZATION

The 2010 format is based mostly on the 2004 format and somewhat on the 2007 format. The file header, page map, section map, compression are the same as in R2004. The bit coding is the same as in R2007 (see chapter 2), with the exception of the Object Type being encoded differently (see paragraph 2.12). Like the R2007 format, the data, strings and handles are separated in header and objects sections.

7 R2013 DWG FILE FORMAT ORGANIZATION

The 2013 format is based mostly on the 2010 format. The file header, summary info, page map, section map, compression are the same as in R2004. The bit coding is the same as in R2010. Like the R2007 format, the data, strings and handles are separated in header and objects sections. The changes in the Header section are minor (only 2 added fields).

A new data section was introduced, the data storage section (AcDb:AcDsPrototype_1b). At this moment (December 2012), this sections contains information about Acis data (regions, solids). See chapter 24 for more details about this section.

Note that at the point of writing (22 March 2013) known valid values for acad maintenance version are 6 and 8. The ODA currently writes value 8.

8 R2018 DWG FILE FORMAT ORGANIZATION

The AutoCAD 2018 format is almost identical to the 2013 format. Structurally they are identical.

Below is a summary of the changes:

- Three shorts (int16) with value zero have been added to end of the auxiliary file header (see chapter 27).
- In the AcDb:Header nothing changed, but note that the unknown 32-bit int at the start, directly following the section size that was present for R2010/R2013 for acad maintenance version greater than 3, is also present for R2018 (see chapter 9).
- Additions/changes in the following entities:
    - ACAD_PROXY_ENTITY (paragraph 20.4.90),
    - ATTRIB (paragraph 20.4.4),
    - ATTDEF (paragraph 20.4.5),
    - MTEXT (see paragraph 20.4.46).
- Object MLINESTYLE (paragraph 20.4.73) references line types in its element by their handle rather than by index.

9 Data section AcDb:Header (HEADER VARIABLES)

The header contains all header (system) variables, except the MEASUREMENT variable, which is present in the AcDb:Template section, see chapter 22.

The header variables section indicated by section-locator 0 has the following form:

```
    Beginning sentinel
    Size of the section (a 4 byte long)
    R2010/R2013 (only present if the maintenance version is greater than 3!) or R2018+:
       Unknown (4 byte long), might be part of a 64-bit size.
    Data (system variables and possibly other data at the beginning)
    CRC (covers the stepper and the data)
    Ending sentinel
```

This data section appear as one long stream, with no gaps. Most are bit coded. (See the BIT CODES section.) The header is padded with random bits to the next byte boundary.

The following 16 byte sentinel introduces this section:

```
0xCF,0x7B,0x1F,0x23,0xFD,0xDE,0x38,0xA9,0x5F,0x7C,0x68,0xB8,0x4E,0x6D,0x33,0x5F
    RL : Size of the section.
```

Next come the data items, as listed below:

```
      TYPE    DESCRIPTION
R2007 Only:
         RL  :  Size in bits
R2013+:
        BLL  :  Variabele REQUIREDVERSIONS, default value 0, read only.
Common:
         BD  :  Unknown, default value 412148564080.0
         BD  :  Unknown, default value 1.0
         BD  :  Unknown, default value 1.0
         BD  :  Unknown, default value 1.0
         TV  :  Unknown text string, default ""
         TV  :  Unknown text string, default ""
         TV  :  Unknown text string, default ""
         TV  :  Unknown text string, default ""
         BL  :  Unknown long, default value 24L
         BL  :  Unknown long, default value 0L;
R13-R14 Only:
```

```
         BS  : Unknown short, default value 0
Pre-2004 Only:
          H  : Handle of the current viewport entity header (hard pointer)
Common:
          B  : DIMASO
          B  : DIMSHO
R13-R14 Only:
          B  : DIMSAV  Undocumented.
Common:
          B  : PLINEGEN
          B  : ORTHOMODE
          B  : REGENMODE
          B  : FILLMODE
          B  : QTEXTMODE
          B  : PSLTSCALE
          B  : LIMCHECK
R13-R14 Only (stored in registry from R15 onwards):
          B  : BLIPMODE
R2004+:
          B  : Undocumented
Common:
          B  : USRTIMER (User timer on/off).
          B  : SKPOLY
          B  : ANGDIR
          B  : SPLFRAME
R13-R14 Only (stored in registry from R15 onwards):
          B  : ATTREQ
          B  : ATTDIA
Common:
          B  : MIRRTEXT
          B  : WORLDVIEW
R13-R14 Only:
          B  : WIREFRAME  Undocumented.
Common:
          B  : TILEMODE
          B  : PLIMCHECK
          B  : VISRETAIN
R13-R14 Only (stored in registry from R15 onwards):
          B  : DELOBJ
Common:
```

```
          B  : DISPSILH
          B  : PELLIPSE (not present in DXF)
         BS  : PROXYGRAPHICS
R13-R14 Only (stored in registry from R15 onwards):
         BS  : DRAGMODE
Common:
         BS  : TREEDEPTH
         BS  : LUNITS
         BS  : LUPREC
         BS  : AUNITS
         BS  : AUPREC
R13-R14 Only Only (stored in registry from R15 onwards):
         BS  : OSMODE
Common:
         BS  : ATTMODE
R13-R14 Only Only (stored in registry from R15 onwards):
         BS  : COORDS
Common:
         BS  : PDMODE
R13-R14 Only Only (stored in registry from R15 onwards):
         BS  : PICKSTYLE
R2004+:
         BL  : Unknown
         BL: Unknown
         BL  : Unknown
Common:
         BS  : USERI1
         BS  : USERI2
         BS  : USERI3
         BS  : USERI4
         BS  : USERI5
         BS  : SPLINESEGS
         BS  : SURFU
         BS  : SURFV
         BS  : SURFTYPE
         BS  : SURFTAB1
         BS  : SURFTAB2
         BS  : SPLINETYPE
         BS  : SHADEDGE
         BS  : SHADEDIF
         BS : UNITMODE
         BS : MAXACTVP
         BS : ISOLINES
         BS : CMLJUST
         BS : TEXTQLTY
         BD : LTSCALE
         BD : TEXTSIZE
         BD : TRACEWID
         BD : SKETCHINC
         BD : FILLETRAD
         BD : THICKNESS
         BD : ANGBASE
         BD : PDSIZE
         BD : PLINEWID
         BD : USERR1
         BD : USERR2
         BD : USERR3
         BD : USERR4
         BD : USERR5
         BD : CHAMFERA
         BD : CHAMFERB
         BD : CHAMFERC
         BD : CHAMFERD
         BD : FACETRES
         BD : CMLSCALE
         BD : CELTSCALE
R13-R18:
         TV : MENUNAME
Common:
         BL : TDCREATE   (Julian day)
         BL : TDCREATE   (Milliseconds into the day)
         BL : TDUPDATE   (Julian day)
         BL : TDUPDATE   (Milliseconds into the day)
R2004+:
         BL : Unknown
         BL : Unknown
         BL : Unknown
Common:
         BL : TDINDWG    (Days)
         BL : TDINDWG    (Milliseconds into the day)
         BL  :  TDUSRTIMER (Days)

         BL  :  TDUSRTIMER (Milliseconds into the day)

        CMC  :  CECOLOR

          H  :  HANDSEED  The next handle, with an 8-bit length specifier preceding the handle
                bytes (standard hex handle form) (code 0). The HANDSEED is not part of the handle
                stream, but of the normal data stream (relevant for R21 and later).

          H  :  CLAYER (hard pointer)

          H  :  TEXTSTYLE (hard pointer)

          H  :  CELTYPE (hard pointer)

R2007+ Only:

          H  :  CMATERIAL (hard pointer)

Common:

          H  :  DIMSTYLE (hard pointer)

          H  :  CMLSTYLE (hard pointer)

R2000+ Only:

         BD  :  PSVPSCALE

Common:

        3BD  :  INSBASE                           (PSPACE)

        3BD  :  EXTMIN                            (PSPACE)

        3BD  :  EXTMAX                            (PSPACE)

        2RD  :  LIMMIN                            (PSPACE)

        2RD  :  LIMMAX                            (PSPACE)

         BD  :  ELEVATION                         (PSPACE)

        3BD  :  UCSORG                            (PSPACE)

        3BD  :  UCSXDIR                           (PSPACE)

        3BD  :  UCSYDIR                           (PSPACE)

          H  :  UCSNAME                           (PSPACE)  (hard pointer)

R2000+ Only:

          H  :  PUCSORTHOREF (hard pointer)

         BS  :  PUCSORTHOVIEW

          H  :  PUCSBASE (hard pointer)

        3BD  :  PUCSORGTOP

        3BD  :  PUCSORGBOTTOM

        3BD  :  PUCSORGLEFT

        3BD  :  PUCSORGRIGHT

        3BD  :  PUCSORGFRONT

        3BD  :  PUCSORGBACK

Common:

        3BD  :  INSBASE                           (MSPACE)

        3BD  :  EXTMIN                            (MSPACE)

        3BD  :  EXTMAX                            (MSPACE)
        2RD  :  LIMMIN                              (MSPACE)
        2RD  :  LIMMAX                              (MSPACE)
         BD  :  ELEVATION                           (MSPACE)
        3BD  :  UCSORG                              (MSPACE)
        3BD  :  UCSXDIR                             (MSPACE)
        3BD  :  UCSYDIR                             (MSPACE)
          H  :  UCSNAME                             (MSPACE)  (hard pointer)
R2000+ Only:
          H  :   UCSORTHOREF (hard pointer)
         BS  :  UCSORTHOVIEW
          H  :  UCSBASE (hard pointer)
        3BD  :  UCSORGTOP
        3BD  :  UCSORGBOTTOM
        3BD  :  UCSORGLEFT
        3BD  :  UCSORGRIGHT
        3BD  :  UCSORGFRONT
        3BD  :  UCSORGBACK
         TV  :  DIMPOST
         TV  :  DIMAPOST
R13-R14 Only:
          B  :  DIMTOL
          B  :  DIMLIM
          B  :  DIMTIH
          B  :  DIMTOH
          B  :  DIMSE1
          B  :  DIMSE2
          B  :  DIMALT
          B  :  DIMTOFL
          B  :  DIMSAH
          B  :  DIMTIX
          B  :  DIMSOXD
         RC  :  DIMALTD
         RC  :  DIMZIN
          B  :  DIMSD1
          B  :  DIMSD2
         RC  :  DIMTOLJ
         RC  :  DIMJUST
         RC  :  DIMFIT
          B  :  DIMUPT
         RC  :  DIMTZIN
         RC  : DIMALTZ
         RC  : DIMALTTZ
         RC  : DIMTAD
         BS  : DIMUNIT
         BS  : DIMAUNIT
         BS  : DIMDEC
         BS  : DIMTDEC
         BS  : DIMALTU
         BS  : DIMALTTD
          H  : DIMTXSTY (hard pointer)
Common:
         BD  : DIMSCALE
         BD  : DIMASZ
         BD  : DIMEXO
         BD  : DIMDLI
         BD  : DIMEXE
         BD  : DIMRND
         BD  : DIMDLE
         BD  : DIMTP
         BD  : DIMTM
R2007+ Only:
         BD  : DIMFXL
         BD  : DIMJOGANG
         BS  : DIMTFILL
        CMC  : DIMTFILLCLR
R2000+ Only:
          B  : DIMTOL
          B  : DIMLIM
          B  : DIMTIH
          B  : DIMTOH
          B  : DIMSE1
          B  : DIMSE2
         BS  : DIMTAD
         BS  : DIMZIN
         BS  : DIMAZIN
R2007+ Only:
         BS  : DIMARCSYM
Common:
         BD  : DIMTXT
         BD  : DIMCEN
         BD  : DIMTSZ
         BD  : DIMALTF
         BD  : DIMLFAC
         BD  : DIMTVP
         BD  : DIMTFAC
         BD  : DIMGAP
R13-R14 Only:
          T  : DIMPOST
          T  : DIMAPOST
          T  : DIMBLK
          T  : DIMBLK1
          T  : DIMBLK2
R2000+ Only:
         BD  : DIMALTRND
          B  : DIMALT
         BS  : DIMALTD
          B  : DIMTOFL
          B  : DIMSAH
          B  : DIMTIX
          B  : DIMSOXD
Common:
        CMC  : DIMCLRD
        CMC  : DIMCLRE
        CMC  : DIMCLRT
R2000+ Only:
         BS  : DIMADEC
         BS  : DIMDEC
         BS  : DIMTDEC
         BS  : DIMALTU
         BS  : DIMALTTD
         BS  : DIMAUNIT
         BS  : DIMFRAC
         BS  : DIMLUNIT
         BS  : DIMDSEP
         BS  : DIMTMOVE
         BS  : DIMJUST
          B  : DIMSD1
          B  : DIMSD2
         BS  : DIMTOLJ
         BS  : DIMTZIN
         BS  : DIMALTZ
         BS  : DIMALTTZ
          B  : DIMUPT
         BS  : DIMATFIT
R2007+ Only:
          B  : DIMFXLON
R2010+ Only:
          B  : DIMTXTDIRECTION
         BD  : DIMALTMZF
          T  : DIMALTMZS
         BD  : DIMMZF
          T  : DIMMZS
R2000+ Only:
          H  : DIMTXSTY (hard pointer)
          H  : DIMLDRBLK (hard pointer)
          H  : DIMBLK (hard pointer)
          H  : DIMBLK1 (hard pointer)
          H  : DIMBLK2 (hard pointer)
R2007+ Only:
          H  : DIMLTYPE (hard pointer)
          H  : DIMLTEX1 (hard pointer)
          H  : DIMLTEX2 (hard pointer)
R2000+ Only:
         BS  : DIMLWD
         BS  : DIMLWE
Common:
          H  : BLOCK CONTROL OBJECT (hard owner)
          H  : LAYER CONTROL OBJECT (hard owner)
          H  : STYLE CONTROL OBJECT (hard owner)
          H  : LINETYPE CONTROL OBJECT (hard owner)
          H  : VIEW CONTROL OBJECT (hard owner)
          H  : UCS CONTROL OBJECT (hard owner)
          H  : VPORT CONTROL OBJECT (hard owner)
          H  : APPID CONTROL OBJECT (hard owner)
          H  : DIMSTYLE CONTROL OBJECT (hard owner)
R13-R15 Only:
          H  : VIEWPORT ENTITY HEADER CONTROL OBJECT (hard owner)
Common:
          H  : DICTIONARY (ACAD_GROUP) (hard pointer)
          H  : DICTIONARY (ACAD_MLINESTYLE) (hard pointer)
          H  : DICTIONARY (NAMED OBJECTS) (hard owner)
R2000+ Only:
         BS  : TSTACKALIGN, default = 1 (not present in DXF)
         BS  : TSTACKSIZE, default = 70 (not present in DXF)
         TV  : HYPERLINKBASE
         TV  : STYLESHEET
          H  : DICTIONARY (LAYOUTS) (hard pointer)
          H  : DICTIONARY (PLOTSETTINGS) (hard pointer)
          H  : DICTIONARY (PLOTSTYLES) (hard pointer)
R2004+:
          H  : DICTIONARY (MATERIALS) (hard pointer)
          H  : DICTIONARY (COLORS) (hard pointer)
R2007+:
          H  : DICTIONARY (VISUALSTYLE) (hard pointer)
R2013+:
          H  : UNKNOWN (hard pointer)
R2000+:
         BL  : Flags:
                      CELWEIGHT      Flags & 0x001F
                      ENDCAPS        Flags & 0x0060
                      JOINSTYLE      Flags & 0x0180
                      LWDISPLAY      !(Flags & 0x0200)
                      XEDIT          !(Flags & 0x0400)
                      EXTNAMES       Flags & 0x0800
                      PSTYLEMODE     Flags & 0x2000
                      OLESTARTUP     Flags & 0x4000
         BS  : INSUNITS
         BS  : CEPSNTYPE
          H  : CPSNID  (present only if CEPSNTYPE == 3) (hard pointer)
         TV  : FINGERPRINTGUID
         TV  : VERSIONGUID
R2004+:
         RC  : SORTENTS
         RC  : INDEXCTL
         RC  : HIDETEXT
         RC  : XCLIPFRAME, before R2010 the value can be 0 or 1 only.
         RC  : DIMASSOC
         RC  : HALOGAP
         BS  : OBSCUREDCOLOR
         BS  : INTERSECTIONCOLOR
         RC  : OBSCUREDLTYPE
```

```
        RC  :  INTERSECTIONDISPLAY
        TV  :  PROJECTNAME
Common:
         H  :  BLOCK_RECORD (*PAPER_SPACE) (hard pointer)
         H  :  BLOCK_RECORD (*MODEL_SPACE) (hard pointer)
         H  :  LTYPE (BYLAYER) (hard pointer)
         H  :  LTYPE (BYBLOCK) (hard pointer)
         H  :  LTYPE (CONTINUOUS) (hard pointer)
R2007+:
         B  :  CAMERADISPLAY
        BL  :  unknown
        BL  :  unknown
        BD  :  unknown
        BD  :  STEPSPERSEC
        BD  :  STEPSIZE
        BD  :  3DDWFPREC
        BD  :  LENSLENGTH
        BD  :  CAMERAHEIGHT
        RC  :  SOLIDHIST
        RC  :  SHOWHIST
        BD  :  PSOLWIDTH
        BD  :  PSOLHEIGHT
        BD  :  LOFTANG1
        BD  :  LOFTANG2
        BD  :  LOFTMAG1
        BD  :  LOFTMAG2
        BS  :  LOFTPARAM
        RC  :  LOFTNORMALS
        BD  :  LATITUDE
        BD  :  LONGITUDE
        BD  :  NORTHDIRECTION
        BL  :  TIMEZONE
        RC  :  LIGHTGLYPHDISPLAY
        RC  :  TILEMODELIGHTSYNCH
        RC  :  DWFFRAME
        RC  :  DGNFRAME
         B  :  unknown
       CMC  :  INTERFERECOLOR
         H  :  INTERFEREOBJVS (hard pointer)
         H  :  INTERFEREVPVS (hard pointer)
```

```
          H  : DRAGVS (hard pointer)
          RC : CSHADOW
          BD : unknown
R14+:
          BS : unknown short (type 5/6 only)  these do not seem to be required,
          BS : unknown short (type 5/6 only)  even for type 5.
          BS : unknown short (type 5/6 only)
          BS : unknown short (type 5/6 only)
Common:
          RS : CRC for the data section, starting after the sentinel. Use 0xC0C1 for the initial
               value.
```

This following 16-byte sentinel appears after the CRC:

`0x30,0x84,0xE0,0xDC,0x02,0x21,0xC7,0x56,0xA0,0x83,0x97,0x47,0xB1,0x92,0xCC,0xA0`

Here is a dump of a complete R14 header:

```
empty14.dwg  02/24/98  11:40:03
          0  1  2  3  4  5  6  7
    00000 41 43 31 30 31 34 00 00   AC1014..   0100 0001 0100 0011 0011 0001 0011 0000 0011 0001 0011 0100 0000 0000 0000 0000

    00008 00 00 00 00 01 3F 0C 00   .....?..   0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0011 1111 0000 1100 0000 0000

    00010 00 00 00 1E 00 05 00 00   ........   0000 0000 0000 0000 0000 0000 0001 1110 0000 0000 0000 0101 0000 0000 0000 0000

    00018 00 00 58 00 00 00 ED 01   ..X.....   0000 0000 0000 0000 0101 1000 0000 0000 0000 0000 0000 0000 1110 1101 0000 0001

    00020 00 00 01 45 02 00 00 26   ...E...&   0000 0000 0000 0000 0000 0001 0100 0101 0000 0010 0000 0000 0000 0000 0010 0110

    00028 00 00 00 02 27 0B 00 00   ....'...   0000 0000 0000 0000 0000 0000 0000 0010 0010 0111 0000 1011 0000 0000 0000 0000

    00030 50 00 00 00 03 77 0B 00   P....w..   0101 0000 0000 0000 0000 0000 0000 0000 0000 0011 0111 0111 0000 1011 0000 0000

    00038 00 35 00 00 00 04 3B 0C   .5....;.   0000 0000 0011 0101 0000 0000 0000 0000 0000 0000 0000 0100 0011 1011 0000 1100
          0  1  2  3  4  5  6  7
    00040 00 00 04 00 00 00 2D 5C   ......-\   0000 0000 0000 0000 0000 0100 0000 0000 0000 0000 0000 0000 0010 1101 0101 1100

    00048 95 A0 4E 28 99 82 1A E5   ..N(....   1001 0101 1010 0000 0100 1110 0010 1000 1001 1001 1000 0010 0001 1010 1110 0101

    00050 5E 41 E0 5F 9D 3A 4D 00   ^A._.:M.   0101 1110 0100 0001 1110 0000 0101 1111 1001 1101 0011 1010 0100 1101 0000 0000

    00058 CF 7B 1F 23 FD DE 38 A9   .{.#..8.   1100 1111 0111 1011 0001 1111 0010 0011 1111 1101 1101 1110 0011 1000 1010 1001

    00060 5F 7C 68 B8 4E 6D 33 5F   _|h.Nm3_   0101 1111 0111 1100 0110 1000 1011 1000 0100 1110 0110 1101 0011 0011 0101 1111

    00068 C7 01 00 00 00 00 07 00   ........   1100 0111 0000 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0111 0000 0000

    00070 1F BF 55 D0 95 40 5B 6A   ..U..@[j   0001 1111 1011 1111 0101 0101 1101 0000 1001 0101 0100 0000 0101 1011 0110 1010

    00078 51 A9 43 1A 65 AC 40 50   Q.C.e.@P   0101 0001 1010 1001 0100 0011 0001 1010 0110 0101 1010 1100 0100 0000 0101 0000

          0  1  2  3  4  5  6  7
    00080 23 30 2D 02 41 2A 40 50   #0-.A*@P   0010 0011 0011 0000 0010 1101 0000 0010 0100 0001 0010 1010 0100 0000 0101 0000

    00088 19 01 AA 90 84 19 06 41   .......A   0001 1001 0000 0001 1010 1010 1001 0000 1000 0100 0001 1001 0000 0110 0100 0001
00090 90 64 19 06 40 D4 69 30   .d..@.i0   1001 0000 0110 0100 0001 1001 0000 0110 0100 0000 1101 0100 0110 1001 0011 0000

00098 41 24 C9 26 A6 66 66 66   A$.&.fff   0100 0001 0010 0100 1100 1001 0010 0110 1010 0110 0110 0110 0110 0110 0110 0110

000A0 66 72 4F C9 A9 99 99 99   frO.....   0110 0110 0111 0010 0100 1111 1100 1001 1010 1001 1001 1001 1001 1001 1001 1001

000A8 99 9A 93 F2 6A 66 66 66   ....jfff   1001 1001 1001 1010 1001 0011 1111 0010 0110 1010 0110 0110 0110 0110 0110 0110

000B0 66 66 E4 FC 00 00 00 00   ff......   0110 0110 0110 0110 1110 0100 1111 1100 0000 0000 0000 0000 0000 0000 0000 0000

000B8 00 00 E0 3F AA AA 80 00   ...?....   0000 0000 0000 0000 1110 0000 0011 1111 1010 1010 1010 1010 1000 0000 0000 0000

       0  1  2  3  4  5  6  7
000C0 00 00 00 00 0E 03 F0 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 1110 0000 0011 1111 0000 0000 0000

000C8 00 00 00 00 03 80 FD 80   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0011 1000 0000 1111 1101 1000 0000

000D0 00 00 00 00 00 0E 03 F5   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1110 0000 0011 1111 0101

000D8 40 4B 8B 56 52 50 02 D1   @K.VRP..   0100 0000 0100 1011 1000 1011 0101 0110 0101 0010 0101 0000 0000 0010 1101 0001

000E0 A6 00 08 B5 65 25 00 20   ....e%.    1010 0110 0000 0000 0000 1000 1011 0101 0110 0101 0010 0101 0000 0000 0010 0000

000E8 29 E0 00 A3 30 F4 00 02   )...0...   0010 1001 1110 0000 0000 0000 1010 0011 0011 0000 1111 0100 0000 0000 0000 0010

000F0 33 0F 40 00 30 14 D5 10   3.@.0...   0011 0011 0000 1111 0100 0000 0000 0000 0011 0000 0001 0100 1101 0101 0001 0000

000F8 F5 11 05 11 45 11 D5 11   ....E...   1111 0101 0001 0001 0000 0101 0001 0001 0100 0101 0001 0001 1101 0101 0001 0001

       0  1  2  3  4  5  6  7
00100 CA 84 08 CB 57 81 DA F1   ....W...   1100 1010 1000 0100 0000 1000 1100 1011 0101 0111 1000 0001 1101 1010 1111 0001

00108 54 41 02 32 D5 E0 76 BC   TA.2..v.   0101 0100 0100 0001 0000 0010 0011 0010 1101 0101 1110 0000 0111 0110 1011 1100

00110 55 10 40 8C B5 78 1D AF   U.@..x..   0101 0101 0001 0000 0100 0000 1000 1100 1011 0101 0111 1000 0001 1101 1010 1111

00118 15 44 10 23 2D 5E 07 6B   .D.#-^.k   0001 0101 0100 0100 0001 0000 0010 0011 0010 1101 0101 1110 0000 0111 0110 1011

00120 C5 71 04 08 CB 57 81 DA   .q...W..   1100 0101 0111 0001 0000 0100 0000 1000 1100 1011 0101 0111 1000 0001 1101 1010

00128 F1 5C 41 02 32 D5 E0 76   .\A.2..v   1111 0001 0101 1100 0100 0001 0000 0010 0011 0010 1101 0101 1110 0000 0111 0110

00130 BC 57 10 00 00 00 00 00   .W......   1011 1100 0101 0111 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

00138 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

       0  1  2  3  4  5  6  7
00140 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

00148 00 A1 00 00 00 00 00 00   ........   0000 0000 1010 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

00150 00 89 02 A9 A9 94 2A 10   ......*.   0000 0000 1000 1001 0000 0010 1010 1001 1010 1001 1001 0100 0010 1010 0001 0000

00158 23 2D 5E 07 6B C5 51 04   #-^.k.Q.   0010 0011 0010 1101 0101 1110 0000 0111 0110 1011 1100 0101 0101 0001 0000 0100

00160 08 CB 57 81 DA F1 54 41   ..W...TA   0000 1000 1100 1011 0101 0111 1000 0001 1101 1010 1111 0001 0101 0100 0100 0001

00168 02 32 D5 E0 76 BC 55 10   .2..v.U.   0000 0010 0011 0010 1101 0101 1110 0000 0111 0110 1011 1100 0101 0101 0001 0000

00170 40 8C B5 78 1D AF 15 C4   @..x....   0100 0000 1000 1100 1011 0101 0111 1000 0001 1101 1010 1111 0001 0101 1100 0100

00178 10 23 2D 5E 07 6B C5 71   .#-^.k.q   0001 0000 0010 0011 0010 1101 0101 1110 0000 0111 0110 1011 1100 0101 0111 0001
```

```
      0  1  2  3  4  5  6  7
00180 04 08 CB 57 81 DA F1 5C   ...W...\   0000 0100 0000 1000 1100 1011 0101 0111 1000 0001 1101 1010 1111 0001 0101 1100

00188 40 00 00 00 00 00 00 00   @.......   0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

00190 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

00198 00 00 00 00 00 00 02 84   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 1000 0100

001A0 00 00 00 00 00 00 02 24   .......$   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 0010 0100

001A8 0A A6 A6 50 30 00 40 00   ...P0.@.   0000 1010 1010 0110 1010 0110 0101 0000 0011 0000 0000 0000 0100 0000 0000 0000

001B0 08 00 18 00 00 00 01 02   ........   0000 1000 0000 0000 0001 1000 0000 0000 0000 0000 0000 0000 0000 0001 0000 0010

001B8 90 44 11 02 40 94 44 10   .D..@.D.   1001 0000 0100 0100 0001 0001 0000 0010 0100 0000 1001 0100 0100 0100 0001 0000

      0  1  2  3  4  5  6  7
001C0 2B 5E 8D C0 F4 2B 1C FC   +^...+..   0010 1011 0101 1110 1000 1101 1100 0000 1111 0100 0010 1011 0001 1100 1111 1100

001C8 00 00 00 00 00 00 B0 3F   .......?   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1011 0000 0011 1111

001D0 14 AE 07 A1 7A D4 76 0F   ....z.v.   0001 0100 1010 1110 0000 0111 1010 0001 0111 1010 1101 0100 0111 0110 0000 1111

001D8 C0 AD 7A 37 03 D0 AC 73   ..z7...s   1100 0000 1010 1101 0111 1010 0011 0111 0000 0011 1101 0000 1010 1100 0111 0011

001E0 FA A0 2B 5E 8D C0 F4 2B   ..+^...+   1111 1010 1010 0000 0010 1011 0101 1110 1000 1101 1100 0000 1111 0100 0010 1011

001E8 1C FC 0A D7 A3 70 3D 0A   .....p=.   0001 1100 1111 1100 0000 1010 1101 0111 1010 0011 0111 0000 0011 1101 0000 1010

001F0 B7 3F 86 66 66 66 66 66   .?.fffff   1011 0111 0011 1111 1000 0110 0110 0110 0110 0110 0110 0110 0110 0110 0110 0110

001F8 63 94 06 40 AD 7A 37 03   c..@.z7.   0110 0011 1001 0100 0000 0110 0100 0000 1010 1101 0111 1010 0011 0111 0000 0011

      0  1  2  3  4  5  6  7
00200 D0 AB 73 FA AA A3 10 13   ..s.....   1101 0000 1010 1011 0111 0011 1111 1010 1010 1010 1010 0011 0001 0000 0001 0011

00208 10 23 10 33 10 53 10 63   .#.3.S.c   0001 0000 0010 0011 0001 0000 0011 0011 0001 0000 0101 0011 0001 0000 0110 0011

00210 10 73 10 83 10 93 10 A3   .s......   0001 0000 0111 0011 0001 0000 1000 0011 0001 0000 1001 0011 0001 0000 1010 0011

00218 10 B5 10 D5 10 E3 10 C5   ........   0001 0000 1011 0101 0001 0000 1101 0101 0001 0000 1110 0011 0001 0000 1100 0101

00220 11 65 11 95 11 45 11 35   .e...E.5   0001 0001 0110 0101 0001 0001 1001 0101 0001 0001 0100 0101 0001 0001 0011 0101

00228 11 51 D5 58 D4 A0 34 26   .Q.X..4&   0001 0001 0101 0001 1101 0101 0101 1000 1101 0100 1010 0000 0011 0100 0010 0110

00230 4B 76 E0 5B 27 30 84 E0   Kv.['0..   0100 1011 0111 0110 1110 0000 0101 1011 0010 0111 0011 0000 1000 0100 1110 0000

00238 DC 02 21 C7 56 A0 83 97   ..!.V...   1101 1100 0000 0010 0010 0001 1100 0111 0101 0110 1010 0000 1000 0011 1001 0111

      0  1  2  3  4  5  6  7
00240 47 B1 92 CC A0            G....      0100 0111 1011 0001 1001 0010 1100 1100 1010 0000
```

10 Data section AcDb:Classes

10.1 R13-R15

This section contains the defined classes for the drawing.

```
        SN  :  0x8D 0xA1 0xC4 0xB8 0xC4 0xA9 0xF8 0xC5 0xC0 0xDC 0xF4 0x5F 0xE7 0xCF 0xB6 0x8A.
        RL  :  size of class data area.
```

Then follows the class data:

```
        BS  :  classnum
        BS  :  version – in R14, becomes a flag indicating whether objects can be moved, edited,
               etc.  We are still examining this.
        TV  :  appname
        TV  :  cplusplusclassname
        TV  :  classdxfname
         B  :  wasazombie
        BS  :  itemclassid -- 0x1F2 for classes which produce entities, 0x1F3 for classes which
               produce objects.
```

We read sets of these until we exhaust the data.

```
        RS  :  CRC
```

This following 16-byte sentinel appears after the CRC:

```
0x72,0x5E,0x3B,0x47,0x3B,0x56,0x07,0x3A,0x3F,0x23,0x0B,0xA0,0x18,0x30,0x49,0x75
```

```
For R18 and later 8 unknown bytes follow. The ODA writes 0 bytes.
```

10.2 R18+

This section is compressed and contains the standard 32 byte section header.

This section contains the defined classes for the drawing.

```
        SN  :  0x8D 0xA1 0xC4 0xB8 0xC4 0xA9 0xF8 0xC5 0xC0 0xDC 0xF4 0x5F 0xE7 0xCF 0xB6 0x8A.
        RL  :  size of class data area.
R2010+ (only present if the maintenance version is greater than 3!)
        RL  :  unknown, possibly the high 32 bits of a 64-bit size?
R2004+
        BS  :  Maxiumum class number
        RC  :  0x00
        RC  :  0x00
         B  :  true
```

Then follows the class data (note that strings are in the string stream for R2007+):

```
        BS  :  classnum
        BS  :  Proxy flags:
               Erase allowed = 1,
               transform allowed = 2,
               color change allowed = 4,
               layer change allowed = 8,
               line type change allowed = 16,
               line type scale change allowed = 32,
               visibility change allowed = 64,
               cloning allowed = 128,
               Lineweight change allowed = 256,
               Plot Style Name change allowed = 512,
               Disables proxy warning dialog = 1024,
               is R13 format proxy= 32768
        TV  :  appname
        TV  :  cplusplusclassname
        TV  :  classdxfname
         B  :  wasazombie
        BS  :  itemclassid -- 0x1F2 for classes which produce entities, 0x1F3 for classes which
               produce objects.
        BL  :  Number of objects created of this type in the current DB (DXF 91).
        BS  :  Dwg Version
        BS  :  Maintenance release version.
        BL  :  Unknown (normally 0L)
        BL  :  Unknown (normally 0L)
```

We read sets of these until we exhaust the data.

```
        RS  :  CRC
```

This following 16-byte sentinel appears after the CRC:

```
0x72,0x5E,0x3B,0x47,0x3B,0x56,0x07,0x3A,0x3F,0x23,0x0B,0xA0,0x18,0x30,0x49,0x75
```

11 PADDING (R13C3 AND LATER)

0x200 bytes of padding. Can be ignored. When writing, the Open Design Toolkit writes all 0s. Occasionally AutoCAD will use the first 4 bytes of this area to store the value of the “measurement” variable. This padding was evidently required to allow pre-R13C3 versions of AutoCAD to read files produced by R13C3 and later.

12 Data section: “”

The empty data section was introduced in R18. This section contains no data.

Section property	Value
Name	“”
Section ID	Always 0
Compressed	2
Page size	0x7400
Encrypted	0

13 Data section AcDb:SummaryInfo Section

Section property	Value
Name	AcDb:SummaryInfo
Compressed	1
Encrypted	0 if not encrypted, 1 if encrypted.
Page size	0x100

This section contains summary information about the drawing. Strings are encoded as a 16-bit length, followed by the character bytes (0-terminated).

Type	Length	Description
String	2 + n	Title
String	2 + n	Subject
String	2 + n	Author
String	2 + n	Keywords
String	2 + n	Comments
String	2 + n	Last saved by
String	2 + n	Revision number
String	2 + n	Hyperlink base
?	8	Total editing time (ODA writes two zero Int32's)
Julian date	8	Create date time
Julian date	8	Modified date time
Int16	2 + 2 * (2 +	Property count, followed by PropertyCount key/value string pairs.

	n)	
Int32	4	Unknown (write 0)
Int32	4	Unknown (write 0)

14 Data section AcDb:Preview

14.1 PRE-R13C3

Section property	Value
Name	AcDb:Preview
Compressed	1
Encrypted	0 if not encrypted, 1 if encrypted.
Page size	If a thumbnail image is present, then header + image data size + sentinels and size info (0x40 bytes) + section alignment padding If no thumbnail image is present, the value is 0x400.

The BMP (or, sometimes, WMF) image of this file, if any. Only stored here for pre-R13C3 files. Later files place the data at the end. The format of this data is discussed in the section illustrating where R13C4 and beyond store it.

14.2 R13C3 AND LATER

Start sentinel

```
{0x1F,0x25,0x6D,0x07,0xD4,0x36,0x28,0x28,0x9D,0x57,0xCA,0x3F,0x9D,0x44,0x10,0x2B }
       overall size         RL            overall size of image area
       imagespresent        RC            counter indicating what is present here
Repeat imagespresent times {
       Code                 RC            code indicating what follows
  if (code==1) {
       header data start    RL            start of header data
       header data size     RL            size of header data
  }
  if (code == 2) {
       start of bmp         RL            start of bmp data
       size of bmp          RL            size of bmp data
  }
  if (code == 3) {
        start of wmf            RL              start of wmf data
        size of wmf             RL              size of wmf data
  }
}
if (bmpdata is present) {
        bmp data                RC              (there are "size of bmp" bytes of data)
}
if (wmfdata is present) {
        wmf data                RC              (there are "size of wmf" bytes of data)
}
end sentinel
0xE0,0xDA,0x92,0xF8,0x2B,0xc9,0xD7,0xD7,0x62,0xA8,0x35,0xC0,0x62,0xBB,0xEF,0xD4 };
```

15 Data section AcDb:VBAProject Section

The VBA project section is optional.

Section property	Value
Name	AcDb:VBAProject
Compressed	1
Encrypted	2 (meaning unknown).
Page size	Project data size + 0x80 + section alignment padding

The contents are currently unknown.The ODA reads and writes the contents of this section as is:

Type	Length	Description
byte	16	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x19, 0x00, 0x00, 0x00
byte	n	The VBA project data
Int32	4	0

16 Data section AcDb:AppInfo

Contains information about the application that wrote the .dwg file. This section is optional.

Section property	Value
Name	AcDb:AppInfo
Compressed	1
Encrypted	0
Page size	0x80

The AppInfo format depends on the application version (Acad version that wrote the file) in the file header. So a R18 .dwg file might have an R21 AppInfo section.

16.1 R18

In R18 the app info section consists of the following fields. Strings are encoded as a 16-bit length, followed by the character bytes (0-terminated).

Type	Length	Description
String	2 + n	App info name, ODA writes “AppInfoDataList”
UInt32	4	Unknown, ODA writes 2
String	2 + n	Unknown, ODA writes “4001”
String	2 + n	App info product XML element, e.g. ODA writes “<ProductInformation name ="Teigha" build_version="0.0" registry_version="3.3" install_id_string="ODA" registry_localeID="1033"/>"
String	2 + n	App info version, e.g. ODA writes "2.7.2.0".

16.2 R21-27

In R21 (and also R24, R27) the app info section consists of the following fields. Strings are encoded as a 16-bit length, followed by the character bytes (0-terminated), using unicode encoding (2 bytes per character).

Type	Length	Description
UInt32	4	Unknown (ODA writes 2)
String	2 + 2 * n + 2	App info name, ODA writes “AppInfoDataList”
UInt32	4	Unknown (ODA writes 3)
Byte[]	16	Version data (checksum, ODA writes zeroes)
String	2 + 2 * n + 2	Version
Byte[]	16	Comment data (checksum, ODA writes zeroes)
String	2 + 2 * n + 2	Comment
Byte[]	16	Product data (checksum, ODA writes zeroes)
String	2 + 2 * n + 2	Product
String	2 + n	App info version, e.g. ODA writes "2.7.2.0".

17 Data section AcDb:FileDepList

Contains file dependencies (e.g. IMAGE files, or fonts used by STYLE).

Section property	Value
Name	AcDb:FileDepList
Compressed	1
Encrypted	2 (meaning unknown)
Page size	0x80 if number of entries is 0 or 1. If more than 1, then 0x80 x number of entries.

In R18 the app info section consists of the following fields. Strings are encoded as a 32-bit length, followed by the character bytes (without trailing 0).

Type	Length	Description
Int32	4	Feature count (ftc)
String32	ftc * (4 + n)	Feature name list. A feature name is one of the following: “Acad:XRef” (for block table record) “Acad:Image” (for image definition) “Acad:PlotConfig” (for plotsetting) “Acad:Text” (for text style)
Int32	4	File count

Then follows an array of features (repeated file count times). The feature name + the full filename constitute the lookup key of a file dependency:

Type	Length	Description
String32	4 + n	Full filename
String32	4 + n	Found path, path at which file was found

String32	4 + n	Fingerprint GUID (applies to xref's only)
String32	4 + n	Version GUID (applies to xref's only)
Int32	4	Feature index in the feature list above.
Int32	4	Timestamp (Seconds since 1/1/1980)
Int32	4	Filesize
Int16	2	Affects graphics (1 = true, 0 = false)
Int32	4	Reference count

18 Data section AcDb:RevHistory

Section property	Value
Name	AcDb:RevHistory
Compressed	2
Encrypted	0
Page size	0x7400

The contents of this section are unknown. In the following paragraphs is described what the ODA writes in this section.

18.1 R18

Type	Length	Description
UInt32	4	Unknown (ODA writes 0)
UInt32	4	Unknown (ODA writes 0)
UInt32	4	Unknown (ODA writes 0)

More unknown bytes may follow.

18.2 R21

Type	Length	Description
UInt32	4	Unknown (ODA writes 0)
UInt32	4	Unknown (ODA writes 0)
UInt32	4	Unknown (ODA writes 1)
UInt32	4	Unknown (ODA writes 0)

More unknown bytes may follow.

19 Data section AcDb:Security

Section property	Value
Name	AcDb:Security
Compressed	1
Encrypted	0
Page size	0x7400

This section was introduced in R18. The AcDb:Security section is optional in the file—it is present if the file was saved with a password.

R18: The section is present in the file if the SecurityType entry at location 0x18 in the file is greater than 0.

Strings are prefixed with a 32-bit length (not zero terminated).

Type	Length	Description
Int32	4	Unknown (ODA writes 0x0c)
Int32	4	Unknown (ODA writes 0x0)
Int32	4	Unknown (ODA writes 0xabcdabcd)
UInt32	4	Cryptographic provider ID
String32	4 + n	Croptographic provider name
UInt32	4	Algorithm ID
UInt32	4	Encryption key length
Int32	4	Buffer size of following buffer
Byte[]	n	Encrypted string "SamirBajajSamirB"

Using the indicated provider and algorithm (and password obtained from the client for this drawing), the encryption password can be verified by decrypting the Test Encrypted Sequence. If the result is

"SamirBajajSamirB" (0x53, 0x61, 0x6d, 0x69, 0x72, 0x42, 0x61, 0x6a, 0x61, 0x6a, 0x53, 0x61, 0x6d, 0x69, 0x72, 0x42), then the password is correct.

The algorithm is RC4 (this is a symmetric encryption algorithm). The algorithm is used in DWG file format version 2004 and 2007.

Parameters are:

- Password (provided by user).

- Provider id: e.g. 0x0d.

- Provider name: e.g. "Microsoft Base DSS and Diffie-Hellman Cryptographic Provider".

- Key length: default value is 40.

- Flags: no salt.

The password bytes (convert unicode password string to bytes, 2 bytes per character) are hashed (using MD5). A session key is derived from the password hash (using no salt). This session key is then used for both encryption and decryption.

20 Data section AcDb:AcDbObjects

Section property	Value
Name	AcDb:AcDbObjects
Compressed	2
Encrypted	0 if not encrypted, 1 if encrypted
Page size	0x7400

This region holds the actual objects in the drawing. These can be entities, table entries, dictionary entries, and objects. This second use of objects is somewhat confusing; all items stored in the file are "objects", but only some of them are object objects. Others are entities, table entries, etc. The objects in this section can appear in any order.

Not all objects present in the file are actually used. All used objects can eventually be traced back to handle references in the Header section. So the proper way to read a file is to start reading the header and then tracing all references from there until all references have been followed. Very occasionally a file contains e.g. two APPID objects with the same name, of which one is used, and the other is not. Reading both would be incorrect due to a name clash. To complicate matters more, files also exist with table records with duplicate names. This is incorrect, and the software should rename the record to be unique upon reading.

For R18 and later the section data (right after the page header) starts with a RL value of 0x0dca (meaning unknown).

20.1 Common non-entity object format

Objects (non-entities) have the following general format:

Version	**Field type**	**DXF grou p**	**Description**
	MS		Size in bytes of object, not including the CRC
R2010+			
	MC		Size in bits of the handle stream (unsigned, 0x40 is not interpreted as sign). This includes the padding bits at the end of the handle stream (the padding bits make sure the object stream ends on a byte boundary).
Commmon			
	OT		Object type
R2000-R2007			
	RL		Size of object data in bits (number of bits before the handles), or the "endbit" of the pre-handles section.
Common:			

	H	5	Object's handle
	BS		Size of extended object data, if any
	X		Extended object data, if any. See EED section, chapter 28.
R13-R14			
	RL		Size of object data in bits
	BL		Number of persistent reactors attached to this object
R2004+			
	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
R2013+			
	B		Indicates whether the object has associated binary data in the data store section (see chapter 24 for more details about this section).
Common			
	X		Object data (varies by type of object)
R2007+			
	X		String data (optional)
	B		String stream present bit (last bit in pre-handles section). If 1, then the "endbit" location should be decremented by 16 bytes, and a short should be read at location endbit – 128 (bits), call this short strDataSize. If this short has the 0x8000 bit set, then decrement endbit by an additional 16 bytes, strip the 0x8000 bit off of strDataSize, and read the short at this new location, calling it hiSize. Then set strDataSize to (strDataSize | (hiSize << 15)). "endbit" should then be decremented by this final strDataSize value, and this bit location marks the start of the "string stream" within this object. All unicode strings in this object are located in the "string stream", and should be read from this stream, even though the location of the TV type fields in the object descriptions list these fields in among the normal object data.
Common			
			Below begins the handles stream, this begins at offset specified by number of bits before handles above
	H		Parent handle (soft pointer)
	H		[Reactors (soft pointer)], repeated as many times as specified by the number of persistent reactors
	H		xdictionary (hard owner), present if the has xdictionary flag is true
	X		Object specific handles
	B*		Padding bits are added until the next byte boundary is reached.
	RS		CRC

The CRC includes the size bytes.

20.2 Common entity format

Drawing entities, which are of course objects, have the same format as objects, with some additional standard items:

```
MS  : Size of object, not including the CRC
```

```
R2010+:
         MC  : Size in bits of the handle stream (unsigned, 0x40 is not interpreted as sign).
Commmon:
         OT  : Object type
R2000+ Only:
         RL  : Size of object data in bits
Common:
          H  : Object’s handle
         BS  : Size of extended object data, if any
          X  : Extended object data, if any
          B  : Flag indicating presence of graphic image.
               if (graphicimageflag is 1) {
R13-R007:
                 RL: Size of graphic image in bytes
R2010+:
                 BLL: Size of graphic image in bytes
Common:
                  X: The graphic image
               }
R13-R14 Only:
         RL  : Size of object data in bits
         6B  : Flags
         6B  : Common parameters
R2000+ Only:
          B  : 0 if the previous and next linkers are present; 1 if they are BOTH defaults (1
               back and 1 forward).
        ENC  : Entity color
         BD  : Linetype Scale
         BB  : Line type flags
               00 – BYLAYER linetype
               01 – BYBLOCK linetype
               10 – CONTINUOUS linetype
               11 – Indicates that a linetype handle will be stored in the handles section of the
               entity.
         BB  : Plotstyle flags:
               00 – BYLAYER plotstyle
               01 – BYBLOCK plotstyle
               10 – CONTINUOUS plotstyle
               11 – Indicates that a plotstyle handle will be stored in the handles section of
               the entity.
R2007+:
```

```
         BB  : Material flags:
               00 – BYLAYER material
               01 – BYBLOCK material
               10 – global material?
               11 – Indicates that a material handle will be stored in the handles section of the
               entity.
         RC  : Shadow flags
R2010+:
          B  : Has full visual style
          B  : Has face visual style
          B  : Has edge visual style
Common:
         BS  : Invisible flag (bit 0: 0 = visible, 1 = invisible)
R2000+:
         RC  : Entity lineweight flag
Common:
          X  : Object data (varies by type of object)
          X  : Handles associated with this object
         B*  : Padding bits are added until the next byte boundary is reached.
         RS  : CRC
```

The R13-R14 FLAGS area (6 bits) indicates which handle references are present in the HANDLE REFS area. They are as follows:

FEDCBA

```
         FE  : Entity mode (entmode).  Generally, this indicates whether or not the owner
               relative handle reference is present.  The values go as follows:
               00 : The owner relative handle reference is present.
                    Applies to the following:
                       VERTEX, ATTRIB, and SEQEND.
                       BLOCK, ENDBLK, and the defining entities in all
                       block defs except *MODEL_SPACE and *PAPER_SPACE.
               01 : PSPACE entity without a owner relative handle ref.
               10 : MSPACE entity without a owner relative handle ref.
               11 : Not used.

         DC  : This is the number of reactors attached to an entity as a bitshort. This feature
               may have been dormant in R13, but it appears in R14, and in files saved as R13 by
               R14.
          B  : 0 if a linetype reference is present; 1 if it's not (the default being BYLAYER --
               even though there IS a BYLAYER linetype entity and it has a handle).
          A  : 0 if the previous and next linkers are present; 1 if they are BOTH defaults (1
               back and 1 forward).
```

The COMMON PARAMETERS (6 bits):

CCSSII

```
CC  :  Color bitshort
SS  :  Linetype scale bitdouble
II  :  "Invisible" flag bitshort (bit 0: 0 = visible, 1 = invisible).
```

The ENTITY-SPECIFIC PARAMETERS area is coded with bitcodes. Each entity has its own parameter prescription. Some parameters ALWAYS appear in raw form -- even if bitcode abbreviations could be used (the 10 and 11 points in TEXT, for example). Generally the raw form is used in conditions wherein it cannot reasonably be assumed that the likely value for the particular parameter is one of the compressible values.

One method for loading these objects is to follow the object map. Doing so will cause each object to be loaded once and only once. Alternatively one can try to scan the objects as they are found, and replace objects with duplicated object handles with the ones found later in the file. The Teigha Classic for .dwg files Toolkit uses a hybrid approach, loading the control objects first, then the objects they contain.

20.3 Object types

Some object types have fixed values, others have values which vary with the drawing. Here are the fixed values:

UNUSED	0	RAY	0x28
TEXT	1	XLINE	0x29
ATTRIB	2	DICTIONARY	0x2A
ATTDEF	3	OLEFRAME	0x2B
BLOCK	4	MTEXT	0x2C
ENDBLK	5	LEADER	0x2D
SEQEND	6	TOLERANCE	0x2E
INSERT	7	MLINE	0x2F
MINSERT	8	BLOCK CONTROL OBJ	0x30
	9	BLOCK HEADER	0x31
VERTEX (2D)	0x0A	LAYER CONTROL OBJ	0x32
VERTEX (3D)	0x0B	LAYER	0x33
VERTEX (MESH)	0x0C	STYLE CONTROL OBJ	0x34
VERTEX (PFACE)	0x0D	STYLE	0x35
VERTEX (PFACE FACE)	0x0E		0x36
POLYLINE (2D)	0x0F		0x37
POLYLINE (3D)	0x10	LTYPE CONTROL OBJ	0x38
ARC	0x11	LTYPE	0x39

CIRCLE	0x12		0x3A
LINE	0x13		0x3B
DIMENSION (ORDINATE)	0x14	VIEW CONTROL OBJ	0x3C
DIMENSION (LINEAR)	0x15	VIEW	0x3D
DIMENSION (ALIGNED)	0x16	UCS CONTROL OBJ	0x3E
DIMENSION (ANG 3-Pt)	0x17	UCS	0x3F
DIMENSION (ANG 2-Ln)	0x18	VPORT CONTROL OBJ	0x40
DIMENSION (RADIUS)	0x19	VPORT	0x41
DIMENSION (DIAMETER)	0x1A	APPID CONTROL OBJ	0x42
POINT	0x1B	APPID	0x43
3DFACE	0x1C	DIMSTYLE CONTROL OBJ	0x44
POLYLINE (PFACE)	0x1D	DIMSTYLE	0x45
POLYLINE (MESH)	0x1E	VP ENT HDR CTRL OBJ	0x46
SOLID	0x1F	VP ENT HDR	0x47
TRACE	0x20	GROUP	0x48
SHAPE	0x21	MLINESTYLE	0x49
VIEWPORT	0x22	OLE2FRAME	0x4A
ELLIPSE	0x23	(DUMMY)	0x4B
SPLINE	0x24	LONG_TRANSACTION	0x4C
REGION	0x25	LWPOLYLINE	0x4D
3DSOLID	0x26	HATCH	0x4E
BODY	0x27	XRECORD	0x4F
		ACDBPLACEHOLDER	0x50
		VBA_PROJECT	0x51
		LAYOUT	0x52
		ACAD_PROXY_ENTITY	0x1f2
		ACAD_PROXY_OBJECT	0x1f3

There are a number of objects with non-fixed values. These are:

```
ACAD_TABLE
CELLSTYLEMAP
DBCOLOR
DICTIONARYVAR
DICTIONARYWDFLT
FIELD
GROUP
HATCH
IDBUFFER
IMAGE
IMAGEDEF
IMAGEDEFREACTOR
LAYER_INDEX
LAYOUT
LWPLINE
MATERIAL
```

```
MLEADER
MLEADERSTYLE
OLE2FRAME
PLACEHOLDER
PLOTSETTINGS
RASTERVARIABLES
SCALE
SORTENTSTABLE
SPATIAL_FILTER
SPATIAL_INDEX
TABLEGEOMETRY
TABLESTYLES
VBA_PROJECT
VISUALSTYLE
WIPEOUTVARIABLE
XRECORD
```

For objects with non-fixed values, taking the object type minus 500 gives an index into the class list, which then determines the type of object. For instance, an object type of 501 means that this object is of the class which is second in the class list; the **classdxfname** field determines the type of the object.

See the sections on EED a description of that areas.

20.4 OBJECT PRESCRIPTIONS

The object prescriptions are given in the following form:

```
ITEM    TYPE-CODE    DXF-CODE    DESCRIPTION
```

See the top of this document for the key to the data types used here.

20.4.1 Common Entity Data

The following data appears at the beginning of each entity in the file, and will be referred to as Common Entity Data in the subsequent entity descriptions.

```
       Length                MS    ---     Entity length (not counting itself or CRC).
       Type                  BS      0     1 (internal DWG type code).
R2000+ Only:
       Obj size              RL            size of object in bits, not including end handles
Common:
       Handle                 H      5     code 0, length followed by the handle bytes.
       EED size              BS            size of extended entity data, if any
       EED                    X     -3     See EED section.
       Graphic present Flag   B            1 if a graphic is present
       Graphics               X            if graphicpresentflag is 1, the graphic goes here.
```

			See the section on Proxy Entity Graphics for the format of this section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Entmode	BB		entity mode
Numreactors	BL		number of persistent reactors attached to this object
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
R2013+:			
Has DS binary data	B		If 1 then this object has associated binary data stored in the data store. See for more details chapter 24.
R13-R14 Only:			
Isbylayerlt	B		1 if bylayer linetype, else 0
Common:			
Nolinks	B		1 if major links are assumed +1, -1, else 0 For R2004+ this always has value 1 (links are not used)
Color	CMC(B)	62	
Ltype scale	BD	48	
R2000+:			
Ltype flags	BB		00 = bylayer, 01 = byblock, 10 = continous, 11 = linetype handle present at end of object
Plotstyle flags	BB		00 = bylayer, 01 = byblock, 11 = plotstyle handle present at end of object
R2007+:			
Material flags	BB		00 = bylayer, 01 = byblock, 11 = material handle present at end of object
Shadow flags	RC		
Common:			
Invisibility	BS	60	
R2000+:			
Lineweight	RC	370	

20.4.2 Common Entity Handle Data

The following data appears in the handles section of each entity, and will be referred to as Common Entity Handle Data in the subsequent entity descriptions.

Handle refs

```
              [Owner ref handle (soft pointer)]
              [Reactors (soft pointer)]
              xdicobjhandle  (hard owner)
R13-R14 Only:
           8  LAYER (hard pointer)
           6  [LTYPE (hard pointer)] (present if Isbylayerlt is 0)
R13-R2000 Only:
              previous/next handles present if Nolinks is 0
              [PREVIOUS ENTITY (relative soft pointer)]
              [NEXT ENTITY (relative soft pointer)]
R2004+:
              [Color book color handle (hard pointer)]
R2000+ Only:
           8  LAYER (hard pointer)
           6  [LTYPE (hard pointer)] present if linetype flags
              were 11
R2007+:
              MATERIAL present if material flags were 11
R2000+:
              PLOTSTYLE (hard pointer) present if plotstyle flags
              were 11
R2010+:
              If has full visual style, the full visual style handle (hard pointer).
              If has face visual style, the face visual style handle (hard pointer).
              If has edge visual style, the full visual style handle (hard pointer).
```

20.4.3 TEXT (1)

```
       Common Entity Data
R13-14 Only:
       Elevation             BD    ---
       Insertion pt         2RD     10
       Alignment pt         2RD     11
       Extrusion            3BD    210
       Thickness             BD     39
       Oblique ang           BD     51
       Rotation ang          BD     50
       Height                BD     40
       Width factor          BD     41
       Text value            TV      1
```

Generation	BS	71	
Horiz align.	BS	72	
Vert align.	BS	73	
R2000+ Only:			
DataFlags	RC		Used to determine presence of subsquent data
Elevation	RD	---	present if !(DataFlags & 0x01)
Insertion pt	2RD	10	
Alignment pt	2DD	11	present if !(DataFlags & 0x02), use 10 & 20 values for 2 default values.
Extrusion	BE	210	
Thickness	BT	39	
Oblique ang	RD	51	present if !(DataFlags & 0x04)
Rotation ang	RD	50	present if !(DataFlags & 0x08)
Height	RD	40	
Width factor	RD	41	present if !(DataFlags & 0x10)
Text value	TV	1	
Generation	BS	71	present if !(DataFlags & 0x20)
Horiz align.	BS	72	present if !(DataFlags & 0x40)
Vert align.	BS	73	present if !(DataFlags & 0x80)
Common:			
Common Entity Handle Data			
	H	7	STYLE (hard pointer)
CRC	X	---	

20.4.3.1 R14 Example:

```
OBJECT: text (1H), len 49H (73), handle: 4C    00559 49 00                     I.        0100 1001 0000 0000
    0055B 40 40 53 20 58 10 00 05   @@S X...   0100 0000 0100 0000 0101 0011 0010 0000 0101 1000 0001 0000 0000 0000 0000 0101
    00563 5B 40 00 00 00 00 00 01   [@......   0101 1011 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001
    0056B 08 00 00 00 00 00 00 02   ........   0000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010
    00573 08 00 00 00 00 00 00 00   ........   0000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    0057B 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    00583 00 14 D4 4D 4C CC CC CC   ...ML...   0000 0000 0001 0100 1101 0100 0100 1101 0100 1100 1100 1100 1100 1100 1100 1100
    0058B CC E4 9F A8 63 A3 43 4B   ....c.CK   1100 1100 1110 0100 1001 1111 1010 1000 0110 0011 1010 0011 0100 0011 0100 1011
    00593 99 03 4B 99 03 A3 2B C3   ..K...+.   1001 1001 0000 0011 0100 1011 1001 1001 0000 0011 1010 0011 0010 1011 1100 0011
    0059B A5 46 0A 21 E8 08 0A 22   .F.!..."   1010 0101 0100 0110 0000 1010 0010 0001 1110 1000 0000 1000 0000 1010 0010 0010
    005A3 00                        .          0000 0000
    005A4 C9 72                     crc
ENDOBJECT
```

20.4.4 ATTRIB (2)

Common TEXT Entity Data			
R2010+:			
Version	RC	?	
R2018+:			

```
       Attribute type        RC      71     1 = Single line,
                                            2 = Multi line (ATTRIB),
                                            4 = Multie line (ATTDEF)
IF Attribute type is multi line
       MTEXT fields           …             Here all fields of an embedded MTEXT object
                                            are written, starting from the Entmode
                                            (entity mode). The owner handle can be 0.

                                            For DXF this is marked with <101, Embedded Object>,
                                            Followed by the MTEXT fields, starting with the
                                            Insertion point (group 10).
       Annotative data size  BS
IF Annotative data size greater than zero
       Annotative data bytes RC             Byte array with length Annotative data size.
       Registered application H             Hard pointer.
       Unknown               BS     72?     Value 0.
END IF
       Tag string            VT       2
       Unknown               BS     73?     Value 0
       Flags                 RC      70     0 = None,
                                            1 = Invisible,
                                            2 = Constant,
                                            4 = Input verification required,
                                            8 = Preset
       Lock position          B     280
END IF
Common:
IF Attribute type is single line
       Tag                   TV       2
       Field length          BS      73     unused
       Flags                 RC      70     NOT bit-pair-coded.
R2007+:
       Lock position flag     B     280
END IF (single line)

Common:
       CRC                    X     ---
```

20.4.4.1 R14 Example:

```
OBJECT: attrib (2H), len 58H (88), handle: 52
```

```
    00614 58 00                    X.        0101 1000 0000 0000
    00616 40 80 54 A3 F8 10 00 01   @.T.....  0100 0000 1000 0000 0101 0100 1010 0011 1111 1000 0001 0000 0000 0000 0000 0001
    0061E 5B 40 00 00 00 00 00 02   [@......  0101 1011 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010
    00626 88 00 00 00 00 00 00 03   ........  1000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0011
    0062E 88 00 00 00 00 00 00 00   ........  1000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    00636 00 00 00 00 00 00 00 00   ........  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    0063E 00 14 D4 4D CC CC CC CC   ...M....  0000 0000 0001 0100 1101 0100 0100 1101 1100 1100 1100 1100 1100 1100 1100 1100
    00646 CC E4 9F 9F FF FF FF FF   ........  1100 1100 1110 0100 1001 1111 1001 1111 1111 1111 1111 1111 1111 1111 1111 1111
    0064E FF FD E7 E8 5B 6B CB 0B   ....[k..  1111 1111 1111 1101 1110 0111 1110 1000 0101 1011 0110 1011 1100 1011 0000 1011
    00656 A3 A1 03 B3 0B 63 AB 2D   .....c.-  1010 0011 1010 0001 0000 0011 1011 0011 0000 1011 0110 0011 1010 1011 0010 1101
    0065E 48 2A 6A CA 0A A2 A4 01   H*j.....  0100 1000 0010 1010 0110 1010 1100 1010 0000 1010 1010 0010 1010 0100 0000 0001
    00666 00 60 A2 1E 80 80 A2 21   .`.....!  0000 0000 0110 0000 1010 0010 0001 1110 1000 0000 1000 0000 1010 0010 0010 0001

    0066E 6F A6                    crc
```

20.4.5 ATTDEF (3)

```
        Common ATTRIB Entity Data
R2010+:
        Version                  RC       ?
Common:
        Prompt                   TV       3


        CRC                       X     ---
```

20.4.5.1 *R14 Example:*

```
spec3.dwg

OBJECT: attdef (3H), len 50H (80), handle: 4C

    00559 50 00                    P.        0101 0000 0000 0000

    0055B 40 C0 53 22 08 10 00 05   @.S"....  0100 0000 1100 0000 0101 0011 0010 0010 0000 1000 0001 0000 0000 0000 0000 0101

    00563 5B 40 00 00 00 00 00 01   [@......  0101 1011 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001

    0056B 08 00 00 00 00 00 00 02   ........  0000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010

    00573 08 00 00 00 00 00 00 00   ........  0000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    0057B 00 00 00 00 00 00 00 00   ........  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    00583 00 14 D4 4D 4C CC CC CC   ...ML...  0000 0000 0001 0100 1101 0100 0100 1101 0100 1100 1100 1100 1100 1100 1100 1100

    0058B CC E4 9F B5 48 2A 6A CA   ....H*j.  1100 1100 1110 0100 1001 1111 1011 0101 0100 1000 0010 1010 0110 1010 1100 1010

    00593 0A A2 A4 00 85 A2 B7 3A   .......:  0000 1010 1010 0010 1010 0100 0000 0000 1000 0101 1010 0010 1011 0111 0011 1010

    0059B 32 B9 10 36 BC B0 BA 3A   2..6...:  0011 0010 1011 1001 0001 0000 0011 0110 1011 1100 1011 0000 1011 1010 0011 1010

    005A3 18 28 87 A0 20 28 88 00   .(.. (..  0001 1000 0010 1000 1000 0111 1010 0000 0010 0000 0010 1000 1000 1000 0000 0000

    005AB 78 53                    crc
```

20.4.6 BLOCK (4)

```
        Common Entity Data
        Block name             TV       2
        Common Entity Handle Data
        CRC                     X     ---
```

20.4.6.1 Example:

```
OBJECT: block (4H), len 16H (22), handle: 4E
    00BC2 16 00                    ..         0001 0110 0000 0000
    00BC4 41 00 53 A3 D8 00 00 01  A.S.....   0100 0001 0000 0000 0101 0011 1010 0011 1101 1000 0000 0000 0000 0000 0000 0001
    00BCC 5B 20 A9 AB 28 49 89 70  [ ..(I.p   0101 1011 0010 0000 1010 1001 1010 1011 0010 1000 0100 1001 1000 1001 0111 0000
    00BD4 06 0A 21 E8 08 00        ..!...     0000 0110 0000 1010 0010 0001 1110 1000 0000 1000 0000 0000
    00BDA 39 F3                    crc
```

NOTES: The BLOCK_RECORD entity seems to have all the goodies that show up in a BLOCK entget -- except for the common parameters. The actual BLOCK entity seems to be almost a dummy.

20.4.7 ENDBLK (5)

```
        Common Entity Data
        Common Entity Handle Data
        CRC                     X     ---
```

20.4.7.1 Example:

```
OBJECT: endblk (5H), len FH (15), handle: 1B
    00685 0F 00                    ..         0000 1111 0000 0000
    00687 41 40 46 E2 48 00 00 05  A@F.H...   0100 0001 0100 0000 0100 0110 1110 0010 0100 1000 0000 0000 0000 0000 0000 0101
    0068F 5B 18 28 87 A0 20 20     [.(..      0101 1011 0001 1000 0010 1000 1000 0111 1010 0000 0010 0000 0010 0000
    00696 2E 8B                    crc
```

20.4.8 SEQEND (6)

```
        Common Entity Data
        Common Entity Handle Data
        CRC                     X     ---
```

20.4.8.1 Example:

```
OBJECT: seqend (6H), len 11H (17), handle: 53
    00670 11 00                    ..         0001 0001 0000 0000
    00672 41 80 54 E2 48 00 00 01  A.T.H...   0100 0001 1000 0000 0101 0100 1110 0010 0100 1000 0000 0000 0000 0000 0000 0001
    0067A 5B 60 81 18 28 87 A0 20  [`..(..    0101 1011 0110 0000 1000 0001 0001 1000 0010 1000 1000 0111 1010 0000 0010 0000
    00682 08                       .          0000 1000
    00683 88 C7                    crc
```

20.4.9 INSERT (7)

```
        Common Entity Data
        Ins pt                3BD      10
R13-R14 Only:
        X Scale                BD      41
        Y Scale                BD      42
```

Z Scale	BD	43	
R2000+ Only:			
Data flags	BB		
Scale Data			Varies with Data flags: 11 - scale is (1.0, 1.0, 1.0), no data stored. 01 – 41 value is 1.0, 2 DD’s are present, each using 1.0 as the default value, representing the 42 and 43 values. 10 – 41 value stored as a RD, and 42 & 43 values are not stored, assumed equal to 41 value. 00 – 41 value stored as a RD, followed by a 42 value stored as DD (use 41 for default value), and a 43 value stored as a DD (use 41 value for default value).
Common:			
Rotation	BD	50	
Extrusion	3BD	210	
Has ATTRIBs	B	66	Single bit; 1 if ATTRIBs follow.
R2004+:			
Owned Object Count	BL		Number of objects owned by this object.
Common:			
Common Entity Handle Data			
	H	2	BLOCK HEADER (hard pointer)
R13-R200:			
	H		[1st ATTRIB (soft pointer)] if 66 bit set; can be NULL
	H		[last ATTRIB](soft pointer)] if 66 bit set; can be NULL
R2004:			
	H		[ATTRIB (hard owner)] Repeats “Owned Object Count” times.
Common:			
	H		[SEQEND (hard owner)] if 66 bit set
CRC	X	---	

20.4.9.1 R14 Example:

```
OBJECT: insert (7H), len 29H (41), handle: 51
    005E7 29 00                    ).        0010 1001 0000 0000    005E9 41 C0 54 66 F0 00 00 05   A.Tf....   0100 0001 1100 0000 0101
0100 0110 0110 1111 0000 0000 0000 0000 0000 0000 0101
    005F1 5B 00 00 00 00 00 00 01   [.......   0101 1011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001
    005F9 08 00 00 00 00 00 00 00   ........   0000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    00601 82 04 AD 4C C1 44 3D 01   ...L.D=.   1000 0010 0000 0100 1010 1101 0100 1100 1100 0001 0100 0100 0011 1101 0000 0001
    00609 01 45 35 05 49 05 48 C5   .E5.I.H.   0000 0001 0100 0101 0011 0101 0000 0101 0100 1001 0000 0101 0100 1000 1100 0101
    00611 4C                       L         0100 1100
    00612 CB 54                    crc
```

20.4.10 MINSERT (8)

```
        Common Entity Data
        Ins pt              3BD     10
R13-R14 Only:
        X Scale              BD     41
        Y Scale              BD     42
        Z Scale              BD     43
R2000+ Only:
        Data flags           BB
        Scale Data                        Varies with Data flags:
                                          11 - scale is (1.0, 1.0, 1.0), no data stored.
                                          01 – 41 value is 1.0, 2 DD’s are present, each using
                                          1.0 as the default value, representing the 42 and 43
                                          values.
                                          10 – 41 value stored as a RD, and 42 & 43 values are
                                          not stored, assumed equal to 41 value.
                                          00 – 41 value stored as a RD, followed by a 42 value
                                          stored as DD (use 41 for default value), and a 43
                                          value stored as a DD (use 41 value for default
                                          value).
Common:
        Rotation             BD     50
        Extrusion           3BD    210
        Has ATTRIBs           B     66    Single bit; 1 if ATTRIBs follow.
R2004+:
        Owned Object Count   BL           Number of objects owned by this object.
Common:
        Numcols              BS     70
        Numrows              BS     71
        Col spacing          BD     44
        Row spacing          BD     45
        Common Entity Handle Data
                              H      2    BLOCK HEADER (hard pointer)
R13-R2000:
                              H           [1st  ATTRIB (soft pointer)]  if 66 bit set; can be
                                          NULL
                              H           [last ATTRIB](soft pointer)]  if 66 bit set; can be
                                          NULL
R2004+:
                              H           [ATTRIB (soft pointer)] Repeats “Owned Object Count”
                                          times.
Common:
                              H           [SEQEND (hard owner)]       if 66 bit set
        CRC                       X     ---
```

20.4.10.1 R14 Example:

```
OBJECT: minsert (8H), len 36H (54), handle: 59
   0069E 36 00                   6.        0011 0110 0000 0000
   006A0 42 00 56 63 B0 08 00 05   B.Vc....   0100 0010 0000 0000 0101 0110 0110 0011 1011 0000 0000 1000 0000 0000 0000 0101
   006A8 5B 00 00 00 00 00 00 00   [.......   0101 1011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
   006B0 08 00 00 00 00 00 00 00   ........   0000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
   006B8 42 04 AD 49 04 40 C0 00   B..I.@..   0100 0010 0000 0100 1010 1101 0100 1001 0000 0100 0100 0000 1100 0000 0000 0000
   006C0 00 00 00 00 00 84 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 1000 0100 0000 0000 0000 0000
   006C8 00 00 00 00 00 01 00 C1   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 1100 0001
   006D0 44 3D 01 01 45 44         D=..ED     0100 0100 0011 1101 0000 0001 0000 0001 0100 0101 0100 0100
   006D6 84 2E                     crc
```

20.4.11 VERTEX (2D) (10)

```
        Common Entity Data
        Flags                    EC      70      NOT bit-pair-coded.
        Point                   3BD      10      NOTE THAT THE Z SEEMS TO ALWAYS BE 0.0! The Z must
                                                 be taken from the 2D POLYLINE elevation.
        Start width              BD      40      If it's negative, use the abs val for start AND end
                                                 widths (and note that no end width will be present).
                                                 This is a compression trick for cases where the
                                                 start and end widths are identical and non-0.
        End width                BD      41      Not present if the start width is < 0.0; see above.
        Bulge                    BD      42
R2010+:
        Vertex ID                BL      91
Common:
        Tangent dir              BD      50
        Common Entity Handle Data
        CRC                       X     ---
```

20.4.11.1 Example:

```
OBJECT: pline vert (AH), len 22H (34), handle: 4D

   00B39 22 00                   ".        0010 0010 0000 0000

   00B3B 42 80 53 66 F8 00 00 01   B.Sf....   0100 0010 1000 0000 0101 0011 0110 0110 1111 1000 0000 0000 0000 0000 0000 0001

   00B43 5B 00 00 00 00 00 00 00   [.......   0101 1011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   00B4B 00 08 00 00 00 00 00 00   ........   0000 0000 0000 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   00B53 00 02 05 55 00 60 A2 1E   ...U.`..   0000 0000 0000 0010 0000 0101 0101 0101 0000 0000 0110 0000 1010 0010 0001 1110

   00B5B 80 C1                     ..         1000 0000 1100 0001

   00B5D B2 FC                     crc
```

NOTES: Neither elevation nor thickness are present in the 2D VERTEX data. Both should be taken from the 2D POLYLINE entity (15).

20.4.12 VERTEX (3D) (11)

```
Common Entity Data
Flags                 EC      70      NOT bit-pair-coded.
Point                3BD      10
Common Entity Handle Data
CRC                    X     ---
```

20.4.12.1 Example:

```
OBJECT: 3d pline vert (BH), len 1AH (26), handle: 62

   00D74 1A 00                    ..        0001 1010 0000 0000

   00D76 42 C0 58 A4 B8 00 00 01   B.X.....   0100 0010 1100 0000 0101 1000 1010 0100 1011 1000 0000 0000 0000 0000 0000 0001

   00D7E 5B 10 00 00 00 00 00 00   [.......   0101 1011 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   00D86 00 08 0D 82 08 60 A2 1F   .....`..   0000 0000 0000 1000 0000 1101 1000 0010 0000 1000 0110 0000 1010 0010 0001 1111

   00D8E 00 80                    ..        0000 0000 1000 0000

   00D90 8C 03                    crc
```

20.4.13 VERTEX (MESH) (12)

Same as VERTEX (3D) (11) except for type code.

20.4.13.1 Example:

```
OBJECT: 3d surf sol vert (CH), len 21H (33), handle: 67

   00E36 21 00                    !.        0010 0001 0000 0000

   00E38 43 00 59 E6 B8 00 00 01   C.Y.....   0100 0011 0000 0000 0101 1001 1110 0110 1011 1000 0000 0000 0000 0000 0000 0001

   00E40 5B 20 19 1D 70 D1 7F E3   [ ..p...   0101 1011 0010 0000 0001 1001 0001 1101 0111 0000 1101 0001 0111 1111 1110 0011

   00E48 FF 47 E1 72 DB 05 A8 C4   .G.r....   1111 1111 0100 0111 1110 0001 0111 0010 1101 1011 0000 0101 1010 1000 1100 0100

   00E50 58 CA 05 00 60 A2 1E 80   X...`...   0101 1000 1100 1010 0000 0101 0000 0000 0110 0000 1010 0010 0001 1110 1000 0000

   00E58 C0                       .         1100 0000

   00E59 B3 50                    crc
```

20.4.14 VERTEX (PFACE) (13)

Same as VERTEX (3D) (11) except for type code.

R13 .dwg files seem to have color and linetype data for all PFACE VERTEXs (both types), but R12 and SAVEASR12 seem to omit color and linetype when writing out the location VERTEXs.

20.4.14.1 Example:

```
OBJECT: pface pt (DH), len 21H (33), handle: 56

   00BDD 21 00                    !.        0010 0001 0000 0000
   00BDF 43 40 55 A6 B8 00 00 01   C@U.....   0100 0011 0100 0000 0101 0101 1010 0110 1011 1000 0000 0000 0000 0000 0000 0001

   00BE7 5B 60 20 00 00 00 00 00   [` .....   0101 1011 0110 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   00BEF 00 02 00 00 00 00 00 00   ........   0000 0000 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   00BF7 00 10 81 00 60 A2 1E 80   ....`...   0000 0000 0001 0000 1000 0001 0000 0000 0110 0000 1010 0010 0001 1110 1000 0000

   00BFF C1                        .          1100 0001

   00C00 3D 1E                     crc
```

20.4.15 VERTEX (PFACE FACE) (14)

```
        Common Entity Data
        Vert index            BS      71      1-based vertex index (see DXF doc)
        Vert index            BS      72      1-based vertex index (see DXF doc)
        Vert index            BS      73      1-based vertex index (see DXF doc)
        Vert index            BS      74      1-based vertex index (see DXF doc)
        Common Entity Handle Data
        CRC                    X     ---
```

20.4.15.1 Example:

```
OBJECT: pface face def (EH), len 13H (19), handle: 5A

   00C7E 13 00                     ..         0001 0011 0000 0000

   00C80 43 80 56 A3 48 00 00 01   C.V.H...   0100 0011 1000 0000 0101 0110 1010 0011 0100 1000 0000 0000 0000 0000 0000 0001

   00C88 7B 20 28 1A 05 60 82 98   { (..`..   0111 1011 0010 0000 0010 1000 0001 1010 0000 0101 0110 0000 1000 0010 1001 1000

   00C90 28 87 80                  (..        0010 1000 1000 0111 1000 0000

   00C93 C3 BA                     crc
```

20.4.16 2D POLYLINE (15)

```
        Common Entity Data
        Flags                 BS      70
        Curve type            BS      75      Curve and smooth surface type.
        Start width           BD      40      Default start width
        End width             BD      41      Default end width
        Thickness             BT      39
        Elevation             BD      10      The 10-pt is (0,0,elev)
        Extrusion             BE     210
R2004+:
        Owned Object Count    BL              Number of objects owned by this object.
Common:
        Common Entity Handle Data
```

R13-R2000:			
	H		1st VERTEX (soft pointer)
	H		last VERTEX (soft pointer)
R2004+:			
	H		[VERTEX (hard owner)] Repeats "Owned Object Count" times.
Common:			
	H		SEQEND (hard owner)
CRC	X	---	

20.4.16.1 R14 Example:

```
OBJECT: pline st (FH), len 18H (24), handle: 4C

   00B1D 18 00                    ..        0001 1000 0000 0000

   00B1F 43 C0 53 22 D8 00 00 05   C.S"....   0100 0011 1100 0000 0101 0011 0010 0010 1101 1000 0000 0000 0000 0000 0000 0101

   00B27 5B 55 55 26 0A 21 E8 14   [UU&.!..   0101 1011 0101 0101 0101 0101 0010 0110 0000 1010 0010 0001 1110 1000 0001 0100

   00B2F 21 28 29 A8 29 E6 2A 01   !().).*.   0010 0001 0010 1000 0010 1001 1010 1000 0010 1001 1110 0110 0010 1010 0000 0001

   00B37 13 EA                    crc
```

20.4.17 3D POLYLINE (16)

Common Entity Data			
Flags	RC	70	NOT DIRECTLY THE 75. Bit-coded (76543210):
		75	0 : Splined (75 value is 5)
			1 : Splined (75 value is 6)
			(If either is set, set 70 bit 2 (4) to indicate splined.)
Flags	RC	70	NOT DIRECTLY THE 70. Bit-coded (76543210):
			0 : Closed (70 bit 0 (1))
			(Set 70 bit 3 (8) because this is a 3D POLYLINE.)
R2004+:			
Owned Object Count	BL		Number of objects owned by this object.
Common:			
Common Entity Handle Data			
R13-R2000:			
	H		first VERTEX (soft pointer)
	H		last VERTEX (soft pointer)
R2004+:			
	H		[VERTEX (hard owner)] Repeats "Owned Object Count" times.
Common:			
	H		SEQEND (hard owner)

CRC	X	---

20.4.17.1 Example:

```
OBJECT: 3d poly start (10H), len 19H (25), handle: 5E

    00CDA 19 00                    ..        0001 1001 0000 0000

    00CDC 44 00 57 A2 C8 00 00 05   D.W.....   0100 0100 0000 0000 0101 0111 1010 0010 1100 1000 0000 0000 0000 0000 0000 0101

    00CE4 5B 00 00 18 28 87 E0 84   [...(...   0101 1011 0000 0000 0000 0000 0001 1000 0010 1000 1000 0111 1110 0000 1000 0100

    00CEC D0 83 20 AF A0 B1 18 B1   .. .....   1101 0000 1000 0011 0010 0000 1010 1111 1010 0000 1011 0001 0001 1000 1011 0001

    00CF4 80                       .         1000 0000

    00CF5 4A A6                    crc
```

20.4.18 ARC (17)

Common Entity Data		
Center	3BD	10
Radius	BD	40
Thickness	BT	39
Extrusion	BE	210
Start angle	BD	50
End angle	BD	51
Common Entity Handle Data		
CRC	X	---

20.4.18.1 R14 Example:

```
OBJECT: arc (11H), len 3AH (58), handle: 64

    00DA7 3A 00                    :.        0011 1010 0000 0000

    00DA9 44 40 59 24 E8 08 00 05   D@Y$....   0100 0100 0100 0000 0101 1001 0010 0100 1110 1000 0000 1000 0000 0000 0000 0101

    00DB1 5B 0F 61 AA 41 EB F9 A0   [.a.A...   0101 1011 0000 1111 0110 0001 1010 1010 0100 0001 1110 1011 1111 1001 1010 0000

    00DB9 88 05 DD 50 53 3A 0A 70   ...PS:.p   1000 1000 0000 0101 1101 1101 0101 0000 0101 0011 0011 1010 0000 1010 0111 0000

    00DC1 EA 04 13 B4 FD AC 6D CB   ......m.   1110 1010 0000 0100 0001 0011 1011 0100 1111 1101 1010 1100 0110 1101 1100 1011

    00DC9 7A 9F D4 88 6D E1 F9 BC   z...m...   0111 1010 1001 1111 1101 0100 1000 1000 0110 1101 1110 0001 1111 1001 1011 1100

    00DD1 BC 60 08 00 27 5B 70 E5   .`..'[p.   1011 1100 0110 0000 0000 1000 0000 0000 0010 0111 0101 1011 0111 0000 1110 0101

    00DD9 02 68 7A 01 82 88 7E 08   .hz...~.   0000 0010 0110 1000 0111 1010 0000 0001 1000 0010 1000 1000 0111 1110 0000 1000

    00DE1 33 05                    3.        0011 0011 0000 0101

    00DE3 91 5F                    crc
```

20.4.19 ARC_DIMENSION

Class properties:

App name	ObjectDBX Classes
Class number	Dynamic (>= 500)
DWG version	R18
Maintenance version	0
Class proxy flags	0x401
C++ class name	AcDbArcDimension
DXF name	ARC_DIMENSION

The arc length dimension was introduced in AutoCAD 2004. The DXF format is slightly different from the other dimension entities. The entity type in DXF is `ARC_DIMENSION`, rather than `DIMENSION`.

```
 Common Entity Data
 Common Dimension Data                         See paragraph 20.4.22.
Common:
 Dim line arc point          3BD    10
 Extension line 1 point      3BD    13
 Extension line 2 point      3BD    14
 Arc center                  3BD    15
 Is partial?                   B    70
 Start angle (radians)        BD    40
 End angle (radians)          BD    41
 Has leader?                   B    71
 Leader point 1              3BD    16
 Leader point 2              3BD    17
 Common Entity Handle Data
                               H     3       DIMSTYLE (hard pointer)
                               H     2       anonymous BLOCK (hard pointer)
 CRC                           X   ---
```

20.4.20 CIRCLE (18)

```
 Common Entity Data
 Center              3BD      10
 Radius               BD      40
 Thickness            BT      39
 Extrusion            BE     210
 Common Entity Handle Data
 CRC                   X     ---
```

20.4.20.1 R14 Example:

```
OBJECT: circle (12H), len 2BH (43), handle: 92
```

```
   0154E 2B 00                   +.        0010 1011 0000 0000

   01550 44 80 64 A0 C8 08 00 05   D.d.....  0100 0100 1000 0000 0110 0100 1010 0000 1100 1000 0000 1000 0000 0000 0000 0101

   01558 5B 0A 88 A1 BF 90 3F C3   [.....?.  0101 1011 0000 1010 1000 1000 1010 0001 1011 1111 1001 0000 0011 1111 1100 0011

   01560 48 00 45 2D C2 C7 6F 28   H.E-..o(  0100 1000 0000 0000 0100 0101 0010 1101 1100 0010 1100 0111 0110 1111 0010 1000

   01568 FA 04 6A 9D CD 75 A2 1A   ..j..u..  1111 1010 0000 0100 0110 1010 1001 1101 1100 1101 0111 0101 1010 0010 0001 1010

   01570 72 9F D4 98 28 87 E0 96   r...(...  0111 0010 1001 1111 1101 0100 1001 1000 0010 1000 1000 0111 1110 0000 1001 0110

   01578 50 86 6D                P.m       0101 0000 1000 0110 0110 1101

   0157B 36 1C                   crc
```

20.4.21 LINE (19)

```
       Common Entity Data
R13-R14 Only:
       Start pt             3BD     10
       End   pt             3BD     11
R2000+:
       Z's are zero bit       B
       Start Point x         RD     10
       End Point x           DD     11     Use 10 value for default
       Start Point y         RD     20
       End Point y           DD     21     Use 20 value for default
       Start Point z         RD     30     Present only if "Z's are zero bit" is 0
       End Point z           DD     31     Present only if "Z's are zero bit" is 0, use 30
                                           value for default.

Common:
       Thickness             BT     39
       Extrusion             BE    210
       Common Entity Handle Data
       CRC                    X    ---
```

20.4.21.1 R14 Example:

```
OBJECT: line (13H), len 35H (53), handle: CC

   004A5 35 00                   5.        0011 0101 0000 0000

   004A7 44 C0 73 22 E8 08 00 01   D.s"....  0100 0100 1100 0000 0111 0011 0010 0010 1110 1000 0000 1000 0000 0000 0000 0001

   004AF 13 00 6B B5 95 B2 D9 24   ..k....$  0001 0011 0000 0000 0110 1011 1011 0101 1001 0101 1011 0010 1101 1001 0010 0100

   004B7 08 04 88 93 FD FD 9A 00   ........  0000 1000 0000 0100 1000 1000 1001 0011 1111 1101 1111 1101 1001 1010 0000 0000

   004BF FA 04 53 E6 F4 DB B6 B6   ..S.....  1111 1010 0000 0100 0101 0011 1110 0110 1111 0100 1101 1011 1011 0110 1011 0110
```

```
004C7 90 20 12 02 4F F7 F6 68   . ..O..h   1001 0000 0010 0000 0001 0010 0000 0010 0100 1111 1111 0111 1111 0110 0110 1000

004CF 03 E8 15 4E 08 11 82 88   ...N....   0000 0011 1110 1000 0001 0101 0100 1110 0000 1000 0001 0001 1000 0010 1000 1000

004D7 7A 88 9A 03 06            z....      0111 1010 1000 1000 1001 1010 0000 0011 0000 0110

004DC FA FE                     crc
```

20.4.22 COMMON DIMENSION DATA

R2010:

Version	RC	280	0 = R2010

Common:

Extrusion	3BD	210	
Text midpt	2RD	11	See DXF documentation.
Elevation	BD	11	Z-coord for the ECS points (11, 12, 16).
		12	(The 16 remains (0,0,0) in entgets of this entity, since the 16 is not used in this type of dimension and is not present in the binary form here.)
Flags 1	RC	70	Non-bit-pair-coded. NOT the 70 group, but helps define it. Apparently only the two lowest bit are used: 76543210: Bit 0 : The OPPOSITE of bit 7 (128) of 70. Bit 1 : Same as bit 5 (32) of the 70 (but 32 is not doc'd by ACAD). The actual 70-group value comes from 3 things: 6 for being an ordinate DIMENSION, plus whatever bits "Flags 1" and "Flags 2" specify.
User text	TV	1	
Text rot	BD	53	See DXF documentation.
Horiz dir	BD	51	See DXF documentation.
Ins X-scale	BD	41	Undoc'd. These apply to the insertion of the
Ins Y-scale	BD	42	anonymous block. None of them can be
Ins Z-scale	BD	43	dealt with via entget/entmake/entmod.
Ins rotation	BD	54	The last 2 (43 and 54) are reported by DXFOUT (when not default values). ALL OF THEM can be set via DXFIN, however.

R2000+:

Attachment Point	BS	71	
Linespacing Style	BS	72	
Linespacing Factor	BD	41	
Actual Measurement	BD	42	

R2007+:

Unknown	B	73	
Flip arrow1	B	74	

Flip arrow2	B	75	
Common:			
12-pt	2RD	12	See DXF documentation.

20.4.23 DIMENSION (ORDINATE) (20)

Common Entity Data			
Common Dimension Data			See paragraph 20.4.22.
Common:			
10-pt	3BD	10	See DXF documentation.
13-pt	3BD	13	See DXF documentation.
14-pt	3BD	14	See DXF documentation.
Flags 2	RC	70	Non-bit-pair-coded. NOT the 70 group, but helps define it. Apparently only the lowest bit is used; it's bit 6 (64) of the 70 group.
Common Entity Handle Data			
	H	3	DIMSTYLE (hard pointer)
	H	2	anonymous BLOCK (hard pointer)
CRC	X	---	

20.4.23.1 R14 Example:

```
OBJECT: dim ordinate (14H), len 5CH (92), handle: 9E

    0157D 5C 00                    \.        0101 1100 0000 0000

    0157F 45 00 67 A4 08 10 00 05   E.g.....   0100 0101 0000 0000 0110 0111 1010 0100 0000 1000 0001 0000 0000 0000 0000 0101

    01587 5B 52 6B 24 C2 1F B9 8C   [Rk$....   0101 1011 0101 0010 0110 1011 0010 0100 1100 0010 0001 1111 1011 1001 1000 1100

    0158F 32 80 21 6E 4C 98 C7 73   2.!nL..s   0011 0010 1000 0000 0010 0001 0110 1110 0100 1100 1001 1000 1100 0111 0111 0011

    01597 F0 7F 05 D4 AC 00 00 00   ........   1111 0000 0111 1111 0000 0101 1101 0100 1010 1100 0000 0000 0000 0000 0000 0000

    0159F 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    015A7 00 00 00 00 01 50 D8 84   .....P..   0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0101 0000 1101 1000 1000 0100

    015AF 7F 51 B7 94 26 80 2C 78   .Q..&.,x   0111 1111 0101 0001 1011 0111 1001 0100 0010 0110 1000 0000 0010 1100 0111 1000

    015B7 71 23 C3 5B 81 20 40 61   q#.[. @a   0111 0001 0010 0011 1100 0011 0101 1011 1000 0001 0010 0000 0100 0000 0110 0001

    015BF B6 92 67 34 E8 BA 00 21   ..g4...!   1011 0110 1001 0010 0110 0111 0011 0100 1110 1000 1011 1010 0000 0000 0010 0001

    015C7 6E 4C 98 C7 73 F0 7F 00   nL..s...   0110 1110 0100 1100 1001 1000 1100 0111 0111 0011 1111 0000 0111 1111 0000 0000

    015CF 18 28 87 E0 86 50 87 28   .(...P.(   0001 1000 0010 1000 1000 0111 1110 0000 1000 0110 0101 0000 1000 0111 0010 1000

    015D7 8E A8 C9 80               ....       1000 1110 1010 1000 1100 1001 1000 0000

    015DB 8E 48                    crc
```

20.4.24 DIMENSION (LINEAR) (21)

```
        Common Entity Data
        Common Dimension Data                 See paragraph 20.4.22.
Common:
        13-pt                 3BD      13     See DXF documentation.
        14-pt                 3BD      14     See DXF documentation.
        10-pt                 3BD      10     See DXF documentation.
        Ext ln rot             BD      52     Extension line rotation; see DXF documentation.
        Dim rot                BD      50     Linear dimension rotation; see DXF documentation.
        Common Entity Handle Data
                               H        3     DIMSTYLE (hard pointer)
                               H        2     anonymous BLOCK (hard pointer)
        CRC                    X      ---
```

20.4.24.1 R14 Example:

```
OBJECT: dim linear (15H), len 6BH (107), handle: AC

    015DD 6B 00                    k.         0110 1011 0000 0000

    015DF 45 40 6B 27 E8 10 00 05   E@k'....   0100 0101 0100 0000 0110 1011 0010 0111 1110 1000 0001 0000 0000 0000 0000 0101

    015E7 5B 52 A8 5F BD 44 3D 70   [R._.D=p   0101 1011 0101 0010 1010 1000 0101 1111 1011 1101 0100 0100 0011 1101 0111 0000

    015EF 3C 80 80 18 62 E8 57 62   <...b.Wb   0011 1100 1000 0000 1000 0000 0001 1000 0110 0010 1110 1000 0101 0111 0110 0010

    015F7 24 81 05 D4 AC 00 00 00   $.......   0010 0100 1000 0001 0000 0101 1101 0100 1010 1100 0000 0000 0000 0000 0000 0000

    015FF 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    01607 00 00 00 00 00 72 6E 2A   .....rn*   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0111 0010 0110 1110 0010 1010

    0160F 01 C0 D2 8D 20 09 11 EC   .... ...   0000 0001 1100 0000 1101 0010 1000 1101 0010 0000 0000 1001 0001 0001 1110 1100

    01617 04 B1 82 01 48 11 C5 80   ....H...   0000 0100 1011 0001 1000 0010 0000 0001 0100 1000 0001 0001 1100 0101 1000 0000

    0161F 66 42 BC CA 42 80 5C 7C   fB..B.\|   0110 0110 0100 0010 1011 1100 1100 1010 0100 0010 1000 0000 0101 1100 0111 1100

    01627 B9 38 1C BB 05 20 47 16   .8... G.   1011 1001 0011 1000 0001 1100 1011 1011 0000 0101 0010 0000 0100 0111 0001 0110

    0162F 01 99 0A F3 29 0A 00 80   ....)...   0000 0001 1001 1001 0000 1010 1111 0011 0010 1001 0000 1010 0000 0000 1000 0000

    01637 18 62 E8 57 62 24 81 51   .b.Wb$.Q   0001 1000 0110 0010 1110 1000 0101 0111 0110 0010 0010 0100 1000 0001 0101 0001

    0163F 82 88 7E 08 75 08 72 88   ..~.u.r.   1000 0010 1000 1000 0111 1110 0000 1000 0111 0101 0000 1000 0111 0010 1000 1000

    01647 EA 8C FB                 ...        1110 1010 1000 1100 1111 1011

    0164A 48 DA                    crc
```

20.4.25 DIMENSION (ALIGNED) (22)

```
        Common Entity Data
        Common Dimension Data                 See paragraph 20.4.22.
```

Common:

13-pt	3BD	13	See DXF documentation.
14-pt	3BD	14	See DXF documentation.
10-pt	3BD	10	See DXF documentation.
Ext ln rot	BD	52	Extension line rotation; see DXF documentation.
Common Entity Handle Data			
	H	3	DIMSTYLE (hard pointer)
	H	2	anonymous BLOCK (hard pointer)
CRC	X	---	

20.4.25.1 R14 Example:

```
OBJECT: dim aligned (16H), len 6BH (107), handle: BA

    0164C 6B 00                    k.         0110 1011 0000 0000

    0164E 45 80 6E A7 D8 10 00 05   E.n.....   0100 0101 1000 0000 0110 1110 1010 0111 1101 1000 0001 0000 0000 0000 0000 0101

    01656 5B 53 B7 92 B9 9A CA CA   [S......   0101 1011 0101 0011 1011 0111 1001 0010 1011 1001 1001 1010 1100 1010 1100 1010

    0165E 1C 81 55 6D 19 67 3E 90   ..Um.g>.   0001 1100 1000 0001 0101 0101 0110 1101 0001 1001 0110 0111 0011 1110 1001 0000

    01666 28 81 05 D4 AC 00 00 00   (.......   0010 1000 1000 0001 0000 0101 1101 0100 1010 1100 0000 0000 0000 0000 0000 0000

    0166E 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    01676 00 00 00 00 00 2A 41 59   .....*AY   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 1010 0100 0001 0101 1001

    0167E E6 59 20 09 20 04 E7 DE   .Y . ...   1110 0110 0101 1001 0010 0000 0000 1001 0010 0000 0000 0100 1110 0111 1101 1110

    01686 65 A9 1D 81 E8 11 E8 B7   e.......   0110 0101 1010 1001 0001 1101 1000 0001 1110 1000 0001 0001 1110 1000 1011 0111

    0168E 57 AB F5 B4 22 80 6E 48   W...".nH   0101 0111 1010 1011 1111 0101 1011 0100 0010 0010 1000 0000 0110 1110 0100 1000

    01696 CB DF EC 81 08 20 46 F7   ..... F.   1100 1011 1101 1111 1110 1100 1000 0001 0000 1000 0010 0000 0100 0110 1111 0111

    0169E 1E 19 C7 7A E8 92 00 60   ...z...`   0001 1110 0001 1001 1100 0111 0111 1010 1110 1000 1001 0010 0000 0000 0110 0000

    016A6 DD 30 19 D6 34 28 81 46   .0..4(.F   1101 1101 0011 0000 0001 1001 1101 0110 0011 0100 0010 1000 1000 0001 0100 0110

    016AE 0A 21 F8 21 D4 21 EA 23   .!.!.!.#   0000 1010 0010 0001 1111 1000 0010 0001 1101 0100 0010 0001 1110 1010 0010 0011

    016B6 AA 35 BB                 .5.        1010 1010 0011 0101 1011 1011

    016B9 EA 25                    crc
```

20.4.26 DIMENSION (ANGULAR, 3-PT) (23)

Common Entity Data			
Common Dimension Data			See paragraph 20.4.22.

Common:

10-pt	3BD	10	See DXF documentation.
13-pt	3BD	13	See DXF documentation.
14-pt	3BD	14	See DXF documentation.

15-pt	3BD	15	See DXF documentation.
Common Entity Handle Data			
	H	3	DIMSTYLE (hard pointer)
	H	2	anonymous BLOCK (hard pointer)
CRC	X	---	

20.4.26.1 R14 Example:

```
OBJECT: dim angular (17H), len 7BH (123), handle: C9

    016BB 7B 00                    {.         0111 1011 0000 0000

    016BD 45 C0 72 63 F8 18 00 05   E.rc....   0100 0101 1100 0000 0111 0010 0110 0011 1111 1000 0001 1000 0000 0000 0000 0101

    016C5 5B 53 DC 3A 57 CD 05 40   [S.:W..@   0101 1011 0101 0011 1101 1100 0011 1010 0101 0111 1100 1101 0000 0101 0100 0000

    016CD 2E 80 C0 5E B2 D6 6F 22   ...^..o"   0010 1110 1000 0000 1100 0000 0101 1110 1011 0010 1101 0110 0110 1111 0010 0010

    016D5 40 81 05 D4 AC 00 00 00   @.......   0100 0000 1000 0001 0000 0101 1101 0100 1010 1100 0000 0000 0000 0000 0000 0000

    016DD 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    016E5 00 00 00 00 00 68 08 AF   .....h..   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0110 1000 0000 1000 1010 1111

    016ED 7C 2C 5E 0C A0 11 C0 E9   |,^.....   0111 1100 0010 1100 0101 1110 0000 1100 1010 0000 0001 0001 1100 0000 1110 1001

    016F5 18 9D 34 04 28 10 D2 BA   ..4.(...   0001 1000 1001 1101 0011 0100 0000 0100 0010 1000 0001 0000 1101 0010 1011 1010

    016FD AD A6 B9 2C 3A 80 61 CA   ...,:.a.   1010 1101 1010 0110 1011 1001 0010 1100 0011 1010 1000 0000 0110 0001 1100 1010

    01705 13 3A 13 1C 90 20 45 65   .:... Ee   0001 0011 0011 1010 0001 0011 0001 1100 1001 0000 0010 0000 0100 0101 0110 0101

    0170D 3A 06 5E 80 38 EA 00 D8   :.^.8...   0011 1010 0000 0110 0101 1110 1000 0000 0011 1000 1110 1010 0000 0000 1101 1000

    01715 3B 7A 98 A2 88 3A 81 0A   ;z...:..   0011 1011 0111 1010 1001 1000 1010 0010 1000 1000 0011 1010 1000 0001 0000 1010

    0171D 88 A1 BF 90 3F C3 48 00   ....?.H.   1000 1000 1010 0001 1011 1111 1001 0000 0011 1111 1100 0011 0100 1000 0000 0000

    01725 55 2D C2 C7 6F 28 FA 04   U-..o(..   0101 0101 0010 1101 1100 0010 1100 0111 0110 1111 0010 1000 1111 1010 0000 0100

    0172D 60 A2 1F 82 1F 42 18 A2   `....B..   0110 0000 1010 0010 0001 1111 1000 0010 0001 1111 0100 0010 0001 1000 1010 0010

    01735 3A A3 76                 :.v        0011 1010 1010 0011 0111 0110

    01738 42 38                    crc
```

20.4.27 DIMENSION (ANGULAR, 2-LINE) (24)

Common Entity Data			
Common Dimension Data			See paragraph 20.4.22.

Common:

16-pt	2RD	16	See DXF documentation.
13-pt	3BD	13	See DXF documentation.
14-pt	3BD	14	See DXF documentation.
15-pt	3BD	15	See DXF documentation.
10-pt	3BD	10	See DXF documentation.

Common Entity Handle Data			
	H	3	DIMSTYLE (hard pointer)
	H	2	anonymous BLOCK (hard pointer)
CRC	X	---	

20.4.28 DIMENSION (RADIUS) (25)

Common Entity Data			
Common Dimension Data			See paragraph 20.4.22.
Common:			
10-pt	3BD	10	See DXF documentation.
15-pt	3BD	15	See DXF documentation.
Leader len	D	40	Leader length.
Common Entity Handle Data			
	H	3	DIMSTYLE (hard pointer)
	H	2	anonymous BLOCK (hard pointer)
CRC	X	---	

20.4.28.1 R14 Example:

```
OBJECT: dim radial (19H), len 71H (113), handle: D5

   0173A 71 00                 q.        0111 0001 0000 0000

   0173C 46 40 75 51 45 11 10 00   F@uQE...   0100 0110 0100 0000 0111 0101 0101 0001 0100 0101 0001 0001 0001 0000 0000 0000

   01744 60 01 E4 45 35 45 94 C4   `..E5E..   0110 0000 0000 0001 1110 0100 0100 0101 0011 0101 0100 0101 1001 0100 1100 0100

   0174C 50 20 04 62 00 14 60 10   P .b..`.   0101 0000 0010 0000 0000 0100 0110 0010 0000 0000 0001 0100 0110 0000 0001 0000

   01754 00 20 18 5E 06 00 01 56   . .^...V   0000 0000 0010 0000 0001 1000 0101 1110 0000 0110 0000 0000 0000 0001 0101 0110

   0175C D4 BE B8 AD 7A BB 82 11   ....z...   1101 0100 1011 1110 1011 1000 1010 1101 0111 1010 1011 1011 1000 0010 0001 0001

   01764 20 48 89 3F DF D9 A0 0F    H.?....   0010 0000 0100 1000 1000 1001 0011 1111 1101 1111 1101 1001 1010 0000 0000 1111

   0176C A0 41 55 2B 00 00 00 00   .AU+....   1010 0000 0100 0001 0101 0101 0010 1011 0000 0000 0000 0000 0000 0000 0000 0000

   01774 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   0177C 00 00 00 00 0A A8 A1 BF   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 1010 1010 1000 1010 0001 1011 1111

   01784 90 3F C3 48 00 55 2D C2   .?.H.U-.   1001 0000 0011 1111 1100 0011 0100 1000 0000 0000 0101 0101 0010 1101 1100 0010

   0178C C7 6F 28 FA 04 27 F9 9E   .o(..'..   1100 0111 0110 1111 0010 1000 1111 1010 0000 0100 0010 0111 1111 1001 1001 1110

   01794 65 FB 50 0E A0 07 FF E7   e.P.....   0110 0101 1111 1011 0101 0000 0000 1110 1010 0000 0000 0111 1111 1111 1110 0111

   0179C 46 14 F3 63 E8 14 60 A2   F..c..`.   0100 0110 0001 0100 1111 0011 0110 0011 1110 1000 0001 0100 0110 0000 1010 0010

   017A4 1F 82 19 42 18 A2 3A A3   ...B..:.   0001 1111 1000 0010 0001 1001 0100 0010 0001 1000 1010 0010 0011 1010 1010 0011

   017AC 94                    .         1001 0100
   017AD EA 1E                   crc
```

20.4.29 DIMENSION (DIAMETER) (26)

```
        Common Entity Data
        Common Dimension Data                  See paragraph 20.4.22.
Common:
        15-pt                3BD      15       See DXF documentation.
        10-pt                3BD      10       See DXF documentation.
        Leader len            BD      40       Leader length.
        Common Entity Handle Data
                               H       3       DIMSTYLE (hard pointer)
                               H       2       anonymous BLOCK (hard pointer)
        CRC                    X     ---
```

20.4.29.1 R14 Example:

```
OBJECT: dim diameter (1AH), len 70H (112), handle: E1

   017AF 70 00                   p.        0111 0000 0000 0000

   017B1 46 80 78 51 45 11 10 00   F.xQE...   0100 0110 1000 0000 0111 1000 0101 0001 0100 0101 0001 0001 0001 0000 0000 0000

   017B9 60 01 E4 45 35 45 94 C4   `..E5E..   0110 0000 0000 0001 1110 0100 0100 0101 0011 0101 0100 0101 1001 0100 1100 0100

   017C1 50 20 04 62 00 14 60 10   P .b..`.   0101 0000 0010 0000 0000 0100 0110 0010 0000 0000 0001 0100 0110 0000 0001 0000

   017C9 00 20 18 5E 06 00 01 56   . .^...V   0000 0000 0010 0000 0001 1000 0101 1110 0000 0110 0000 0000 0000 0001 0101 0110

   017D1 D4 AE 72 3A F7 9A B2 10   ..r:....   1101 0100 1010 1110 0111 0010 0011 1010 1111 0111 1001 1010 1011 0010 0001 0000

   017D9 A0 4A 92 A4 03 41 DC 0E   .J...A..   1010 0000 0100 1010 1001 0010 1010 0100 0000 0011 0100 0001 1101 1100 0000 1110

   017E1 20 41 55 2B 00 00 00 00    AU+....   0010 0000 0100 0001 0101 0101 0010 1011 0000 0000 0000 0000 0000 0000 0000 0000

   017E9 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   017F1 00 00 00 00 1B 6E ED 97   .....n..   0000 0000 0000 0000 0000 0000 0000 0000 0001 1011 0110 1110 1110 1101 1001 0111

   017F9 4E 85 E3 A8 06 3F D6 3A   N....?.:   0100 1110 1000 0101 1110 0011 1010 1000 0000 0110 0011 1111 1101 0110 0011 1010

   01801 B1 4B 40 F2 04 65 89 57   .K@..e.W   1011 0001 0100 1011 0100 0000 1111 0010 0000 0100 0110 0101 1000 1001 0101 0111

   01809 1E C7 E6 8C 20 14 94 EA   .... ...   0001 1110 1100 0111 1110 0110 1000 1100 0010 0000 0001 0100 1001 0100 1110 1010

   01811 95 BB 16 24 08 14 60 A2   ...$..`.   1001 0101 1011 1011 0001 0110 0010 0100 0000 1000 0001 0100 0110 0000 1010 0010

   01819 1F 82 18 C0 A2 3A A3 AD   .....:..   0001 1111 1000 0010 0001 1000 1100 0000 1010 0010 0011 1010 1010 0011 1010 1101

   01821 37 B4                   crc
```

20.4.30 LARGE_RADIAL_DIMENSION

Class properties:

App name	ObjectDBX Classes

Class number	Dynamic (>= 500)
DWG version	R18
Maintenance version	0
Class proxy flags	0x401
C++ class name	AcDbRadialDimensionLarge
DXF name	LARGE_RADIAL_DIMENSION

The large radial dimension was introduced in AutoCAD 2004. The DXF format is slightly different from the other dimension entities. The entity type in DXF is `LARGE_RADIAL_DIMENSION`, rather than `DIMENSION`.

```
 Common Entity Data
 Common Dimension Data                        See paragraph 20.4.22.
Common:
 Center point                3BD   10
 Chord point                 3BD   13
 Unknown                      BD   40         Value 0
 Override center             3BD   14
 Jog point                   3BD   15
 Common Entity Handle Data
                               H    3         DIMSTYLE (hard pointer)
                               H    2         anonymous BLOCK (hard pointer)
 CRC                           X   ---
```

20.4.31 POINT (27)

```
 Common Entity Data
 Point                3BD      10
 Thickness             BT      39
 Extrusion             BE     210
 X-axis ang            BD      50     See DXF documentation
 Common Entity Handle Data
 CRC                    X     ---
```

20.4.31.1 R14 Example:

```
OBJECT: point (1BH), len 23H (35), handle: D2

   0062A 23 00                     #.        0010 0011 0000 0000

   0062C 46 C0 74 A6 C8 00 00 01   F.t.....  0100 0110 1100 0000 0111 0100 1010 0110 1100 1000 0000 0000 0000 0000 0000 0001

   00634 33 09 FE 67 99 7E D4 03   3..g.~..  0011 0011 0000 1001 1111 1110 0110 0111 1001 1001 0111 1110 1101 0100 0000 0011

   0063C A8 01 FF F9 D1 85 3C D8   ......<.  1010 1000 0000 0001 1111 1111 1111 1001 1101 0001 1000 0101 0011 1100 1101 1000

   00644 FA 05 53 60 84 18 28 CC   ..S`..(.  1111 1010 0000 0101 0101 0011 0110 0000 1000 0100 0001 1000 0010 1000 1100 1100

   0064C A8 89 86                  ...       1010 1000 1000 1001 1000 0110

   0064F 09 DF                     crc
```

20.4.32 3DFACE (28)

```
        Common Entity Data
R13-R14 Only:
        1st corner           3BD     10
        2nd corner           3BD     11
        3rd corner           3BD     12
        4th corner           3BD     13
        Invis flags           BS     70      Invisible edge flags
R2000+:
        Has no flag ind.       B
        Z is zero bit          B
        1st corner x          RD     10
        1st corner y          RD     20
        1st corner z          RD     30      Present only if "Z is zero bit" is 0.
        2nd corner           3DD     11      Use 10 value as default point
        3rd corner           3DD     12      Use 11 value as default point
        4th corner           3DD     13      Use 12 value as default point
        Invis flags           BS     70      Present it "Has no flag ind." is 0.
Common:
        Common Entity Handle Data
        CRC                    X    ---
```

20.4.32.1 R14 Example:

```
OBJECT: 3d face (1CH), len 50H (80), handle: E3

   01846 50 00                 P.        0101 0000 0000 0000

   01848 47 00 78 E3 18 10 00 05   G.x.....   0100 0111 0000 0000 0111 1000 1110 0011 0001 1000 0001 0000 0000 0000 0000 0101

   01850 7B 06 54 B1 62 D9 BA E4   {.T.b...   0111 1011 0000 0110 0101 0100 1011 0001 0110 0010 1101 1001 1011 1010 1110 0100

   01858 28 00 02 01 84 E7 8E 80   (.......   0010 1000 0000 0000 0000 0010 0000 0001 1000 0100 1110 0111 1000 1110 1000 0000

   01860 12 04 4F 73 C2 29 98 53   ..Os.).S   0001 0010 0000 0100 0100 1111 0111 0011 1100 0010 0010 1001 1001 1000 0101 0011

   01868 12 20 16 8C 3C B6 E3 69   . ..<..i   0001 0010 0010 0000 0001 0110 1000 1100 0011 1100 1011 0110 1110 0011 0110 1001

   01870 E2 08 10 14 77 8D 5D FA   ....w.].   1110 0010 0000 1000 0001 0000 0001 0100 0111 0111 1000 1101 0101 1101 1111 1010

   01878 52 4C 80 50 0B 36 A4 30   RL.P.6.0   0101 0010 0100 1100 1000 0000 0101 0000 0000 1011 0011 0110 1010 0100 0011 0000

   01880 F0 FF 1F C5 51 57 68 85   ....QWh.   1111 0000 1111 1111 0001 1111 1100 0101 0101 0001 0101 0111 0110 1000 1000 0101

   01888 48 51 12 00 40 84 CA FB   HQ..@...   0100 1000 0101 0001 0001 0010 0000 0000 0100 0000 1000 0100 1100 1010 1111 1011

   01890 AF DF EC 7F 46 0A 21 FA   ....F.!.   1010 1111 1101 1111 1110 1100 0111 1111 0100 0110 0000 1010 0010 0001 1111 1010

   01898 1A A6                 crc
```

20.4.33 POLYLINE (PFACE) (29)

```
        Common Entity Data
        Numverts               BS      71      Number of vertices in the mesh.
        Numfaces               BS      72      Number of faces
R2004+:
        Owned Object Count     BL              Number of objects owned by this object.
Common:
        Common Entity Handle Data
R13-R2000:
                                H              first VERTEX (soft pointer)
                                H              last  VERTEX (soft pointer)
R2004+:
                                H              [VERTEX (soft pointer)] Repeats "Owned Object Count"
                                               times.
Common:
                                H              SEQEND (hard owner)
        CRC                     X     ---
```

20.4.33.1 Example:

```
OBJECT: pface start (1DH), len 19H (25), handle: 55

   00BC0 19 00                   ..        0001 1001 0000 0000

   00BC2 47 40 55 62 E8 00 00 05   G@Ub....   0100 0111 0100 0000 0101 0101 0110 0010 1110 1000 0000 0000 0000 0000 0000 0101

   00BCA 5B 20 88 19 82 88 7E 08   [ ....~.   0101 1011 0010 0000 1000 1000 0001 1001 1000 0010 1000 1000 0111 1110 0000 1000

   00BD2 4D 08 4A 0A B2 0A E1 8A   M.J.....   0100 1101 0000 1000 0100 1010 0000 1010 1011 0010 0000 1010 1110 0001 1000 1010

   00BDA E8                      .         1110 1000

   00BDB D7 3E                   crc
```

20.4.34 POLYLINE (MESH) (30)

```
        Common Entity Data
        Flags                  BS      70
        Curve type             BS      75      Curve and smooth surface type.
        M vert count           BS      71      M vertex count
        N vert count           BS      72      N vertex count
        M density              BS      73      M vertex count
        N density              BS      74      N vertex count
R2004+:
        Owned Object Count     BL              Number of objects owned by this object.
Common:
        Common Entity Handle Data
```

R13-R2000:			
	H		FIRST VERTEX (soft pointer)
	H		LAST VERTEX (soft pointer)
R2004+:			
	H		[VERTEX (soft pointer)] Repeats “Owned Object Count” times.
Common:			
	H		SEQEND (CODE 3)
CRC	X	---	

20.4.34.1 Example:

```
OBJECT: 3d surf sol st (1EH), len 1AH (26), handle: 66

   00E18 1A 00                   ..        0001 1010 0000 0000

   00E1A 47 80 59 A3 68 00 00 05   G.Y.h...   0100 0111 1000 0000 0101 1001 1010 0011 0110 1000 0000 0000 0000 0000 0000 0101

   00E22 5B 22 32 0C 83 D1 82 88   ["2.....   0101 1011 0010 0010 0011 0010 0000 1100 1000 0011 1101 0001 1000 0010 1000 1000

   00E2A 7C 05 09 62 0B 3A 0C 81   |..b.:..   0111 1100 0000 0101 0000 1001 0110 0010 0000 1011 0011 1010 0000 1100 1000 0001

   00E32 8C 8C                   ..        1000 1100 1000 1100

   00E34 3C E7                   crc
```

20.4.35 SOLID (31)

Common Entity Data			
Thickness	BT	39	
Elevation	BD	---	Z for 10 - 13.
1st corner	2RD	10	
2nd corner	2RD	11	
3rd corner	2RD	12	
4th corner	2RD	13	
Extrusion	BE	210	
Common Entity Handle Data			
CR	X	---	

20.4.35.1 R14 Example:

```
OBJECT: solid (1FH), len 52H (82), handle: CF

   00566 52 00                   R.        0101 0010 0000 0000

   00568 47 C0 73 E2 98 10 00 01   G.s.....   0100 0111 1100 0000 0111 0011 1110 0010 1001 1000 0001 0000 0000 0000 0000 0001

   00570 33 50 A8 BE 18 24 52 D8   3P...$R.   0011 0011 0101 0000 1010 1000 1011 1110 0001 1000 0010 0100 0101 0010 1101 1000

   00578 F2 07 52 C3 01 40 1D 30   ..R..@.0   1111 0010 0000 0111 0101 0010 1100 0011 0000 0001 0100 0000 0001 1101 0011 0000

   00580 FA 00 FF 31 0A 96 82 A0   ...1....   1111 1010 0000 0000 1111 1111 0011 0001 0000 1010 1001 0110 1000 0010 1010 0000
```

```
    00588 F2 06 8B FA 70 47 8B 40   ....pG.@   1111 0010 0000 0110 1000 1011 1111 1010 0111 0000 0100 0111 1000 1011 0100 0000
    00590 FA 02 7F 99 E6 5F B5 00   ....._..   1111 1010 0000 0010 0111 1111 1001 1001 1110 0110 0101 1111 1011 0101 0000 0000
    00598 EA 01 FF F9 D1 85 3C D8   ......<.   1110 1010 0000 0001 1111 1111 1111 1001 1101 0001 1000 0101 0011 1100 1101 1000
    005A0 FA 02 7F 99 E6 5F B5 00   ....._..   1111 1010 0000 0010 0111 1111 1001 1001 1110 0110 0101 1111 1011 0101 0000 0000
    005A8 EA 01 FF F9 D1 85 3C D8   ......<.   1110 1010 0000 0001 1111 1111 1111 1001 1101 0001 1000 0101 0011 1100 1101 1000
    005B0 FA 05 38 20 A6 0A 21 EA   ..8 ..!.   1111 1010 0000 0101 0011 1000 0010 0000 1010 0110 0000 1010 0010 0001 1110 1010
    005B8 22 6A                     "j         0010 0010 0110 1010
    005BA 18 03                     crc
```

20.4.36 TRACE (32)

```
        Common Entity Data
        Thickness             BT      39
        Elevation             BD     ---      Z for 10 - 13.
        1st corner           2RD      10
        2nd corner           2RD      11
        3rd corner           2RD      12
        4th corner           2RD      13
        Extrusion             BE     210
        Common Entity Handle Data
        CRC                    X     ---
```

20.4.36.1 R14 Example:

```
OBJECT: trace (20H), len 51H (81), handle: E7

    018EF 51 00                     Q.         0101 0001 0000 0000
    018F1 48 00 79 E2 98 10 00 05   H.y.....   0100 1000 0000 0000 0111 1001 1110 0010 1001 1000 0001 0000 0000 0000 0000 0101
    018F9 5B 53 70 DA A0 AD EE C1   [Sp.....   0101 1011 0101 0011 0111 0000 1101 1010 1010 0000 1010 1101 1110 1110 1100 0001
    01901 42 05 BA E0 2A DA A9 60   B...*..`   0100 0010 0000 0101 1011 1010 1110 0000 0010 1010 1101 1010 1010 1001 0110 0000
    01909 02 05 75 29 DE 3E FF 89   ..u).>..   0000 0010 0000 0101 0111 0101 0010 1001 1101 1110 0011 1110 1111 1111 1000 1001
    01911 42 03 4E 20 B3 8F 50 C0   B.N ..P.   0100 0010 0000 0011 0100 1110 0010 0000 1011 0011 1000 1111 0101 0000 1100 0000
    01919 02 00 4B 2A 12 65 70 A9   ..K*.ep.   0000 0010 0000 0000 0100 1011 0010 1010 0001 0010 0110 0101 0111 0000 1010 1001
    01921 52 04 9E 9B A5 92 BF 40   R......@   0101 0010 0000 0100 1001 1110 1001 1011 1010 0101 1001 0010 1011 1111 0100 0000
    01929 A2 06 1D 16 C2 A5 61 81   ......a.   1010 0010 0000 0110 0001 1101 0001 0110 1100 0010 1010 0101 0110 0001 1000 0001
    01931 52 02 6F 4E 85 D6 E7 88   R.oN....   0101 0010 0000 0010 0110 1111 0100 1110 1000 0101 1101 0110 1110 0111 1000 1000
    01939 A2 05 26 0A 21 F8 20 6C   ..&.!. l   1010 0010 0000 0101 0010 0110 0000 1010 0010 0001 1111 1000 0010 0000 0110 1100
    01941 1A                        .          0001 1010
```

```
   01942 7E C2                   crc
```

20.4.37 SHAPE (33)

Common Entity Data			
Ins pt	3BD	10	
Scale	BD	40	Scale factor, default value 1.
Rotation	BD	50	Rotation in radians, default value 0.
Width factor	BD	41	Width factor, default value 1.
Oblique	BD	51	Oblique angle in radians, default value 0.
Thickness	BD	39	
Shapeno	BS	2	This is the shape index. In DXF the shape name is stored. When reading from DXF, the shape is found by iterating over all the text styles (SHAPEFILE, see paragraph 20.4.56) and when the text style contains a shape file, iterating over all the shapes until the one with the matching name is found.
Extrusion	3BD	210	
Common Entity Handle Data			
	H		SHAPEFILE (hard pointer)
CRC	X	--	

20.4.37.1 Example:

```
OBJECT: shape (21H), len 26H (38), handle: F5

   008BC 26 00                   &.        0010 0110 0000 0000

   008BE 48 40 7D 67 48 00 00 01   H@}gH...  0100 1000 0100 0000 0111 1101 0110 0111 0100 1000 0000 0000 0000 0000 0000 0001

   008C6 5B 14 AF 3D 96 39 59 A1   [..=.9Y.  0101 1011 0001 0100 1010 1111 0011 1101 1001 0110 0011 1001 0101 1001 1010 0001

   008CE 48 04 20 D5 14 35 41 08   H. ..5A.  0100 1000 0000 0100 0010 0000 1101 0101 0001 0100 0011 0101 0100 0001 0000 1000

   008D6 8A 04 CD 32 F4 C0 18 28   ...2...(  1000 1010 0000 0100 1100 1101 0011 0010 1111 0100 1100 0000 0001 1000 0010 1000

   008DE 87 A0 30 28 F9 ED         ..0(..    1000 0111 1010 0000 0011 0000 0010 1000 1111 1001 1110 1101

   008E4 38 74                   crc
```

20.4.38 VIEWPORT ENTITY (34)

Common Entity Data		
Center	3BD	10
Width	BD	40
Height	BD	41

R2000+:

View Target	3BD	17
View Direction	3BD	16
View Twist Angle	BD	51

	View Height	BD	45	
	Lens Length	BD	42	
	Front Clip Z	BD	43	
	Back Clip Z	BD	44	
	Snap Angle	BD	50	
	View Center	2RD	12	
	Snap Base	2RD	13	
	Snap Spacing	2RD	14	
	Grid Spacing	2RD	15	
	Circle Zoom	BS	72	
R2007+:				
	Grid Major	BS	61	
R2000+:				
	Frozen Layer Count	BL		
	Status Flags	BL	90	
	Style Sheet	TV	1	
	Render Mode	RC	281	
	UCS at origin	B	74	
	UCS per Viewport	B	71	
	UCS Origin	3BD	110	
	UCS X Axis	3BD	111	
	UCS Y Axis	3BD	112	
	UCS Elevation	BD	146	
	UCS Ortho View Type	BS	79	
R2004+:				
	ShadePlot Mode	BS	170	
R2007+:				
	Use def. lights	B	292	
	Def. lighting type	RC	282	
	Brightness	BD	141	
	Contrast	BD	142	
	Ambient light color	CMC	63	
Common:				
	Common Entity Handle Data			
R13-R14 Only:				
		H		VIEWPORT ENT HEADER (hard pointer)
R2000+:				
		H	341	Frozen Layer Handles (use count from above) (hard pointer until R2000, soft pointer from R2004 onwards)
		H	340	Clip boundary handle (soft pointer)

R2000:			
	H		VIEWPORT ENT HEADER ((hard pointer))
R2000+:			
	H	345	Named UCS Handle (hard pointer)
	H	346	Base UCS Handle (hard pointer)
R2007+:			
	H	332	Background (soft pointer)
	H	348	Visual Style (hard pointer)
	H	333	Shadeplot ID (soft pointer)
	H	361	Sun (hard owner)

20.4.38.1 R14 Example:

```
OBJECT: vpent (22H), len 117H (279), handle: 01 26

   03934 17 01                    ..        0001 0111 0000 0001

   03936 48 80 80 49 9D F5 11 10   H..I....   0100 1000 1000 0000 1000 0000 0100 1001 1001 1101 1111 0101 0001 0001 0001 0000

   0393E 00 50 01 E4 D5 64 94 55   .P...d.U   0000 0000 0101 0000 0000 0001 1110 0100 1101 0101 0110 0100 1001 0100 0101 0101

   03946 70 20 04 61 00 00 A0 00   p .a....   0111 0000 0010 0000 0000 0100 0110 0001 0000 0000 0000 0000 1010 0000 0000 0000

   0394E 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   03956 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   0395E 00 00 00 00 00 00 00 A0   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1010 0000

   03966 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   0396E 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   03976 00 00 00 00 00 0F 03 F2   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1111 0000 0011 1111 0010

   0397E 80 00 00 00 00 00 00 00   ........   1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   03986 02 80 00 00 00 00 00 02   ........   0000 0010 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010

   0398E 24 02 87 89 21 A6 5A CA   $...!.Z.   0010 0100 0000 0010 1000 0111 1000 1001 0010 0001 1010 0110 0101 1010 1100 1010

   03996 21 A4 02 80 00 00 00 00   !.......   0010 0001 1010 0100 0000 0010 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   0399E 00 01 24 02 80 00 00 00   ..$.....   0000 0000 0000 0001 0010 0100 0000 0010 1000 0000 0000 0000 0000 0000 0000 0000

   039A6 00 00 04 94 02 80 00 00   ........   0000 0000 0000 0000 0000 0100 1001 0100 0000 0010 1000 0000 0000 0000 0000 0000

   039AE 00 00 00 00 00 02 80 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 1000 0000 0000 0000

   039B6 00 00 00 00 00 00 04 60   .......`   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 0110 0000

   039BE 00 04 66 40 04 60 10 04   ..f@.`..   0000 0000 0000 0100 0110 0110 0100 0000 0000 0100 0110 0000 0001 0000 0000 0100

   039C6 60 10 04 60 00 04 60 00   `..`..`.   0110 0000 0001 0000 0000 0100 0110 0000 0000 0000 0000 0100 0110 0000 0000 0000

   039CE 04 60 00 04 60 00 02 80   .`..`...   0000 0100 0110 0000 0000 0000 0000 0100 0110 0000 0000 0000 0000 0010 1000 0000
```

```
    039D6 00 00 00 00 00 00 00 02   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010

    039DE 80 00 00 00 00 00 00 00   ........   1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    039E6 02 80 00 00 00 00 00 00   ........   0000 0010 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    039EE 00 02 80 00 00 00 00 00   ........   0000 0000 0000 0010 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    039F6 0E 03 F2 80 00 00 00 00   ........   0000 1110 0000 0011 1111 0010 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    039FE 00 0E 03 F2 80 00 00 00   ........   0000 0000 0000 1110 0000 0011 1111 0010 1000 0000 0000 0000 0000 0000 0000 0000

    03A06 00 00 0E 03 F2 80 00 00   ........   0000 0000 0000 0000 0000 1110 0000 0011 1111 0010 1000 0000 0000 0000 0000 0000

    03A0E 00 00 00 0E 03 F4 60 00   ......`.   0000 0000 0000 0000 0000 0000 0000 1110 0000 0011 1111 0100 0110 0000 0000 0000

    03A16 00 20 00 20 10 20 18 DA   . . . ..   0000 0000 0010 0000 0000 0000 0010 0000 0001 0000 0010 0000 0001 1000 1101 1010

    03A1E 10 00 00 D6 C3 C4 90 D3   ........   0001 0000 0000 0000 0000 0000 1101 0110 1100 0011 1100 0100 1001 0000 1101 0011

    03A26 2D 65 10 D2 00 00 00 00   -e......   0010 1101 0110 0101 0001 0000 1101 0010 0000 0000 0000 0000 0000 0000 0000 0000

    03A2E 00 00 00 24 81 0F 12 43   ...$...C   0000 0000 0000 0000 0000 0000 0010 0100 1000 0001 0000 1111 0001 0010 0100 0011

    03A36 4C B5 94 45 48 00 00 00   L..EH...   0100 1100 1011 0101 1001 0100 0100 0101 0100 1000 0000 0000 0000 0000 0000 0000

    03A3E 00 00 00 01 12 01 82 88   ........   0000 0000 0000 0000 0000 0000 0000 0001 0001 0010 0000 0001 1000 0010 1000 1000

    03A46 7A 05 08 12 90 09 28      z.....(    0111 1010 0000 0101 0000 1000 0001 0010 1001 0000 0000 1001 0010 1000

    03A4D 6C 19                     crc
```

20.4.39 ELLIPSE (35)

Note that the 10 pt and the 11 vector are WCS -- even though an ellipse is planar and has an extrusion vector (210-group).

Common Entity Data			
Center	3BD	10	(WCS)
SM axis vec	3BD	11	Semi-major axis vector (WCS)
Extrusion	3BD	210	
Axis ratio	BD	40	Minor/major axis ratio
Beg angle	BD	41	Starting angle (eccentric anomaly, radians)
End angle	BD	42	Ending angle (eccentric anomaly, radians)
Common Entity Handle Data			
CRC	X	---	

20.4.39.1 Example:

```
OBJECT: ellipse (23H), len 4CH (76), handle: 01 22

    0381E 4C 00                     L.         0100 1100 0000 0000

    03820 48 C0 80 48 A1 48 10 00   H..H.H..   0100 1000 1100 0000 1000 0000 0100 1000 1010 0001 0100 1000 0001 0000 0000 0000

    03828 05 5B 0C 0A 03 29 8A E7   .[...)..   0000 0101 0101 1011 0000 1100 0000 1010 0000 0011 0010 1001 1000 1010 1110 0111
```

```
03830 42 48 01 F0 9F BC 53 10   BH....S.   0100 0010 0100 1000 0000 0001 1111 0000 1001 1111 1011 1100 0101 0011 0001 0000

03838 40 DA 04 51 23 D0 F1 D6   @..Q#...   0100 0000 1101 1010 0000 0100 0101 0001 0010 0011 1101 0000 1111 0001 1101 0110

03840 AF 7B 9F 9A 89 15 EA 36   .{.....6   1010 1111 0111 1011 1001 1111 1001 1010 1000 1001 0001 0101 1110 1010 0011 0110

03848 B2 DD 17 F5 00 20 00 00   ..... ..   1011 0010 1101 1101 0001 0111 1111 0101 0000 0000 0010 0000 0000 0000 0000 0000

03850 00 00 1E 07 E5 D2 A4 7D   .......}   0000 0000 0000 0000 0001 1110 0000 0111 1110 0101 1101 0010 1010 0100 0111 1101

03858 B0 4C 5E F9 FC 0C 16 A2   .L^.....   1011 0000 0100 1100 0101 1110 1111 1001 1111 1100 0000 1100 0001 0110 1010 0010

03860 2A 7D 90 8C A0 18 28 87   *}....(.   0010 1010 0111 1101 1001 0000 1000 1100 1010 0000 0001 1000 0010 1000 1000 0111

03868 E0 83 A0 69               ...i       1110 0000 1000 0011 1010 0000 0110 1001

0386C ED 08                     crc
```

20.4.40 SPLINE (36)

	Common Entity Data			
	Scenario	BL		a flag which is 2 for fitpts only, 1 for ctrlpts/knots. In 2013 the meaning is somehwat more sophisticated, see knot parameter below.
R2013+:				
	Spline flags 1	BL		Spline flags 1: method fit points = 1, CV frame show = 2, Is closed = 4. At this point the regular spline flags closed bit is made equal to this bit. Value is overwritten below in scenario 2 though, Use knot parameter = 8

	Knot parameter	BL		Knot parameter:
				
				Chord = 0,
				Square root = 1,
				Uniform = 2,
				Custom = 15
				
				The scenario flag becomes 1 if the knot parameter is Custom or has no fit data, otherwise 2. If the spline does not have fit data, then the knot parameter should become Custom.

Common:				
	Degree	BL		degree of this spline
If (scenario==2) {				
	Fit Tol	BD	44	
	Beg tan vec	3BD	12	Beginning tangent direction vector (normalized).
	End tan vec	3BD	13	Ending tangent direction vector (normalized).
	num fit pts	BL	74	Number of fit points. Stored as a LONG, although it is defined in DXF as a short. You can see this if you create a spline with >=256 fit points

```
}
if (scenario==1) {
       Rational                B            flag bit 2
       Closed                  B            flag bit 0
       Periodic                B            flag bit 1
       Knot tol               BD      42
       Ctrl tol               BD      43
       Numknots               BL      72    This is stored as a LONG, although it is defined in
                                            DXF as a short. You can see this if you create a
                                            spline with >=256 knots.
       Numctrlpts             BL      73    Number of 10's (and 41's, if weighted) that follow.
                                            Same, stored as LONG, defined in DXF as a short.
       Weight                  B            Seems to be an echo of the 4 bit on the flag for
                                            "weights present".
}
Repeat numknots times {
       Knot                   BD            knot value
}
Repeat numctrlpts times {
       Control pt            3BD      10
       Weight                  D      41    if present as indicated by 4 bit on flag
}
Repeat numfitpts times {
       Fit pt                3BD
}
       Common Entity Handle Data
       CRC                     X     ---
```

20.4.40.1 Example:

```
OBJECT: spline (24H), len 61H (97), handle: 01 01

   01AC5 61 00                   a.        0110 0001 0000 0000

   01AC7 49 00 80 40 66 A8 10 00   I..@f...   0100 1001 0000 0000 1000 0000 0100 0000 0110 0110 1010 1000 0001 0000 0000 0000

   01ACF 05 5B 20 48 19 77 7B AF   .[ H.w{.   0000 0101 0101 1011 0010 0000 0100 1000 0001 1001 0111 0111 0111 1011 1010 1111

   01AD7 B3 BE F9 B6 7B 55 48 21   ....{UH!   1011 0011 1011 1110 1111 1001 1011 0110 0111 1011 0101 0101 0100 1000 0010 0001

   01ADF D1 F6 EC 49 3D 16 1C 80   ...I=...   1101 0001 1111 0110 1110 1100 0100 1001 0011 1101 0001 0110 0001 1100 1000 0000

   01AE7 60 3C 07 63 43 16 F8 9F   `<.cC...   0110 0000 0011 1100 0000 0111 0110 0011 0100 0011 0001 0110 1111 1000 1001 1111

   01AEF C0 E4 4E 53 64 CA 30 B2   ..NSd.0.   1100 0000 1110 0100 0100 1110 0101 0011 0110 0100 1100 1010 0011 0000 1011 0010

   01AF7 01 F0 33 3C 1C A7 C2 0E   ..3<....   0000 0001 1111 0000 0011 0011 0011 1100 0001 1100 1010 0111 1100 0010 0000 1110
```

```
    01AFF 81 01 85 80 9A FE 6F 63   ......oc   1000 0001 0000 0001 1000 0101 1000 0000 1001 1010 1111 1110 0110 1111 0110 0011

    01B07 88 02 07 89 BE 3C 1B 4F   .....<.O   1000 1000 0000 0010 0000 0111 1000 1001 1011 1110 0011 1100 0001 1011 0100 1111

    01B0F 51 FC 5F 51 14 FA 2F CF   Q._Q../.   0101 0001 1111 1100 0101 1111 0101 0001 0001 0100 1111 1010 0010 1111 1100 1111

    01B17 94 20 04 18 CB 8B BB C6   . ......   1001 0100 0010 0000 0000 0100 0001 1000 1100 1011 1000 1011 1011 1011 1100 0110

    01B1F 9D 67 F1 82 88 7E 08 13   .g...~..   1001 1101 0110 0111 1111 0001 1000 0010 1000 1000 0111 1110 0000 1000 0001 0011

    01B27 05                        .          0000 0101

    01B28 99 F5                     crc

OBJECT: spline (24H), len BBH (187), handle: 01 02

    01B2A BB 00                     ..         1011 1011 0000 0000

    01B2C 49 00 80 40 A5 C8 28 00   I..@..(.   0100 1001 0000 0000 1000 0000 0100 0000 1010 0101 1100 1000 0010 1000 0000 0000

    01B34 05 7B 20 28 18 12 2B EF   .{ (..+.   0000 0101 0111 1011 0010 0000 0010 1000 0001 1000 0001 0010 0010 1011 1110 1111

    01B3C 26 BC B5 DE 8F 84 8A FB   &.......   0010 0110 1011 1100 1011 0101 1101 1110 1000 1111 1000 0100 1000 1010 1111 1011

    01B44 C9 AF 2D 77 A3 E4 29 06   ..-w..).   1100 1001 1010 1111 0010 1101 0111 0111 1010 0011 1110 0100 0010 1001 0000 0110

    01B4C 55 01 E2 91 A5 D0 17 80   U.......   0101 0101 0000 0001 1110 0010 1001 0001 1010 0101 1101 0000 0001 0111 1000 0000

    01B54 88 04 31 35 FD 44 D0 08   ..15.D..   1000 1000 0000 0100 0011 0001 0011 0101 1111 1101 0100 0100 1101 0000 0000 1000

    01B5C AA 01 79 09 BE 48 77 C4   ..y..Hw.   1010 1010 0000 0001 0111 1001 0000 1001 1011 1110 0100 1000 0111 0111 1100 0100

    01B64 48 80 5E 42 6F 92 1D F1   H.^Bo...   0100 1000 1000 0000 0101 1110 0100 0010 0110 1111 1001 0010 0001 1101 1111 0001

    01B6C 12 20 17 90 9B E4 87 7C   . .....|   0001 0010 0010 0000 0001 0111 1001 0000 1001 1011 1110 0100 1000 0111 0111 1100

    01B74 44 88 05 E4 26 F9 21 DF   D...&.!.   0100 0100 1000 1000 0000 0101 1110 0100 0010 0110 1111 1001 0010 0001 1101 1111

    01B7C 11 22 00 51 81 5C A9 30   .".Q.\.0   0001 0001 0010 0010 0000 0000 0101 0001 1000 0001 0101 1100 1010 1001 0011 0000

    01B84 EA 18 80 40 1B 4B CF 66   ...@.K.f   1110 1010 0001 1000 1000 0000 0100 0000 0001 1011 0100 1011 1100 1111 0110 0110

    01B8C F3 7D 9F C4 63 6D AF 9B   .}..cm..   1111 0011 0111 1101 1001 1111 1100 0100 0110 0011 0110 1101 1010 1111 1001 1011

    01B94 7D A8 82 00 51 75 80 5C   }...Qu.\   0111 1101 1010 1000 1000 0010 0000 0000 0101 0001 0111 0101 1000 0000 0101 1100

    01B9C B0 1C 0C 81 09 D0 81 4C   .......L   1011 0000 0001 1100 0000 1100 1000 0001 0000 1001 1101 0000 1000 0001 0100 1100

    01BA4 07 B7 62 A8 05 58 F7 A1   ..b..X..   0000 0111 1011 0111 0110 0010 1010 1000 0000 0101 0101 1000 1111 0111 1010 0001

    01BAC 59 01 E8 9A 04 5B 36 58   Y....[6X   0101 1001 0000 0001 1110 1000 1001 1010 0000 0100 0101 1011 0011 0110 0101 1000

    01BB4 8C 6B 16 0E 20 1F C2 20   .k.. ..    1000 1100 0110 1011 0001 0110 0000 1110 0010 0000 0001 1111 1100 0010 0010 0000

    01BBC 5C 89 E6 DA 97 F1 0C 22   \......"   0101 1100 1000 1001 1110 0110 1101 1010 1001 0111 1111 0001 0000 1100 0010 0010

    01BC4 B6 2B 1A 3A 48 80 23 05   .+.:H.#.   1011 0110 0010 1011 0001 1010 0011 1010 0100 1000 1000 0000 0010 0011 0000 0101

    01BCC 1E F2 8C 22 74 9F C6 74   ..."t..t   0001 1110 1111 0010 1000 1100 0010 0010 0111 0100 1001 1111 1100 0110 0111 0100

    01BD4 99 BC 06 F0 C9 42 01 A0   .....B..   1001 1001 1011 1100 0000 0110 1111 0000 1100 1001 0100 0010 0000 0001 1010 0000

    01BDC 40 6E 0F 6C A7 F0 7F 18   @n.l....   0100 0000 0110 1110 0000 1111 0110 1100 1010 0111 1111 0000 0111 1111 0001 1000
```

```
   01BE4 28 87 D7                (..       0010 1000 1000 0111 1101 0111

   01BE7 E3 F3                   crc
```

20.4.41 REGION (37) 3DSOLID (38) BODY (39)

These are all ACIS entities. We do not have a complete decryption of these, although we can step them, and write them, properly.

```
        Common Entity Data
```

After this, data are read as groups starting with a short which seems to indicate the type of data. This is not completely understood. The current algorithm is:

```
        ACIS Empty bit B       X       If 1, then no data follows

        Unknown bit    B       X

        Version        BS              Can be 1 or 2.

Version == 1 (following 2 items repeat until Block Size is 0):
        Block Size             BL       X      Number of bytes of SAT data in this block. if value
                                               is between 0x20 and 0x7E, calculate 0x9F-the value
                                               to get the real character.  If it's a tab, we
                                               convert to a space.

        SAT data       RC      X       Length is specified by the above count.

Version == 2:

        Immediately following will be an acis file.  Header value of "ACIS BinaryFile" indicates
SAB, otherwise it is a text SAT file.  No length is given.  SAB files will end with
"End\x0E\x02of\x0E\x04ACIS\x0D\x04data".  SAT files must be parsed to find the end.



Common:
        Wireframe data present  B       X      True if wireframe data is present
Wireframe == true:
        Point present           B       X      If true, following point is present, otherwise
                                               assume 0,0,0 for point
        Point                 3BD       X      Present if above bit is 1.
        Num IsoLines           BL       X
        IsoLines present        B       X      If true, isoline data is present.
        Num Wires              BL       X      Number of ISO lines that follow.
Repeat Num Wires times:
        Wire type              RC       X
        Wire selection marker  BL       X
        Wire color             BS       X
        Wire Acis Index        BL       X
        Wire # of points       BL       X
        Point                3BD      X      Repeats "Wire # of points" times.
        Transform present      B      X
If "Transform present" == 1:
        X Axis               3BD      X
        Y Axis               3BD      X
        Z Axis               3BD      X
        Translation          3BD      X
        Scale                 BD      X
        Has rotation           B      X
        Has reflection         B      X
        Has shear              B      X
End If
End Repeat
        Num. silhouettes      BL      X
Repeat "Num. silhouettes" times:
        VP id                 BL      X
        VP Target            3BD      X
        VP dir. From target  3BD      X
        VP up dir.           3BD      X
        VP perspective         B      X
        Num Wires             BL      X
Repeat "Num Wires" times:
        Same as above
End Repeat
        ACIS Empty bit         B      X      Normally 1.  If 0, then acis data follows in the
                                             same format as described above, except no wireframe
                                             of silhouette data will be present (no empty bits
                                             for these items either).
R2007+:
        Unknown               BL
Common:
        Common Entity Handle Data
R2007+:
                               H    350      History ID
Common:
        CRC                    X    ---
```

20.4.41.1 Example:

```
OBJECT: region (25H), len 22DH (557), handle: 01 03

   01BE9 2D 02                 -.        0010 1101 0000 0010

   01BEB 49 40 80 40 E2 48 88 00   I@.@.H..   0100 1001 0100 0000 1000 0000 0100 0000 1110 0010 0100 1000 1000 1000 0000 0000
```

```
01BF3 05 7B 28 08 0C 04 00 00   .{(.....   0000 0101 0111 1011 0010 1000 0000 1000 0000 1100 0000 0100 0000 0000 0000 0000

01BFB DC DE D2 40 DC DC 40 DC   ...@..@.   1101 1100 1101 1110 1101 0010 0100 0000 1101 1100 1101 1100 0100 0000 1101 1100

01C03 40 DE 40 40 40 40 40 40   @.@@@@@@   0100 0000 1101 1110 0100 0000 0100 0000 0100 0000 0100 0000 0100 0000 0100 0000

01C0B 40 40 40 40 1A 14 7A 60   @@@@..z`   0100 0000 0100 0000 0100 0000 0100 0000 0001 1010 0001 0100 0111 1010 0110 0000

01C13 76 4C 40 F6 E4 DC 40 F6   vL@...@.   0111 0110 0100 1100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110

01C1B DC 40 F6 E4 DC 40 F6 E4   .@...@..   1101 1100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1110 0100

01C23 DC 40 F8 1A 14 66 54 64   .@...fTd   1101 1100 0100 0000 1111 1000 0001 1010 0001 0100 0110 0110 0101 0100 0110 0100

01C2B 5E 40 F6 E4 DC 40 F6 E4   ^@...@..   0101 1110 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1110 0100

01C33 DC 40 F6 DA 40 F6 DE 40   .@..@..@   1101 1100 0100 0000 1111 0110 1101 1010 0100 0000 1111 0110 1101 1110 0100 0000

01C3B F8 1A 14 58 6E 74 66 66   ...Xntff   1111 1000 0001 1010 0001 0100 0101 1000 0110 1110 0111 0100 0110 0110 0110 0110

01C43 40 F6 E4 DC 40 F6 E4 DC   @...@...   0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1110 0100 1101 1100

01C4B 40 F6 E4 DC 40 F6 D8 40   @...@..@   0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1101 1000 0100 0000

01C53 F6 DC 40 F8 1A 14 72 7C   ..@...r|   1111 0110 1101 1100 0100 0000 1111 1000 0001 1010 0001 0100 0111 0010 0111 1100

01C5B 78 74 40 F6 E4 DC 40 F6   xt@...@.   0111 1000 0111 0100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110

01C63 E4 DC 40 F6 D6 40 F6 DA   ..@..@..   1110 0100 1101 1100 0100 0000 1111 0110 1101 0110 0100 0000 1111 0110 1101 1010

01C6B 40 F6 E4 DC 40 F6 D4 40   @...@..@   0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1101 0100 0100 0000

01C73 72 60 5A 50 7C 5A 76 40   r`ZP|Zv@   0111 0010 0110 0000 0101 1010 0101 0000 0111 1100 0101 1010 0111 0110 0100 0000

01C7B 76 60 54 7A 66 74 40 60   v`Tzft@`   0111 0110 0110 0000 0101 0100 0111 1010 0110 0110 0111 0100 0100 0000 0110 0000

01C83 54 56 40 F8 1A 14 66 60   TV@...f`   0101 0100 0101 0110 0100 0000 1111 1000 0001 1010 0001 0100 0110 0110 0110 0000

01C8B 60 5E 40 F6 E4 DC 40 F6   `^@...@.   0110 0000 0101 1110 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110

01C93 E4 DC 40 F6 D2 40 F6 D8   ..@..@..   1110 0100 1101 1100 0100 0000 1111 0110 1101 0010 0100 0000 1111 0110 1101 1000

01C9B 40 F8 1A 14 5E 66 7C 62   @...^f|b   0100 0000 1111 1000 0001 1010 0001 0100 0101 1110 0110 0110 0111 1100 0110 0010

01CA3 74 E4 58 54 5A 72 7C 78   t.XTZr|x   0111 0100 1110 0100 0101 1000 0101 0100 0101 1010 0111 0010 0111 1100 0111 1000

01CAB 74 40 F6 E4 DC 40 CE E2   t@...@..   0111 0100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1100 1110 1110 0010

01CB3 DC DE DC CC D0 D2 D8 DC   ........   1101 1100 1101 1110 1101 1100 1100 1100 1101 0000 1101 0010 1101 1000 1101 1100

01CBB DA D6 D6 D0 D6 CC CC D4   ........   1101 1010 1101 0110 1101 0110 1101 0000 1101 0110 1100 1100 1100 1100 1101 0100

01CC3 40 DC E2 CE D0 D6 DC D0   @.......   0100 0000 1101 1100 1110 0010 1100 1110 1101 0000 1101 0110 1101 1100 1101 0000

01CCB DC D2 DC CE D4 D4 DA DE   ........   1101 1100 1101 0010 1101 1100 1100 1110 1101 0100 1101 0100 1101 1010 1101 1110

01CD3 D2 DC D2 40 DE 40 DE 40   ...@.@.@   1101 0010 1101 1100 1101 0010 0100 0000 1101 1110 0100 0000 1101 1110 0100 0000

01CDB DE 40 DC 40 DC 40 DE 40   .@.@.@.@   1101 1110 0100 0000 1101 1100 0100 0000 1101 1100 0100 0000 1101 1110 0100 0000

01CE3 DE 40 DE 40 AC 40 AC 40   .@.@.@.@   1101 1110 0100 0000 1101 1110 0100 0000 1010 1100 0100 0000 1010 1100 0100 0000

01CEB AC 40 AC 40 F8 1A 14 78   .@.@...x   1010 1100 0100 0000 1010 1100 0100 0000 1111 1000 0001 1010 0001 0100 0111 1000
```

```
01CF3 60 74 76 70 74 40 F6 E4   `tvpt@..   0110 0000 0111 0100 0111 0110 0111 0000 0111 0100 0100 0000 1111 0110 1110 0100

01CFB DC 40 F6 D2 40 F6 D2 40   .@..@..@   1101 1100 0100 0000 1111 0110 1101 0010 0100 0000 1111 0110 1101 0010 0100 0000

01D03 F6 E4 DC 40 F6 D0 40 DE   ...@..@.   1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1101 0000 0100 0000 1101 1110

01D0B 40 F6 D6 40 F6 E4 DC 40   @..@...@   0100 0000 1111 0110 1101 0110 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000

01D13 F8 1A 14 74 76 70 74 40   ...tvpt@   1111 1000 0001 1010 0001 0100 0111 0100 0111 0110 0111 0000 0111 0100 0100 0000

01D1B F6 E4 DC 40 F6 CE 40 F6   ...@..@.   1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1100 1110 0100 0000 1111 0110

01D23 CE 40 F6 D2 40 F6 CC 40   .@..@..@   1100 1110 0100 0000 1111 0110 1101 0010 0100 0000 1111 0110 1100 1100 0100 0000

01D2B DE 40 F8 1A 14 52 74 5A   .@...RtZ   1101 1110 0100 0000 1111 1000 0001 1010 0001 0100 0101 0010 0111 0100 0101 1010

01D33 56 74 4E 40 F6 E4 DC 40   VtN@...@   0101 0110 0111 0100 0100 1110 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000

01D3B F6 D0 40 F6 DC DE 40 F8   ..@...@.   1111 0110 1101 0000 0100 0000 1111 0110 1101 1100 1101 1110 0100 0000 1111 1000

01D43 1A 14 74 66 66 6C 5E 58   ..tffl^X   0001 1010 0001 0100 0111 0100 0110 0110 0110 0110 0110 1100 0101 1110 0101 1000

01D4B 74 E4 78 54 5A 52 74 40   t.xTZRt@   0111 0100 1110 0100 0111 1000 0101 0100 0101 1010 0101 0010 0111 0100 0100 0000

01D53 F6 E4 DC 40 CE E2 DC DE   ...@....   1111 0110 1110 0100 1101 1100 0100 0000 1100 1110 1110 0010 1101 1100 1101 1110

01D5B DC CC D0 D2 D8 DC DA D6   ........   1101 1100 1100 1100 1101 0000 1101 0010 1101 1000 1101 1100 1101 1010 1101 0110

01D63 D6 D0 D4 DE DC D8 40 DC   ......@.   1101 0110 1101 0000 1101 0100 1101 1110 1101 1100 1101 1000 0100 0000 1101 1100

01D6B E2 CE D0 D6 DC D0 DC D2   ........   1110 0010 1100 1110 1101 0000 1101 0110 1101 1100 1101 0000 1101 1100 1101 0010

01D73 DC CE D4 D4 DA DE D2 D8   ........   1101 1100 1100 1110 1101 0100 1101 0100 1101 1010 1101 1110 1101 0010 1101 1000

01D7B D6 40 DE 40 DE 40 DE 40   .@.@.@.@   1101 0110 0100 0000 1101 1110 0100 0000 1101 1110 0100 0000 1101 1110 0100 0000

01D83 DC 40 DE E2 D2 D8 D0 D4   .@......   1101 1100 0100 0000 1101 1110 1110 0010 1101 0010 1101 1000 1101 0000 1101 0100

01D8B DE D2 D8 CE DA D2 D0 D4   ........   1101 1110 1101 0010 1101 1000 1100 1110 1101 1010 1101 0010 1101 0000 1101 0100

01D93 D6 DA CE DC D6 40 E4 DE   .....@..   1101 0110 1101 1010 1100 1110 1101 1100 1101 0110 0100 0000 1110 0100 1101 1110

01D9B E2 CC DA DA D0 DA D0 CC   ........   1110 0010 1100 1100 1101 1010 1101 1010 1101 0000 1101 1010 1101 0000 1100 1100

01DA3 DE DA D2 D6 D8 D6 D8 D0   ........   1101 1110 1101 1010 1101 0010 1101 0110 1101 1000 1101 0110 1101 1000 1101 0000

01DAB DA CC 40 DE 40 DE E2 D6   ..@.@...   1101 1010 1100 1100 0100 0000 1101 1110 0100 0000 1101 1110 1110 0010 1101 0110

01DB3 D4 D0 D4 DA D4 CC D4 DC   ........   1101 0100 1101 0000 1101 0100 1101 1010 1101 0100 1100 1100 1101 0100 1101 1100

01DBB D8 D0 CC DC DA D8 D4 D4   ........   1101 1000 1101 0000 1100 1100 1101 1100 1101 1010 1101 1000 1101 0100 1101 0100

01DC3 40 AC 40 AC 40 F8 1A 14   @.@.@...   0100 0000 1010 1100 0100 0000 1010 1100 0100 0000 1111 1000 0001 1010 0001 0100

01DCB 5E 60 6C 62 56 40 F6 E4   ^`lbV@..   0101 1110 0110 0000 0110 1100 0110 0010 0101 0110 0100 0000 1111 0110 1110 0100

01DD3 DC 40 CE E2 D0 D8 CC D6   .@......   1101 1100 0100 0000 1100 1110 1110 0010 1101 0000 1101 1000 1100 1100 1101 0110

01DDB CE DA D2 CC D4 DC DA DA   ........   1100 1110 1101 1010 1101 0010 1100 1100 1101 0100 1101 1100 1101 1010 1101 1010

01DE3 CC DA CE D0 40 DE E2 CC   ....@...   1100 1100 1101 1010 1100 1110 1101 0000 0100 0000 1101 1110 1110 0010 1100 1100

01DEB D4 DC D6 D6 D8 D0 DC D4   ........   1101 0100 1101 1100 1101 0110 1101 0110 1101 1000 1101 0000 1101 1100 1101 0100
```

```
    01DF3 CC DE CE D2 DA D2 DE CC   ........   1100 1100 1101 1110 1100 1110 1101 0010 1101 1010 1101 0010 1101 1110 1100 1100
    01DFB 40 DE 40 F8 1A 15 63 F6   @.@...c.   0100 0000 1101 1110 0100 0000 1111 1000 0001 1010 0001 0101 0110 0011 1111 0110
    01E03 D9 E9 E9 B1 A1 02 00 50   .......P   1101 1001 1110 1001 1110 1001 1011 0001 1010 0001 0000 0010 0000 0000 0101 0000
    01E0B B8 18 C3 37 F9 FA 7F 20   ...7...    1011 1000 0001 1000 1100 0011 0011 0111 1111 1001 1111 1010 0111 1111 0010 0000
    01E13 9A 98 28 87 80            ..(..      1001 1010 1001 1000 0010 1000 1000 0111 1000 0000
    01E18 07 33                     crc
OBJECT: 3d solid (26H), len 334H (820), handle: 01 04
    01E1A 34 03                     4.         0011 0100 0000 0011
    01E1C 49 80 80 41 24 30 C8 00   I..A$0..   0100 1001 1000 0000 1000 0000 0100 0001 0010 0100 0011 0000 1100 1000 0000 0000
    01E24 05 7B 28 0B E4 DC DE D2   .{(.....   0000 0101 0111 1011 0010 1000 0000 1011 1110 0100 1101 1100 1101 1110 1101 0010
    01E2C 40 D4 40 DC 40 DE 40 40   @.@.@.@@   0100 0000 1101 0100 0100 0000 1101 1100 0100 0000 1101 1110 0100 0000 0100 0000
    01E34 40 40 40 40 40 40 40 40   @@@@@@@@   0100 0000 0100 0000 0100 0000 0100 0000 0100 0000 0100 0000 0100 0000 0100 0000
    01E3C 40 1A 14 7A 60 76 4C 40   @..z`vL@   0100 0000 0001 1010 0001 0100 0111 1010 0110 0000 0111 0110 0100 1100 0100 0000
    01E44 F6 E4 DC 40 F6 DC 40 F6   ...@..@.   1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1101 1100 0100 0000 1111 0110
    01E4C E4 DC 40 F6 E4 DC 40 F8   ..@...@.   1110 0100 1101 1100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 1000
    01E54 1A 14 66 54 64 5E 40 F6   ..fTd^@.   0001 1010 0001 0100 0110 0110 0101 0100 0110 0100 0101 1110 0100 0000 1111 0110
    01E5C E4 DC 40 F6 E4 DC 40 F6   ..@...@.   1110 0100 1101 1100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110
    01E64 DA 40 F6 DE 40 F8 1A 14   .@..@...   1101 1010 0100 0000 1111 0110 1101 1110 0100 0000 1111 1000 0001 1010 0001 0100
    01E6C 58 6E 74 66 66 40 F6 E4   Xntff@..   0101 1000 0110 1110 0111 0100 0110 0110 0110 0110 0100 0000 1111 0110 1110 0100
    01E74 DC 40 F6 E4 DC 40 F6 E4   .@...@..   1101 1100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1110 0100
    01E7C DC 40 F6 D8 40 F6 DC 40   .@..@..@   1101 1100 0100 0000 1111 0110 1101 1000 0100 0000 1111 0110 1101 1100 0100 0000
    01E84 F8 1A 14 72 7C 78 74 40   ...r|xt@   1111 1000 0001 1010 0001 0100 0111 0010 0111 1100 0111 1000 0111 0100 0100 0000
    01E8C F6 E4 DC 40 F6 E4 DC 40   ...@...@   1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1110 0100 1101 1100 0100 0000
    01E94 F6 E4 DC 40 F6 DA 40 F6   ...@..@.   1111 0110 1110 0100 1101 1100 0100 0000 1111 0110 1101 1010 0100 0000 1111 0110
    01E9C E4 DC 40 F6 D6 40 72 60   ..@..@r`   1110 0100 1101 1100 0100 0000 1111 0110 1101 0110 0100 0000 0111 0010 0110 0000
    01EA4 5A 50 7C 5A 76 40 58 6C   ZP|Zv@Xl   0101 1010 0101 0000 0111 1100 0101 1010 0111 0110 0100 0000 0101 1000 0110 1100
    01EAC 62 70 66 74 40 F8 1A 14   bpft@...   0110 0010 0111 0000 0110 0110 0111 0100 0100 0000 1111 1000 0001 1010 0001 0100
    01EB4 58 5E 6E 74 5A 74 E4 58   X^ntZt.X   0101 1000 0101 1110 0110 1110 0111 0100 0101 1010 0111 0100 1110 0100 0101 1000
    01EBC 54 5A 72 7C 78 74 40 F6   TZr|xt@.   0101 0100 0101 1010 0111 0010 0111 1100 0111 1000 0111 0100 0100 0000 1111 0110
    01EC4 E4 DC 40 DC DE E2 DA D2   ..@.....   1110 0100 1101 1100 0100 0000 1101 1100 1101 1110 1110 0010 1101 1010 1101 0010
    01ECC DA D4 DE D8 D8 D6 D8 CC   ........   1101 1010 1101 0100 1101 1110 1101 1000 1101 1000 1101 0110 1101 1000 1100 1100
    01ED4 CE D8 DC DA D8 40 CE E2   .....@..   1100 1110 1101 1000 1101 1100 1101 1010 1101 1000 0100 0000 1100 1110 1110 0010
    01EDC D6 D6 D2 DC CE D6 DE D6   ........   1101 0110 1101 0110 1101 0010 1101 1100 1100 1110 1101 0110 1101 1110 1101 0110

    01EE4 CC D0 DE CC D4 CC D0 40   .......@   1100 1100 1101 0000 1101 1110 1100 1100 1101 0100 1100 1100 1101 0000 0100 0000

    01EEC DE 40 DE E2 CE D6 D8 DE   .@......   1101 1110 0100 0000 1101 1110 1110 0010 1100 1110 1101 0110 1101 1000 1101 1110

    01EF4 D4 D4 D2 D4 CE D4 DC DE   ........   1101 0100 1101 0100 1101 0010 1101 0100 1100 1110 1101 0100 1101 1100 1101 1110

    01EFC DC CC CC CC DA 40 DC 40   .....@.@   1101 1100 1100 1100 1100 1100 1100 1100 1101 1010 0100 0000 1101 1100 0100 0000

    01F04 DE 40 DE 40 DE 40 DE 40   .@.@.@.@   1101 1110 0100 0000 1101 1110 0100 0000 1101 1110 0100 0000 1101 1110 0100 0000

    01F0C DC 40 DE 40 AC 40 AC 40   .@.@.@.@   1101 1100 0100 0000 1101 1110 0100 0000 1010 1100 0100 0000 1010 1100 0100 0000

    01F14 AC 40 AC 40 F8 1A 15 60   .@.@...`   1010 1100 0100 0000 1010 1100 0100 0000 1111 1000 0001 1010 0001 0101 0110 0000

    01F1C 77 FC D6 B3 34 31 22 01   w...41".   0111 0111 1111 1100 1101 0110 1011 0011 0011 0100 0011 0001 0010 0010 0000 0001

    01F24 8D FE B4 78 E5 C8 40 81   ...x..@.   1000 1101 1111 1110 1011 0100 0111 1000 1110 0101 1100 1000 0100 0000 1000 0001

    01F2C 20 94 1C 0C FF FF FF FF    .......   0010 0000 1001 0100 0001 1100 0000 1100 1111 1111 1111 1111 1111 1111 1111 1111

    01F34 CF FF FF FF F4 0C 0E FF   ........   1100 1111 1111 1111 1111 1111 1111 1111 1111 0100 0000 1100 0000 1110 1111 1111

    01F3C 9A D6 66 86 24 40 32 3F   ..f.$@2?   1001 1010 1101 0110 0110 0110 1000 0110 0010 0100 0100 0000 0011 0010 0011 1111

    01F44 D6 8F 1C B9 08 10 02 84   ........   1101 0110 1000 1111 0001 1100 1011 1001 0000 1000 0001 0000 0000 0010 1000 0100

    01F4C A4 0D C4 FF AE AB F0 33   .......3   1010 0100 0000 1101 1100 0100 1111 1111 1010 1110 1010 1011 1111 0000 0011 0011

    01F54 FE 6B 59 9A 18 91 00 47   .kY....G   1111 1110 0110 1011 0101 1001 1001 1010 0001 1000 1001 0001 0000 0000 0100 0111

    01F5C F6 2D 7D 9A 69 1E 40 80   .-}.i.@.   1111 0110 0010 1101 0111 1101 1001 1010 0110 1001 0001 1110 0100 0000 1000 0000

    01F64 EF F9 AD 66 68 62 44 03   ...fhbD.   1110 1111 1111 1001 1010 1101 0110 0110 0110 1000 0110 0010 0100 0100 0000 0011

    01F6C 23 FD 68 F1 CB 90 81 00   #.h.....   0010 0011 1111 1101 0110 1000 1111 0001 1100 1011 1001 0000 1000 0001 0000 0000

    01F74 28 4A 40 DC 4F FA EA 3F   (J@.O..?   0010 1000 0100 1010 0100 0000 1101 1100 0100 1111 1111 1010 1110 1010 0011 1111

    01F7C 01 9F FF FF FF F9 FF FF   ........   0000 0001 1001 1111 1111 1111 1111 1111 1111 1111 1111 1001 1111 1111 1111 1111

    01F84 FF FE 81 81 9F F3 5A CC   ......Z.   1111 1111 1111 1110 1000 0001 1000 0001 1001 1111 1111 0011 0101 1010 1100 1100

    01F8C D0 C4 88 06 37 FA D1 E3   ....7...   1101 0000 1100 0100 1000 1000 0000 0110 0011 0111 1111 1010 1101 0001 1110 0011

    01F94 97 21 02 00 60 94 81 B8   .!..`...   1001 0111 0010 0001 0000 0010 0000 0000 0110 0000 1001 0100 1000 0001 1011 1000

    01F9C 9F F5 D5 7E 58 81 AF EA   ...~X...   1001 1111 1111 0101 1101 0101 0111 1110 0101 1000 1000 0001 1010 1111 1110 1010

    01FA4 05 9B 13 20 18 DF EB 47   ... ...G   0000 0101 1001 1011 0001 0011 0010 0000 0001 1000 1101 1111 1110 1011 0100 0111

    01FAC 8E 5C 84 08 10 19 FF 35   .\.....5   1000 1110 0101 1100 1000 0100 0000 1000 0001 0000 0001 1001 1111 1111 0011 0101

    01FB4 AC CD 0C 48 80 63 7F AD   ...H.c..   1010 1100 1100 1101 0000 1100 0100 1000 1000 0000 0110 0011 0111 1111 1010 1101

    01FBC 1E 39 72 10 20 06 09 48   .9r. ..H   0001 1110 0011 1001 0111 0010 0001 0000 0010 0000 0000 0110 0000 1001 0100 1000

    01FC4 1B 89 FF 5D 47 E0 33 FF   ...]G.3.   0001 1011 1000 1001 1111 1111 0101 1101 0100 0111 1110 0000 0011 0011 1111 1111

    01FCC FF FF FF 3F FF FF FF D0   ...?....   1111 1111 1111 1111 1111 1111 0011 1111 1111 1111 1111 1111 1111 1111 1101 0000

    01FD4 30 3B FE 6B 59 9A 18 91   0;.kY...   0011 0000 0011 1011 1111 1110 0110 1011 0101 1001 1001 1010 0001 1000 1001 0001
```

```
01FDC 00 C4 FF 5A 3C 72 E4 20   ...Z<r.    0000 0000 1100 0100 1111 1111 0101 1010 0011 1100 0111 0010 1110 0100 0010 0000

01FE4 40 0C 12 90 37 13 FE BA   @...7...   0100 0000 0000 1100 0001 0010 1001 0000 0011 0111 0001 0011 1111 1110 1011 1010

01FEC AF C0 CF F9 AD 66 68 62   .....fhb   1010 1111 1100 0000 1100 1111 1111 1001 1010 1101 0110 0110 0110 1000 0110 0010

01FF4 44 01 A4 10 7C E8 5E 50   D...|.^P   0100 0100 0000 0001 1010 0100 0001 0000 0111 1100 1110 1000 0101 1110 0101 0000

01FFC 89 02 03 BF E6 B5 99 A1   ........   1000 1001 0000 0010 0000 0011 1011 1111 1110 0110 1011 0101 1001 1001 1010 0001

02004 89 10 0C 4F F5 A3 C7 2E   ...O....   1000 1001 0001 0000 0000 1100 0100 1111 1111 0101 1010 0011 1100 0111 0010 1110

0200C 42 04 00 C1 29 03 71 3F   B...).q?   0100 0010 0000 0100 0000 0000 1100 0001 0010 1001 0000 0011 0111 0001 0011 1111

02014 EB A8 FC 06 7F FF FF FF   ........   1110 1011 1010 1000 1111 1100 0000 0110 0111 1111 1111 1111 1111 1111 1111 1111

0201C E7 FF FF FF FA 06 08 7F   ........   1110 0111 1111 1111 1111 1111 1111 1111 1111 1010 0000 0110 0000 1000 0111 1111

02024 CD 6B 33 43 12 20 18 DF   .k3C. ..   1100 1101 0110 1011 0011 0011 0100 0011 0001 0010 0010 0000 0001 1000 1101 1111

0202C EB 47 8E 5C 84 08 01 82   .G.\....   1110 1011 0100 0111 1000 1110 0101 1100 1000 0100 0000 1000 0000 0001 1000 0010

02034 52 06 E2 7F D7 55 F8 D7   R....U..   0101 0010 0000 0110 1110 0010 0111 1111 1101 0111 0101 0101 1111 1000 1101 0111

0203C F5 AD B1 83 AC 44 80 61   .....D.a   1111 0101 1010 1101 1011 0001 1000 0011 1010 1100 0100 0100 1000 0000 0110 0001

02044 FF AD 1E 39 72 10 20 40   ...9r. @   1111 1111 1010 1101 0001 1110 0011 1001 0111 0010 0001 0000 0010 0000 0100 0000

0204C 87 FC D6 B3 34 31 22 01   ....41".   1000 0111 1111 1100 1101 0110 1011 0011 0011 0100 0011 0001 0010 0010 0000 0001

02054 8D FE B4 78 E5 C8 40 80   ...x..@.   1000 1101 1111 1110 1011 0100 0111 1000 1110 0101 1100 1000 0100 0000 1000 0000

0205C 18 25 20 6E 27 FD 75 1F   .% n'.u.   0001 1000 0010 0101 0010 0000 0110 1110 0010 0111 1111 1101 0111 0101 0001 1111

02064 80 8F FF FF FF FC FF FF   ........   1000 0000 1000 1111 1111 1111 1111 1111 1111 1111 1111 1100 1111 1111 1111 1111

0206C FF FF 40 CA A3 08 AD 62   ..@....b   1111 1111 1111 1111 0100 0000 1100 1010 1010 0011 0000 1000 1010 1101 0110 0010

02074 E5 52 34 03 23 FD 68 F1   .R4.#.h.   1110 0101 0101 0010 0011 0100 0000 0011 0010 0011 1111 1101 0110 1000 1111 0001

0207C CB 90 81 00 5B E6 0C 01   ....[...   1100 1011 1001 0000 1000 0001 0000 0000 0101 1011 1110 0110 0000 1100 0000 0001

02084 80 13 E3 BF 37 25 E4 B5   ....7%..   1000 0000 0001 0011 1110 0011 1011 1111 0011 0111 0010 0101 1110 0100 1011 0101

0208C B2 BB 48 D0 0F 62 D3 7F   ..H..b..   1011 0010 1011 1011 0100 1000 1101 0000 0000 1111 0110 0010 1101 0011 0111 1111

02094 FC 5E C2 14 01 6F 98 30   .^...o.0   1111 1100 0101 1110 1100 0010 0001 0100 0000 0001 0110 1111 1001 1000 0011 0000

0209C 06 00 4F 8E FC DC 97 92   ..O.....   0000 0110 0000 0000 0100 1111 1000 1110 1111 1100 1101 1100 1001 0111 1001 0010

020A4 D6 CA ED 23 40 0B 28 FF   ...#@.(.   1101 0110 1100 1010 1110 1101 0010 0011 0100 0000 0000 1011 0010 1000 1111 1111

020AC 7C 8F 2E 07 D0 05 BE 60   |......`   0111 1100 1000 1111 0010 1110 0000 0111 1101 0000 0000 0101 1011 1110 0110 0000

020B4 C0 18 01 3E 3B F0 11 FF   ...>;...   1100 0000 0001 1000 0000 0001 0011 1110 0011 1011 1111 0000 0001 0001 1111 1111

020BC FF FF FF 9F FF FF FF E8   ........   1111 1111 1111 1111 1111 1111 1001 1111 1111 1111 1111 1111 1111 1111 1110 1000

020C4 18 D7 F5 AD B1 83 AC 44   .......D   0001 1000 1101 0111 1111 0101 1010 1101 1011 0001 1000 0011 1010 1100 0100 0100

020CC 80 64 FF AD 1E 39 72 10   .d...9r.   1000 0000 0110 0100 1111 1111 1010 1101 0001 1110 0011 1001 0111 0010 0001 0000

020D4 20 45 EF E5 C2 BC A5 71    E.....q   0010 0000 0100 0101 1110 1111 1110 0101 1100 0010 1011 1100 1010 0101 0111 0001
```

```
   020DC 1A 00 8B 93 7D CC 84 B4   ....}...   0001 1010 0000 0000 1000 1011 1001 0011 0111 1101 1100 1100 1000 0100 1011 0100

   020E4 44 81 17 9F 97 0A F2 95   D.......   0100 0100 1000 0001 0001 0111 1001 1111 1001 0111 0000 1010 1111 0010 1001 0101

   020EC C4 68 04 73 67 71 1A 1E   .h.sgq..   1100 0100 0110 1000 0000 0100 0111 0011 0110 0111 0111 0001 0001 1010 0001 1110

   020F4 E8 F2 04 02 3F FF FF FF   ....?...   1110 1000 1111 0010 0000 0100 0000 0010 0011 1111 1111 1111 1111 1111 1111 1111

   020FC F3 FF FF FF FD 03 2A 8C   ......*.   1111 0011 1111 1111 1111 1111 1111 1111 1111 1101 0000 0011 0010 1010 1000 1100

   02104 22 B5 8B 95 48 D0 0C 8F   "...H...   0010 0010 1011 0101 1000 1011 1001 0101 0100 1000 1101 0000 0000 1100 1000 1111

   0210C F5 A3 C7 2E 42 04 01 6F   ....B..o   1111 0101 1010 0011 1100 0111 0010 1110 0100 0010 0000 0100 0000 0001 0110 1111

   02114 98 30 06 00 4F 8C FC DC   .0..O...   1001 1000 0011 0000 0000 0110 0000 0000 0100 1111 1000 1100 1111 1100 1101 1100

   0211C 97 92 D6 CA ED 23 40 3D   .....#@=   1001 0111 1001 0010 1101 0110 1100 1010 1110 1101 0010 0011 0100 0000 0011 1101

   02124 8B 4D FF F1 7B 08 50 05   .M..{.P.   1000 1011 0100 1101 1111 1111 1111 0001 0111 1011 0000 1000 0101 0000 0000 0101

   0212C BE 60 C0 18 01 3E 33 F3   .`...>3.   1011 1110 0110 0000 1100 0000 0001 1000 0000 0001 0011 1110 0011 0011 1111 0011

   02134 72 5E 4B 5B 2B B4 8D 00   r^K[+...   0111 0010 0101 1110 0100 1011 0101 1011 0010 1011 1011 0100 1000 1101 0000 0000

   0213C 2C A3 FD F2 3C B8 1F 40   ,...<..@   0010 1100 1010 0011 1111 1101 1111 0010 0011 1100 1011 1000 0001 1111 0100 0000

   02144 16 F9 83 00 60 04 F8 CF   ....`...   0001 0110 1111 1001 1000 0011 0000 0000 0110 0000 0000 0100 1111 1000 1100 1111

   0214C D4 C1 44 3E               ..D>       1101 0100 1100 0001 0100 0100 0011 1110

   02150 5A C5                     crc
```

20.4.42 RAY (40)

Common Entity Data		
Point	3BD	10
Vector	3BD	11
Common Entity Handle Data		
CRC	X	---

20.4.42.1 Example:

```
OBJECT: ray (28H), len 2FH (47), handle: 01 06

   02185 2F 00                     /.         0010 1111 0000 0000

   02187 4A 00 80 41 A2 E8 08 00   J..A....   0100 1010 0000 0000 1000 0000 0100 0001 1010 0010 1110 1000 0000 1000 0000 0000

   0218F 05 7B 12 4C 98 47 CA EF   .{.L.G..   0000 0101 0111 1011 0001 0010 0100 1100 1001 1000 0100 0111 1100 1010 1110 1111

   02197 C4 A8 00 84 0B FC 98 72   .......r   1100 0100 1010 1000 0000 0000 1000 0100 0000 1011 1111 1100 1001 1000 0111 0010

   0219F 3F F9 FC 0F 91 CC D7 E5   ?.......   0011 1111 1111 1001 1111 1100 0000 1111 1001 0001 1100 1100 1101 0111 1110 0101

   021A7 CD 71 5F 99 7D 7E 4D 05   .q_.}~M.   1100 1101 0111 0001 0101 1111 1001 1001 0111 1101 0111 1110 0100 1101 0000 0101

   021AF C1 3D 47 F1 82 88 78      .=G...x    1100 0001 0011 1101 0100 0111 1111 0001 1000 0010 1000 1000 0111 1000

   021B6 AD CF                     crc
```

20.4.43 XLINE (41)

Same as RAY (40) — except for type code.

20.4.43.1 Example:

```
OBJECT: const line (29H), len 2FH (47), handle: 01 05
    02152 2F 00                    /.        0010 1111 0000 0000
    02154 4A 40 80 41 62 E8 08 00  J@.Ab...  0100 1010 0100 0000 1000 0000 0100 0001 0110 0010 1110 1000 0000 1000 0000 0000
    0215C 05 7B 09 95 83 71 C8 B2  .{...q..  0000 0101 0111 1011 0000 1001 1001 0101 1000 0011 0111 0001 1100 1000 1011 0010
    02164 03 E8 03 C1 22 6C 91 8A  ...."l..  0000 0011 1110 1000 0000 0011 1100 0001 0010 0010 0110 1100 1001 0001 1000 1010
    0216C 28 4A 04 28 B1 0A 5D F6  (J.(..].  0010 1000 0100 1010 0000 0100 0010 1000 1011 0001 0000 1010 0101 1101 1111 0110
    02174 48 76 1F 8D E0 E8 59 D8  Hv....Y.  0100 1000 0111 0110 0001 1111 1000 1101 1110 0000 1110 1000 0101 1001 1101 1000
    0217C 7A FB 87 F1 82 88 78     z.....x   0111 1010 1111 1011 1000 0111 1111 0001 1000 0010 1000 1000 0111 1000
    02183 FE 9C                    crc
```

20.4.44 DICTIONARY (42)

Basically a list of pairs of string/objhandle that constitute the dictionary entries.

	Length	MS	---	Entity length (not counting itself or CRC).
	Type	S	0	42 (internal DWG type code).
R2000+:				
	Obj size	RL		size of object in bits, not including end handles
Common:				
	Handle	H	5	Length (char) followed by the handle bytes.
	EED	X	-3	See EED section.
R13-R14 Only:				
	Obj size	RL		size of object in bits, not including end handles
Common:				
	Numreactors	S		number of reactors in this object
R2004+:				
	XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:				
	Numitems	L		number of dictonary items
R14 Only:				
	Unknown R14	RC		Unknown R14 byte, has always been 0
R2000+:				
	Cloning flag	BS	281	
	Hard Owner flag	RC	280	
Common:				
	Text	TV		string name of dictionary entry, numitems entries
	Handle refs	H		parenthandle (soft relative pointer)
				[Reactors (soft pointer)]
				xdicobjhandle (hard owner)

			itemhandles (soft owner)

20.4.44.1 *Example:*

```
OBJECT: dictionary (2AH), len 2CH (44), handle: 0C

   0254B 2C 00                   ,.        0010 1100 0000 0000

   0254D 4A 80 43 22 C0 10 00 09   J.C"....   0100 1010 1000 0000 0100 0011 0010 0010 1100 0000 0001 0000 0000 0000 0000 1001

   02555 02 00 42 90 50 D0 51 17   ..B.P.Q.   0000 0010 0000 0000 0100 0010 1001 0000 0101 0000 1101 0000 0101 0001 0001 0111

   0255D D1 D4 93 D5 54 10 F4 14   ....T...   1101 0001 1101 0100 1001 0011 1101 0101 0101 0100 0001 0000 1111 0100 0001 0100

   02565 34 14 45 F4 D4 C4 94 E4   4.E.....   0011 0100 0001 0100 0100 0101 1111 0100 1101 0100 1100 0100 1001 0100 1110 0100

   0256D 55 35 45 94 C4 54 03 02   U5E..T..   0101 0101 0011 0101 0100 0101 1001 0100 1100 0100 0101 0100 0000 0011 0000 0010

   02575 10 D2 10 EC               ....       0001 0000 1101 0010 0001 0000 1110 1100

   02579 D2 36                     crc
```

20.4.45 DICTIONARYWDFLT

Same as the DICTIONARY object with the following additional fields:

	H	7	Default entry (hard pointer)

20.4.46 MTEXT (44)

Common Entity (Handle) Data			
Insertion pt3	BD	10	First picked point. (Location relative to text depends on attachment point (71).)
Extrusion	3BD	210	Undocumented; appears in DXF and entget, but ACAD doesn't even bother to adjust it to unit length.
X-axis dir	3BD	11	Apparently the text x-axis vector. (Why not just a rotation?) ACAD maintains it as a unit vector.
Common:			
Rect width	BD	41	Reference rectangle width (width picked by the user).
R2007+:			
Rect height	BD	46	Reference rectangle height.
Common:			
Text height	BD	40	Undocumented
Attachment	BS	71	Similar to justification; see DXF doc
Drawing dir	BS	72	Left to right, etc.; see DXF doc
Extents	ht	BD	--- Undocumented and not present in DXF or entget
Extents wid	BD	---	Undocumented and not present in DXF or entget

			The extents rectangle, when rotated the same as the text, fits the actual text image on the screen (altough we've seen it include an extra row of text in height).
Text	TV	1	All text in one long string (without '\n's
		3	for line wrapping). ACAD seems to add braces ({ }) and backslash-P's to indicate paragraphs based on the "\r\n"'s found in the imported file. But, all the text is in this one long string -- not broken into 1- and 3-groups as in DXF and entget.
			ACAD's entget breaks this string into 250-char pieces (not 255 as doc'd) – even if it's mid-word. The 1-group always gets the tag end; therefore, the 3's are always 250 chars long.
	H	7	STYLE (hard pointer)
R2000+:			
Linespacing Style	BS	73	
Linespacing Factor	BD	44	
Unknown bit	B		
R2004+:			
Background flags	BL	90	0 = no background, 1 = background fill, 2 = background fill with drawing fill color, 0x10 = text frame (R2018+)
IF background flags has bit 0x01 set, or in case of R2018 bit 0x10:			
Background scale factor	BL	45	default = 1.5
Background color	CMC	63	
Background transparency	BL	441	
END IF background flags 0x01/0x10			
R2018+			
Is NOT annotative	B		
IF MTEXT is not annotative			
Version	BS		Default 0
Default flag	B		Default true
BEGIN REDUNDANT FIELDS (see above for descriptions)			
Registered application	H		Hard pointer
Attachment point	BL		
X-axis dir	3BD	10	
Insertion point	3BD	11	
Rect width	BD	40	
Rect height	BD	41	
Extents width	BD	42	
Extents height	BD	43	
END REDUNDANT FIELDS			

```
       Column type            BS      71      0 = No columns, 1 = static columns, 2 = dynamic
                                              columns
IF Has Columns data (column type is not 0)
       Column height count    BL      72
       Columnn width          BD      44
       Gutter                 BD      45
       Auto height?            B      73
       Flow reversed?          B      74
IF not auto height and column type is dynamic columns
REPEAT Column heights
       Column height          BD      46
END REPEAT
END IF (has column heights)
END IF (has columns data)
END IF (not annotative)
Common:
       CRC                     X     ---
```

20.4.46.1 Example:

```
OBJECT: mtext (2CH), len 4DH (77), handle: CE

   00515 4D 00                    M.        0100 1101 0000 0000

   00517 4B 00 73 A0 C8 10 00 01   K.s.....   0100 1011 0000 0000 0111 0011 1010 0000 1100 1000 0001 0000 0000 0000 0000 0001

   0051F 33 0F AE 2B 5E AE E0 84   3..+^...   0011 0011 0000 1111 1010 1110 0010 1011 0101 1110 1010 1110 1110 0000 1000 0100

   00527 48 04 88 93 FD FD 9A 00   H.......   0100 1000 0000 0100 1000 1000 1001 0011 1111 1101 1111 1101 1001 1010 0000 0000

   0052F FA 05 4B 50 15 AF 46 E0   ..KP..F.   1111 1010 0000 0101 0100 1011 0101 0000 0001 0101 1010 1111 0100 0110 1110 0000

   00537 7A 15 8E 7E 82 A0 20 6E   z..~.. n   0111 1010 0001 0101 1000 1110 0111 1110 1000 0010 1010 0000 0010 0000 0110 1110

   0053F BD 1B 81 E8 56 39 F9 9B   ....V9..   1011 1101 0001 1011 1000 0001 1110 1000 0101 0110 0011 1001 1111 1001 1001 1011

   00547 99 99 99 99 99 E0 7E 85   ......~.   1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1110 0000 0111 1110 1000 0101

   0054F AE 20 98 9D A9 18 17 1B   . ......   1010 1110 0010 0000 1001 1000 1001 1101 1010 1001 0001 1000 0001 0111 0001 1011

   00557 1B 19 1B 60 82 18 28 87   ...`..(.   0001 1011 0001 1001 0001 1011 0110 0000 1000 0010 0001 1000 0010 1000 1000 0111

   0055F A8 89 A8 88 00           .....      1010 1000 1000 1001 1010 1000 1000 1000 0000 0000

   00564 6F F0                    crc
```

20.4.47 LEADER (45)

```
       Common Entity Data
       Unknown bit             B     ---      Always seems to be 0.
       Annot type             BS     ---      Annotation type (NOT bit-coded):
                                              Value 0 : MTEXT
```

				Value 1 : TOLERANCE
				Value 2 : INSERT
				Value 3 : None
	path type	BS	---	
	numpts	BL	---	number of points
	point	3BD	10	As many as counter above specifies.
	Origin	3BD	---	The leader plane origin (by default it’s the first point).
	Extrusion	3BD	210	
	x direction	3BD	211	
	offsettoblockinspt	3BD	212	Used when the BLOCK option is used. Seems to be an unused feature.
R14+:				
	Endptproj	3BD	---	A non-planar leader gives a point that projects the endpoint back to the annotation. It's the offset from the endpoint of the leader to the annotation, taking into account the extrusion direction.
R13-R14 Only:				
	DIMGAP	BD	---	The value of DIMGAP in the associated DIMSTYLE at the time of creation, multiplied by the dimscale in that dimstyle.
Common:				
	Box height	BD	40	MTEXT extents height. (A text box is slightly taller, probably by some DIMvar amount.)
	Box width	BD	41	MTEXT extents width. (A text box is slightly wider, probably by some DIMvar amount.)
	Hooklineonxdir	B		hook line is on x direction if 1
	Arrowheadon	B		arrowhead on indicator
R13-R14 Only:				
	Arrowheadtype	BS		arrowhead type
	Dimasz	BD		DIMASZ at the time of creation, multiplied by DIMSCALE
	Unknown	B		
	Unknown	B		
	Unknown	BS		
	Byblockcolor	BS		
	Unknown	B		
	Unknown	B		
R2000+:				
	Unknown	BS		
	Unknown	B		
	Unknown	B		
Common:				
	Common Entity Handle Data			

```
                              H     340     Associated annotation
                                            activated in R14. (hard pointer)
                              H       2     DIMSTYLE (hard pointer)
     CRC                      X     ---
```

20.4.47.1 Example:

```
OBJECT: leader (2DH), len 80H (128), handle: 01 09

   02213 80 00                   ..        1000 0000 0000 0000

   02215 4B 40 80 42 65 20 18 00   K@.Be ..   0100 1011 0100 0000 1000 0000 0100 0010 0110 0101 0010 0000 0001 1000 0000 0000

   0221D 05 5B 29 03 25 AD 59 2D   .[).%.Y-   0000 0101 0101 1011 0010 1001 0000 0011 0010 0101 1010 1101 0101 1001 0010 1101

   02225 08 7D C9 50 04 41 FF AB   .}.P.A..   0000 1000 0111 1101 1100 1001 0101 0000 0000 0100 0100 0001 1111 1111 1010 1011

   0222D AF A2 81 04 08 2E E6 9D   ........   1010 1111 1010 0010 1000 0001 0000 0100 0000 1000 0010 1110 1110 0110 1001 1101

   02235 29 5D 0C 21 40 1C 3C C0   )].!@.<.   0010 1001 0101 1101 0000 1100 0010 0001 0100 0000 0001 1100 0011 1100 1100 0000

   0223D 0F B5 ED 05 D0 20 50 04   ..... P.   0000 1111 1011 0101 1110 1101 0000 0101 1101 0000 0010 0000 0101 0000 0000 0100

   02245 84 77 1E 34 65 00 78 56   .w.4e.xV   1000 0100 0111 0111 0001 1110 0011 0100 0110 0101 0000 0000 0111 1000 0101 0110

   0224D 18 21 BF AB 15 40 89 6B   .!...@.k   0001 1000 0010 0001 1011 1111 1010 1011 0001 0101 0100 0000 1000 1001 0110 1011

   02255 56 4B 42 1F 72 54 01 10   VKB.rT..   0101 0110 0100 1011 0100 0010 0001 1111 0111 0010 0101 0100 0000 0001 0001 0000

   0225D 7F EA EB E8 A0 41 02 A5   .....A..   0111 1111 1110 1010 1110 1011 1110 1000 1010 0000 0100 0001 0000 0010 1010 0101

   02265 AA AA 02 B5 E8 DC 0F 42   .......B   1010 1010 1010 1010 0000 0010 1011 0101 1110 1000 1101 1100 0000 1111 0100 0010

   0226D AD CF C0 AD 7A 37 03 D0   ....z7..   1010 1101 1100 1111 1100 0000 1010 1101 0111 1010 0011 0111 0000 0011 1101 0000

   02275 AC 73 F3 0B D4 A1 72 3F   .s....r?   1010 1100 0111 0011 1111 0011 0000 1011 1101 0100 1010 0001 0111 0010 0011 1111

   0227D 0B B4 FF 80 AD 7A 37 03   .....z7.   0000 1011 1011 0100 1111 1111 1000 0000 1010 1101 0111 1010 0011 0111 0000 0011

   02285 D0 AC 73 F2 C3 05 10 FC   ..s.....   1101 0000 1010 1100 0111 0011 1111 0010 1100 0011 0000 0101 0001 0000 1111 1100

   0228D 10 26 05 20 10 A5 11 D6   .&. ....   0001 0000 0010 0110 0000 0101 0010 0000 0001 0000 1010 0101 0001 0001 1101 0110

   02295 6E AB                   crc
```

20.4.48 MLEADER

This entity was introduced in version 21. A significant portion (content block/text and leaders) of the multileader entity is stored in the MLeaderAnnotContext object (see paragraph 20.4.86), which is embedded into this object (stream).

Version	**Field type**	**DXF group code**	**Description**
	…		Common entity data.
R2010+			
	BS	270	Version (expected to be 2).

Common			
	…		MLeaderAnnotContext fields (see paragraph 20.4.86). This contains the mleader content (block/text) and the leaders.
	H	340	Leader style handle (hard pointer)
	BL	90	Override flags: 1 << 0 = Leader line type, 1 << 1 = Leader line color, 1 << 2 = Leader line type handle, 1 << 3 = Leader line weight, 1 << 4 = Enabled landing, 1 << 5 = Landing gap, 1 << 6 = Enabled dog-leg, 1 << 7 = Dog-leg length, 1 << 8 = Arrow symbol handle, 1 << 9 = Arrow size, 1 << 10 = Conent type, 1 << 11 = Text style handle, 1 << 12 = Text left attachment type (of MTEXT), 1 << 13 = Text angle type (of MTEXT), 1 << 14 = Text alignment type (of MTEXT), 1 << 15 = Text color (of MTEXT), 1 << 16 = Text height (of MTEXT), 1 << 17 = Enable text frame, 1 << 18 = Enable use of default MTEXT (from MLEADERSTYLE), 1 << 19 = Content block handle, 1 << 20 = Block content color, 1 << 21 = Block content scale, 1 << 22 = Block content rotation, 1 << 23 = Block connection type, 1 << 24 = Scale, 1 << 25 = Text right attachment type (of MTEXT), 1 << 26 = Text switch alignment type (of MTEXT), 1 << 27 = Text attachment direction (of MTEXT), 1 << 28 = Text top attachment type (of MTEXT), 1 << 29 = Text bottom attachment type (of MTEXT)
	BS	170	Leader type (0 = inivisible leader, 1 = straight line leader, 2 = spline leader).
	CMC	91	Leader color
	H	341	Leader line type handle (hard pointer)
	BL	171	Line weight
	B	290	Landing enabled
	B	291	Dog-leg enabled
	BD	41	Landing distance
	H	342	Arrow head handle (hard pointer)
	BD	42	Default arrow head size
	BS	172	Style content type:

			0 = None, 1 = Block content, 2 = MTEXT content, 3 = TOLERANCE content
	H	343	Style text style handle (hard pointer)
	BS	173	Style left text attachment type. Values 0-8 are used for the left/right attachment point (attachment direction is horizontal), values 9-10 are used for the top/bottom attachment points (attachment direction is vertical). Attachment point is: • 0 = top of top text line, • 1 = middle of top text line, • 2 = middle of text, • 3 = middle of bottom text line, • 4 = bottom of bottom text line, • 5 = bottom text line, • 6 = bottom of top text line. Underline bottom line • 7 = bottom of top text line. Underline top line, • 8 = bottom of top text line. Underline all content, • 9 = center of text (y-coordinate only), • 10 = center of text (y-coordinate only), and overline top/underline bottom content.
	BS	95	Style right text attachment type. See also style left text attachment type.
	BS	174	Style text angle type: 0 = text angle is equal to last leader line segment angle, 1 = text is horizontal, 2 = text angle is equal to last leader line segment angle, but potentially rotated by 180 degrees so the right side is up for readability.
	BS	175	Unknown
	CMC	92	Style text color
	B	292	Style text frame enabled
	H	[344]	Style block handle (hard pointer) (DXF group is optional)
	CMC	93	Style block color
	3BD	10	Style block scale vector
	BD	43	Style block rotation (radians)
	BS	176	Style attachment type (0 = center extents, 1 = insertion point)
	B	293	Is annotative
-R2007			
	BL	-	Number of arrow heads
			BEGIN REPEAT arrow heads
	B	94	Is default?
	H	345	Arrow head handle (hard pointer)
			END REPEAT arrow heads
	BL	-	Number of block labels
			BEGIN REPEAT block labels

	H	330	Attribute definition (ATTDEF) handle (soft pointer)
	TV	302	Label text
	BS	177	UI index (sequential index of the label in the collection)
	BD	44	Width
			END REPEAT block labels
	B	294	Is text direction negative
	BS	178	IPE align (meaning unknown)
	BS	179	Justification (1 = left, 2 = center, 3 = righ)
	BD	45	Scale factor
R2010+			
	BS	271	Attachment direction (0 = horizontal, 1 = vertical). This defines whether the leaders attach to the left/right of the content block/text, or attach to the top/bottom.
	BS	273	Style top text attachment. See also style left text attachment type.
	BS	272	Style bottom text attachment type. See also style left text attachment type.
R2013+	B	295	Leader extended to text

20.4.49 TOLERANCE (46)

```
        Common Entity Data
R13-R14 Only:
        Unknown short          S
        Height                BD      --
        Dimgap(?)             BD             dimgap at time of creation, * dimscale
Common:
        Ins pt               3BD      10
        X direction          3BD      11
        Extrusion            3BD     210     etc.
        Text string           BS       1
        Common Entity Handle Data
                               H             DIMSTYLE (hard pointer)
```

20.4.49.1 Example:

```
OBJECT: tolerance (2EH), len 65H (101), handle: 01 0C

    022EB 65 00                   e.        0110 0101 0000 0000

    022ED 4B 80 80 43 27 18 10 00   K..C'...   0100 1011 1000 0000 1000 0000 0100 0011 0010 0111 0001 1000 0001 0000 0000 0000

    022F5 05 5B 40 56 BD 1B 81 E8   .[@V....   0000 0101 0101 1011 0100 0000 0101 0110 1011 1101 0001 1011 1000 0001 1110 1000

    022FD 56 39 F8 15 AF 46 E0 7A   V9...F.z   0101 0110 0011 1001 1111 1000 0001 0101 1010 1111 0100 0110 1110 0000 0111 1010

    02305 15 6E 7E 51 19 A0 47 00   .n~Q..G.   0001 0101 0110 1110 0111 1110 0101 0001 0001 1001 1010 0000 0100 0111 0000 0000

    0230D C7 13 A0 18 09 38 21 8A   .....8!.   1100 0111 0001 0011 1010 0000 0001 1000 0000 1001 0011 1000 0010 0001 1000 1010

    02315 5E A1 48 13 54 A5 CF 6B   ^.H.T..k   0101 1110 1010 0001 0100 1000 0001 0011 0101 0100 1010 0101 1100 1111 0110 1011
```

```
0231D 88 CC EC 8E 87 6D 4F A4   .....mO.   1000 1000 1100 1100 1110 1100 1000 1110 1000 0111 0110 1101 0100 1111 1010 0100
02325 A4 AE CF 6B 88 CC EC 8E   ...k....   1010 0100 1010 1110 1100 1111 0110 1011 1000 1000 1100 1100 1110 1100 1000 1110
0232D 87 6D CF AC 2E 6C 8C CF   .m...l..   1000 0111 0110 1101 1100 1111 1010 1100 0010 1110 0110 1100 1000 1100 1100 1111
02335 6B 88 CC EC 8E 87 6D AF   k.....m.   0110 1011 1000 1000 1100 1100 1110 1100 1000 1110 1000 0111 0110 1101 1010 1111
0233D A4 A4 AE C4 A4 AE C4 A4   ........   1010 0100 1010 0100 1010 1110 1100 0100 1010 0100 1010 1110 1100 0100 1010 0100
02345 AE C4 A4 AE C6 0A 21 F8   ......!.   1010 1110 1100 0100 1010 0100 1010 1110 1100 0110 0000 1010 0010 0001 1111 1000
0234D 20 4C 0A 23 B4             L.#.      0010 0000 0100 1100 0000 1010 0010 0011 1011 0100
02352 45 2F                     crc
```

20.4.50 MLINE (47)

```
Common Entity Data
Scale                  BD      40
Just                   EC             top (0), bottom(2), or center(1)
Base point            3BD      10
Extrusion             3BD     210     etc.
Openclosed             BS             open (1), closed(3)
Linesinstyle           RC      73
Numverts               BS      72
do numverts times {
  vertex    3BD
  vertex direction 3BD
  miter direction  3BD
  do lineinstyle times {
    numsegparms    BS
    do numsegparms times {
      segparm         BD    segment parameter
    }
    numareafillparms BS
    do num area fill parms times {
      areafillparm   BD    area fill parameter
}
}
}

Common Entity Handle Data
                        H             mline style oject handle (hard pointer)
```

20.4.50.1 Example:

```
OBJECT: mline (2FH), len E4H (228), handle: 01 0D

    02354 E4 00                    ..         1110 0100 0000 0000

    02356 4B C0 80 43 66 C8 30 00   K..Cf.0.   0100 1011 1100 0000 1000 0000 0100 0011 0110 0110 1100 1000 0011 0000 0000 0000

    0235E 05 5B 20 04 61 AD 1E F2   .[ .a...   0000 0101 0101 1011 0010 0000 0000 0100 0110 0001 1010 1101 0001 1110 1111 0010

    02366 13 A8 8A 01 81 93 3C 67   ......<g   0001 0011 1010 1000 1000 1010 0000 0001 1000 0001 1001 0011 0011 1100 0110 0111

    0236E D4 2B C0 7F 52 80 81 20   .+..R..    1101 0100 0010 1011 1100 0000 0111 1111 0101 0010 1000 0000 1000 0001 0010 0000

    02376 64 61 AD 1E F2 13 A8 8A   da......   0110 0100 0110 0001 1010 1101 0001 1110 1111 0010 0001 0011 1010 1000 1000 1010

    0237E 01 81 93 3C 67 D4 2B C0   ...<g.+.   0000 0001 1000 0001 1001 0011 0011 1100 0110 0111 1101 0100 0010 1011 1100 0000

    02386 7F 13 E9 CF 70 DE 47 3C   ....p.G<   0111 1111 0001 0011 1110 1001 1100 1111 0111 0000 1101 1110 0100 0111 0011 1100

    0238E C7 E4 F8 6A 7B 9C 00 2F   ...j{../   1100 0111 1110 0100 1111 1000 0110 1010 0111 1011 1001 1100 0000 0000 0010 1111

    02396 39 FC 4E 86 A7 B9 C0 02   9.N.....   0011 1001 1111 1100 0100 1110 1000 0110 1010 0111 1011 1001 1100 0000 0000 0010

    0239E F3 DF 93 E9 CF 70 DE 47   .....p.G   1111 0011 1101 1111 1001 0011 1110 1001 1100 1111 0111 0000 1101 1110 0100 0111

    023A6 3C C7 F2 05 52 04 00 00   <...R...   0011 1100 1100 0111 1111 0010 0000 0101 0101 0010 0000 0100 0000 0000 0000 0000

    023AE 00 00 00 00 78 5F D0 38   ....x_.8   0000 0000 0000 0000 0000 0000 0000 0000 0111 1000 0101 1111 1101 0000 0011 1000

    023B6 DD 69 B0 78 DE 38 80 44   .i.x.8.D   1101 1101 0110 1001 1011 0000 0111 1000 1101 1110 0011 1000 1000 0000 0100 0100

    023BE 0A 1C 33 BD E1 05 20 40   ..3... @   0000 1010 0001 1100 0011 0011 1011 1101 1110 0001 0000 0101 0010 0000 0100 0000

    023C6 62 C6 53 15 C9 57 71 F9   b.S..Wq.   0110 0010 1100 0110 0101 0011 0001 0101 1100 1001 0101 0111 0111 0001 1111 1001

    023CE FB 6C F0 9B 58 B3 AB 7F   .l..X...   1111 1011 0110 1100 1111 0000 1001 1011 0101 1000 1011 0011 1010 1011 0111 1111

    023D6 05 47 3C F4 7B 0B 79 B7   .G<.{.y.   0000 0101 0100 0111 0011 1100 1111 0100 0111 1011 0000 1011 0111 1001 1011 0111

    023DE E0 8F 92 1D DC D1 2F 79   ....../y   1110 0000 1000 1111 1001 0010 0001 1101 1101 1100 1101 0001 0010 1111 0111 1001

    023E6 FC 81 54 81 18 D6 BA 82   ..T.....   1111 1100 1000 0001 0101 0100 1000 0001 0001 1000 1101 0110 1011 1010 1000 0010

    023EE C0 20 DE 77 F4 27 50 A1   . .w.'P.   1100 0000 0010 0000 1101 1110 0111 0111 1111 0100 0010 0111 0101 0000 1010 0001

    023F6 65 46 02 92 A0 13 16 15   eF......   0110 0101 0100 0110 0000 0010 1001 0010 1010 0000 0001 0011 0001 0110 0001 0101

    023FE 43 DA 24 00 28 10 1C B1   C.$.(...   0100 0011 1101 1010 0010 0100 0000 0000 0010 1000 0001 0000 0001 1100 1011 0001

    02406 94 C5 72 55 DC 7E 7F 5B   ..rU.~.[   1001 0100 1100 0101 0111 0010 0101 0101 1101 1100 0111 1110 0111 1111 0101 1011

    0240E 3C 26 D6 2C EA DF C7 65   <&.,...e   0011 1100 0010 0110 1101 0110 0010 1100 1110 1010 1101 1111 1100 0111 0110 0101

    02416 B3 C2 6D 62 CE A9 F8 20   ..mb...    1011 0011 1100 0010 0110 1101 0110 0010 1100 1110 1010 1001 1111 1000 0010 0000

    0241E B1 94 C5 72 55 DC 7F 20   ...rU..    1011 0001 1001 0100 1100 0101 0111 0010 0101 0101 1101 1100 0111 1111 0010 0000

    02426 55 20 40 00 00 00 00 00   U @.....   0101 0101 0010 0000 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    0242E 07 85 FD 18 28 87 C0 50   ....(..P   0000 0111 1000 0101 1111 1101 0001 1000 0010 1000 1000 0111 1100 0000 0101 0000

    02436 87 28 8E 4C               .(.L       1000 0111 0010 1000 1000 1110 0100 1100
   0243A 91 88                  crc
```

20.4.51 BLOCK CONTROL (48)

```
       Length                 MS     ---     Object length (not counting itself or CRC).
       Type                   BS     0&2     48 (internal DWG type code).
R2000+:
       Obj size               RL             size of object in bits, not including end handles
Common:
       Handle                  H       5     Owner handle (soft pointer) of root object (0).
       EED                     X      -3     See EED section.
R13-R14 Only:
       Obj size               RL             size of object in bits, not including end handles
Common:
       Numreactors             L             Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag       B             If 1, no XDictionary handle is stored for this
                                             object, otherwise XDictionary handle is stored as in
                                             R2000 and earlier.
Common:
       Numentries             BL      70     Doesn't count *MODEL_SPACE and *PAPER_SPACE.
       Handle refs             H             NULL (soft pointer)
                                             xdicobjhandle (hard owner)
                                             numentries handles of blockheaders in the file (soft
                                             owner), then *MODEL_SPACE and *PAPER_SPACE (hard
                                             owner).
       CRC                     X     ---
```

20.4.51.1 Example:

```
OBJECT: blk ctrl (30H), len 20H (32), handle: 01

   00464 20 00                    .        0010 0000 0000 0000

   00466 4C 00 40 64 80 00 00 09   L.@d....   0100 1100 0000 0000 0100 0000 0110 0100 1000 0000 0000 0000 0000 0000 0000 1001

   0046E 08 40 30 21 93 21 9F 21   .@0!.!.!   0000 1000 0100 0000 0011 0000 0010 0001 1001 0011 0010 0001 1001 1111 0010 0001

   00476 AD 21 BB 21 CA 21 D6 21   .!.!.!.!   1010 1101 0010 0001 1011 1011 0010 0001 1100 1010 0010 0001 1101 0110 0010 0001

   0047E F4 22 01 13 31 19 31 16   ."..1.1.   1111 0100 0010 0010 0000 0001 0001 0011 0011 0001 0001 1001 0011 0001 0001 0110

   00486 C1 3A                  crc
```

20.4.52 BLOCK HEADER (49)

```
       Length                 MS     ---     Object length (not counting itself or CRC).
       Type                   BS     0&2     49 (internal DWG type code).
R2000+:
       Obj size              RL            size of object in bits, not including end handles
Common:
       Handle                 H       5    code 0, length followed by the handle bytes.
       EED                    X      -3    See EED section.
R13-R14 Only:
       Obj size              RL            size of object in bits, not including end handles
Common:
       Numreactors            L            Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag      B            If 1, no XDictionary handle is stored for this
                                           object, otherwise XDictionary handle is stored as in
                                           R2000 and earlier.
Common:
       Entry name            TV       2
       64-flag                B      70    The 64-bit of the 70 group.
       xrefindex+1           BS      70    subtract one from this value when read.  After that,
                                           -1 indicates that this reference did not come from
                                           an xref, otherwise this value indicates the index of
                                           the blockheader for the xref from which this came.
       Xdep                   B      70    block is dependent on an xref. (16 bit)
       Anonymous              B       1    if this is an anonymous block  (1 bit)
       Hasatts                B       1    if block contains attdefs    (2 bit)
       Blkisxref              B       1    if block is xref             (4 bit)
       Xrefoverlaid           B       1    if an overlaid xref          (8 bit)
R2000+:
       Loaded Bit             B            0 indicates loaded for an xref
R2004+:
       Owned Object Count    BL            Number of objects owned by this object.
Common:
       Base pt              3BD      10    Base point of block.
       Xref pname            TV       1    Xref pathname.  That's right: DXF 1 AND 3!
                                      3    1 appears in a tblnext/search elist; 3 appears in an
                                           entget.
R2000+:
       Insert Count          RC            A sequence of zero or more non-zero RC's, followed
                                           by a terminating 0 RC.  The total number of these
                                           indicates how many insert handles will be present.
       Block Description     TV       4    Block description.
       Size of preview data  BL            Indicates number of bytes of data following.
       Binary Preview Data N*RC     310
R2007+:
       Insert units          BS      70
       Explodable             B     280
       Block scaling          RC     281
Common:
       Handle refs             H             Block control handle (soft pointer)
                                             [Reactors (soft pointer)]
                                             xdicobjhandle (hard owner)
                                             NULL (hard pointer)
                                             BLOCK entity. (hard owner)
R13-R2000:
                                             if (!blkisxref && !xrefisoverlaid) {
                                               first entity in the def. (soft pointer)
                                               last entity in the def. (soft pointer)
                                             }
R2004+:
                               H             [ENTITY (hard owner)] Repeats "Owned Object Count"
                                             times.
Common:
                                             ENDBLK entity. (hard owner)
R2000+:
       Insert Handles          H             N insert handles, where N corresponds to the number
                                             of insert count entries above (soft pointer).
       Layout Handle           H             (hard pointer)
Common:
       CRC                     X     ---
```

20.4.52.1 Example:

```
OBJECT: blk hdr (31H), len 19H (25), handle: CA

   00488 19 00                   ..        0001 1001 0000 0000

   0048A 4C 40 72 A6 80 00 00 09   L@r.....   0100 1100 0100 0000 0111 0010 1010 0110 1000 0000 0000 0000 0000 0000 0000 1001

   00492 02 2A 44 C8 AA 41 01 30   .*D..A.0   0000 0010 0010 1010 0100 0100 1100 1000 1010 1010 0100 0001 0000 0001 0011 0000

   0049A 50 31 CB 41 CC 41 D3 31   P1.A.A.1   0101 0000 0011 0001 1100 1011 0100 0001 1100 1100 0100 0001 1101 0011 0011 0001

   004A2 D4                      .         1101 0100

   004A3 E5 AA                   crc
```

20.4.53 LAYER CONTROL (50) (UNDOCUMENTED)

```
       Length                 MS     ---     Object length (not counting itself or CRC).
       Type                   BS     0&2     50 (internal DWG type code).
R2000+:
       Obj size               RL             size of object in bits, not including end handles
Common:
       Handle                  H       5     Owner handle (soft pointer) of root object (0).
```

EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Numentries	BL	70	Counts layer "0", too.
Handle refs	H		NULL (soft pointer)
			xdicobjhandle(hard owner)
			layer objhandles (soft owner)
CRC	X	---	

20.4.53.1 Example:

```
OBJECT: layer ctrl (32H), len FH (15), handle: 02

   024B1 0F 00                  ..        0000 1111 0000 0000

   024B3 4C 80 40 A4 80 00 00 09   L.@.....   0100 1100 1000 0000 0100 0000 1010 0100 1000 0000 0000 0000 0000 0000 0000 1001

   024BB 02 40 30 21 0F 21 99      .@0!.!.    0000 0010 0100 0000 0011 0000 0010 0001 0000 1111 0010 0001 1001 1001

   024C2 C3 1D                  crc
```

20.4.54 LAYER (51)

Length	MS	---	Object length (not counting itself or CRC).
Type	BS	0&2	51 (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	code 0, length followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Entry name	TV	2	

64-flag	B	70	The 64-bit of the 70 group.
xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
Xdep	B	70	dependent on an xref. (16 bit)
R13-R14 Only:			
Frozen	B	70	if frozen (1 bit)
On	B		if on. Normal Autodesk (and Open Design Toolkit) policy is not to report this per se, but rather to negate the color if the layer is off.
Frz in new	B	70	if frozen by default in new viewports (2 bit)
Locked	B	70	if locked (4 bit)
R2000+:			
Values	BS	70,290,370	contains frozen (1 bit), on (2 bit), frozen by default in new viewports (4 bit), locked (8 bit), plotting flag (16 bit), and lineweight (mask with 0x03E0)
Common:			
Color	CMC	62	
Handle refs	H		Layer control (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)
			External reference block handle (hard pointer)
R2000+:			
	H	390	Plotstyle (hard pointer), by default points to PLACEHOLDER with handle 0x0f.
R2007+:			
	H	347	Material
Common:			
	H	6	linetype (hard pointer)
	H		Unknown handle (hard pointer). Always seems to be NULL.
CRC	X	---	

20.4.54.1 Example:

```
OBJECT: layer (33H), len 1BH (27), handle: 99

   02F91 1B 00                   ..        0001 1011 0000 0000

   02F93 4C C0 66 6A 20 00 00 09  L.fj ...  0100 1100 1100 0000 0110 0110 0110 1010 0010 0000 0000 0000 0000 0000 0000 1001

   02F9B 09 44 45 46 50 4F 49 4E  .DEFPOIN  0000 1001 0100 0100 0100 0101 0100 0110 0101 0000 0100 1111 0100 1001 0100 1110

   02FA3 54 53 C0 41 D0 40 8C 14  TS.A.@..  0101 0100 0101 0011 1100 0000 0100 0001 1101 0000 0100 0000 1000 1100 0001 0100

   02FAB 14 45 48                .EH       0001 0100 0100 0101 0100 1000
```

```
    02FAE 34 8F                 crc
```

20.4.55 SHAPEFILE CONTROL (52) (UNDOCUMENTED)

```
       Length               MS     ---     Object length (not counting itself or CRC).
       Type                 BS     0&2     52 (internal DWG type code).
R2000+:
       Obj size             RL             size of object in bits, not including end handles
Common:
       Handle                H       5     Owner handle (soft pointer) of root object (0).
       EED                   X      -3     See EED section.
R13-R14 Only:
       Obj size             RL             size of object in bits, not including end handles
Common:
       Numreactors          BL             Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag     B             If 1, no XDictionary handle is stored for this
                                           object, otherwise XDictionary handle is stored as in
                                           R2000 and earlier.
Common:
       Numentries           BL      70     number of style handles in refs section.
       Handle refs           H             NULL (soft pointer)
                                           xdicobjhandle (hard owner)
                                           shapefile objhandles (soft owner)
       CRC                   X     ---
```

20.4.55.1 Example:

```
OBJECT: shpfile ctrl (34H), len FH (15), handle: 03

    024C4 0F 00                 ..        0000 1111 0000 0000

    024C6 4D 00 40 E4 80 00 00 09   M.@.....  0100 1101 0000 0000 0100 0000 1110 0100 1000 0000 0000 0000 0000 0000 0000 1001

    024CE 02 40 30 21 10 21 F3      .@0!.!.   0000 0010 0100 0000 0011 0000 0010 0001 0001 0000 0010 0001 1111 0011

    024D5 33 8B                 crc
```

20.4.56 SHAPEFILE (53)

This contains a text style for the TEXT or MTEXT entity. Mostly the font information is stored in fields Font name and Big font name, but sometimes (for reasons unknown) some true type font information is contained in the table record’s extended data (see paragraph 28). The true type descriptor is stored as follows in the extended data:

Group code (Value type)	Value
1001 (String)	Font file name
1002 (Bracket)	‘{‘ (optional)
1071 (Int32)	Flags:

	Bold = 0x02000000, Italic = 0x01000000, Pitch (bitmask) = 0x00000003, Font family (bitmask) = 0x000000f0, Character set (bitmask) = 0x0000ff00
1002 (Bracket)	'}' (optional)

	Length	MS	---	Object length (not counting itself or CRC).
	Type	BS	0&2	53 (internal DWG type code).
R2000+:				
	Obj size	RL		size of object in bits, not including end handles
Common:				
	Handle	H	5	code 0, length followed by the handle bytes.
	EED	X	-3	See EED section.
R13-R14 Only:				
	Obj size	RL		size of object in bits, not including end handles
Common:				
	Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:				
	XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:				
	Entry name	TV	2	
	64-flag	B	70	The 64-bit of the 70 group.
	xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
	Xdep	B	70	dependent on an xref. (16 bit)
	Vertical	B	1	if vertical (1 bit of flag)
	shape file	B	1	if a shape file rather than a font (4 bit)
	Fixed height	BD	40	
	Width factor	BD	41	
	Oblique ang	BD	50	
	Generation	RC	71	Generation flags (not bit-pair coded).
	Last height	BD	42	
	Font name	TV	3	
	Bigfont name	TV	4	
	Handle refs	H		Shapefile control (soft pointer)
				[Reactors (soft pointer)]
				xdicobjhandle (hard owner)
				External reference block handle (hard pointer)

```
       CRC                      X     ---
```

20.4.56.1 Example:

```
OBJECT: shpfile (35H), len 25H (37), handle: 10

   02FB0 25 00                   %.        0010 0101 0000 0000

   02FB2 4D 40 44 20 20 10 00 09   M@D  ...  0100 1101 0100 0000 0100 0100 0010 0000 0010 0000 0001 0000 0000 0000 0000 1001

   02FBA 08 53 54 41 4E 44 41 52   .STANDAR  0000 1000 0101 0011 0101 0100 0100 0001 0100 1110 0100 0100 0100 0001 0101 0010

   02FC2 44 C2 60 02 6A 66 66 66   D.`.jfff  0100 0100 1100 0010 0110 0000 0000 0010 0110 1010 0110 0110 0110 0110 0110 0110

   02FCA 66 67 24 FD 03 74 78 74   fg$..txt  0110 0110 0110 0111 0010 0100 1111 1101 0000 0011 0111 0100 0111 1000 0111 0100

   02FD2 90 40 CC 14 28            .@..(     1001 0000 0100 0000 1100 1100 0001 0100 0010 1000

   02FD7 EC 6E                   crc
```

20.4.57 LINETYPE CONTROL (56) (UNDOCUMENTED)

```
       Length                  MS     ---      Object length (not counting itself or CRC).
       Type                    BS     0&2      56 (internal DWG type code).
R2000 Only:
       Obj size                RL              size of object in bits, not including end handles
Common:
       Handle                   H       5      Owner handle (soft pointer) of root object (0).
       EED                      X      -3      See EED section.
R13-R14 Only:
       Obj size                RL              size of object in bits, not including end handles
Common:
       Numreactors             BL              Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag        B              If 1, no XDictionary handle is stored for this
                                               object, otherwise XDictionary handle is stored as in
                                               R2000 and earlier.
Common:
       Numentries              BL      70      Doesn't count BYBLOCK and BYLAYER even though they
                                               both have entries.  Counts the soft owner ones.
       Handle refs              H              NULL (soft pointer)
                                               xdicobjhandle (hard owner)
                                               the linetypes, ending with BYLAYER and BYBLOCK.
                                               all are soft owner references except BYLAYER and
                                               BYBLOCK, which are hard owner references.
       CRC                      X     ---
```

20.4.57.1 Example:

```
OBJECT: ltype ctrl (38H), len 11H (17), handle: 05
```

```
024D7 11 00                   ..        0001 0001 0000 0000

024D9 4E 00 41 64 80 00 00 09   N.Ad....   0100 1110 0000 0000 0100 0001 0110 0100 1000 0000 0000 0000 0000 0000 0000 1001

024E1 01 40 30 21 15 31 13 31   .@0!.1.1   0000 0001 0100 0000 0011 0000 0010 0001 0001 0101 0011 0001 0001 0011 0011 0001

024E9 14                      .         0001 0100

024EA 82 54                   crc
```

20.4.58 LTYPE (57)

```
       Length               MS    ---     Object length (not counting itself or CRC).
       Type                 BS    0&2     57 (internal DWG type code).
R2000 Only:
       Obj size             RL            size of object in bits, not including end handles
Common:
       Handle                H      5     code 0, length followed by the handle bytes.
       EED                   X     -3     See EED section.
R13-R14 Only:
       Obj size             RL            size of object in bits, not including end handles
Common:
       Numreactors          BL            Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag     B            If 1, no XDictionary handle is stored for this
                                          object, otherwise XDictionary handle is stored as in
                                          R2000 and earlier.
Common:
       Entry name           TV      2
       64-flag               B     70     The 64-bit of the 70 group.
       xrefindex+1          BS     70     subtract one from this value when read.  After that,
                                          -1 indicates that this reference did not come from
                                          an xref, otherwise this value indicates the index of
                                          the blockheader for the xref from which this came.
       Xdep                  B     70     dependent on an xref.  (16 bit)
       Description          TV      3
       Pattern Len          BD     40
       Alignment            RC     72     Always 'A'.
       Numdashes            RC     73     The number of repetitions of the 49...74 data.
repeat numdashes times {
       Dash length          BD     49     Dash or dot specifier.
       Complex shapecode    BS     75     Shape number if shapeflag is 2, or index into the
                                          string area if shapeflag is 4.
       X-offset             RD     44     (0.0 for a simple dash.)
       Y-offset             RD     45     (0.0 for a simple dash.)
       Scale                BD     46     (1.0 for a simple dash.)
```

Rotation	BD	50	(0.0 for a simple dash.)
Shapeflag	BS	74	bit coded:
			if (shapeflag & 1), text is rotated 0 degrees, otherwise it follows the segment
			if (shapeflag & 2), complexshapecode holds the index of the shape to be drawn
			if (shapeflag & 4), complexshapecode holds the index into the text area of the string to be drawn.

NOTE: Teigha Classic for .dwg files Toolkit does not present the data this way. It uses a separate variable called stroffset which indicates the offets into the text string area. This is done in order to attempt to make the data easier to understand.

}

R2004 and earlier:

Strings area	X	9	256 bytes of text area. The complex dashes that have text use this area via the 75-group indices. It's basically a pile of 0-terminated strings. First byte is always 0 for R13 and data starts at byte 1. In R14 it is not a valid data start from byte 0.
			(The 9-group is undocumented.)

R2007+:

	X	9	512 bytes of text area, if the 0x02 bit is set on any ShapeFlag (DXF 74) value above. Otherwise no data is present.

Common:

Handle refs	H		Ltype control (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)
			External reference block handle (hard pointer)
		340	shapefile for dash/shape (1 each) (hard pointer)
CRC	X	---	

20.4.59 VIEW CONTROL (60) (UNDOCUMENTED)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0&2	60 (internal DWG type code).

R2000+:

Obj size	RL		size of object in bits, not including end handles

Common:

Handle	H	5	Owner handle (soft pointer) of root object (0).
EED	X	-3	See EED section.

R13-R14 Only:

Obj size	RL		size of object in bits, not including end handles

Common:				
	Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:				
	XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:				
	Numentries	BL	70	
	Handle refs	H		NULL (soft pointer)
				xdicobjhandle (hard owner)
				the views (soft owner)
	CRC	X	---	

20.4.59.1 Example:

```
OBJECT: view ctrl (3CH), len DH (13), handle: 06

   00A04 0D 00                 ..        0000 1101 0000 0000

   00A06 4F 00 41 A4 80 00 00 09   O.A.....   0100 1111 0000 0000 0100 0001 1010 0100 1000 0000 0000 0000 0000 0000 0000 1001

   00A0E 01 40 30 21 3F        .@0!?      0000 0001 0100 0000 0011 0000 0010 0001 0011 1111

   00A13 E1 20                 crc
```

20.4.60 VIEW (61)

	Length	MS	---	Entity length (not counting itself or CRC).
	Type	BS	0	61 (internal DWG type code).
R2000+:				
	Obj size	RL		size of object in bits, not including end handles
Common:				
	Handle	H	5	Length (char) followed by the handle bytes.
	EED	X	-3	See EED section.
R13-R14 Only:				
	Obj size	RL		size of object in bits, not including end handles
Common:				
	Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:				
	XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:				
	Entry name	TV	2	
	64-flag	B	70	The 64-bit of the 70 group.
	xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from

			an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
Xdep	B	70	dependent on an xref. (16 bit)
View height	BD	40	
View width	BD	41	
View center	2RD	10	(Not bit-pair coded.)
Target	3BD	12	
View dir	3BD	11	DXF doc suggests from target toward camera.
Twist angle	BD	50	Radians
Lens length	BD	42	
Front clip	BD	43	
Back clip	BD	44	
View mode	X	71	4 bits: 0123
			0 : 71's bit 0 (1)
			1 : 71's bit 1 (2)
			2 : 71's bit 2 (4)
			3 : OPPOSITE of 71's bit 4 (16)
			Note that only bits 0, 1, 2, and 4 of the 71 can be specified -- not bit 3 (8).
R2000+:			
Render Mode	RC	281	
R2007+:			
Use default lights	B	?	Default value is true
Default lighting Type	RC	?	Default value is 1
Brightness	BD	?	Default value is 0
Contrast	BD	?	Default value is 0
Abient color	CMC	?	Default value is AutoCAD indexed color 250
Common:			
Pspace flag	B	70	Bit 0 (1) of the 70-group.
R2000+:			
Associated UCS	B	72	
Origin	3BD	10	This and next 4 R2000 items are present only if 72 value is 1.
X-direction	3BD	11	
Y-direction	3BD	12	
Elevation	BD	146	
OrthographicViewType	BS	79	
R2007+:			
Camera plottable	B	73	
Common:			

Handle refs	H		view control object (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)
			External reference block handle (hard pointer)
R2007:			
Background handle	H	332	soft pointer
Visual style	H	348	hard pointer
Sun	H	361	hard owner
R2000+:			
Base UCS Handle	H	346	hard pointer
Named UCS Handle	H	345	hard pointer
R2007+:			
Live section	H	334	soft pointer
Common:			
CRC	X	---	

20.4.60.1 Example:

```
OBJECT: view (3DH), len 40H (64), handle: 3F

    01409 40 00                  @.        0100 0000 0000 0000

    0140B 4F 40 4F ED 90 10 00 09   O@O.....   0100 1111 0100 0000 0100 1111 1110 1101 1001 0000 0001 0000 0000 0000 0000 1001

    01413 06 4D 59 56 49 45 57 C2   .MYVIEW.   0000 0110 0100 1101 0101 1001 0101 0110 0100 1001 0100 0101 0101 0111 1100 0010

    0141B F1 38 4A E7 EB B4 A9 00   .8J.....   1111 0001 0011 1000 0100 1010 1110 0111 1110 1011 1011 0100 1010 1001 0000 0000

    01423 9E EA 45 5D 73 27 34 40   ..E]s'4@   1001 1110 1110 1010 0100 0101 0101 1101 0111 0011 0010 0111 0011 0100 0100 0000

    0142B 9D EA 45 5D 73 27 24 40   ..E]s'$@   1001 1101 1110 1010 0100 0101 0101 1101 0111 0011 0010 0111 0010 0100 0100 0000

    01433 BC 4E 12 B9 FA ED 1A 40   .N.....@   1011 1100 0100 1110 0001 0010 1011 1001 1111 1010 1110 1101 0001 1010 0100 0000

    0143B AA 98 00 00 00 00 00 00   ........   1010 1010 1001 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    01443 49 40 A1 20 83 18 28 00   I@. ..(.   0100 1001 0100 0000 1010 0001 0010 0000 1000 0011 0001 1000 0010 1000 0000 0000

    0144B 0C 90                  crc
```

20.4.61 UCS CONTROL (62) (UNDOCUMENTED)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0&2	62 (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Owner handle (soft pointer) of root object (0).
EED	X	-3	See EED section.
R13-R14 Only:			

Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Numentries	BL	70	
Handle refs	H		NULL (soft pointer)
			xdicobjhandle (hard owner)
			the ucs's (soft owner)
CRC	X	---	

20.4.61.1 Example:

```
OBJECT: ucs ctrl (3EH), len DH (13), handle: 07


   0350B 0D 00                    ..        0000 1101 0000 0000

   0350D 4F 80 41 E4 80 00 00 09   O.A.....  0100 1111 1000 0000 0100 0001 1110 0100 1000 0000 0000 0000 0000 0000 0000 1001

   03515 01 40 30 21 4C           .@0!L      0000 0001 0100 0000 0011 0000 0010 0001 0100 1100

   0351A A0 6F                    crc
```

20.4.62 UCS (63)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	63 (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Entry name	TV	2	
64-flag	B	70	The 64-bit of the 70 group.

	xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
	Xdep	B	70	dependent on an xref. (16 bit)
	Origin	3BD	10	
	X-direction	3BD	11	
	Y-direction	3BD	12	
R2000+:				
	Elevation	BD	146	
	OrthographicViewType	BS	79	
	OrthographicType	BS	71	
Common:				
	Handle refs	H		ucs control object (soft pointer)
				[Reactors (soft pointer)]
				xdicobjhandle (hard owner)
				External reference block handle (hard pointer)
R2000+:				
	Base UCS Handle	H	346	hard pointer
	Named UCS Handle	H	-	hard pointer, not present in DXF
Common:				
	CRC	X	---	

20.4.62.1 Example:

```
OBJECT: ucs (3FH), len 45H (69), handle: 4C

   03EB1 45 00                   E.        0100 0101 0000 0000

   03EB3 4F C0 53 20 60 20 00 09   O.S ` ..   0100 1111 1100 0000 0101 0011 0010 0000 0110 0000 0010 0000 0000 0000 0000 1001

   03EBB 05 4D 59 55 43 53 CA 8F   .MYUCS..   0000 0101 0100 1101 0101 1001 0101 0101 0100 0011 0101 0011 1100 1010 1000 1111

   03EC3 DF FF FF FF FF FE 73 F2   ......s.   1101 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1110 0111 0011 1111 0010

   03ECB 14 E5 08 BB 73 23 90 FC   ....s#..   0001 0100 1110 0101 0000 1000 1011 1011 0111 0011 0010 0011 1001 0000 1111 1100

   03ED3 EC FF FF FF FF FF BF BF   ........   1110 1100 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1011 1111 1011 1111

   03EDB 2B D3 16 3A 1E AD B6 EF   +..:....   0010 1011 1101 0011 0001 0110 0011 1010 0001 1110 1010 1101 1011 0110 1110 1111

   03EE3 CF 8F FF FF FF FF FE 33   .......3   1100 1111 1000 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1110 0011 0011

   03EEB F2 18 E5 08 BB 73 23 90   .....s#.   1111 0010 0001 1000 1110 0101 0000 1000 1011 1011 0111 0011 0010 0011 1001 0000

   03EF3 FD 04 1C C1 40            ....@      1111 1101 0000 0100 0001 1100 1100 0001 0100 0000

   03EF8 BE 62                   crc
```

20.4.63 TABLE (VPORT) (64) (UNDOCUMENTED)

Length	MS	---	Entity length (not counting itself or CRC).

Type	BS	0&2	64 (internal DWG type code).
R2000:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Owner handle (soft pointer) of root object (0).
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Numentries	BL	70	Counts all 0010.refs -- even the null ones (0010.0000). The actual 70-group value from an entget doesn't count the null ones.
Handle refs	H		NULL (soft pointer)
			xdicobjhandle (hard owner)
			the vports (soft owner)
CRC	X	---	

20.4.63.1 *Example:*

```
OBJECT: vport ctrl (40H), len 12H (18), handle: 08

   0351C 12 00                   ..        0001 0010 0000 0000

   0351E 50 00 42 24 80 00 00 09   P.B$....  0101 0000 0000 0000 0100 0010 0010 0100 1000 0000 0000 0000 0000 0000 0000 1001

   03526 04 40 30 20 21 4E 21 4F   .@0 !N!O  0000 0100 0100 0000 0011 0000 0010 0000 0010 0001 0100 1110 0010 0001 0100 1111

   0352E 21 50                   !P        0010 0001 0101 0000

   03530 9E 1F                   crc
```

20.4.64 VPORT (65)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	65 (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles

Common:

Numreactors	BL		Number of persistent reactors attached to this obj

R2004+:

XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.

Common:

Entry name	TV	2	
64-flag	B	70	The 64-bit of the 70 group.
xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
Xdep	B	70	dependent on an xref. (16 bit)
View height	BD	40	
Aspect ratio	BD	41	The number stored here is actually the aspect ratio times the view height (40), so this number must be divided by the 40-value to produce the aspect ratio that entget gives. (R13 quirk; R12 has just the aspect ratio.)
View Center	2RD	12	DCS. (If it's plan view, add the view target (17) to get the WCS coordinates. Careful! Sometimes you have to SAVE/OPEN to update the .dwg file.) Note that it's WSC in R12.
View target	3BD	17	
View dir	3BD	16	
View twist	BD	51	
Lens length	BD	42	
Front clip	BD	43	
Back clip	BD	44	
View mode	X	71	4 bits: 0123 0 : 71's bit 0 (1) 1 : 71's bit 1 (2) 2 : 71's bit 2 (4) 3 : OPPOSITE of 71's bit 4 (16) Note that only bits 0, 1, 2, and 4 are given here; see UCSFOLLOW below for bit 3 (8) of the 71.

R2000+:

Render Mode	RC	281	

R2007+:

Use default lights	B	292	
Default lighting type	RC	282	
Brightness	BD	141	
Constrast	BD	142	
Ambient Color	CMC	63	

```
Common:
        Lower left           2RD      10      In fractions of screen width and height.
        Upper right          2RD      11      In fractions of screen width and height.
        UCSFOLLOW              B      71      UCSFOLLOW.  Bit 3 (8) of the 71-group.
        Circle zoom           BS      72      Circle zoom percent.
        Fast zoom              B      73
        UCSICON                X      74      2 bits: 01
                                                0 : 74's bit 0 (1)
                                                1 : 74's bit 1 (2)
        Grid on/off            B      76
        Grd spacing          2RD      15
        Snap on/off            B      75
        Snap style             B      77
        Snap isopair          BS      78
        Snap rot              BD      50
        Snap base            2RD      13
        Snp spacing          2RD      14
R2000+:
        Unknown                B
        UCS per Viewport       B      71
        UCS Origin           3BD     110
        UCS X Axis           3BD     111
        UCS Y Axis           3BD     112
        UCS Elevation         BD     146
        UCS Orthographic type BS      79
R2007+:
        Grid flags            BS      60
        Grid major            BS      61
Common:
        Handle refs            H              Vport control (soft pointer)
                                              [Reactors (soft pointer)]
                                              xdicobjhandle (hard owner)
                                              External reference block handle (hard pointer)
R2007+:
        Background handle      H     332      soft pointer
        Visual Style handle    H     348      hard pointer
        Sun handle             H     361      hard owner
R2000+:
        Named UCS Handle       H     345      hard pointer
        Base UCS Handle        H     346      hard pointer
```

Common:

CRC	X	---	

20.4.64.1 Example:

```
OBJECT: vport (41H), len 93H (147), handle: 4E

    03EFA 93 00                    ..        1001 0011 0000 0000

    03EFC 50 40 53 A7 50 40 00 09   P@S.P@..   0101 0000 0100 0000 0101 0011 1010 0111 0101 0000 0100 0000 0000 0000 0000 1001

    03F04 07 2A 41 43 54 49 56 45   .*ACTIVE   0000 0111 0010 1010 0100 0001 0100 0011 0101 0100 0100 1001 0101 0110 0100 0101

    03F0C C2 1E 94 3B 21 CD A4 CD   ...;!...   1100 0010 0001 1110 1001 0100 0011 1011 0010 0001 1100 1101 1010 0100 1100 1101

    03F14 00 A5 86 68 4A 2C 0E 2D   ...hJ,.-   0000 0000 1010 0101 1000 0110 0110 1000 0100 1010 0010 1100 0000 1110 0010 1101

    03F1C 40 A5 86 68 4A 2C 0E 1D   @..hJ,..   0100 0000 1010 0101 1000 0110 0110 1000 0100 1010 0010 1100 0000 1110 0001 1101

    03F24 40 87 A5 0E C8 73 69 23   @....si#   0100 0000 1000 0111 1010 0101 0000 1110 1100 1000 0111 0011 0110 1001 0010 0011

    03F2C 40 AA 98 00 00 00 00 00   @.......   0100 0000 1010 1010 1001 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    03F34 00 49 40 A1 00 00 00 00   .I@.....   0000 0000 0100 1001 0100 0000 1010 0001 0000 0000 0000 0000 0000 0000 0000 0000

    03F3C 00 00 E0 3F 00 00 00 00   ...?....   0000 0000 0000 0000 1110 0000 0011 1111 0000 0000 0000 0000 0000 0000 0000 0000

    03F44 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    03F4C 00 00 F0 3F 00 00 00 00   ...?....   0000 0000 0000 0000 1111 0000 0011 1111 0000 0000 0000 0000 0000 0000 0000 0000

    03F54 00 00 F0 3F 2C 98 00 00   ...?,...   0000 0000 0000 0000 1111 0000 0011 1111 0010 1100 1001 1000 0000 0000 0000 0000

    03F5C 00 00 00 01 C0 7E 00 00   .....~..   0000 0000 0000 0000 0000 0000 0000 0001 1100 0000 0111 1110 0000 0000 0000 0000

    03F64 00 00 00 01 C0 7E 50 00   .....~P.   0000 0000 0000 0000 0000 0000 0000 0001 1100 0000 0111 1110 0101 0000 0000 0000

    03F6C 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    03F74 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    03F7C 00 00 00 00 07 01 F8 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0111 0000 0001 1111 1000 0000 0000

    03F84 00 00 00 00 07 01 FA 08   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0111 0000 0001 1111 1010 0000 1000

    03F8C 41 82 80                 A..        0100 0001 1000 0010 1000 0000

    03F8F 7D 31                    crc
```

20.4.65 TABLE (APPID) (66) (UNDOCUMENTED)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0&2	66 (internal DWG type code).

R2000+:

Obj size	RL		size of object in bits, not including end handles

Common:

Handle	H	5	Owner handle (soft pointer) of root object (0).
EED	X	-3	See EED section.

R13-R14 Only:

Obj size	RL		size of object in bits, not including end handles

Common:

Numreactors	BL		Number of persistent reactors attached to this obj

R2004+:

XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.

Common:

Numentries	BL	70	
Handle refs	H		NULL (soft pointer)
			xdicobjhandle (hard owner)
			the apps (soft owner)
CRC	X	---	

20.4.65.1 Example:

```
OBJECT: regapp ctrl (42H), len FH (15), handle: 09

   03532 0F 00                   ..        0000 1111 0000 0000

   03534 50 80 42 64 80 00 00 09   P.Bd....   0101 0000 1000 0000 0100 0010 0110 0100 1000 0000 0000 0000 0000 0000 0000 1001

   0353C 02 40 30 21 11 21 86      .@0!.!.    0000 0010 0100 0000 0011 0000 0010 0001 0001 0001 0010 0001 1000 0110

   03543 FA D9                   crc
```

20.4.66 APPID (67)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	67 (internal DWG type code).

R2000+:

Obj size	RL		size of object in bits, not including end handles

Common:

Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.

R13-R14 Only:

Obj size	RL		size of object in bits, not including end handles

Common:

Numreactors	BL		Number of persistent reactors attached to this obj

R2004+:

XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.

Common:

Entry name	TV	2	
64-flag	B	70	The 64-bit of the 70 group.

xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
Xdep	B	70	dependent on an xref. (16 bit)
Unknown	RC	71	Undoc'd 71-group; doesn't even appear in DXF or an entget if it's 0.
Handle refs	H		The app control (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)
			External reference block handle (hard pointer)
CRC	X	---	

20.4.66.1 Example:

```
OBJECT: regapp (43H), len 13H (19), handle: 11

   040BF 13 00                    ..        0001 0011 0000 0000

   040C1 50 C0 44 67 40 00 00 09   P.Dg@...   0101 0000 1100 0000 0100 0100 0110 0111 0100 0000 0000 0000 0000 0000 0000 1001

   040C9 04 41 43 41 44 C0 0C 10   .ACAD...   0000 0100 0100 0001 0100 0011 0100 0001 0100 0100 1100 0000 0000 1100 0001 0000

   040D1 83 05 0A                 ...        1000 0011 0000 0101 0000 1010

   040D4 8C E9                    crc
```

20.4.67 DIMSTYLE CONTROL (68) (UNDOCUMENTED)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0&2	68 (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Owner handle (soft pointer) of root object (0).
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Numentries	BL	70	
Handle refs	H		NULL (soft pointer)
			xdicobjhandle (hard owner)

			the dimstyles (soft owner)
CRC	X	---	

20.4.67.1 Example:

```
OBJECT: dimstyle ctrl (44H), len 10H (16), handle: 0A

   03545 10 00                  ..        0001 0000 0000 0000

   03547 51 00 42 A4 80 00 00 09   Q.B.....   0101 0001 0000 0000 0100 0010 1010 0100 1000 0000 0000 0000 0000 0000 0000 1001

   0354F 03 40 30 21 1D 21 4D 20   .@0!.!M    0000 0011 0100 0000 0011 0000 0010 0001 0001 1101 0010 0001 0100 1101 0010 0000

   03557 BA 14                  crc
```

20.4.68 DIMSTYLE (69)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	69 (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Entry name	TV	2	
64-flag	B	70	The 64-bit of the 70 group.
xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
Xdep	B	70	dependent on an xref. (16 bit)
R13 & R14 Only:			
DIMTOL	B	71	
DIMLIM	B	72	
DIMTIH	B	73	
DIMTOH	B	74	
DIMSE1	B	75	
DIMSE2	B	76	

DIMALT	B	170
DIMTOFL	B	172
DIMSAH	B	173
DIMTIX	B	174
DIMSOXD	B	175
DIMALTD	RC	171
DIMZIN	RC	78
DIMSD1	B	281
DIMSD2	B	282
DIMTOLJ	RC	283
DIMJUST	RC	280
DIMFIT	RC	287
DIMUPT	B	288
DIMTZIN	RC	284
DIMALTZ	RC	285
DIMALTTZ	RC	286
DIMTAD	RC	77
DIMUNIT	BS	270
DIMAUNIT	BS	275
DIMDEC	BS	271
DIMTDEC	BS	272
DIMALTU	BS	273
DIMALTTD	BS	274
DIMSCALE	BD	40
DIMASZ	BD	41
DIMEXO	BD	42
DIMDLI	BD	43
DIMEXE	BD	44
DIMRND	BD	45
DIMDLE	BD	46
DIMTP	BD	47
DIMTM	BD	48
DIMTXT	BD	140
DIMCEN	BD	141
DIMTSZ	BD	142
DIMALTF	BD	143
DIMLFAC	BD	144
DIMTVP	BD	145
DIMTFAC	BD	146
DIMGAP	BD	147

```
        DIMPOST             T      3
        DIMAPOST            T      4
        DIMBLK              T      5
        DIMBLK1             T      6
        DIMBLK2             T      7
        DIMCLRD            BS    176
        DIMCLRE            BS    177
        DIMCLRT            BS    178
R2000+:
        DIMPOST            TV      3
        DIMAPOST           TV      4
        DIMSCALE           BD     40
        DIMASZ             BD     41
        DIMEXO             BD     42
        DIMDLI             BD     43
        DIMEXE             BD     44
        DIMRND             BD     45
        DIMDLE             BD     46
        DIMTP              BD     47
        DIMTM              BD     48
R2007+:
        DIMFXL             BD     49
        DIMJOGANG          BD     50
        DIMTFILL           BS     69
        DIMTFILLCLR       CMC     70
R2000+:
        DIMTOL              B     71
        DIMLIM              B     72
        DIMTIH              B     73
        DIMTOH              B     74
        DIMSE1              B     75
        DIMSE2              B     76
        DIMTAD             BS     77
        DIMZIN             BS     78
        DIMAZIN            BS     79
R2007+:
        DIMARCSYM          BS     90
R2000+:
        DIMTXT             BD    140
        DIMCEN             BD    141
```

```
          DIMTSZ                BD    142
          DIMALTF               BD    143
          DIMLFAC               BD    144
          DIMTVP                BD    145
          DIMTFAC               BD    146
          DIMGAP                BD    147
          DIMALTRND             BD    148
          DIMALT                 B    170
          DIMALTD               BS    171
          DIMTOFL                B    172
          DIMSAH                 B    173
          DIMTIX                 B    174
          DIMSOXD                B    175
          DIMCLRD               BS    176
          DIMCLRE               BS    177
          DIMCLRT               BS    178
          DIMADEC               BS    179
          DIMDEC                BS    271
          DIMTDEC               BS    272
          DIMALTU               BS    273
          DIMALTTD              BS    274
          DIMAUNIT              BS    275
          DIMFRAC               BS    276
          DIMLUNIT              BS    277
          DIMDSEP               BS    278
          DIMTMOVE              BS    279
          DIMJUST               BS    280
          DIMSD1                 B    281
          DIMSD2                 B    282
          DIMTOLJ               BS    283
          DIMTZIN               BS    284
          DIMALTZ               BS    285
          DIMALTTZ              BS    286
          DIMUPT                 B    288
          DIMFIT                BS    287
R2007+:
          DIMFXLON               B    290
R2010+:
          DIMTXTDIRECTION        B    295
          DIMALTMZF             BD      ?
```

DIMALTMZS	T	?	DIMMZF	BD	?
DIMMZS	T	?			
R2000+:					
DIMLWD	BS	371			
DIMLWE	BS	372			
Common:					
Unknown	B	70	Seems to set the 0-bit (1) of the 70-group.		
Handle refs	H		Dimstyle control (soft pointer)		
			[Reactors (soft pointer)]		
			xdicobjhandle (hard owner)		
			External reference block handle (hard pointer)		
		340	shapefile (DIMTXSTY) (hard pointer)		
R2000+:					
		341	leader block (DIMLDRBLK) (hard pointer)		
		342	dimblk (DIMBLK) (hard pointer)		
		343	dimblk1 (DIMBLK1) (hard pointer)		
		344	dimblk2 (DIMBLK2) (hard pointer)		
R2007+:					
		345	dimltype (hard pointer)		
		346	dimltex1 (hard pointer)		
		347	dimltex2 (hard pointer)		
Common:					
CRC	X	---			

20.4.68.1 Example:

```
OBJECT: dimstyle (45H), len 70H (112), handle: 1D


   040F0 70 00                    p.         0111 0000 0000 0000

   040F2 51 40 47 64 90 30 00 09   Q@Gd.0..   0101 0001 0100 0000 0100 0111 0110 0100 1001 0000 0011 0000 0000 0000 0000 1001

   040FA 08 53 54 41 4E 44 41 52   .STANDAR   0000 1000 0101 0011 0101 0100 0100 0001 0100 1110 0100 0100 0100 0001 0101 0010

   04102 44 C3 00 04 00 00 80 01   D.......   0100 0100 1100 0011 0000 0000 0000 0100 0000 0000 0000 0000 1000 0000 0000 0001

   0410A 80 00 00 00 10 29 04 41   .....).A   1000 0000 0000 0000 0000 0000 0000 0000 0001 0000 0010 1001 0000 0100 0100 0001

   04112 10 24 09 02 B5 E8 DC 0F   .$......   0001 0000 0010 0100 0000 1001 0000 0010 1011 0101 1110 1000 1101 1100 0000 1111

   0411A 42 B1 CF C0 00 00 00 00   B.......   0100 0010 1011 0001 1100 1111 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000

   04122 00 0B 03 F1 4A E0 7A 17   ....J.z.   0000 0000 0000 1011 0000 0011 1111 0001 0100 1010 1110 0000 0111 1010 0001 0111

   0412A AD 47 60 FC 0A D7 A3 70   .G`....p   1010 1101 0100 0111 0110 0000 1111 1100 0000 1010 1101 0111 1010 0011 0111 0000

   04132 3D 0A C7 3F AA 02 B5 E8   =..?....   0011 1101 0000 1010 1100 0111 0011 1111 1010 1010 0000 0010 1011 0101 1110 1000
```

```
0413A DC 0F 42 B1 CF C0 AD 7A   ..B....z   1101 1100 0000 1111 0100 0010 1011 0001 1100 1111 1100 0000 1010 1101 0111 1010

04142 37 03 D0 AB 73 F8 66 66   7...s.ff   0011 0111 0000 0011 1101 0000 1010 1011 0111 0011 1111 1000 0110 0110 0110 0110

0414A 66 66 66 66 39 40 64 0A   ffff9@d.   0110 0110 0110 0110 0110 0110 0110 0110 0011 1001 0100 0000 0110 0100 0000 1010

04152 D7 A3 70 3D 0A B7 3F AA   ..p=..?.   1101 0111 1010 0011 0111 0000 0011 1101 0000 1010 1011 0111 0011 1111 1010 1010

0415A AA 20 85 18 28 28 88 00   . ..((..   1010 1010 0010 0000 1000 0101 0001 1000 0010 1000 0010 1000 1000 1000 0000 0000

04162 CC 33                     crc
```

20.4.69 VIEWPORT ENTITY CONTROL (70) (UNDOCUMENTED)

```
       Length                 MS     ---      Entity length (not counting itself or CRC).
       Type                   BS     0&2      70 (internal DWG type code).
R2000+:
       Obj size               RL              size of object in bits, not including end handles
Common:
       Handle                  H       5      Owner handle (soft pointer) of root object (0).
       EED                     X      -3      See EED section.
R13-R14 Only:
       Obj size               RL              size of object in bits, not including end handles
Common:
       Numreactors             B       L      Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag       B              If 1, no XDictionary handle is stored for this
                                              object, otherwise XDictionary handle is stored as in
                                              R2000 and earlier.
Common:
       Numentries             BL      70
       Handle refs             H              NULL (soft pointer)
                                              xdicobjhandle (hard owner)
                                              the viewport entity headers (soft owner)
       CRC                     X     ---
```

20.4.69.1 Example:

```
OBJECT: vpent ctrl (46H), len 17H (23), handle: 0B


03559 17 00                     ..         0001 0111 0000 0000

0355B 51 80 42 E4 80 00 00 09   Q.B.....   0101 0001 1000 0000 0100 0010 1110 0100 1000 0000 0000 0000 0000 0000 0000 1001

03563 06 40 30 21 51 21 52 21   .@0!Q!R!   0000 0110 0100 0000 0011 0000 0010 0001 0101 0001 0010 0001 0101 0010 0010 0001

0356B 54 21 56 21 58 21 5A      T!V!X!Z    0101 0100 0010 0001 0101 0110 0010 0001 0101 1000 0010 0001 0101 1010

03572 9E 84                     crc
```

20.4.70 VIEWPORT ENTITY HEADER (71)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	71 (internal DWG type code).
R2000:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Entry name	TV	2	
64-flag	B	70	The 64-bit of the 70 group.
xrefindex+1	BS	70	subtract one from this value when read. After that, -1 indicates that this reference did not come from an xref, otherwise this value indicates the index of the blockheader for the xref from which this came.
Xdep	B	70	dependent on an xref. (16 bit)
1 flag	B		The 1 bit of the 70 group
Handle refs	H		viewport entity control (soft pointer)
			xdicobjhandle (hard owner)
			External reference block handle (hard pointer)
			the corresponding viewport entity (soft owner) objhandle of previous vport ent header in chain (hard pointer) sometimes points to self; I change those to NULL. NULL indicates end of chain.
CRC	X	---	

20.4.70.1 Example:

```
OBJECT: vpent hdr (47H), len 11H (17), handle: 58


   03574 11 00                   ..        0001 0001 0000 0000

   03576 51 C0 56 24 50 00 00 0A   Q.V$P...   0101 0001 1100 0000 0101 0110 0010 0100 0101 0000 0000 0000 0000 0000 0000 1010

   0357E CA 08 59 82 82 0A CA 8A   ..Y.....   1100 1010 0000 1000 0101 1001 1000 0010 1000 0010 0000 1010 1100 1010 1000 1010

   03586 B4                      .         1011 0100

   03587 2F 9E                   crc
```

20.4.71 AcDbAnnotScaleObjectContextData

This class inherits from class AcDbObjectContextData (see paragraph 20.4.89).

Version	Field type	DXF group code	Description
	…		Common AcDbObjectContextData data (see paragraph 20.4.89).
	H	340	Handle to scale (AcDbScale) object (hard pointer). See paragraph 20.4.92.

20.4.72 GROUP (72): Group of ACAD entities

```
        Length                MS    ---     Entity length (not counting itself or CRC).
        Type                  BS      0     72 (internal DWG type code).
R2000+:
        Obj size              RL            size of object in bits, not including end handles
Common:
        Handle                 H      5     Length (char) followed by the handle bytes.
        EED                    X     -3     See EED section.
R13-R14 Only:
        Obj size              RL            size of object in bits, not including end handles
Common:
        Numreactors           BL            Number of persistent reactors attached to this obj
R2004+:
        XDic Missing Flag      B            If 1, no XDictionary handle is stored for this
                                            object, otherwise XDictionary handle is stored as in
                                            R2000 and earlier.
Common:
        Str                   TV            name of group
        Unnamed               BS      1     if group has no name
        Selectable            BS      1     if group selectable
        Numhandles            BL            # objhandles in this group
        Handle refs            H            parenthandle (soft pointer)
                                            [Reactors (soft pointer)]
                                            xdicobjhandle (hard owner)
                                            the entries in the group (hard pointer)
```

20.4.72.1 Example:

```
OBJECT: group (48H), len 27H (39), handle: 7B

   0431E 27 00                   '.        0010 0111 0000 0000

   04320 52 00 5E ED E0 00 00 04   R.^.....   0101 0010 0000 0000 0101 1110 1110 1101 1110 0000 0000 0000 0000 0000 0000 0100

   04328 05 0F 74 68 69 73 20 69   ..this i   0000 0101 0000 1111 0111 0100 0110 1000 0110 1001 0111 0011 0010 0000 0110 1001

   04330 73 20 6D 79 67 72 6F 75   s mygrou   0111 0011 0010 0000 0110 1101 0111 1001 0110 0111 0111 0010 0110 1111 0111 0101
```

```
04338 70 90 14 0D 04 35 04 34   p....5.4   0111 0000 1001 0000 0001 0100 0000 1101 0000 0100 0011 0101 0000 0100 0011 0100

04340 C1 45 E9 45 B5 45 A1      .E.E.E.    1100 0001 0100 0101 1110 1001 0100 0101 1011 0101 0100 0101 1010 0001

04347 35 69                     crc
```

20.4.73 MLINESTYLE (73):

```
       Length                 MS     ---      Entity length (not counting itself or CRC).
       Type                   BS       0      73 (internal DWG type code).
R2000+:
       Obj size               RL              size of object in bits, not including end handles
Common:
       Handle                  H       5      Length (char) followed by the handle bytes.
       EED                     X      -3      See EED section.
R13-R14 Only:
       Obj size               RL              size of object in bits, not including end handles
Common:
       Numreactors            BL              Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag       B              If 1, no XDictionary handle is stored for this
                                              object, otherwise XDictionary handle is stored as in
                                              R2000 and earlier.
Common:
       Name                   TV              Name of this style
       Desc                   TV              Description of this style
       Flags                  BS              A short which reconstitutes the mlinestyle flags as
                                              defined in DXF.  Here are the bits as they relate to
                                              DXF:
                                              DWG bit     goes with     DXF bit
                                                 1                         2
                                                 2                         1
                                                16                        16
                                                32                        64
                                                64                        32
                                               256                       256
                                               512                      1024
                                              1024                       512
       fillcolor             CMC              Fill color for this style
       startang               BD              Start angle
       endang                 BD              End angle
       linesinstyle           RC              Number of lines in this style

REPEAT 'linesinstyle' times:
       Offset                BD           Offset of this segment
       Color                CMC           Color of this segment
Before R2018:
       Ltindex               BS           Linetype index (yes, index)
R2018+:
       Line type handle       H           Line type handle (hard pointer)
END REPEAT

Common:
       Handle refs            H           parenthandle (soft pointer)
                                          [Reactors (soft pointer)]
                                          xdicobjhandle (hard owner)
```

20.4.73.1 Example:

```
OBJECT: mlstyle (49H), len 55H (85), handle: 74

   0439B 55 00                     U.        0101 0101 0000 0000

   0439D 52 40 5D 27 C0 20 00 04   R@]'. ..  0101 0010 0100 0000 0101 1101 0010 0111 1100 0000 0010 0000 0000 0000 0000 0100

   043A5 05 09 4D 59 4D 4C 53 54   ..MYMLST  0000 0101 0000 1001 0100 1101 0101 1001 0100 1101 0100 1100 0101 0011 0101 0100

   043AD 59 4C 45 44 9B 5E 48 1B   YLED.^H.  0101 1001 0100 1100 0100 0101 0100 0100 1001 1011 0101 1110 0100 1000 0001 1011

   043B5 5D 5B 1D 1A 5B 1A 5B 99   ][..[.[.  0101 1101 0101 1011 0001 1101 0001 1010 0101 1011 0001 1010 0101 1011 1001 1001

   043BD 48 1C DD 1E 5B 19 68 18   H...[.h.  0100 1000 0001 1100 1101 1101 0001 1110 0101 1011 0001 1001 0110 1000 0001 1000

   043C5 2D 44 54 FB 21 F9 3F 06   -DT.!.?.  0010 1101 0100 0100 0101 0100 1111 1011 0010 0001 1111 1001 0011 1111 0000 0110

   043CD 0B 51 15 3E C8 7E 4F C0   .Q.>.~O.  0000 1011 0101 0001 0001 0101 0011 1110 1100 1000 0111 1110 0100 1111 1100 0000

   043D5 C0 00 00 00 00 00 0E 03   ........  1100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1110 0000 0011

   043DD FD 01 90 44 08 00 00 00   ...D....  1111 1101 0000 0001 1001 0000 0100 0100 0000 1000 0000 0000 0000 0000 0000 0000

   043E5 00 00 00 E0 BF 40 90 34   .....@.4  0000 0000 0000 0000 0000 0000 1110 0000 1011 1111 0100 0000 1001 0000 0011 0100

   043ED 10 E4 10 E3 00            .....     0001 0000 1110 0100 0001 0000 1110 0011 0000 0000

   043F2 8F AA                     crc
```

NOTE: OBJECTS LISTED AFTER THIS POINT DO NOT HAVE FIXED TYPES. THEIR TYPES ARE DETERMINED BY FINDING THE CLASS ENTRY WHOSE POSITION IN THE CLASS LIST + 500 EQUALS THE TYPE OF THIS OBJECT

20.4.74 DICTIONARYVAR (varies)

```
       Length                MS    ---    Entity length (not counting itself or CRC).
       Type                  BS      0    typecode (internal DWG type code).
R2000+:
       Obj size              RL           size of object in bits, not including end handles
```

Common:

Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.

R13-R14 Only:

Obj size	RL		size of object in bits, not including end handles

Common:

Numreactors	BL		Number of persistent reactors attached to this obj

R2004+:

XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.

Common:

Intval	RC		an integer value
Str	BS		a string
Handle refs	H		parenthandle (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)

20.4.74.1 Example:

```
OBJECT: proxy (1F9H), len 12H (18), handle: 01 EA

   0CDB4 12 00                   ..        0001 0010 0000 0000

   0CDB6 3E 40 40 80 7A A7 00 00   >@@.z...   0011 1110 0100 0000 0100 0000 1000 0000 0111 1010 1010 0111 0000 0000 0000 0000

   0CDBE 00 04 04 01 01 33 40 41   .....3@A   0000 0000 0000 0100 0000 0100 0000 0001 0000 0001 0011 0011 0100 0000 0100 0001

   0CDC6 A2 30                   .0        1010 0010 0011 0000

   0CDC8 AC DA                   crc
```

20.4.75 HATCH (varies)

Common Entity Data

R2004+:

Is Gradient Fill	BL	450	Non-zero indicates a gradient fill is used.
Reserved	BL	451	
Gradient Angle	BD	460	
Gradient Shift	BD	461	
Single Color Grad.	BL	452	
Gradient Tint	BD	462	
# of Gradient Colors	BL	453	

Repeats # of Gradient Colors time:

Unknown double	BD	463	
Unknown short	BS		
RGB Color	BL	63,421	

```
       Ignored color byte    RC
End Repeat
       Gradient Name         TV    470
Common:
       Z coord               BD     30     X, Y always 0.0
       Extrusion            3BD    210
       Name                  TV      2     name of hatch
       Solidfill              B     70     1 if solidfill, else 0
       Associative            B     71     1 if associative, else 0
       Numpaths              BL     91     Number of paths enclosing the hatch

/* definitions of the hatch boundaries */

Repeat numpaths times:
  Pathflag     BL  92  Path flag
  if (!(pathflag & 2)) {
    Numpathsegs   BL  93  number of segments in this path
    Repeat numpathsegs times:
      pathtypestatus   RC  72  type of path
      if (pathtypestatus==1) {    /* LINE */
        pt0              2RD  10  first endpoint
        pt1              2RD  11  second endpoint
      }
      else if (pathtypestatus==2) {  /* CIRCULAR ARC */
        pt0              2RD  10  center
        radius            BD  40  radius
        startangle        BD  50  start angle
        endangle          BD  51  endangle
        isccw             B   73  1 if counter clockwise, otherwise 0
      }
      else if (pathtypestatus==3) {  /* ELLIPTICAL ARC */
        pt0              2RD  10  center
        endpoint         2RD  11  endpoint of major axis
        minormajoratio    BD  40  ratio of minor to major axis
        startangle        BD  50  start angle
        endangle          BD  51  endangle
        isccw              B  73  1 if counter clockwise, otherwise 0
      }
      else if (pathtypestatus==4) {  /* SPLINE */
        degree            BL  94  degree of the spline
        isrational         B  73  1 if rational (has weights), else 0
        isperiodic         B  74  1 if periodic, else 0
        numknots          BL  95  number of knots
        numctlpts         BL  96  number of control points
        Repeat numknots times:
           knot               BD  40  knot value
        End repeat
        Repeat numctlpts times:
          pt0                2RD  10  control point
          if (isrational)
            weight            BD  40  weight
          endif
        End repeat
R24:
        Numfitpoints          BL  97  number of fit points
        Begin repeat numfitpoints times:
          Fitpoint           2RD  11
        End repeat
        Start tangent        2RD  12
        End tangent          2RD  13
Common:
      }
    End repeat (numpathsegs)
  } /* (!(pathflag & 2)) */
  else {  /* POLYLINE PATH */
    bulgespresent         B  72  bulges are present if 1
    closed                B  73  1 if closed
    numpathsegs          BL  91  number of path segments
    Repeat numpathsegs times:
      pt0                2RD  10  point on polyline
      if (bulgespresent) {
        bulge                BD  42  bulge
      }
    End repeat
  }  /* pathflag & 2 */
  numboundaryobjhandles     BL  97  Number of boundary object handles for this path
End repeat (numpaths)

/* below this point is the definition of the hatch itself */
```

```
       style                BS      75      style of hatch  0==odd parity, 1==outermost,
                                            2==whole area

       patterntype          BS      76      pattern type  0==user-defined, 1==predefined,
                                            2==custom
if (!solidfill) {
  angle          BD  52  hatch angle
  scaleorspacing BD  41  scale or spacing (pattern fill only)
  doublehatch     B  77  1 for double hatch
  numdeflines    BS  78  number of definition lines
  Repeat numdeflines times:
    angle           BD  53  line angle
    pt0            2BD  43/44  pattern through this point (X,Y)
    offset         2BD  45/56  pattern line offset
    numdashes       BS  79  number of dash length items
    Repeat numdashes times:
      dashlength      BD  49  dash length
    End repeat
  End repeat
}
if (ANY of the pathflags & 4) {
  pixelsize        BD  47  pixel size
}
numseedpoints      BL  98  number of seed points
Repeat numseedpoints times:
  pt0               2RD  10  seed point
End repeat

       Common Entity Handle Data
Repeat totalbounditems (sum of all "numboundaryitems") times
  boundaryhandle    H  330  boundary handle (soft pointer)
End repeat

       CRC                    X     ---
```

20.4.75.1 Example:

```
OBJECT: proxy (1F5H), len E2H (226), handle: 68

   069C4 E2 00                   ..        1110 0010 0000 0000

   069C6 3D 40 40 5A 26 70 30 00   =@@Z&p0.   0011 1101 0100 0000 0100 0000 0101 1010 0010 0110 0111 0000 0011 0000 0000 0000

   069CE 02 80 DB 54 A0 C8 29 CA   ...T..).   0000 0010 1000 0000 1101 1011 0101 0100 1010 0000 1100 1000 0010 1001 1100 1010

   069D6 69 26 66 22 02 80 A0 80   i&f"....   0110 1001 0010 0110 0110 0110 0010 0010 0000 0010 1000 0000 1010 0000 1000 0000
```

```
069DE 28 03 02 5A E2 89 80 68   (..Z...h   0010 1000 0000 0011 0000 0010 0101 1010 1110 0010 1000 1001 1000 0000 0110 1000

069E6 0D 0F 09 03 C3 C3 C0 88   ........   0000 1101 0000 1111 0000 1001 0000 0011 1100 0011 1100 0011 1100 0000 1000 1000

069EE 1C 12 5E 96 B9 91 BC A7   ..^.....   0001 1100 0001 0010 0101 1110 1001 0110 1011 1001 1001 0001 1011 1100 1010 0111

069F6 F6 1A C1 03 D6 06 A0 88   ........   1111 0110 0001 1010 1100 0001 0000 0011 1101 0110 0000 0110 1010 0000 1000 1000

069FE 00 3C 12 5E 96 B9 91 BC   .<.^....   0000 0000 0011 1100 0001 0010 0101 1110 1001 0110 1011 1001 1001 0001 1011 1100

06A06 A7 F6 1A C1 03 D6 06 A0   ........   1010 0111 1111 0110 0001 1010 1100 0001 0000 0011 1101 0110 0000 0110 1010 0000

06A0E 88 1A 0F A5 B5 4C A8 1F   .....L..   1000 1000 0001 1010 0000 1111 1010 0101 1011 0101 0100 1100 1010 1000 0001 1111

06A16 47 EC 11 0B 35 26 AC 5D   G...5&.]   0100 0111 1110 1100 0001 0001 0000 1011 0011 0101 0010 0110 1010 1100 0101 1101

06A1E C7 E0 3A 0F A5 B5 4C A8   ..:...L.   1100 0111 1110 0000 0011 1010 0000 1111 1010 0101 1011 0101 0100 1100 1010 1000

06A26 1F 47 EC 11 0B 35 26 AC   .G...5&.   0001 1111 0100 0111 1110 1100 0001 0001 0000 1011 0011 0101 0010 0110 1010 1100

06A2E 5D C7 EA 06 B5 C1 81 D1   ].......   0101 1101 1100 0111 1110 1010 0000 0110 1011 0101 1100 0001 1000 0001 1101 0001

06A36 80 28 10 04 08 44 05 F1   .(...D..   1000 0000 0010 1000 0001 0000 0000 0100 0000 1000 0100 0100 0000 0101 1111 0001

06A3E 1E 87 E0 2A 06 B5 C1 81   ...*....   0001 1110 1000 0111 1110 0000 0010 1010 0000 0110 1011 0101 1100 0001 1000 0001

06A46 D1 80 28 10 04 08 44 05   ..(...D.   1101 0001 1000 0000 0010 1000 0001 0000 0000 0100 0000 1000 0100 0100 0000 0101

06A4E F1 1E 87 E8 03 02 5A E2   ......Z.   1111 0001 0001 1110 1000 0111 1110 1000 0000 0011 0000 0010 0101 1010 1110 0010

06A56 89 80 68 0D 0F 09 03 C3   ..h.....   1000 1001 1000 0000 0110 1000 0000 1101 0000 1111 0000 1001 0000 0011 1100 0011

06A5E C3 C0 88 14 80 83 05 A8   ........   1100 0011 1100 0000 1000 1000 0001 0100 1000 0000 1000 0011 0000 0101 1010 1000

06A66 8A 9F 64 3F 27 E4 D4 CC   ..d?'...   1000 1010 1001 1111 0110 0100 0011 1111 0010 0111 1110 0100 1101 0100 1100 1100

06A6E CC CC CC CD C9 F9 01 34   .......4   1100 1100 1100 1100 1100 1100 1100 1101 1100 1001 1111 1001 0000 0001 0011 0100

06A76 88 4C DF DF 36 40 90 28   .L..6@.(   1000 1000 0100 1100 1101 1111 1101 1111 0011 0110 0100 0000 1001 0000 0010 1000

06A7E 0B 63 FF 51 18 1A 82 BF   .c.Q....   0000 1011 0110 0011 1111 1111 0101 0001 0001 1000 0001 1010 1000 0010 1011 1111

06A86 02 98 FF D4 46 06 A0 AF   ....F...   0000 0010 1001 1000 1111 1111 1101 0100 0100 0110 0000 0110 1010 0000 1010 1111

06A8E E4 04 00 00 00 00 00 00   ........   1110 0100 0000 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

06A96 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

06A9E 00 01 05 EC C1 44 3E 02   .....D>.   0000 0000 0000 0001 0000 0101 1110 1100 1100 0001 0100 0100 0011 1110 0000 0010

06AA6 84 17                     ..         1000 0100 0001 0111

06AA8 C2 EA                     crc
```

The proxy flags for the class are 0x480.

20.4.76 FIELD

Class properties:

App name	ObjectDBX Classes
Class number	Dynamic (>= 500)
DWG version	R18
Maintenance version	0
Class proxy flags	0x480
C++ class name	AcDbField
DXF name	FIELD

Fields are referenced from the field list of a drawing (paragraph 20.4.77).

Version	**Field type**	**DXF group code**	**Description**
	…		Common object data (paragraph 20.1).
	TV	1	Evaluator ID
	TV	2,3	Field code (in DXF strings longer than 255 characters are written in chunks of 255 characters in one 2 group and one or more 3 groups).
	BL	90	Number of child fields
			Begin repeat child fields
	H	360	Child field handle (hard owner)
			End repeat child fields
	BL	97	Number of field objects
			Begin repeat field objects
	H	331	Field object handle (soft pointer)
			End repeat field objects
-R2004	TV	4	Format string. After R2004 the format became part of the value object.
Common	BL	91	Evaluation option flags: Never = 0, On open = 1, On save = 2, On plot = 4, When packed for eTransmit = 8, On regeneration = 16, On demand = 32
	BL	92	Filing option flags: None = 0, Don't file field result = 1

	BL	94	Field state flags:
			Unknown = 0,
			Initialized = 1,
			Compiled = 2,
			Modified = 4,

			Evaluated = 8,
			Cached = 16

	BL	95	Evaluation status flags: Not evaluated = 1, Success = 2, Evaluator not found = 4, Syntax error = 8, Invalid code = 16, Invalid context = 32, Other error = 64
	BL	96	Evaluation error code
	TV	300	Evaluation error message
	…	…	The field value, see paragraph 20.4.99.
	TV	301, 9	Value string (DXF: written in 255 character chunks)
	TV	98	Value string length
	BL	93	Number of child fields
			Begin repeat child fields
	TV	6	Child field key
	…	…	The field value, see paragraph 20.4.99.
			End repeat child fields

20.4.77 FIELDLIST

Class properties:

App name	ObjectDBX Classes
Class number	Dynamic (>= 500)
DWG version	R18
Maintenance version	0
Class proxy flags	0x480
C++ class name	AcDbFieldList, inherits AcDbIdSet
DXF name	FIELDLIST

Fields (paragraph 20.4.76) are referenced from the field list of a drawing. The field list is stored in the root dictionary entry ACAD_FIELDLIST.

Version	**Field type**	**DXF group code**	**Description**
	…		Common object data (paragraph 20.1).
	BL		Number of fields
	B		Unknown
			Begin repeat fields

	H	330	Field handle (soft pointer)
			End repeat fields

20.4.78 GEODATA

Class properties:

App name	ObjectDBX Classes
Class number	Dynamic (>= 500)
DWG version	R21
Maintenance version	45
Class proxy flags	0xFFF
C++ class name	AcDbGeoData
DXF name	GEODATA

The geo data object was introduced in AutoCAD 2009. The format changed considerably in AutoCAD 2010. The objectVersion field discerns between the formats (1 = AutoCAD 2009, 2 = AutoCAD 2010, 3 = AutoCAD 2013, but the format is the same as 2010).

Version	**Field type**	**DXF group code**	**Description**
	…		Common object data (paragraph 20.1).
	BL		Object version formats (1 = AutoCAD 2009, 2 = AutoCAD 2010, 3 = AutoCAD 2013, but the format is the same as 2010)
	H		Soft pointer to host block (model space layout owner block)
	BS		Design coordinate type (0 = unknown, local grid = 1, projected grid = 2, geographic (defined by latitude/longitude) = 3)
			If version is AutoCAD 2010 or later
	3BD		Design point
	3BD		Reference point
	BD		Unit scale factor horizontal

	BL		Units value horizontal
			Undefined = 0,
			Inches = 1,
			Feet = 2,
			Miles = 3,
			Millimeters = 4,
			Centimeters = 5,
			Meters = 6,
			Kilometers = 7,
			Micro inches = 8,
			Mils = 9,
			Yards = 10,
			Angstroms = 11,
			Nanometers = 12,

			Microns = 13,
			Decimeters = 14,
			Dekameters = 15,
			Hectometers = 16,
			Gigameters = 17,
			Astronomical = 18,
			Light years = 19,
			Parsecs = 20

	BD		Unit scale factor vertical
	BL		Units value vertical (same enumeration as for the units value horizontal)
	3BD		Up direction
	3RD		North direction
	BL		Scale estimation method: None = 1, User specified scale factor = 2, Grid scale at reference point = 3, Prismodial = 4
	BD		User specified scale factor
	B		Do sea level correction
	BD		Sea level elevation
	BD		Coordinate projection radius
	VT		Coordinate system definition . In AutoCAD 2010 this is a map guide XML string.
	VT		Geo RSS tag.
			Else (version is earlier than AutoCAD 2010)
	3BD		Reference point
	BL		Units value horizontal (see above for enum definition).
	3BD		Design point
	3BD		Obsolete, ODA writes (0, 0, 0)
	3BD		Up direction
	BD		Angle of north direction (radians, angle measured clockwise from the (0, 1) vector).
	3BD		Obsolete, ODA writes (1, 1, 1)
	VT		Coordinate system definition. In AutoCAD 2009 this is a “Well known text” (WKT) string containing a projected coordinate system (PROJCS).
	VT		Geo RSS tag
	BD		Unit scale factor horizontal
	VT		Obsolete, coordinate system datum name
	VT		Obsolete: coordinate system WKT
			End if version is AutoCAD 2010 or later
	VT		Observation from tag
	VT		Observation to tag
	VT		Observation coverage tag
	BL		Number of geo mesh points

			Repeat for each geo mesh point
	2RD		Source point
	2RD		Destination point
			End repeat geo mesh points
	BL		Number of geo mesh faces
			Repeat for each geo mesh face
	BL		Face index 1
	BL		Face index 2
	BL		Face index 3
			End repeat geo mesh faces
			If DWG version is R21 or lower
			Below is CIVIL data. AutoCAD 2010 always writes civil data.
	B		Has civil data? (true)
	B		False
	RD		Reference point Y
	RD		Reference point X
	RD		Reference point Y
	RD		Reference point X
	BL		0
	BL		0
	2RD		(0, 0)
	2RD		(0, 0)
	B		false
	BD		North direction angle (degrees)
	BD		North direction angle (radians)
	BL		Scale estimation method
	BD		User specified scale factor
	B		Do sea level correction
	BD		Sea level elevation
	BD		Coordinate projection radius
			End if

20.4.79 IDBUFFER (varies)

```
(holds list of references to an xref)
       Length               MS    ---     Entity length (not counting itself or CRC).
       Type                 BS      0     typecode (internal DWG type code).
R2000+:
       Obj size             RL            size of object in bits, not including end handles
Common:
       Handle                H      5     Length (char) followed by the handle bytes.
       EED                   X     -3     See EED section.
```

R13-R14 Only:				
	Obj size	RL		size of object in bits, not including end handles
Common:				
	Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:				
	XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:				
	Unknown	RC		always 0?
	Numobjids	BL		number of object ids
	Handle refs	H		parenthandle (soft pointer)
				[Reactors (soft pointer)]
				xdicobjhandle (hard owner)
			330	objids (soft pointer)

20.4.79.1 *Example:*

```
OBJECT: proxy (1FAH), len 12H (18), handle: 8B

   04437 12 00                 ..        0001 0010 0000 0000

   04439 3E 80 40 62 E6 00 00 00   >.@b....   0011 1110 1000 0000 0100 0000 0110 0010 1110 0110 0000 0000 0000 0000 0000 0000

   04441 04 04 01 01 80 41 8A 30   .....A.0   0000 0100 0000 0100 0000 0001 0000 0001 1000 0000 0100 0001 1000 1010 0011 0000

   04449 41 89                 A.        0100 0001 1000 1001

   0444B C9 64                 crc
```

20.4.80 IMAGE (varies)

	Common Entity Data			
	Classversion	BL	90	class version
	pt0	3BD	10	insertion point
	uvec	3BD	11	u direction vector
	vvec	3BD	12	v direction vector
	size	2RD	13	size of image
	displayprops	BS	70	display properties (bit coded), 1==show image, 2==show image when not aligned with screen, 4==use clipping boundary, 8==transparency on
	clipping	B	280	1 if on
	brightness	RC	281	brightness value (0—100, default 50)
	contrast	RC	282	contrast value (0—100, default 50)
	fade	RC	283	fade value (0—100, default 0)
R2010+:				
	Clip mode	B	290	0 = outside, 1 = inside
Common:				

```
       clipbndtype            BS      71      type of clipping boundary, 1==rect, 2==polygon
if (clipbndtype==1) {
       pt0                   2RD      14      first corner of clip boundary
       pt1                   2RD      14      second corner of clip boundary
}
else {
       numclipverts           BL      91      number of vertices in clipping polygon
  Repeat numclipverts times:
       pt0                   2RD      14      a point on the polygon
  End repeat
}

       Common Entity Handle Data
                              H               imagedef (hard pointer)
                              H               imagedefreactor (hard owner)
       CRC                    X     ---
```

20.4.80.1 Example:

```
OBJECT: proxy (1F9H), len 109H (265), handle: 6D

    02D3E 09 01                    ..        0000 1001 0000 0001

    02D40 3E 40 40 5B 6C 60 00 00   >@@[l`..   0011 1110 0100 0000 0100 0000 0101 1011 0110 1100 0110 0000 0000 0000 0000 0000

    02D48 04 60 00 00 00 08 00 00   .`......   0000 0100 0110 0000 0000 0000 0000 0000 0000 0000 0000 1000 0000 0000 0000 0000

    02D50 04 20 00 00 00 30 00 00   . ...0..   0000 0100 0010 0000 0000 0000 0000 0000 0000 0000 0011 0000 0000 0000 0000 0000

    02D58 00 28 00 00 06 B5 6E 62   .(....nb   0000 0000 0010 1000 0000 0000 0000 0000 0000 0110 1011 0101 0110 1110 0110 0010

    02D60 0B AA E1 02 00 03 72 C5   ......r.   0000 1011 1010 1010 1110 0001 0000 0010 0000 0000 0000 0011 0111 0010 1100 0101

    02D68 63 F8 D8 AA 00 00 00 00   c.......   0110 0011 1111 1000 1101 1000 1010 1010 0000 0000 0000 0000 0000 0000 0000 0000

    02D70 00 00 00 00 06 B5 6E 62   ......nb   0000 0000 0000 0000 0000 0000 0000 0000 0000 0110 1011 0101 0110 1110 0110 0010


    02D78 0B AA E1 02 00 03 72 C5   ......r.   0000 1011 1010 1010 1110 0001 0000 0010 0000 0000 0000 0011 0111 0010 1100 0101

    02D80 63 F8 D8 CA 00 00 00 00   c.......   0110 0011 1111 1000 1101 1000 1100 1010 0000 0000 0000 0000 0000 0000 0000 0000

    02D88 00 00 00 00 06 B5 6E 62   ......nb   0000 0000 0000 0000 0000 0000 0000 0000 0000 0110 1011 0101 0110 1110 0110 0010

    02D90 0B AA E1 12 00 03 72 C5   ......r.   0000 1011 1010 1010 1110 0001 0001 0010 0000 0000 0000 0011 0111 0010 1100 0101

    02D98 63 F8 D8 CA 00 00 00 00   c.......   0110 0011 1111 1000 1101 1000 1100 1010 0000 0000 0000 0000 0000 0000 0000 0000

    02DA0 00 00 00 00 06 B5 6E 62   ......nb   0000 0000 0000 0000 0000 0000 0000 0000 0000 0110 1011 0101 0110 1110 0110 0010

    02DA8 0B AA E1 12 00 03 72 C5   ......r.   0000 1011 1010 1010 1110 0001 0001 0010 0000 0000 0000 0011 0111 0010 1100 0101

    02DB0 63 F8 D8 AA 00 00 00 00   c.......   0110 0011 1111 1000 1101 1000 1010 1010 0000 0000 0000 0000 0000 0000 0000 0000
```

```
02DB8 00 00 00 00 06 B5 6E 62   ......nb   0000 0000 0000 0000 0000 0000 0000 0000 0000 0110 1011 0101 0110 1110 0110 0010

02DC0 0B AA E1 02 00 03 72 C5   ......r.   0000 1011 1010 1010 1110 0001 0000 0010 0000 0000 0000 0011 0111 0010 1100 0101

02DC8 63 F8 D8 AA 00 00 00 00   c.......   0110 0011 1111 1000 1101 1000 1010 1010 0000 0000 0000 0000 0000 0000 0000 0000

02DD0 00 00 00 00 06 D0 38 00   ......8.   0000 0000 0000 0000 0000 0000 0000 0000 0000 0110 1101 0000 0011 1000 0000 0000

02DD8 02 80 DB 46 B5 6E 62 0B   ...F.nb.   0000 0010 1000 0000 1101 1011 0100 0110 1011 0101 0110 1110 0110 0010 0000 1011

02DE0 AA E1 02 00 00 DC B1 58   .......X   1010 1010 1110 0001 0000 0010 0000 0000 0000 0000 1101 1100 1011 0001 0101 1000

02DE8 FE 36 2A 81 00 00 00 00   .6*.....   1111 1110 0011 0110 0010 1010 1000 0001 0000 0000 0000 0000 0000 0000 0000 0000


02DF0 00 00 10 07 F4 00 00 00   ........   0000 0000 0000 0000 0001 0000 0000 0111 1111 0100 0000 0000 0000 0000 0000 0000

02DF8 00 00 53 10 9E 00 00 00   ..S.....   0000 0000 0000 0000 0101 0011 0001 0000 1001 1110 0000 0000 0000 0000 0000 0000

02E00 00 00 00 10 07 F0 00 00   ........   0000 0000 0000 0000 0000 0000 0001 0000 0000 0111 1111 0000 0000 0000 0000 0000

02E08 00 00 00 03 02 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0011 0000 0010 0000 0000 0000 0000 0000 0000

02E10 00 00 00 03 02 02 0E 32   .......2   0000 0000 0000 0000 0000 0000 0000 0011 0000 0010 0000 0010 0000 1110 0011 0010

02E18 32 00 40 40 00 00 00 00   2.@@....   0011 0010 0000 0000 0100 0000 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000

02E20 00 38 2F C0 00 00 00 00   .8/.....   0000 0000 0011 1000 0010 1111 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000

02E28 00 38 2F C0 00 00 00 00   .8/.....   0000 0000 0011 1000 0010 1111 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000

02E30 38 17 D0 00 00 00 00 00   8.......   0011 1000 0001 0111 1101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

02E38 38 17 D0 10 5E CC 14 43   8...^..C   0011 1000 0001 0111 1101 0000 0001 0000 0101 1110 1100 1100 0001 0100 0100 0011

02E40 F0 41 68 43 54 5A CC 5B   .AhCTZ.[   1111 0000 0100 0001 0110 1000 0100 0011 0101 0100 0101 1010 1100 1100 0101 1011

02E48 3F                        ?          0011 1111

02E49 0D 2A                     crc
```

20.4.81 IMAGEDEF (varies)

(used in conjunction with IMAGE entities)

```
        Length              MS     ---     Entity length (not counting itself or CRC).
        Type                BS       0     typecode (internal DWG type code).
R2000+:
        Obj size            RL             size of object in bits, not including end handles
Common:
        Handle               H       5     Length (char) followed by the handle bytes.
        EED                  X      -3     See EED section.
R13-R14 Only:
        Obj size            RL             size of object in bits, not including end handles
Common:
        Numreactors         BL             Number of persistent reactors attached to this obj
```

R2004+:

XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.

Common:

Clsver	BL	0	class version
Imgsize	2RD	10	size of image in pixels
Filepath	TV	1	path to file
Isloaded	B	280	0==no, 1==yes
Resunits	RC	281	0==none, 2==centimeters, 5==inches
Pixelsize	2RD	11	size of one pixel in AutoCAD units
Handle refs	H		parenthandle (hard owner)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)

20.4.81.1 Example:

```
OBJECT: proxy (1F7H), len 4EH (78), handle: 6B

   04349 4E 00                    N.        0100 1110 0000 0000

   0434B 3D C0 40 5A E3 B0 20 00  =.@Z.. .  0011 1101 1100 0000 0100 0000 0101 1010 1110 0011 1011 0000 0010 0000 0000 0000

   04353 04 0A 00 00 00 00 00 00  ........  0000 0100 0000 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   0435B 60 40 00 00 00 00 00 00  `@......  0110 0000 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

   04363 60 40 46 D0 CE 97 15 D2  `@F.....  0110 0000 0100 0000 0100 0110 1101 0000 1100 1110 1001 0111 0001 0101 1101 0010

   0436B 53 93 95 17 11 99 58 5D  S.....X]  0101 0011 1001 0011 1001 0101 0001 0111 0001 0001 1001 1001 0101 1000 0101 1101

   04373 1A 19 5C 95 19 5E 1D 1D  ..\..^..  0001 1010 0001 1001 0101 1100 1001 0101 0001 1001 0101 1110 0001 1101 0001 1101

   0437B 5C 99 4B 98 9B 5C 20 5E  \.K..\ ^  0101 1100 1001 1001 0100 1011 1001 1000 1001 1011 0101 1100 0010 0000 0101 1110

   04383 25 D4 EB 07 52 BA C7 FE  %...R...  0010 0101 1101 0100 1110 1011 0000 0111 0101 0010 1011 1010 1100 0111 1111 1110

   0438B 25 D4 EB 07 52 BA C7 F0  %...R...  0010 0101 1101 0100 1110 1011 0000 0111 0101 0010 1011 1010 1100 0111 1111 0000

   04393 08 2D 48 2D 86 11        .-H-..    0000 1000 0010 1101 0100 1000 0010 1101 1000 0110 0001 0001

   04399 E8 23                    crc
```

20.4.82 IMAGEDEFREACTOR (varies)

(used in conjunction with IMAGE entities)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	typecode (internal DWG type code).

R2000+:

Obj size	RL		size of object in bits, not including end handles

Common:

Handle	H	5	Length (char) followed by the handle bytes.

EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Classver	BL	90	class version
Handle refs	H		parenthandle (soft pointer)
			[Reactors (soft pointer])
			xdicobjhandle (hard owner)

20.4.82.1 *Example:*

```
OBJECT: proxy (1F8H), len CH (12), handle: 6C


   02E4B 0C 00                   ..        0000 1100 0000 0000


   02E4D 3E 00 40 5B 25 00 00 00   >.@[%...   0011 1110 0000 0000 0100 0000 0101 1011 0010 0101 0000 0000 0000 0000 0000 0000

   02E55 09 02 60 30             ..`0      0000 1001 0000 0010 0110 0000 0011 0000

   02E59 A1 13                   crc
```

20.4.83 LAYER_INDEX

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	typecode (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.

```
Common:
       timestamp1            BL      40
       timestamp2            BL      40
       numentries            BL              the number of entries
Repeat numentries times:
       Indexlong             BL              a long
       Indexstr              TV       8      a layer name
End repeat

       Handle refs            H              parenthandle (soft pointer)
                                             [Reactors (soft pointer])
                                             xdicobjhandle (hard owner)
                                             entry handles, 1 per entry
```

20.4.83.1 *Example:*

```
OBJECT: proxy (1FFH), len 59H (89), handle: 01 F8



   0D1CD 59 00                    Y.        0101 1001 0000 0000

   0D1CF 3F C0 40 80 7E 20 C0 20   ?.@.~ .   0011 1111 1100 0000 0100 0000 1000 0000 0111 1110 0010 0000 1100 0000 0010 0000

   0D1D7 00 04 04 61 65 25 00 3B   ...ae%.;  0000 0000 0000 0100 0000 0100 0110 0001 0110 0101 0010 0101 0000 0000 0011 1011

   0D1DF 3A 89 80 90 64 DD 01 30   :...d..0  0011 1010 1000 1001 1000 0000 1001 0000 0110 0100 1101 1101 0000 0001 0011 0000

   0D1E7 42 50 64 15 34 84 14 44   BPd.4..D  0100 0010 0101 0000 0110 0100 0001 0101 0011 0100 1000 0100 0001 0100 0100 0100

   0D1EF 54 05 08 55 52 4C 4C 41   T..URLLA  0101 0100 0000 0101 0000 1000 0101 0101 0101 0010 0100 1100 0100 1100 0100 0001

   0D1F7 59 45 52 90 94 44 54 65   YER..DTe  0101 1001 0100 0101 0101 0010 1001 0000 1001 0100 0100 0100 0101 0100 0110 0101

   0D1FF 04 F4 94 E5 45 34 1D 03   ....E4..  0000 0100 1111 0100 1001 0100 1110 0101 0100 0101 0011 0100 0001 1101 0000 0011

   0D207 52 45 44 41 10 44 24 C5   REDA.D$.  0101 0010 0100 0101 0100 0100 0100 0001 0001 0000 0100 0100 0010 0100 1100 0101

   0D20F 54 58 04 20 1F 73 03 20   TX. .s.   0101 0100 0101 1000 0000 0100 0010 0000 0001 1111 0111 0011 0000 0011 0010 0000

   0D217 1F A3 20 1F B3 20 1F C3   .. .. ..  0001 1111 1010 0011 0010 0000 0001 1111 1011 0011 0010 0000 0001 1111 1100 0011

   0D21F 20 1F D3 20 1F E3 20 1F    .. .. .  0010 0000 0001 1111 1101 0011 0010 0000 0001 1111 1110 0011 0010 0000 0001 1111

   0D227 FE                       .         1111 1110

   0D228 46 E8                    crc
```

20.4.84 LAYOUT (varies)

```
       Length                MS     ---      Entity length (not counting itself or CRC).
       Type                  BS       0      typecode (internal DWG type code).
R2000+:
```

Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Page setup name	TV	1	plotsettings page setup name
Printer/Config	TV	2	plotsettings printer or configuration file
Plot layout flags	BS	70	plotsettings plot layout flag
Left Margin	BD	40	plotsettings left margin in millimeters
Bottom Margin	BD	41	plotsettings bottom margin in millimeters
Right Margin	BD	42	plotsettings right margin in millimeters
Top Margin	BD	43	plotsettings top margin in millimeters
Paper Width	BD	44	plotsettings paper width in millimeters
Paper Height	BD	45	plotsettings paper height in millimeters
Paper Size	TV	4	plotsettings paper size
Plot origin	2BD	46,47	plotsettings origin offset in millimeters
Paper units	BS	72	plotsettings plot paper units
Plot rotation	BS	73	plotsettings plot rotation
Plot type	BS	74	plotsettings plot type
Window min	2BD	48,49	plotsettings plot window area lower left
Window max	2BD	140,141	plotsettings plot window area upper right
R13-R2000 Only:			
Plot view name	T	6	plotsettings plot view name
Common:			
Real world units	BD	142	plotsettings numerator of custom print scale
Drawing units	BD	143	plotsettings denominator of custom print scale
Current style sheet	TV	7	plotsettings current style sheet
Scale type	BS	75	plotsettings standard scale type
Scale factor	BD	147	plotsettings scale factor
Paper image origin	2BD	148,149	plotsettings paper image origin
R2004+:			
Shade plot mode	BS	76	
Shade plot res. Level	BS	77	

```
       Shade plot custom DPI  BS      78
Common:
       Layout name            TV       1      layout name
       Tab order              BL      71      layout tab order
       Flag                   BS      70      layout flags
       Ucs origin            3BD      13      layout ucs origin
       Limmin                2RD      10      layout minimum limits
       Limmax                2RD      11      layout maximum limits
       Inspoint              3BD      12      layout insertion base point
       Ucs x axis            3BD      16      layout ucs x axis direction
       Ucs y axis            3BD      17      layout ucs y axis direction
       Elevation              BD     146      layout elevation
       Orthoview type         BS      76      layout orthographic view type of UCS
       Extmin                3BD      14      layout extent min
       Extmax                3BD      15      layout extent max
R2004+:
       Viewport count         RL              # of viewports in this layout
Common:
       Handle refs             H              parenthandle (soft pointer)
                                              [Reactors (soft pointer])
                                              xdicobjhandle (hard owner)
R2004+:
                                       6      plot view handle (hard pointer)
R2007+:
                                              Visual Style handle (soft pointer)
Common:
                                     330      associated paperspace block record handle (soft
                                              pointer)
                                     331      last active viewport handle (soft pointer)
                                     346      base ucs handle (hard pointer)
                                     345      named ucs handle (hard pointer)
R2004+:
                                              Viewport handle (repeats Viewport count times) (soft
                                              pointer)
```

20.4.85 LWPLINE (varies)

```
       Common Entity Data
       Flag                   BS      70
if (flag & 4) {
        constwidth            BD      43      Constant width for this lwpline
}
```

```
if (flag & 8) {
       elevation             BD     38     Elevation of this lwpline
}
if (flag & 2) {
       thickness             BD     39     thickness of this lwpline
}
if (flag & 1) {
       normal               3BD    210     extrusion direction
}
       numpoints             BL     90     number of verts
if (flag & 16) {
       numbulges             BL            number of bulges (when present, always same as
                                           number of verts in all examples so fa)r
}
R2010+:
If (flag & 1024) {
       vertexIdCount         BL            number of vertex identifiers (when present, always
                                           the same as number of vertes).
}
if (flag & 32) {
       numwidths             BL            number of width entries (when present, always same
                                           as number of verts in all examples so far.
}
R13-R14 Only:
repeat numpoints times
       pt0                  2RD     10     vertex location
end repeat
R2000+:
       pt0                  2RD     10     first vertex
repeat numpoints-1 times
       x                     DD     10     use previous point x value for default
       y                     DD     20     use previous point y value for default
end repeat
Common:
repeat numbulges times
       bulge                 BD     42     bulge value
end repeat
repeat vertexIdCount times
       vertex id             BL     91     The vertex identifier
end repeat
repeat numwidths times
```

```
        widths                2BD   40/41      start, end widths
end repeat
        Common Entity Handle Data
```

20.4.85.1 Example:

```
OBJECT: proxy (1FBH), len C8H (200), handle: 01 0F

    03E52 C8 00                     ..        1100 1000 0000 0000
    03E54 3E C0 40 80 43 D0 35 20   >.@.C.5    0011 1110 1100 0000 0100 0000 1000 0000 0100 0011 1101 0000 0011 0101 0010 0000
    03E5C 10 94 60 10 08 2A 0C 00   ..`..*..   0001 0000 1001 0100 0110 0000 0001 0000 0000 1000 0010 1010 0000 1100 0000 0000
    03E64 00 5E A0 20 80 12 14 85   .^. ....   0000 0000 0101 1110 1010 0000 0010 0000 1000 0000 0001 0010 0001 0100 1000 0101
    03E6C 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    03E74 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    03E7C 00 00 00 00 00 00 14 A0   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0100 1010 0000
    03E84 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    03E8C 00 00 00 00 00 00 14 A0   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0100 1010 0000
    03E94 00 00 00 00 00 00 78 1F   ......x.   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0111 1000 0001 1111
    03E9C 80 00 00 00 00 40 23 20   .....@#    1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 0000 0010 0011 0010 0000
    03EA4 00 00 00 00 00 00 78 1F   ......x.   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0111 1000 0001 1111
    03EAC 80 00 00 00 00 40 23 20   .....@#    1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 0000 0010 0011 0010 0000
    03EB4 00 00 00 00 00 00 00 20   .......    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 0000
    03EBC 00 00 00 00 00 40 23 20   .....@#    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 0000 0010 0011 0010 0000
    03EC4 00 00 00 00 00 00 1E 20   .......    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 1110 0010 0000
    03ECC 00 00 00 00 00 40 23 20   .....@#    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0100 0000 0010 0011 0010 0000
    03ED4 00 00 00 00 00 00 1E A0   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 1110 1010 0000
    03EDC 00 00 00 00 00 00 14 A0   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0100 1010 0000
    03EE4 00 00 00 00 00 00 1E A0   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 1110 1010 0000
    03EEC 00 00 00 00 00 00 14 A0   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0100 1010 0000
    03EF4 00 00 00 00 00 00 1F 20   .......    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 1111 0010 0000
    03EFC 00 00 00 00 00 00 00 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
    03F04 00 00 00 00 00 00 1F 20   .......    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 1111 0010 0000
    03F0C 55 1F FF FF FF FF FF FD   U.......   0101 0101 0001 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1101
    03F14 F7 F5 56 08 19 82 88 7A   ..V....z   1111 0111 1111 0101 0101 0110 0000 1000 0001 1001 1000 0010 1000 1000 0111 1010
    03F1C 85 93                     crc
```

20.4.86 MLeaderAnnotContext

This is a helper class for the multileader entity (see paragraph 20.4.48), that inherits from class AcDbAnnotScaleObjectContextData (see paragraph 20.4.71).

This object mainly contains a content object, which is either a block or multiline text. To the content object one or two leader roots are attached. They are either attached to the left/right or top/bottom depending on the multileaders attachment direction (horizontal/vertical). Each leader root can contain one more leader lines.

Version	**Field type**	**DXF group code**	**Description**
	…		Common AcDbAnnotScaleObjectContextData data (see paragraph 20.4.71).
		300	DXF: “CONTEXT_DATA{“
	BL	-	Number of leader roots
			Begin repeat leader root
		302	DXF: “LEADER{“
	B	290	Is content valid (ODA writes true)
	B	291	Unknown (ODA writes true)
	3BD	10	Connection point
	3BD	11	Direction
	BL		Number of break start/end point pairs
			Begin repeat break start/end point pairs
	3BD	12	Break start point
	3BD	13	Break end point
			End repeat break start/end point pairs
	BL	90	Leader index
	BD	40	Landing distance
	BL		Number of leader lines
			Begin repeat leader lines
		304	DXF: “LEADER_LINE{“
	BL	-	Number of points
			Begin repeat points
	3BD	10	Point
			End repeat points
	BL		Break info count
	BL	90	Segment index
	BL		Start/end point pair count
			Begin repeat start/end point pairs
	3BD	11	Start Point
	3BD	12	End point
			End repeat start/end point pairs
	BL	91	Leader line index.
R2010			

	BS	170	Leader type (0 = invisible leader, 1 = straight leader, 2 = spline leader)
	CMC	92	Line color
	H	340	Line type handle (hard pointer)
	BL	171	Line weight
	BD	40	Arrow size
	H	341	Arrow symbol handle (hard pointer)
	BL	93	Override flags (1 = leader type, 2 = line color, 4 = line type, 8 = line weight, 16 = arrow size, 32 = arrow symbol (handle)
Common			
	-	305	DXF: “}”
			End repeat leader lines
R2010			
	BS	271	Attachment direction (0 = horizontal, 1 = vertical, default is 0)
	-	303	DXF: “}”
			End repeat leader root
Common			
	BD	40	Overall scale
	3BD	10	Content base point
	BD	41	Text height
	BD	140	Arrow head size
	BD	145	Landing gap
	BS	174	Style left text attachment type. See also MLEADER style left text attachment type for values. Relevant if mleader attachment direction is horizontal.
	BS	175	Style right text attachment type. See also MLEADER style left text attachment type for values. Relevant if mleader attachment direction is horizontal.
	BS	176	Text align type (0 = left, 1 = center, 2 = right)
	BS	177	Attachment type (0 = content extents, 1 = insertion point).
	B	290	Has text contents
			IF Has text contents
	TV	304	Text label
	3BD	11	Normal vector
	H	340	Text style handle (hard pointer)
	3BD	12	Location
	3BD	13	Direction
	BD	42	Rotation (radians)
	BD	43	Boundary width
	BD	44	Boundary height
	BD	45	Line spacing factor
	BS	170	Line spacing style (1 = at least, 2 = exactly)
	CMC	90	Text color
	BS	171	Alignment (1 = left, 2 = center, 3 = right)
	BS	172	Flow direction (1 = horizontal, 3 = vertical, 6 = by style)
	CMC	91	Background fill color
	BD	141	Background scale factor

	BL	92	Background transparency
	B	291	Is background fill enabled
	B	292	Is background mask fill on
	BS	173	Column type (ODA writes 0), *TODO: what meaning for values?
	B	293	Is text height automatic?
	BD	142	Column width
	BD	143	Column gutter
	B	294	Column flow reversed
	BL		Column sizes count
			Begin repeat column sizes
	BD	144	Column size
			End repeat column sizes
	B	295	Word break
	B		Unknown
			ELSE (Has text contents)
	B	296	Has contents block
			IF Has contents block
	H	341	AcDbBlockTableRecord handle (soft pointer)
	3BD	14	Normal vector
	3BD	15	Location
	3BD	16	Scale vector
	BD	46	Rotation (radians)
	CMC	93	Block color
	BD (16)	47	16 doubles containg the complete transformation matrix. Order of transformation is: • Rotation, • OCS to WCS (using normal vector), • Scaling (using scale vector), • Translation (using location)
			END IF Has contents block
			END IF Has text contents
	3BD	110	Base point
	3BD	111	Base direction
	3BD	112	Base vertical
	B	297	Is normal reversed?
R2010			
	BS	273	Style top attachment. See also MLEADER style left text attachment type for values. Relevant if mleader attachment direction is vertical.
	BS	272	Style bottom attachment. See also MLEADER style left text attachment type for values. Relevant if mleader attachment direction is vertical.
	-	301	DXF: “}”

20.4.87 MLEADERSTYLE (AcDbMLeaderStyle)

This class inherits from AcDbObject. The provides a style for the MLEADER entity (see paragraph 20.4.48).

The value of IsNewFormat is true in case the version is R2010 or later, or if the object has extended data for APPID “ACAD_MLEADERVER”.

Version	Field type	DXF group code	Description
	…		Common object data (paragraph 20.1).
R2010			
	BS	179	Version (expected to have value 2)
Common			
	BS	170	Content type (see paragraph on LEADER for more details).
	BS	171	Draw multi-leader order (0 = draw content first, 1 = draw leader first)
	BS	172	Draw leader order (0 = draw leader head first, 1 = draw leader tail first)
	BL	90	Maximum number of points for leader
	BD	40	First segment angle (radians)
	BD	41	Second segment angle (radians)
	BS	173	Leader type (see paragraph on LEADER for more details).
	CMC	91	Leader line color
	H	340	Leader line type handle (hard pointer)
	BL	92	Leader line weight
	B	290	Is landing enabled?
	BD	42	Landing gap
	B	291	Auto include landing (is dog-leg enabled?)
	BD	43	Landing distance
	TV	3	Style description
	H	341	Arrow head block handle (hard pointer)
	BD	44	Arrow head size
	TV	300	Text default
	H	342	Text style handle (hard pointer)
	BS	174	Left attachment (see paragraph on LEADER for more details).
	BS	178	Right attachment (see paragraph on LEADER for more details).
			IF IsNewFormat OR DXF file
	BS	175	Text angle type (see paragraph on LEADER for more details).
			END IF IsNewFormat OR DXF file
	BS	176	Text alignment type
	CMC	93	Text color
	BD	45	Text height
	B	292	Text frame enabled
			IF IsNewFormat OR DXF file
	B	297	Always align text left

			END IF IsNewFormat OR DXF file
	BD	46	Align space
	H	343	Block handle (hard pointer)
	CMC	94	Block color
	3BD	47, 49, 140	Block scale vector
	B	293	Is block scale enabled
	BD	141	Block rotation (radians)
	B	294	Is block rotation enabled
	BS	177	Block connection type (0 = MLeader connects to the block extents, 1 = MLeader connects to the block base point)
	BD	142	Scale factor
	B	295	Property changed, meaning not totally clear, might be set to true if something changed after loading, or might be used to trigger updates in dependent MLeaders.
	B	296	Is annotative?
	BD	143	Break size
R2010+			
	BS	271	Attachment direction (see paragraph on LEADER for more details).
	BS	273	Top attachment (see paragraph on LEADER for more details).
	BS	272	Bottom attachment (see paragraph on LEADER for more details).

20.4.88 OLE2FRAME (varies)

```
       Common Entity Data
       Flags                 BS     70
R2000+:
       Mode                  BS
Common:
       Data Length           BL     --     Bit-pair-coded long giving the length of the data
                                           section that follows.
       Unknown data           -     --     The OLE2 data.
R2000+:
       Unknown               RC
Common:
       Common Entity Handle Data
       CRC                    X    ---
```

20.4.88.1 <<No example>>

20.4.89 AcDbObjectContextData

This class inherits from AcDbObject. The object provides contextual data for another object/entity.

Version	Field type	DXF group code	Description
	…		Common object data (paragraph 20.1).
R2010			
	BS	70	Version (default value is 3).
	B	-	Has file to extension dictionary (default value is true).
	B	290	Default flag (default value is false).

20.4.90 PROXY (varies):

```
       Length                 MS    ---     Entity length (not counting itself or CRC).
       Type                   BS      0     typecode (internal DWG type code).
R2000+:
       Obj size               RL            size of object in bits, not including end handles
Common:
       Handle                  H      5     Length (char) followed by the handle bytes.
       EED                     X     -3     See EED section.
R13-R14 Only:
       Obj size               RL            size of object in bits, not including end handles
Common:
       Numreactors            BL            Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag       B            If 1, no XDictionary handle is stored for this
                                            object, otherwise XDictionary handle is stored as in
                                            R2000 and earlier.
R2000+:
       Class ID               BL     91
Before R2018:
       Object Drawing Format  BL     95     This is a bitwise OR of the version and the
                                            maintenance version, shifted 16 bits to the left.
R2018+:
       Version                BL     71     The AutoCAD version of the object.
       Maintenance version    BL     97     The AutoCAD maintenance version of the object.
R2000+:
       Original Data Format    B     70     0 for dwg, 1 for dxf
Common:
       Databits                X            databits, however many there are to the handles
       Handle refs             H            parenthandle (soft pointer)
                                            [Reactors (soft pointer)]
                                            xdicobjhandle (hard owner)
```

			objid object handles, as many as we can read until we run out of data. These are TYPEDOBJHANDLEs.

20.4.90.1 <<No example>>

20.4.91 RASTERVARIABLES (varies)

(used in conjunction with IMAGE entities)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	typecode (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Classver	BL	90	classversion
Dispfrm	BS	70	displayframe
Dispqual	BS	71	display quality
Units	BS	72	units
Handle refs	H		parenthandle (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)

20.4.91.1 Example:

```
OBJECT: proxy (1F5H), len 11H (17), handle: 5A

    0CD78 11 00                    ..        0001 0001 0000 0000

    0CD7A 3D 40 40 56 A6 60 00 00   =@@V.`..   0011 1101 0100 0000 0100 0000 0101 0110 1010 0110 0110 0000 0000 0000 0000 0000

    0CD82 04 06 40 50 19 01 04 30   ..@P...0   0000 0100 0000 0110 0100 0000 0101 0000 0001 1001 0000 0001 0000 0100 0011 0000

    0CD8A C0                       .         1100 0000

    0CD8B DC D2                    crc
```

20.4.92 SCALE (AcDbScale)

This class inherits from AcDbObject. This represents a ratio of paper units to drawing units, where the drawing units are divided by 10 when using the same distance units (e.g. mm). E.g. a scale of 1 mm to 10 mm is stored as paper units = 1, drawing units = 1. A scale of 1 mm to 1000 mm (= 1 m) is stored as paper units = 1, drawing units = 100.

Version	Field type	DXF group code	Description
	…		Common object data (see paragraph 20.1.
	BS	70	Unknown (ODA writes 0).
	TV	300	Name
	BD	140	Paper units (numerator)
	BD	141	Drawing units (denominator, divided by 10).
	B	290	Has unit scale

20.4.93 SORTENTSTABLE (varies)

```
       Length               MS    ---     Entity length (not counting itself or CRC).
       Type                 BS      0     typecode (internal DWG type code).
R2000+:
       Obj size             RL            size of object in bits, not including end handles
Common:
       Handle                H      5     Length (char) followed by the handle bytes.
       EED                   X     -3     See EED section.
R13-R14 Only:
       Obj size             RL            size of object in bits, not including end handles
Common:
       Numreactors          BL            Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag     B            If 1, no XDictionary handle is stored for this
                                          object, otherwise XDictionary handle is stored as in
                                          R2000 and earlier.
Common:
       Numentries           BL            number of entries
       Sorthandle            H            Sort handle (numentries of these, CODE 0, i.e. part
                                          of the main bit stream, not of the handle bit
                                          stream!). The sort handle does not have to point to
                                          an entity (but it can). This is just the handle used
                                          for determining the drawing order of the entity
                                          specified by the entity handle in the handle bit
                                          stream. When the sortentstable doesn’t have a
```

			mapping from entity handle to sort handle, then the entity's own handle is used for sorting.
Handle refs	H		parenthandle (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)
			owner handle (soft pointer)
			handles of entities (numentries of these, soft pointer)

20.4.93.1 *Example:*

```
OBJECT: proxy (1FAH), len 59H (89), handle: A5

    0D015 59 00                    Y.        0101 1001 0000 0000

    0D017 3E 80 40 69 67 80 10 00   >.@ig...   0011 1110 1000 0000 0100 0000 0110 1001 0110 0111 1000 0000 0001 0000 0000 0000

    0D01F 04 05 12 01 6E 01 68 01   ....n.h.   0000 0100 0000 0101 0001 0010 0000 0001 0110 1110 0000 0001 0110 1000 0000 0001

    0D027 6C 01 5E 01 53 01 6A 01   l.^.S.j.   0110 1100 0000 0001 0101 1110 0000 0001 0101 0011 0000 0001 0110 1010 0000 0001

    0D02F 60 01 95 01 58 01 A6 01   `...X...   0110 0000 0000 0001 1001 0101 0000 0001 0101 1000 0000 0001 1010 0110 0000 0001

    0D037 6F 01 6D 01 54 01 6B 01   o.m.T.k.   0110 1111 0000 0001 0110 1101 0000 0001 0101 0100 0000 0001 0110 1011 0000 0001

    0D03F 56 01 69 01 76 01 55 40   V.i.v.U@   0101 0110 0000 0001 0110 1001 0000 0001 0111 0110 0000 0001 0101 0101 0100 0000

    0D047 41 A4 30 41 19 41 6D 41   A.0A.AmA   0100 0001 1010 0100 0011 0000 0100 0001 0001 1001 0100 0001 0110 1101 0100 0001

    0D04F 60 41 6B 41 56 41 A6 41   `AkAVA.A   0110 0000 0100 0001 0110 1011 0100 0001 0101 0110 0100 0001 1010 0110 0100 0001

    0D057 69 41 58 41 76 41 54 41   iAXAvATA   0110 1001 0100 0001 0101 1000 0100 0001 0111 0110 0100 0001 0101 0100 0100 0001

    0D05F 95 41 6E 41 6C 41 55 41   .AnAlAUA   1001 0101 0100 0001 0110 1110 0100 0001 0110 1100 0100 0001 0101 0101 0100 0001

    0D067 6A 41 53 41 68 41 6F 41   jASAhAoA   0110 1010 0100 0001 0101 0011 0100 0001 0110 1000 0100 0001 0110 1111 0100 0001

    0D06F 5E                       ^         0101 1110

    0D070 D3 A5                    crc
```

20.4.94 SPATIAL_FILTER (varies)

(used to clip external references)

Length	MS	---	Entity length (not counting itself or CRC).
Type	BS	0	typecode (internal DWG type code).
R2000+:			
Obj size	RL		size of object in bits, not including end handles
Common:			
Handle	H	5	Length (char) followed by the handle bytes.
EED	X	-3	See EED section.
R13-R14 Only:			
Obj size	RL		size of object in bits, not including end handles
Common:			

Numreactors	BL		Number of persistent reactors attached to this obj
R2004+:			
XDic Missing Flag	B		If 1, no XDictionary handle is stored for this object, otherwise XDictionary handle is stored as in R2000 and earlier.
Common:			
Numpts	BS	70	number of points /* really long? */
Repeat numpts times:			
pt0	2RD	10	a point on the clip boundary
End repeat			
Extrusion	3BD	210	extrusion
Clipbdorg	3BD	10	clip bound origin
Dispbound	BS	71	display boundary
Frontclipon	BS	72	1 if front clip on
Frontdist	BD	40	front clip dist (present if frontclipon==1)
Backclipon	BS	73	1 if back clip on
Backdist	BD	41	back clip dist (present if backclipon==1)
Invblktr	12BD	40	inverse block transformation matrix
			(double [4][3], column major order)
clipbdtr	12BD	40	clip bound transformation matrix
			(double [4][3], column major order)
Handle refs	H		parenthandle (soft pointer)
			[Reactors (soft pointer)]
			xdicobjhandle (hard owner)

20.4.94.1 Example:

```
OBJECT: proxy (1FDH), len 7BH (123), handle: 02 15

   0D68A 7B 00                  {.        0111 1011 0000 0000

   0D68C 3F 40 40 80 85 6A A0 30   ?@@..j.0   0011 1111 0100 0000 0100 0000 1000 0000 1000 0101 0110 1010 1010 0000 0011 0000

   0D694 00 04 05 05 96 EA 02 5E   .......^   0000 0000 0000 0100 0000 0101 0000 0101 1001 0110 1110 1010 0000 0010 0101 1110

   0D69C 66 70 2E 40 3A AF B1 4B   fp.@:..K   0110 0110 0111 0000 0010 1110 0100 0000 0011 1010 1010 1111 1011 0001 0100 1011

   0D6A4 54 7F 16 40 27 E0 D7 48   T..@'..H   0101 0100 0111 1111 0001 0110 0100 0000 0010 0111 1110 0000 1101 0111 0100 1000

   0D6AC 12 9C 30 40 4A F2 5C DF   ..0@J.\.   0001 0010 1001 1100 0011 0000 0100 0000 0100 1010 1111 0010 0101 1100 1101 1111

   0D6B4 87 03 14 40 B5 AB 90 F2   ...@....   1000 0111 0000 0011 0001 0100 0100 0000 1011 0101 1010 1011 1001 0000 1111 0010

   0D6BC 93 F6 31 40 82 75 1C 3F   ..1@.u.?   1001 0011 1111 0110 0011 0001 0100 0000 1000 0010 0111 0101 0001 1100 0011 1111

   0D6C4 54 3A 17 40 75 79 73 B8   T:.@uys.   0101 0100 0011 1010 0001 0111 0100 0000 0111 0101 0111 1001 0111 0011 1011 1000
   0D6CC 56 D7 32 40 EF 3D 5C 72   V.2@.=\r   0101 0110 1101 0111 0011 0010 0100 0000 1110 1111 0011 1101 0101 1100 0111 0010

   0D6D4 DC 11 20 40 74 94 83 D9   .. @t...   1101 1100 0001 0001 0010 0000 0100 0000 0111 0100 1001 0100 1000 0011 1101 1001

   0D6DC 04 00 2E 40 E7 DF 2E FB   ...@....   0000 0100 0000 0000 0010 1110 0100 0000 1110 0111 1101 1111 0010 1110 1111 1011

   0D6E4 75 A7 20 40 A6 A4 06 9A   u. @....   0111 0101 1010 0111 0010 0000 0100 0000 1010 0110 1010 0100 0000 0110 1001 1010

   0D6EC 0F 88 C4 46 B0 5D 8A 70   ...F.].p   0000 1111 1000 1000 1100 0100 0100 0110 1011 0000 0101 1101 1000 1010 0111 0000

   0D6F4 26 06 E1 49 2C DE A1 C0   &..I,...   0010 0110 0000 0110 1110 0001 0100 1001 0010 1100 1101 1110 1010 0001 1100 0000

   0D6FC 70 29 9A A6 A9 90 10 80   p)......   0111 0000 0010 1001 1001 1010 1010 0110 1010 1001 1001 0000 0001 0000 1000 0000

   0D704 85 0C 10                  ...        1000 0101 0000 1100 0001 0000

   0D707 07 5E                     crc
```

20.4.95 SPATIAL_INDEX (varies):

```
        Length              MS    ---     Entity length (not counting itself or CRC).
        Type                BS      0     typecode (internal DWG type code).
R2000+:
        Obj size            RL            size of object in bits, not including end handles
Common:
        Handle               H      5     Length (char) followed by the handle bytes.
        EED                  X     -3     See EED section.
R13-R14 Only:
        Obj size            RL            size of object in bits, not including end handles
Common:
        Numreactors         BL            Number of persistent reactors attached to this obj
R2004+:
        XDic Missing Flag    B            If 1, no XDictionary handle is stored for this
                                          object, otherwise XDictionary handle is stored as in
                                          R2000 and earlier.
Common:
        timestamp1          BL
        timestamp2          BL
        unknown              X            rest of bits to handles
        Handle refs          H            parenthandle (hard owner)
                                          [Reactors (soft pointer)]
                                          xdictionary (hard owner)
```

20.4.95.1 Example:

```
OBJECT: proxy (200H), len 406H (1030), handle: 01 F9

0D280 06 04                     ..         0000 0110 0000 0100

0D282 00 00 80 80 7E 63 A1 F0   ....~c..   0000 0000 0000 0000 1000 0000 1000 0000 0111 1110 0110 0011 1010 0001 1111 0000
```

```
0D28A 00 04 04 61 65 25 00 3B   ...ae%.;   0000 0000 0000 0100 0000 0100 0110 0001 0110 0101 0010 0101 0000 0000 0011 1011

0D292 3A 89 80 88 F8 D8 33 54   :.....3T   0011 1010 1000 1001 1000 0000 1000 1000 1111 1000 1101 1000 0011 0011 0101 0100

0D29A 4E 3A 94 10 02 D6 3C 73   N:....<s   0100 1110 0011 1010 1001 0100 0001 0000 0000 0010 1101 0110 0011 1100 0111 0011

0D2A2 98 D3 04 FC 1F CD 85 40   .......@   1001 1000 1101 0011 0000 0100 1111 1100 0001 1111 1100 1101 1000 0101 0100 0000

0D2AA 69 D4 B2 41 18 08 F6 18   i..A....   0110 1001 1101 0100 1011 0010 0100 0001 0001 1000 0000 1000 1111 0110 0001 1000

0D2B2 FB 39 79 2F C4 29 C3 30   .9y/.).0   1111 1011 0011 1001 0111 1001 0010 1111 1100 0100 0010 1001 1100 0011 0011 0000

0D2BA E2 0C 6C 84 10 00 00 89   ..l.....   1110 0010 0000 1100 0110 1100 1000 0100 0001 0000 0000 0000 0000 0000 1000 1001

0D2C2 17 FE A4 92 FC 25 03 00   .....%..   0001 0111 1111 1110 1010 0100 1001 0010 1111 1100 0010 0101 0000 0011 0000 0000

0D2CA 00 01 00 00 00 FF FF 00   ........   0000 0000 0000 0001 0000 0000 0000 0000 0000 0000 1111 1111 1111 1111 0000 0000

0D2D2 00 FF FF 00 00 FF FF 01   ........   0000 0000 1111 1111 1111 1111 0000 0000 0000 0000 1111 1111 1111 1111 0000 0001

0D2DA 00 00 04 58 00 D4 08 00   ...X....   0000 0000 0000 0000 0000 0100 0101 1000 0000 0000 1101 0100 0000 1000 0000 0000

0D2E2 00 01 08 00 00 FE 8F 00   ........   0000 0000 0000 0001 0000 1000 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000

0D2EA 00 FE 8F 00 00 FE 8F 00   ........   0000 0000 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000

0D2F2 00 01 08 00 00 FE 50 00   ......P.   0000 0000 0000 0001 0000 1000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0000

0D2FA 00 FE 50 00 00 FE 50 03   ..P...P.   0000 0000 1111 1110 0101 0000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0011

0D302 00 00 09 56 00 8B 01 00   ...V....   0000 0000 0000 0000 0000 1001 0101 0110 0000 0000 1000 1011 0000 0001 0000 0000

0D30A 36 00 00 08 00 00 02 01   6.......   0011 0110 0000 0000 0000 0000 0000 1000 0000 0000 0000 0000 0000 0010 0000 0001

0D312 09 00 3F FE 8F 00 00 FE   ..?.....   0000 1001 0000 0000 0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110

0D31A 50 00 00 FE 50 02 00 00   P...P...   0101 0000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0010 0000 0000 0000 0000

0D322 07 E2 01 00 33 00 36 00   ....3.6.   0000 0111 1110 0010 0000 0001 0000 0000 0011 0011 0000 0000 0011 0110 0000 0000

0D32A 00 00 02 01 0A 00 00 FE   ........   0000 0000 0000 0000 0000 0010 0000 0001 0000 1010 0000 0000 0000 0000 1111 1110

0D332 50 00 3F FE 8F 00 00 FE   P.?.....   0101 0000 0000 0000 0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110

0D33A 50 03 00 00 09 DE 01 00   P.......   0101 0000 0000 0011 0000 0000 0000 0000 0000 1001 1101 1110 0000 0001 0000 0000

0D342 13 00 22 00 00 00 00 00   ..".....   0001 0011 0000 0000 0010 0010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

0D34A 02 01 0B 00 3F FE 8F 00   ....?...   0000 0010 0000 0001 0000 1011 0000 0000 0011 1111 1111 1110 1000 1111 0000 0000

0D352 3F FE 8F 00 00 FE 50 04   ?.....P.   0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110 0101 0000 0000 0100

0D35A 00 00 0B CF 01 00 01 00   ........   0000 0000 0000 0000 0000 1011 1100 1111 0000 0001 0000 0000 0000 0001 0000 0000

0D362 01 00 34 00 00 10 00 00   ..4.....   0000 0001 0000 0000 0011 0100 0000 0000 0000 0000 0001 0000 0000 0000 0000 0000

0D36A 02 01 1A 00 00 FE 8F 00   ........   0000 0010 0000 0001 0001 1010 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000

0D372 3F FE 8F 00 00 FE 50 01   ?.....P.   0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110 0101 0000 0000 0001

0D37A 00 00 05 CD 01 00 01 00   ........   0000 0000 0000 0000 0000 0101 1100 1101 0000 0001 0000 0000 0000 0001 0000 0000

0D382 00 00 02 02 01 09 00 70   .......p   0000 0000 0000 0000 0000 0010 0000 0010 0000 0001 0000 1001 0000 0000 0111 0000
```

```
0D38A FF FF 00 00 FE 8F 00 00   ........   1111 1111 1111 1111 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000 0000 0000

0D392 FE 8F 00 00 01 08 00 70   .......p   1111 1110 1000 1111 0000 0000 0000 0000 0000 0001 0000 1000 0000 0000 0111 0000

0D39A FE C0 00 00 FE 50 00 00   .....P..   1111 1110 1100 0000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0000 0000 0000

0D3A2 FE 50 02 00 00 07 BD 01   .P......   1111 1110 0101 0000 0000 0010 0000 0000 0000 0000 0000 0111 1011 1101 0000 0001

0D3AA 00 22 00 00 00 00 00 02   .".....   0000 0000 0010 0010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010

0D3B2 01 09 00 AF FF FF 00 00   ........   0000 0001 0000 1001 0000 0000 1010 1111 1111 1111 1111 1111 0000 0000 0000 0000

0D3BA FE 50 00 00 FE 50 00 00   .P...P..   1111 1110 0101 0000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0000 0000 0000

0D3C2 02 01 0A 00 70 FE C0 00   ....p...   0000 0010 0000 0001 0000 1010 0000 0000 0111 0000 1111 1110 1100 0000 0000 0000

0D3CA 3F FE 8F 00 00 FE 50 01   ?.....P.   0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110 0101 0000 0000 0001

0D3D2 00 00 05 E0 01 00 22 00   ......".   0000 0000 0000 0000 0000 0101 1110 0000 0000 0001 0000 0000 0010 0010 0000 0000

0D3DA 00 00 02 01 0B 00 AF FF   ........   0000 0000 0000 0000 0000 0010 0000 0001 0000 1011 0000 0000 1010 1111 1111 1111

0D3E2 FF 00 3F FE 8F 00 00 FE   ..?.....   1111 1111 0000 0000 0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110

0D3EA 50 00 00 02 02 01 0A 00   P.......   0101 0000 0000 0000 0000 0000 0000 0010 0000 0010 0000 0001 0000 1010 0000 0000

0D3F2 00 FE 8F 00 70 FF FF 00   ....p...   0000 0000 1111 1110 1000 1111 0000 0000 0111 0000 1111 1111 1111 1111 0000 0000

0D3FA 00 FE 8F 00 00 01 08 00   ........   0000 0000 1111 1110 1000 1111 0000 0000 0000 0000 0000 0001 0000 1000 0000 0000

0D402 00 FE 50 00 70 FE C0 00   ..P.p...   0000 0000 1111 1110 0101 0000 0000 0000 0111 0000 1111 1110 1100 0000 0000 0000

0D40A 00 FE 50 07 00 00 12 95   ..P.....   0000 0000 1111 1110 0101 0000 0000 0111 0000 0000 0000 0000 0001 0010 1001 0101

0D412 01 00 14 00 34 00 25 00   ....4.%.   0000 0001 0000 0000 0001 0100 0000 0000 0011 0100 0000 0000 0010 0101 0000 0000

0D41A 01 00 15 00 C7 01 00 57   .......W   0000 0001 0000 0000 0001 0101 0000 0000 1100 0111 0000 0001 0000 0000 0101 0111

0D422 02 00 00 02 01 09 00 3F   .......?   0000 0010 0000 0000 0000 0000 0000 0010 0000 0001 0000 1001 0000 0000 0011 1111

0D42A FE 8F 00 70 FE C0 00 00   ...p....   1111 1110 1000 1111 0000 0000 0111 0000 1111 1110 1100 0000 0000 0000 0000 0000

0D432 FE 50 03 00 00 09 81 02   .P......   1111 1110 0101 0000 0000 0011 0000 0000 0000 0000 0000 1001 1000 0001 0000 0010

0D43A 00 18 00 01 00 25 00 00   .....%..   0000 0000 0001 1000 0000 0000 0000 0001 0000 0000 0010 0101 0000 0000 0000 0000

0D442 00 02 01 0A 00 00 FE 50   .......P   0000 0000 0000 0010 0000 0001 0000 1010 0000 0000 0000 0000 1111 1110 0101 0000

0D44A 00 AF FF FF 00 00 FE 50   .......P   0000 0000 1010 1111 1111 1111 1111 1111 0000 0000 0000 0000 1111 1110 0101 0000

0D452 0F 00 00 21 D3 01 00 01   ...!....   0000 1111 0000 0000 0000 0000 0010 0001 1101 0011 0000 0001 0000 0000 0000 0001

0D45A 00 01 00 01 00 01 00 29   .......)   0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0010 1001

0D462 00 76 00 01 00 01 00 01   .v......   0000 0000 0111 0110 0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0000 0001

0D46A 00 01 00 01 00 01 00 01   ........   0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0000 0001

0D472 00 5A 00 D0 9A 00 00 02   .Z......   0000 0000 0101 1010 0000 0000 1101 0000 1001 1010 0000 0000 0000 0000 0000 0010

0D47A 01 0B 00 3F FE 8F 00 AF   ...?....   0000 0001 0000 1011 0000 0000 0011 1111 1111 1110 1000 1111 0000 0000 1010 1111

0D482 FF FF 00 00 FE 50 00 00   .....P..   1111 1111 1111 1111 0000 0000 0000 0000 1111 1110 0101 0000 0000 0000 0000 0000
```

```
0D48A 02 01 18 00 00 FE 8F 00   ........   0000 0010 0000 0001 0001 1000 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000

0D492 70 FE C0 00 00 FE 50 01   p.....P.   0111 0000 1111 1110 1100 0000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0001

0D49A 00 00 04 54 00 00 00 00   ...T....   0000 0000 0000 0000 0000 0100 0101 0100 0000 0000 0000 0000 0000 0000 0000 0000

0D4A2 00 02 02 01 0B 00 70 FF   ......p.   0000 0000 0000 0010 0000 0010 0000 0001 0000 1011 0000 0000 0111 0000 1111 1111

0D4AA FF 00 70 FF FF 00 00 FE   ..p.....   1111 1111 0000 0000 0111 0000 1111 1111 1111 1111 0000 0000 0000 0000 1111 1110

0D4B2 8F 00 00 01 08 00 70 FE   ......p.   1000 1111 0000 0000 0000 0000 0000 0001 0000 1000 0000 0000 0111 0000 1111 1110

0D4BA C0 00 70 FE C0 00 00 FE   ..p.....   1100 0000 0000 0000 0111 0000 1111 1110 1100 0000 0000 0000 0000 0000 1111 1110

0D4C2 50 01 00 00 05 84 02 00   P.......   0101 0000 0000 0001 0000 0000 0000 0000 0000 0101 1000 0100 0000 0010 0000 0000

0D4CA 78 00 00 00 02 01 09 00   x.......   0111 1000 0000 0000 0000 0000 0000 0000 0000 0010 0000 0001 0000 1001 0000 0000

0D4D2 AF FF FF 00 70 FE C0 00   ....p...   1010 1111 1111 1111 1111 1111 0000 0000 0111 0000 1111 1110 1100 0000 0000 0000

0D4DA 00 FE 50 07 00 00 11 EE   ..P.....   0000 0000 1111 1110 0101 0000 0000 0111 0000 0000 0000 0000 0001 0001 1110 1110

0D4E2 03 00 01 00 01 00 01 00   ........   0000 0011 0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0000 0001 0000 0000

0D4EA 03 00 01 00 01 00 00 77   .......w   0000 0011 0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0000 0000 0111 0111

0D4F2 00 00 02 01 0A 00 70 FE   ......p.   0000 0000 0000 0000 0000 0010 0000 0001 0000 1010 0000 0000 0111 0000 1111 1110

0D4FA C0 00 AF FF FF 00 00 FE   ........   1100 0000 0000 0000 1010 1111 1111 1111 1111 1111 0000 0000 0000 0000 1111 1110

0D502 50 00 00 02 01 0B 00 AF   P.......   0101 0000 0000 0000 0000 0000 0000 0010 0000 0001 0000 1011 0000 0000 1010 1111

0D50A FF FF 00 AF FF FF 00 00   ........   1111 1111 1111 1111 0000 0000 1010 1111 1111 1111 1111 1111 0000 0000 0000 0000

0D512 FE 50 02 00 00 07 F2 03   .P......   1111 1110 0101 0000 0000 0010 0000 0000 0000 0000 0000 0111 1111 0010 0000 0011

0D51A 00 01 00 FA FC 00 00 02   ........   0000 0000 0000 0001 0000 0000 1111 1010 1111 1100 0000 0000 0000 0000 0000 0010

0D522 02 01 18 00 00 FF FF 00   ........   0000 0010 0000 0001 0001 1000 0000 0000 0000 0000 1111 1111 1111 1111 0000 0000

0D52A 00 FE 8F 00 00 FE 8F 00   ........   0000 0000 1111 1110 1000 1111 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000

0D532 00 01 18 00 00 FF FF 00   ........   0000 0000 0000 0001 0001 1000 0000 0000 0000 0000 1111 1111 1111 1111 0000 0000

0D53A 00 FE 50 00 00 FE 50 01   ..P...P.   0000 0000 1111 1110 0101 0000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0001

0D542 00 00 05 EE 01 00 01 00   ........   0000 0000 0000 0000 0000 0101 1110 1110 0000 0001 0000 0000 0000 0001 0000 0000

0D54A 00 00 02 02 01 1A 00 00   ........   0000 0000 0000 0000 0000 0010 0000 0010 0000 0001 0001 1010 0000 0000 0000 0000

0D552 FF FF 00 70 FF FF 00 00   ...p....   1111 1111 1111 1111 0000 0000 0111 0000 1111 1111 1111 1111 0000 0000 0000 0000

0D55A FE 8F 01 00 00 05 A6 01   ........   1111 1110 1000 1111 0000 0001 0000 0000 0000 0000 0000 0101 1010 0110 0000 0001

0D562 00 00 00 00 00 01 1A 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0001 1010 0000 0000

0D56A 00 FF FF 00 AF FF FF 00   ........   0000 0000 1111 1111 1111 1111 0000 0000 1010 1111 1111 1111 1111 1111 0000 0000

0D572 00 FE 50 01 00 00 04 5E   ..P....^   0000 0000 1111 1110 0101 0000 0000 0001 0000 0000 0000 0000 0000 0100 0101 1110

0D57A 00 00 00 00 00 02 02 01   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 0000 0010 0000 0001

0D582 28 00 00 FE 8F 00 00 FF   (.......   0010 1000 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000 0000 0000 1111 1111
0D58A FF 00 00 FE 8F 00 00 01   ........   1111 1111 0000 0000 0000 0000 1111 1110 1000 1111 0000 0000 0000 0000 0000 0001

0D592 28 00 00 FE 50 00 00 FF   (...P...   0010 1000 0000 0000 0000 0000 1111 1110 0101 0000 0000 0000 0000 0000 1111 1111

0D59A FF 00 00 FE 50 01 00 00   ....P...   1111 1111 0000 0000 0000 0000 1111 1110 0101 0000 0000 0001 0000 0000 0000 0000

0D5A2 04 53 00 00 00 00 00 02   .S......   0000 0100 0101 0011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010

0D5AA 01 29 00 3F FE 8F 00 00   .).?....   0000 0001 0010 1001 0000 0000 0011 1111 1111 1110 1000 1111 0000 0000 0000 0000

0D5B2 FF FF 00 00 FE 50 01 00   .....P..   1111 1111 1111 1111 0000 0000 0000 0000 1111 1110 0101 0000 0000 0001 0000 0000

0D5BA 00 04 55 00 00 00 00 00   ..U.....   0000 0000 0000 0100 0101 0101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

0D5C2 02 02 01 38 00 00 FE 8F   ...8....   0000 0010 0000 0010 0000 0001 0011 1000 0000 0000 0000 0000 1111 1110 1000 1111

0D5CA 00 00 FE 8F 00 00 FF FF   ........   0000 0000 0000 0000 1111 1110 1000 1111 0000 0000 0000 0000 1111 1111 1111 1111

0D5D2 00 00 01 3B 00 3F FE 8F   ...;.?..   0000 0000 0000 0000 0000 0001 0011 1011 0000 0000 0011 1111 1111 1110 1000 1111

0D5DA 00 3F FE 8F 00 00 FF FF   .?......   0000 0000 0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111 1111 1111 1111

0D5E2 01 00 00 04 60 00 00 00   ....`...   0000 0001 0000 0000 0000 0000 0000 0100 0110 0000 0000 0000 0000 0000 0000 0000

0D5EA 00 00 02 02 02 00 42 1D   ......B.   0000 0000 0000 0000 0000 0010 0000 0010 0000 0010 0000 0000 0100 0010 0001 1101

0D5F2 FC 00 00 00 03 40 00 00   .....@..   1111 1100 0000 0000 0000 0000 0000 0000 0000 0011 0100 0000 0000 0000 0000 0000

0D5FA 34 3C 7C 40 32 6D 11 40   4<|@2m.@   0011 0100 0011 1100 0111 1100 0100 0000 0011 0010 0110 1101 0001 0001 0100 0000

0D602 3F FF FF FF F7 3C 7C 40   ?....<|@   0011 1111 1111 1111 1111 1111 1111 1111 1111 0111 0011 1100 0111 1100 0100 0000

0D60A 00 40 00 00 21 0E 21 40   .@..!.!@   0000 0000 0100 0000 0000 0000 0000 0000 0010 0001 0000 1110 0010 0001 0100 0000

0D612 A1 0E 21 40 98 00 77 C0   ..!@..w.   1010 0001 0000 1110 0010 0001 0100 0000 1001 1000 0000 0000 0111 0111 1100 0000

0D61A 06 3C BC 40 03 00 00 00   .<.@....   0000 0110 0011 1100 1011 1100 0100 0000 0000 0011 0000 0000 0000 0000 0000 0000

0D622 08 3C BC 40 05 00 00 00   .<.@....   0000 1000 0011 1100 1011 1100 0100 0000 0000 0101 0000 0000 0000 0000 0000 0000

0D62A 00 00 00 00 00 00 1E 00   ........   0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 1110 0000 0000

0D632 70 88 95 C0 81 00 00 00   p.......   0111 0000 1000 1000 1001 0101 1100 0000 1000 0001 0000 0000 0000 0000 0000 0000

0D63A 11 84 50 C0 03 3C BC 40   ..P..<.@   0001 0001 1000 0100 0101 0000 1100 0000 0000 0011 0011 1100 1011 1100 0100 0000

0D642 3C 07 E4 C2 80 00 00 00   <.......   0011 1100 0000 0111 1110 0100 1100 0010 1000 0000 0000 0000 0000 0000 0000 0000

0D64A 00 00 1E 00 7F FF C0 00   ........   0000 0000 0000 0000 0001 1110 0000 0000 0111 1111 1111 1111 1100 0000 0000 0000

0D652 00 00 00 00 3A 40 00 00   ....:@..   0000 0000 0000 0000 0000 0000 0000 0000 0011 1010 0100 0000 0000 0000 0000 0000

0D65A 14 16 AB 40 00 00 00 00   ...@....   0001 0100 0001 0110 1010 1011 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000

0D662 0F FC BC 40 11 0E D0 30   ...@...0   0000 1111 1111 1100 1011 1100 0100 0000 0001 0001 0000 1110 1101 0000 0011 0000

0D66A 40 90 80 7D CC 10 80 41   @..}...A   0100 0000 1001 0000 1000 0000 0111 1101 1100 1100 0001 0000 1000 0000 0100 0001

0D672 90 80 41 D0 80 42 10 80   ..A..B..   1001 0000 1000 0000 0100 0001 1101 0000 1000 0000 0100 0010 0001 0000 1000 0000

0D67A 48 50 80 48 90 80 48 D0   HP.H..H.   0100 1000 0101 0000 1000 0000 0100 1000 1001 0000 1000 0000 0100 1000 1101 0000

0D682 80 84 90 80 87 4E         .....N     1000 0000 1000 0100 1001 0000 1000 0000 1000 0111 0100 1110
```

```
   0D688 54 B0                   crc
```

20.4.96 TABLE (varies)

The TABLE entity (entity type ACAD_TABLE) was introduced in AutoCAD 2005 (a sub release of R18), and a large number of changes were introduced in AutoCAD 2008 (a sub release of R21). The table entity inherits from the INSERT entity. The geometric results, consisting of table borders, texts and such are created in an anonymous block, similarly to the mechanism in the DIMENSION entity. The anonymous block name prefix is "*T". For the AutoCAD 2008 changes see paragraph 20.4.96.2.

TODO: document roundtrip data with connections to AcDbTableContent and AcDbTableGeometry.

20.4.96.1 *Until R21*

This paragraph describes the table DWG format until R21. In R24 the format was changed to make use of table content to contain all data (AcDbTableContent).

```
       Common Entity Data
       Ins pt              3BD     10
R13-R14 Only:
       X Scale              BD     41
       Y Scale              BD     42
       Z Scale              BD     43
R2000+ Only:
       Data flags           BB
       Scale Data                       Varies with Data flags:
                                        11 - scale is (1.0, 1.0, 1.0), no data stored.
                                        01 – 41 value is 1.0, 2 DD's are present, each using
                                        1.0 as the default value, representing the 42 and 43
                                        values.
                                        10 – 41 value stored as a RD, and 42 & 43 values are
                                        not stored, assumed equal to 41 value.
                                        00 – 41 value stored as a RD, followed by a 42 value
                                        stored as DD (use 41 for default value), and a 43
                                        value stored as a DD (use 41 value for default
                                        value).
Common:
       Rotation             BD     50
       Extrusion           3BD    210
       Has ATTRIBs           B     66   Single bit; 1 if ATTRIBs follow.
R2004+:
```

Owned Object Count	BL		Number of objects owned by this object.

Common:

Flag for table value	BS	90	Bit flags, 0x06 (0x02 + 0x04): has block, 0x10: table direction, 0 = up, 1 = down, 0x20: title suppressed. Normally 0x06 is always set.
Hor. Dir. Vector	3BD	11	
Number of columns	BL	92	
Number of rows	BL	91	
Column widths	BD	142	Repeats "# of columns" times
Row heights	BD	141	Repeats "# of rows" times

Cell data, repeats for all cells in n x m table:

Cell type	BS	171	1 = text, 2 = block. In AutoCAD 2007 a cell can contain either 1 text or 1 block. In AutoCAD 2008 this changed (TODO).
Cell edge flags	RC	172	Specifies which edges have property overrides in a cell, 1 = top, 2 = right, 4 = bottom, 8 = left. Note that if a shared edge between two cells has property overrides, the edge overrides flag is set in both adjacent cells, but in one of them the edge is marked as virtual (see virtual edge flags below). So the virtual edge flag property determines where the override is stored: each property override is stored only once. When a virtual edge flag is set, the override is determined by the adjacent cell, when it is not set it is determined by the cell itself. Normally a property override is stored with the cell on which the user made the modification, but sometimes when the user makes multiple changes in both adjacent cells, e.g. a color modification in cell A, and a line weight modification in cell B (adjacent to cell A), then for the shared edge, the property overrides for both color and line weight are stored in the same cell (either A or B). The reason for this is that the virtual edge flag doesn't allow to discriminate between individual properties, only on the edge level.
Cell merged value	B	173	Determines whether this cell is merged with another cell.
Autofit flag	B	174	
Merged width flag	BL	175	Represents the horizontal number of merged cells.
Merged height flag	BL	176	Represents the vertical number of merged cells.
Rotation value	BD	145	

If cell type == 1 (text cell):

Text string	TV	1	Present only if 344 value below is 0

If cell type == 2 (block cell):

Block scale	BD	144	
Has attributes flag	B		

If has attributes flag == 1:

Attr. Def. count	BS	179	

Attr. Def. index	BS		Not present in dxf
Attr. Def. text	TV	300	

Common to both text and block cells:

has override flag	B		

If has override flag == 1:

Cell flag override	BL	177	
Virtual edge flag	RC	178	Determines which edges are virtual, see also the explanation on the cell edge flags above. When an edge is virtual, that edge has no border overrides. 1 = top, 2 = right, 4 = bottom, 8 = left.
Cell alignment	RS	170	Present only if bit 0x01 is set in cell flag override. Top left = 1, top center = 2, top right = 3, middle left = 4, middle center = 5, middle right = 6, bottom left = 7, bottom center = 8, bottom right = 9.
Background fill none	B	283	Present only if bit 0x02 is set in cell flag override
Background color	CMC	63	Present only if bit 0x04 is set in cell flag override
Content color	CMC	64	Present only if bit 0x08 is set in cell flag override
Text style	H	7	Present only if bit 0x10 is set in cell flag override (hard pointer)
Text height	BD	140	Present only if bit 0x20 is set in cell flag override
Top grid color	CMC	69	Present only if bit 0x00040 is set in cell flag override
Top grid lineweight	BS	279	Present only if bit 0x00400 is set in cell flag override
Top visibility	BS	289	Present only if bit 0x04000 is set in cell flag override (1 = visible).
Right grid color	CMC	65	Present only if bit 0x00080 is set in cell flag override
Right grid lineweight	BS	275	Present only if bit 0x00800 is set in cell flag override
Right visibility	BS	285	Present only if bit 0x08000 is set in cell flag override (1 = visible).
Bottom grid color	CMC	66	Present only if bit 0x00100 is set in cell flag override
Bottom grid lineweight	BS	276	Present only if bit 0x01000 is set in cell flag override
Bottom visibility	BS	286	Present only if bit 0x10000 is set in cell flag override (1 = visible).
Left grid color	CMC	68	Present only if bit 0x00200 is set in cell flag override
Left grid lineweight	BS	278	Present only if bit 0x02000 is set in cell flag override

Left visibility	BS	288	Present only if bit 0x20000 is set in cell flag override (1 = visible).

R2007+:

Unknown	BL		
Value fields	…		See paragraph 20.4.98.

Common:

End Cell Data (remaining data applies to entire table)

Has table overrides	B		

If has table overrides == 1:

Table flag override	BL	93	
Title suppressed	B	280	Present only if bit 0x0001 is set in table overrides flag
Header suppresed	--	281	Always true (do not read any data for this)
Flow direction	BS	70	Present only if bit 0x0004 is set in table overrides flag (0 = down, 1 = up).
Horz. Cell margin	BD	40	Present only if bit 0x0008 is set in table overrides flag
Vert. cell margin	BD	41	Present only if bit 0x0010 is set in table overrides flag
Title row color	CMC	64	Present only if bit 0x0020 is set in table overrides flag
Header row color	CMC	64	Present only if bit 0x0040 is set in table overrides flag
Data row color	CMC	64	Present only if bit 0x0080 is set in table overrides flag
Title row fill none	B	283	Present only if bit 0x0100 is set in table overrides flag
Header row fill none	B	283	Present only if bit 0x0200 is set in table overrides flag
Data row fill none	B	283	Present only if bit 0x0400 is set in table overrides flag
Title row fill color	CMC	63	Present only if bit 0x0800 is set in table overrides flag
Header row fill clr.	CMC	63	Present only if bit 0x1000 is set in table overrides flag
Data row fill color	CMC	63	Present only if bit 0x2000 is set in table overrides flag
Title row align.	BS	170	Present only if bit 0x4000 is set in table overrides flag
Header row align.	BS	170	Present only if bit 0x8000 is set in table overrides flag
Data row align.	BS	170	Present only if bit 0x10000 is set in table overrides flag
Title text style hnd	H	7	Present only if bit 0x20000 is set in table overrides flag (hard pointer)

Title text style hnd	H	7	Present only if bit 0x40000 is set in table overrides flag (hard pointer)
Title text style hnd	H	7	Present only if bit 0x80000 is set in table overrides flag (hard pointer)
Title row height	BD	140	Present only if bit 0x100000 is set in table overrides flag
Header row height	BD	140	Present only if bit 0x200000 is set in table overrides flag
Data row height	BD	140	Present only if bit 0x400000 is set in table overrides flag
End If has table overrides == 1			
Has border color overrides		B	
If has border color overrides == 1:			
Overrides flag	BL	94	Border COLOR overrides
Title hor. Top. col.	CMC	64	Present only if bit 0x01 is set in border color overrides flag
Title hor. ins. col.	CMC	65	Present only if bit 0x02 is set in border color overrides flag
Title hor. bot. col.	CMC	66	Present only if bit 0x04 is set in border color overrides flag
Title ver. left. col.	CMC	63	Present only if bit 0x08 is set in border color overrides flag
Title ver. ins. col.	CMC	68	Present only if bit 0x10 is set in border color overrides flag
Title ver. rt. col.	CMC	69	Present only if bit 0x20 is set in border color overrides flag
Header hor. Top. col.	CMC	64	Present only if bit 0x40 is set in border color overrides flag
Header hor. ins. col.	CMC	65	Present only if bit 0x80 is set in border color overrides flag
Header hor. bot. col.	CMC	66	Present only if bit 0x100 is set in border color overrides flag
Header ver. left. col.	CMC	63	Present only if bit 0x200 is set in border color overrides flag
Header ver. ins. col.	CMC	68	Present only if bit 0x400 is set in border color overrides flag
Header ver. rt. col.	CMC	69	Present only if bit 0x800 is set in border color overrides flag
Data hor. Top. col.	CMC	64	Present only if bit 0x1000 is set in border color overrides flag
Data hor. ins. col.	CMC	65	Present only if bit 0x2000 is set in border color overrides flag
Data hor. bot. col.	CMC	66	Present only if bit 0x4000 is set in border color overrides flag
Data ver. left. col.	CMC	63	Present only if bit 0x8000 is set in border color overrides flag
Data ver. ins. col.	CMC	68	Present only if bit 0x10000 is set in border color overrides flag

```
       Data ver. rt. col.   CMC     69      Present only if bit 0x20000 is set in border color
                                            overrides flag
End If has border color overrides == 1
       Has border lineweight overridesB
If has border lineweight overrides == 1:
       Overrides flag        BL     95      Border LINEWEIGHT overrides
       Title hor. Top. lw.   BS             Present only if bit 0x01 is set in border color
                                            overrides flag
       Title hor. ins. lw.   BS             Present only if bit 0x02 is set in border color
                                            overrides flag
       Title hor. bot. lw.   BS             Present only if bit 0x04 is set in border color
                                            overrides flag
       Title ver. left. lw.  BS             Present only if bit 0x08 is set in border color
                                            overrides flag
       Title ver. ins. lw.   BS             Present only if bit 0x10 is set in border color
                                            overrides flag
       Title ver. rt. lw.    BS             Present only if bit 0x20 is set in border color
                                            overrides flag
       Header hor. Top. lw.  BS             Present only if bit 0x40 is set in border color
                                            overrides flag
       Header hor. ins. lw.  BS             Present only if bit 0x80 is set in border color
                                            overrides flag
       Header hor. bot. lw.  BS             Present only if bit 0x100 is set in border color
                                            overrides flag
       Header ver. left. lw. BS             Present only if bit 0x200 is set in border color
                                            overrides flag
       Header ver. ins. lw.  BS             Present only if bit 0x400 is set in border color
                                            overrides flag
       Header ver. rt. lw.   BS             Present only if bit 0x800 is set in border color
                                            overrides flag
       Data hor. Top. lw.    BS             Present only if bit 0x1000 is set in border color
                                            overrides flag
       Data hor. ins. lw.    BS             Present only if bit 0x2000 is set in border color
                                            overrides flag
       Data hor. bot. lw.    BS             Present only if bit 0x4000 is set in border color
                                            overrides flag
       Data ver. left. lw.   BS             Present only if bit 0x8000 is set in border color
                                            overrides flag
       Data ver. ins. lw.    BS             Present only if bit 0x10000 is set in border color
                                            overrides flag
       Data ver. rt. lw.     BS             Present only if bit 0x20000 is set in border color
                                            overrides flag
End If has border lineweight overrides == 1
       Has border visibility overridesB
If has border visibility overrides == 1:
       Overrides flag        BL     96      Border visibility overrides
       Title hor. Top. vsb.  BS             Present only if bit 0x01 is set in border visibility
                                            overrides flag (0 = visible, 1 = invisible)
```

```
        Title hor. ins. vsb.   BS          Present only if bit 0x02 is set in border visibility
                                           overrides flag (0 = visible, 1 = invisible)
        Title hor. bot. vsb.   BS          Present only if bit 0x04 is set in border visibility
                                           overrides flag (0 = visible, 1 = invisible)
        Title ver. left. vsb.  BS          Present only if bit 0x08 is set in border visibility
                                           overrides flag (0 = visible, 1 = invisible)
        Title ver. ins. vsb.   BS          Present only if bit 0x10 is set in border visibility
                                           overrides flag (0 = visible, 1 = invisible)
        Title ver. rt. vsb.    BS          Present only if bit 0x20 is set in border visibility
                                           overrides flag (0 = visible, 1 = invisible)
        Header hor. Top. vsb.  BS          Present only if bit 0x40 is set in border visibility
                                           overrides flag (0 = visible, 1 = invisible)
        Header hor. ins. vsb.  BS          Present only if bit 0x80 is set in border visibility
                                           overrides flag (0 = visible, 1 = invisible)
        Header hor. bot. vsb.  BS          Present only if bit 0x100 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Header ver. left. vsb. BS          Present only if bit 0x200 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Header ver. ins. vsb.  BS          Present only if bit 0x400 is set in border  (0 =
                                           visible, 1 = invisible)visibility overrides flag
        Header ver. rt. vsb.   BS          Present only if bit 0x800 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Data hor. Top. vsb.    BS          Present only if bit 0x1000 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Data hor. ins. vsb.    BS          Present only if bit 0x2000 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Data hor. bot. vsb.    BS          Present only if bit 0x4000 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Data ver. left. vsb.   BS          Present only if bit 0x8000 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Data ver. ins. vsb.    BS          Present only if bit 0x10000 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
        Data ver. rt. vsb.     BS          Present only if bit 0x20000 is set in border
                                           visibility overrides flag (0 = visible, 1 =
                                           invisible)
End If has border visibility overrides == 1


Common:
        Common Entity Handle Data
                               H      2    BLOCK HEADER (hard pointer)
```

```
R13-R200:
                              H            [1st ATTRIB (soft pointer)]  if 66 bit set; can be
                                           NULL
                              H            [last ATTRIB](soft pointer)] if 66 bit set; can be
                                           NULL
R2004:
                              H            [ATTRIB (soft pointer)] Repeats "Owned Object Count"
                                           times.
Common:
                              H            [SEQEND (hard owner)]      if 66 bit set
                              H     342    Table Style ID (hard pointer)
                              H  Varies    344 for text cell, 340 for block cell (hard pointer)
                              H     331    Attr. Def. ID (soft pointer, present only for block
                                           cells, when additional data flag == 1, and 1 entry
                                           per attr. def.)
                              H       7    Text style override (present only if bit 0x08 is set
                                           in cell flag override), one for each applicable cell
                              H       7    Title row style override (present only if bit
                                           0x20000 is set in table overrides flag
                              H       7    Title row style override (present only if bit
                                           0x40000 is set in table overrides flag
                              H       7    Title row style override (present only if bit
                                           0x80000 is set in table overrides flag


       CRC                    X     ---
```

20.4.96.2 *R24 and later*

In the R24 format the old table data structures were replaced with new data structures, of which the root is the AcDbTableContent class. The old data structures are still used in the DXF format. An R24 DXF file contains both the old and new structures, where the new structures are optionally used. If AutoCAD can store all data just using the old structures it does not always write the new structures in DXF. In an R24 DWG file, always the new structures are used. The table then points to a AcDbTableContent object, which contains most of the actual data. Note that AcDbTableContent was already introduced in AutoCAD 2008 (R21), but in R21 it was indirectly referenced through the tables extension dictionary entry ACAD_XREC_ROUNDTRIP (TODO: describe details on ACAD_ROUNDTRIP_2008_TABLE_ENTITY and for 2007).

Version	**Field type**	**DXF group code**	**Description**
	…		Common entity data.
R2010+			
	RC		Unknown (default 0)
	H		Unknown (soft pointer, default NULL)
	BL		Unknown (default 0)

R2010			
	B		Unknown (default true)
R2013			
	BL		Unknown (default 0)
R2010+			
	…		Here the table content is present (see TABLECONTENT object), without the common OBJECT data. See paragraph 20.4.97.
	BS		Unknown (default 38)
	3BD	11	Horizontal direction
	BL		Has break data flag (0 = no break data, 1 = has break data)
			Begin break data (optional)
	BL		Option flags: Enable breaks = 1, Repeat top labels = 2, Repeat bottom labels = 4, Allow manual positions = 8, Allow manual heights = 16

	BL		Flow direction:
			Right = 1,
			Vertical = 2,
			Left = 4

	BD		Break spacing
	BL		Unknown flags
	BL		Unknown flags
	BL		Number of manual positions (break heights)
			Begin repeat manual positions (break heights)
	3BD		Position
	BD		Height
	BL		Flags (meaning unknown)
			End repeat manual positions (break heights)
			End break data
	BL		Number of break row ranges (there is always at least 1)
			Begin repeat row ranges
	3BD		Position
	BL		Start row index
	BL		End row index
			End repeat row ranges

20.4.97 TABLECONTENT

This represents the table content (AcDbTableContent) that replaces the old table data structures that were introduced in AutoCAD 2005. Table content was introduced in AutoCAD 2008 and supports more advanced features like e.g. multiple contents per cell. In AutoCAD 2008 the table content was written as a separate object in DWG and referenced by roundtrip data in the table entity's extension dictionary. In DXF this is still the case even for R24. In a R24 DWG file, the table content is part of the table entity data and is no longer present as a separate object. Possibly for backwards compatibility with the AutoCAD 2007 (R21) format, this separate data container was created instead of extending the ACAD_TABLE entity.

The table content class inherits from 3 other classes, which never exist independently so they will all be described in this paragraph. AcDbTableContent inherits from AcDbFormattedTableData, which inherits from AcDbLinkedTableData, which inherits from AcDbLinkedData. Class AcDbLinkedTableData contains most of the data (rows, columns, cells, cell contents).

Version	**Field type**	**DXF group code**	**Description**
	…		Common object data.
			AcDbLinkedData fields
	TV	1	Name
	TV	300	Description
			AcDbLinkedTableData fields
	BL	90	Number of columns
			Begin repeat columns
	TV	300	Column name
	BL	91	32-bit integer containing custom data
	…		Custom data collection, see paragraph 20.4.100.
	…		Cell style data, see paragraph 20.4.101.4, this contains cell style overrides for the column.
	BL	90	Cell style ID, points to the cell style in the table's table style that is used as the base cell style for the column. 0 if not present.
	BD	40	Column width.
			End repeat columns
	BL	91	Number of rows.
			Begin repeat rows.
	BL	90	Number of cells in row.
			Begin repeat cells
	BL	90	Cell state flags: Content locked = 0x1, Content readonly = 0x2, Linked = 0x4, Content modified after update = 0x8, Format locked = 0x10, Format readonly = 0x20, Format modified after update = 0x40

	TV	300	Tooltip
	BL	91	32-bit integer containing custom data
	…		Custom data collection, see paragraph 20.4.100.
	BL	92	Has linked data flags, 0 = false, 1 = true
			If has linked data
	H	340	Handle to data link object (hard pointer).
	BL	93	Row count.
	BL	94	Column count.
	BL	96	Unknown.
			End if has linked data
	BL	95	Number of cell contents
			Begin repeat cell contents

	BL	90	Cell content type:
			Unknown = 0,
			Value = 0x1,
			Field = 0x2,
			Block = 0x4

			If cell content type is Value
	…		Write value (see paragraph 20.4.98)
			Else if cell content type is Field
	H	340	Handle to AcDbField object (hard pointer).
			Else if cell content type is Block
	H	340	Handle to block record (hard pointer).
			End if cell content type is Block
	BL	91	Number of attributes
			Begin repeat attributes
	H	330	Handle to attribute definition (ATTDEF), soft pointer.
	TV	301	Attribute value.
	BL	92	Index (starts at 1).
			End repeat attributes
	BS	170	Has content format overrides flag
			If has content format overrides flag is non-zero
	…		The content format overrides, see paragraph 20.4.101.3. By default the cell content uses the cell's cell style, this allows to override properties per content.
			End if has content format overrides flag is non-zero
			End repeat cell contents
	…		Cell style data, see paragraph 20.4.101.4, this contains cell style overrides for the cell.
	BL	90	Cell style ID, points to the cell style in the table's table style that is used as the base cell style for the cell. 0 if not present.
	BL	91	Unknown flag
			If unknown flag is non-zero
	BL	91	Unknown
	BD	40	Unknown

	BD	41	Unknown
	BL		Geometry data flags
	H		Unknown ()
			If geometry data flags is non-zero
	…		Cell content geometry, see paragraph 20.4.98.
			Enf if geometry data flags is non-zero
			End If unknown flag is non-zero
			End repeat cells
	BL	91	32-bit integer containing custom data
	…		Custom data collection, see paragraph 20.4.100.
	…		Cell style data, see paragraph 20.4.101.4, this contains cell style overrides for the row.
	BL	90	Cell style ID, points to the cell style in the table's table style that is used as the base cell style for the row. 0 if not present.
	BD	40	Row height.
			End repeat rows.
	BL	-	Number of cell contents that contain a field reference.
			Begin repeat field references
	H	-	Handle to field (AcDbField), hard owner.
			End repeat field references
			AcDbFormattedTableData fields
	…		The table's cell style override fields (see paragraph 20.4.101.4). The table's base cell style is the table style's overall cell style (present from R24 onwards).
	BL	90	Number of merged cell ranges
			Begin repeat merged cell ranges
	BL	91	Top row index
	BL	92	Left column index
	BL	93	Bottom row index
	BL	94	Right column index
			End repeat merged cell ranges
			AcDbTableContent fields
	H	340	Handle to table style (hard pointer).

20.4.98 Cell content geometry

The table below represents the cell content geometry (does not have to be written)

Version	**Field type**	**DXF group code**	**Description**
	3BD		Distance to top left
	3BD		Distance to center
	BD		Content width
	BD		Content height
	BD		Width

	BD		Height
	BL		Unknown flags

20.4.99 Value

This is a not an entity or object, but a common value that is always part of an entity or object. Since it appears in multiple entities/objects a separate paragraph is dedicated to it.

```
R2007+:
       Flags                 BL      93      Flags & 0x01 => type is kGeneral
Common:
       Data type             BL      90
       Varies by type:                       Not present in case bit 1 in Flags is set
         0 – Unknown         BL
         1 - Long            BL
         2 –Double           BD
         4 –String           TV
         8 –Date                             BL data size N, followed by N bytes (Int64 value)
         16 –Point                           BL data size, followed by 2RD
         32 –3D Point                        BL data size, followed by 3RD
         64 –Object Id        H              Read from appropriate place in handles section (soft
                                             pointer).
         128 –Buffer                         Unknown.
         256 –Result Buffer                  Unknown.
         512 –General                        General, BL containing the byte count followed by a
                                             byte array. (introduced in R2007, use Unknown before
                                             R2007).
R2007+:
       Unit type             BL      94      0 = no units, 1 = distance, 2 = angle, 4 = area, 8 =
                                             volume
       Format String         TV     300
       Value String          TV     302
```

20.4.100 Custom data collection

Table cells, columns and rows may have a collection of custom data items (key/value pairs) associated with them.

Version	Field type	DXF group code	Description
	BL	90	Number of custom data items
			Begin repeat custom data items
	TV	300	Item name
	…		Item value (variant), see paragraph 20.4.98.

			End repeat custom data items

20.4.101 TABLESTYLE

The table style object repesents the style for the table entity. Like the table entity, table style was introduced in AutoCAD 2005. In AutoCAD 2008 new cell style data was introduced, which was stored in a separate container object: CELLSTYLEMAP, see paragraph 20.4.102 for more details. The cellstyle map can contain custom cell styles, whereas the TABLESTYLE only contains the *Table* (R24), *_Title*, *_Header* and *_Data* cell style.

20.4.101.1 *TABLESTYLE format until R21*

```
       Common OBJECT data, see paragraph 20.1.

Common:

       Description           TV      3

       Flow direction        BS      70      0 = down, 1 = up

       Bit flags             BS      71      Meaning unknown.

       Horizontal cell margin BD     40

       Vertical cell margin  BD      41

       Suppress title         B     280

       Suppress header        B     281

Begin repeat 3 times (data, title and header row styles in this order)

       Text style ID          H       7      Hard pointer.

       Text height           BD     140

       Text alignment        BS     170      Top left = 1, top center = 2, top right = 3, middle
                                             left = 4, middle center = 5, middle right = 6,
                                             bottom left = 7, bottom center = 8, bottom right =
                                             9.

       Text color           CMC      62

       Fill color           CMC      63

       Background color enabled      B      283

  Begin repeat 6 times (borders: top, horizontal inside, bottom, left, vertical inside, right, in
                                             this order)

       Line weight           BS 274-279

       Visible                B 284-289      0 = invisible, 1 = visible

       Border color         CMC   64-69

  End repeat borders

R2007+

       Data type             BL      90      As defined in the ACAD_TABLE entity.

       Data unit type        BL      91      As defined in the ACAD_TABLE entity.


       Format string         TV       1

End repeat row styles
```

20.4.101.2 *R24 TABLESTYLE format*

Version	Field type	DXF group code	Description
	RC	-	Unknown
	TV	3	Description
	BL	-	Unknown
	BL	-	Unknown
	H	-	Unknown (hard owner)
	…		The cell style with name “Table”, see paragraph 20.4.101.4.
	BL	90	Cell style ID, 1 = title, 2 = header, 3 = data, 4 = table (new in R24). The cell style ID is used by cells, columns, rows to reference a cell style in the table’s table style. Custom cell style ID’s are numbered starting at 101.
	BL	91	Cell style class, 1= data, 2 = label. The default value is label.
	TV	300	Cell style name
	BL		The number of cell styles (should be 3), the non-custom cell styles are present only in the CELLSTYLEMAP.
			Begin repeat cell styles (for data, title, header in this order)
	…		The cell style fields, see paragraph 20.4.101.4.
	BL	-	Cell style ID, 1 = title, 2 = header, 3 = data, 4 = table (new in R24). The cell style ID is used by cells, columns, rows to reference a cell style in the table’s table style. Custom cell style ID’s are numbered starting at 101.
	BL	-	Cell style class, 1= data, 2 = label. The default value is label.
	TV	-	Cell style name
			End repeat cell styles

20.4.101.3 *Content format*

Content format data is present in the cell style map object, in the table entity and also the table content object.

Version	Field type	DXF group code	Description
	BL	90	Property override flags (is used for both content format and cell style): **Content format properties:** Data type = 0x1, Data format = 0x2, Rotation = 0x4, Block scale = 0x8, Alignment = 0x10, Content color = 0x20, Text style = 0x40, Text height = 0x80, Auto scale = 0x100,

			Cell style properties: Background color = 0x200, Margin left = 0x400, Margin top = 0x800, Margin right = 0x1000, Margin bottom = 0x2000, Content layout = 0x4000, Margin horizontal spacing = 0x20000, Margin vertical spacing = 0x40000, **Row/column properties:** Merge all = 0x8000 **Table properties:** Flow direction bottom to top = 0x10000
	BL	91	Property flags. Contains property bit values for property Auto Scale only (0x100).
	BL	92	Value data type, see also paragraph 20.4.98.
	BL	93	Value unit type, see also paragraph 20.4.98.
	TV	300	Value format string
	BD	40	Rotation
	BD	140	Block scale
	BL	94	Cell alignment: Top left = 1, Top center = 2, Top right = 3, Middle left = 4, Middle center = 5, Middle right = 6, Bottom left = 7, Bottom center = 8, Bottom right = 9
	TC	62	Content color
	H	340	Text style handle (hard pointer)
	BD	144	Text height

20.4.101.4 *Cell style*

Table cell style data is present in the cell style map object, in the table entity and also the table content object. A cell style inherits from content format. Cell style adds amongst others cell border style and margin properties to the content style properties of content format (see paragraph 20.4.101.3).

Version	Field type	DXF group code	Description

	BL	90	Cell style type:
			Cell = 1,
			Row = 2,
			Column = 3,
			Formatted table data = 4,
			Table = 5

	BS	170	Data flags, 0 = no data, 1 = data is present
			If data is present
	BL	91	Property override flags. The definition is the same as the content format propery override flags, see paragraph 20.4.101.3.
	BL	92	Merge flags, but may only for bits 0x8000 and 0x10000.
	TC	62	Background color
	BL	93	Content layout flags: Flow = 1, Stacked horizontal = 2, Stacked vertical = 4
	…		Content format fields (see paragraph 20.4.101.3).
	BS	171	Margin override flags, bit 1 is set if margin overrides are present
			If margin overrides are present
	BD	40	Vertical margin
	BD	40	Horizontal margin
	BD	40	Bottom margin
	BD	40	Right margin
	BD	40	Margin horizontal spacing
	BD	40	Margin vertical spacing
			End if margin overrides are present
	BL	94	Number of borders present (0-6)
			Begin repeat borders
	BL	95	Edge flags: 1 = top, 2 = right, 4 = bottom, 8 = left, 0x10 = inside vertical, 0x20 = inside horizontal
			If edge flags is non-zero
	BL	90	Border property override flags: Border types = 0x1, Line weight = 0x2, Line type = 0x4, Color = 0x8, Invisibility = 0x10, Double line spacing = 0x20

	BL	91	Border type:
			Single = 1,
			Double = 2

	TC	62	Color
	BL	92	Line weight
	H	340	Line type (hard pointer)
	BL	93	Invisibility: 1 = invisible, 0 = visible.
	BD	40	Double line spacing
			End if edge flags is non-zero
			End repeat borders
			End if data is present

20.4.102 CELLSTYLEMAP

The cell style map (AcDbCellStyleMap) is a helper class for TABLESTYLE containing all cell styles. This object was introduced in AutoCAD 2008, together with the new table related classes (like AcDbTableContent). Possibly for backwards compatibility with the AutoCAD 2007 (R21) format, this separate data container was created instead of extending the TABLESTYLE object.

The cell style map is connected to the table style through an extension dictionary entry with name "ACAD_ROUNDTRIP_2008_TABLESTYLE_CELLSTYLEMAP" in the table style's extension dictionary. The dictionary entry value points to the cell style map.

Version	Field type	DXF group code	Description
	…		Common AcDbObject fields, see paragraph 20.1.
	BL	90	Number of cell styles
			Begin repeat cell styles
	…		Cell style fields, see paragraph 20.4.101.4.
	BL	90	Cell style ID, 1 = title, 2 = header, 3 = data, 4 = table (new in R24). The cell style ID is used by cells, columns, rows to reference a cell style in the table's table style. Custom cell style ID's are numbered starting at 101.
	BL	91	Cell style class, 1= data, 2 = label. The default value is label.
	TV	300	Cell style name
			End repeat cell styles

20.4.103 TABLEGEOMETRY

This object represents a table's geometry and was introduced in AutoCAD 2008. It does not need to be present in a DWG file.

Version	Field	DXF	Description

	type	group code	
	…		Common AcDbObject fields, see paragraph 20.1.
	BL	90	Row count
	BL	91	Column count
	BL	92	Row * column count
			Begin repeat rows
			Begin repeat columns
	BL	93	Flags
	BD	40	Width with gap
	BD	41	Height with gap
	H	330	Handle to unknown (soft pointer)
	BL	94	Content count
			Begin repeat contents
	3BD	10,20,30	Distance to top left.
	3BD	11,21,31	Distance to center.
	BD	43	Content width.
	BD	44	Content height.
	BD	45	Width.
	BD	46	Height.
	BD	95	Unknown (0).
			End repeat contents
			End repeat columns
			End repeat rows

20.4.104 XRECORD (varies):

```
       Length               MS    ---     Entity length (not counting itself or CRC).
       Type                 BS      0     typecode (internal DWG type code).
R2000+:
       Obj size             RL            size of object in bits, not including end handles
Common:
       Handle                H      5     Length (char) followed by the handle bytes.
       EED                   X     -3     See EED section.
R13-R14 Only:
       Obj size             RL            size of object in bits, not including end handles
Common:
       Numreactors          BL            Number of persistent reactors attached to this obj
R2004+:
       XDic Missing Flag     B            If 1, no XDictionary handle is stored for this
                                          object, otherwise XDictionary handle is stored as in
                                          R2000 and earlier.
Common:
       Numdatabytes         BL            number of databytes
```

Databytes	X		databytes, however many there are to the handles
R2000+:			
Cloning flag	BS	280	
Common:			
XRECORD data is pairs of:			RS indicator number, then data. The indicator number indicates the DXF number of the data, then the data follows, so for instance an indicator of 1 would be followed by the string length (RC), the dwgcodepage (RC), and then the string, for R13-R2004 files. For R2007+, a string contains a short length N, and then N Unicode characters (2 bytes each). An indicator of 70 would mean a 2 byte short following. An indicator of 10 indicates 3 8-byte doubles following. An indicator of 40 means 1 8-byte double. These indicator numbers all follow the normal AutoCAD DXF convention for group codes.
Handle refs	H		parenthandle (soft pointer)
			[Reactors (soft pointer)]
			xdictionary (hard owner)
			objid object handles, as many as you can read until you run out of data

20.4.104.1 Example:

```
OBJECT: proxy (1F4H), len 65H (101), handle: 28

    00AC1 65 00                    e.        0110 0101 0000 0000

    00AC3 3D 00 40 4A 20 80 30 00  =.@J .0.  0011 1101 0000 0000 0100 0000 0100 1010 0010 0000 1000 0000 0011 0000 0000 0000

    00ACB 04 05 56 01 00 1B 00 0C  ..V.....  0000 0100 0000 0101 0101 0110 0000 0001 0000 0000 0001 1011 0000 0000 0000 1100

    00AD3 54 68 69 73 20 69 73 20  This is   0101 0100 0110 1000 0110 1001 0111 0011 0010 0000 0110 1001 0111 0011 0010 0000

    00ADB 61 20 74 65 73 74 20 78  a test x  0110 0001 0010 0000 0111 0100 0110 0101 0111 0011 0111 0100 0010 0000 0111 1000

    00AE3 72 65 63 6F 72 64 20 6C  record l  0111 0010 0110 0101 0110 0011 0110 1111 0111 0010 0110 0100 0010 0000 0110 1100

    00AEB 69 73 74 0A 00 00 00 00  ist.....  0110 1001 0111 0011 0111 0100 0000 1010 0000 0000 0000 0000 0000 0000 0000 0000

    00AF3 00 00 00 F0 3F 00 00 00  ....?...  0000 0000 0000 0000 0000 0000 1111 0000 0011 1111 0000 0000 0000 0000 0000 0000

    00AFB 00 00 00 00 40 00 00 00  ....@...  0000 0000 0000 0000 0000 0000 0000 0000 0100 0000 0000 0000 0000 0000 0000 0000

    00B03 00 00 00 00 00 28 00 6F  .....(.o  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010 1000 0000 0000 0110 1111

    00B0B 86 1B F0 F9 21 09 40 32  ....!.@2  1000 0110 0001 1011 1111 0000 1111 1001 0010 0001 0000 1001 0100 0000 0011 0010

    00B13 00 D7 35 33 F0 F9 21 09  ..53..!.  0000 0000 1101 0111 0011 0101 0011 0011 1111 0000 1111 1001 0010 0001 0000 1001

    00B1B 40 3E 00 01 00 46 00 B4  @>...F..  0100 0000 0011 1110 0000 0000 0000 0001 0000 0000 0100 0110 0000 0000 1011 0100

    00B23 00 40 41 0C 30           .@A.0     0000 0000 0100 0000 0100 0001 0000 1100 0011 0000

    00B28 45 76                    crc
```

21 Data section AcDb:ObjFreeSpace

The meaning of this section is not completely known. The ODA knows how to write a valid section, but the meaning is not known of every field.

21.1 Until R18

Type	Length	Description
Int32	4	0
UInt32	4	Approximate number of objects in the drawing (number of handles).
Julian datetime	8	If version > R14 then system variable TDUPDATE otherwise TDUUPDATE.
UInt32	4	Offset of the objects section in the stream.
UInt8	1	Number of 64-bit values that follow (ODA writes 4).
UInt32	4	ODA writes 0x00000032.
UInt32	4	ODA writes 0x00000000.
UInt32	4	ODA writes 0x00000064.
UInt32	4	ODA writes 0x00000000.
UInt32	4	ODA writes 0x00000200.
UInt32	4	ODA writes 0x00000000.
UInt32	4	ODA writes 0xffffffff.
UInt32	4	ODA writes 0x00000000.

22 Data section: AcDb:Template

This section is optional in releases 13-15. The section is mandatory in the releases 18 and newer. The template section only contains the MEASUREMENT system variable.

Type	Length	Description
Int16	2	Template description string length in bytes (the ODA always writes 0 here).
byte[]	n	Encoded string bytes of the template description (use the drawing's codepage to encode the bytes).
UInt16	2	MEASUREMENT system variable (0 = English, 1 = Metric).

23 Data section AcDb:Handles (OBJECT MAP)

23.1 R13-15

The Object Map is a table which gives the location of each object in the file This table is broken into sections. It is basically a list of handle/file loc pairs, and goes (something like) this:

```
Set the "last handle" to all 0 and the "last loc" to 0L;
Repeat until section size==2 (the last empty (except the CRC) section):
  Short: size of this section.  Note this is in BIGENDIAN order (MSB
         first)
  Repeat until out of data for this section:
    offset of this handle from last handle as modular char.
    offset of location in file from last loc as modular char.  (note
    that location offsets can be negative, if the terminating byte
    has the 4 bit set).
  End repeat.
  CRC (most significant byte followed by least significant byte)
  End of section
End top repeat
```

Note that each section is cut off at a maximum length of 2032.

23.2 R18

This section is compressed and contains the standard 32 byte section header. The decompressed data in this section is identical to the “Object Map” section data found in R15 and earlier files, excepts that offsets are not absolute file addresses, but are instead offsets into the AcDb:Objects logical section (starting with offset 0 at the beginning of this logical section).

24 Section AcDb:AcDsPrototype_1b (DataStorage)

At this moment (December 2012), this sections contains information about Acis data (regions, solids). The data is stored in a byte stream, not a bit stream like e.g. the objects section.

The data store contains several data segments, and index segments that contain lookup information for finding the data segments and objects within these data segments. The file header contains the stream position of the segment index file segment and the segment indexes for the schema index/data index/search file segments. The segment index file segment is a lookup table for finding the stream position of a file segment by its segment index.

In paragraph 24.3 the default contents of this section is shown when empty.

24.1 File header

Version	Field type	DXF group code	Description
	UInt32		File signature
	Int32		File header size
	Int32		Unknown 1 (always 2?)
	Int32		Version (always 2?)
	Int32		Unknown 2 (always 0?)
	Int32		Data storage revision
	Int32		Segment index offset (the stream off set from the data store's stream start position). See paragraph 24.2.2.1 for the segment index file segment.
	Int32		Segment index unknown
	Int32		Segment index entry count
	Int32		Schema index segment index. This is the index into the segment index entry array (see paragraph 24.2.2.1) for the schema index file segment (see paragraph 24.2.2.4).
	Int32		Data index segment index. This is the index into the segment index entry array (see paragraph 24.2.2.1) for the data index file segment (see paragraph 24.2.2.2).
	Int32		Search segment index
	Int32		Previous save index
	Int32		File size

24.2 File segment

A file segment is a segment containing data. There are several types of file segments containing different types of data.

At the beginning of each segment there is a header. Then there are a number of data bytes specific for the type of file segment. At the end there are padding bytes so the total segment size including the header is a multiple of 0x40 bytes (AutoCAD writes a multiple of 0x80 bytes). The padding consists of an array of 0x70 values.

24.2.1 Header

Version	Field type	DXF group code	Description
	Int16		Signature (always 0xd5ac?)
	byte[6]		Name (6 bytes). Names for the several file segments are: • Segment index: “segidx” • Data index: “datidx” • Data: “_data_” • Schema index: “schidx” • Schema data: “schdat” • Search: “search” • Blob01: “blob01”
	Int32		Segment index
	Int32		Unknown 1 (0 or 1? 1 in blob01 segment).
	Int32		Segment size (multiple of 0x40 bytes (AutoCAD uses 0x80), padded with 0x70 values).
	Int32		Unknown 2 (always 0?)
	Int32		Data storage revision
	Int32		Unknown 3 (always 0?)
	Int32		System data alignment offset (calculate the stream position by shifting left 4 bits and adding to the file segment’s stream start position). This offset is used for schema index and schema data segments. So the name “system data” seems to refer to schema index/schema data.
	Int32		Object data alignment offset (calculate the stream position by shifting left 4 bits and adding to the file segment’s stream start position). This offset is used for the data segment.
	byte[8]		8 alignment bytes (always 8 x 0x55?).

24.2.2 Sub types

In the following sub paragraphs all file segment sub types are described. Their data directly follows upon the file segment header.

24.2.2.1 *Segment index file segment*

This file segment contains information about each file segment’s offset and size in the stream. The segment is looked up by the index in the array.

Version	Field type	DXF group	Description

		code	
			Begin repeat segment index entry count (as present in the file header, see paragraph 24.1)
	UInt64		Offset. This is the offset from the data store's stream start position.
	UInt32		Size
			End repeat segment index entry count

24.2.2.2 *Data index file segment*

This file segment contains index entries for objects within the data file segment (see paragraph 24.2.2.3).

Version	**Field type**	**DXF group code**	**Description**
	Int32		Entry count
	Int32		Unknown (always 0?)
			Begin repeat of entries (entry count)
	UInt32		Segment index (0 means stub entry and can be ignored).
	UInt32		Local offset. This is a local offset in the stream, relative to the file segment's stream start position. This points to a data file segment, see paragraph 24.2.2.3.
	UInt32		Schema index
			End repeat of entries

24.2.2.3 *Data file segment*

The data file segment basically contains a byte array, of which the storage type depends on the size:

- data records, where each data record is a byte array. Relatively small amounts of data are stored directly in the data file segment (up to 0x40000 bytes).

- A data blob references, where each blob reference references one or more other blob file segments. These other file segments represent the pages of the blob (paragraph 24.2.2.3.1). Large byte arrays are stored into multiple of these pages (more than 0x40000 bytes, max 0xfffb0 bytes per page).

For each entity's binary data stored in the data file segment entries have to be created in the schema search data. See paragraph 24.2.2.7.1. When reading the schema search data can be ignored.

For each ACIS entity (REGION, 3DSOLID), a data record is created with the SAB stream of the object. More detailed description of the ACIS/SAB data falls outside the scope of this document. The SAB stream bytes are prefixed with the ASCII encoded bytes of the string "ACIS BinaryFile". When for an ACIS entity a SAB stream is created from SAT, then if the version >= 21800, the bytes are post fixed with the ASCII encoded bytes of the string "End-of-ASM-data", otherwise "End-of-ACIS-data".

Version	Field type	DXF group code	Description
			Begin repeat (data record) headers. Repeats number of local offsets times (this is read earlier from the data index, see paragraph 24.2.2.2). For a particular data file segment, find all data index entries with the segment's segment index and take the local offsets.
			Move the stream position according to the current header local offset, which is relative to this data file segments stream start position.
	UInt32		Entry size
	UInt32		Unknown (ODA writes 1)
	UInt64		Handle
	UInt32		Local offset, a stream offset relative to the data start marker (just after this list of data record headers).
			End repeat (data record) header offsets
			Data start marker, this is the beginning of all data records.
			Begin repeat header entries (that were read above)
			Each data record starts at the data start marker position + local offset. The maxRecordSize of the record is the difference between two consecutive stream offsets. For the last data record the size is the file segment header's (object data alignment offset << 4) + segment size - the record's stream offset (i.e. the file segment end position – the record start position).
	UInt32		dataSize
			If ((dataSize + 4) <= maxRecordSize)
	Byte[]		Data record's bytes of length dataSize
			Else If (dataSize == 0xbb106bb1)
			Data blob reference record, see paragraph 24.2.2.3.1
			End If
			End repeat header entries

24.2.2.3.1 *Data blob reference record*

A data blob reference references one or more other file segments. These other file segments represent the pages of the blob. Each page is stored in a Blob01 file segment, see paragraph 24.2.2.4.

Version	Field type	DXF group code	Description
	UInt64		Total data size
	UInt32		Page count
	UInt32		Record size (the size of this data blob reference record
	UInt32		Page size
	UInt32		Last page size
	UInt32		Unknown 1 (ODA writes 0)

	UInt32		Unknown 2 (ODA writes 0)
			Begin repeat page count
	UInt32		Segment index. The page's blob01 file segment stream position can be found by a lookup in the segment index file segment using the segment index, see paragraph 24.2.2.1.
	UInt32		Size
			End repeat page count

24.2.2.4 ***Blob01 file segment***

Version	Field type	DXF group code	Description
	UInt64		Total data size
	UInt64		Page start offset
	Int32		Page index
	Int32		Page count
	UInt64		Page data size
	byte[]		Binary data (byte array) of size Page data size

24.2.2.5 Schema index file segment

The schema index contains references to objects within the schema data file segment, see paragraph 24.2.2.6.

Version	Field type	DXF group code	Description
	UInt32		Unknown property count
	UInt32		Unknown (0)
			Begin repeat schema unknown property count
	UInt32		Index (starting at 0)
	UInt32		Segment index into the segment index file segment entry table (paragraph 24.2.2.1) of the schema data file segment (paragraph 24.2.2.6)
	UInt32		Local offset of the unknown schema property. This is a local offset in the stream, relative to the schema data file segment's stream start position.
			End repeat schema unknown property count
	Int64		Unknown (0x0af10c)
	UInt32		Property entry count
	UInt32		Unknown (0)
			Begin repeat property entry count
	UInt32		Segment index into the segment index file segment entry table (paragraph 24.2.2.1) of the schema data file segment (paragraph 24.2.2.6).
	UInt32		Local offset of the schema property. This is a local offset in the stream, relative to the schema data file segment's stream start position.
	UInt32		Index
			End repeat property entry count

24.2.2.6 *Schema data file segment*

The schema data file segment contains unknown properties and schemas. The stream offsets of these objects from the start of this file segment are found in the schema index, see paragraph 24.2.2.5.

Version	Field type	DXF group code	Description
			Begin repeat schema unknown properties in the associated schema index file segment (paragraph 24.2.2.4), where the property's segment index is equal to this file segment's segment index (found in the header).
	UInt32		Data size
	UInt32		Unknown flags
			End repeat schema unknown properties
			Begin repeat schema entries in the associated schema index file segment (paragraph 24.2.2.4), where the property's segment index is equal to this file segment's segment index (found in the header).
			A schema, see paragraph 24.2.2.6.1. The stream position is the file segment's start position + the schema entry's local offset.
			End repeat schema entries
	Uint32		Property name count
			Begin repeat property name count
	AnsiString		Property name (zero byte delimited). These names are referred to by the schema's schema property's name index (paragraph 24.2.2.6.1.1). Name strings can be shared between multiple schema properties this way. See paragraph 24.2.2.6.1 for details about the schema.
			End repeat property name count

24.2.2.6.1 *Schema*

A schema is a collection of name value pairs, where the value can have a number of types.

Version	Field type	DXF group code	Description
	UInt16		Index count
			Begin repeat index count
	UInt64		Index
			End repeat index count
	UInt16		Property count
			Begin repeat property count
			Schema property, see paragraph 24.2.2.6.1.1.
			End repeat property count

24.2.2.6.1.1 **Schema property**

This is a schema (see 24.2.2.6.1) property, having a name and a value of a certain type.

Version	Field type	DXF group code	Description
	UInt32	91	Property flags: • 1 = Unknown 1 (if set then all other bits are cleared). • 2 = Has no type. • 8 = Unknown 2 (if set then all other bits are cleared).
	UInt32	2	Name index. Index into a property names array in the schema data file segment (see paragraph 24.2.2.6). In a DXF file the name is directly written instead of indirectly through a table lookup.
			If property flags bit 2 is NOT set
	UInt32	280	Type (0-15)
			If type == 0xe
	UInt32		Custom type size
			Else
			The typeSize is looked up in the following array with the type being the index: 0, 0, 2, 1, 2, 4, 8, 1, 2, 4, 8, 4, 8, 0, 0, 0
			End if type == 0xe
			End If property flags bit 2 is NOT set
			If property flags == 1
	UInt32		Unknown1
			Else if property flags == 8
	UInt32		Unknown2
			End if property flags == 8
	UInt16		Property value count
			If (typeSize != 0)
	Byte[]		Property value, represented by a byte array of size typeSize.
			End if (typeSize != 0)

24.2.2.7 *Search file segment*

Version	Field type	DXF group code	Description
	UInt32		Schema count
			Begin repeat schema count
			Schema search data, see paragraph 24.2.2.7.1.
			End repeat schema count

24.2.2.7.1 *Schema search data*

The purpose of this segment is unknown. It seems to contain redundant data coupling a (sort) index to the objects in the data segment. When reading the schema search data can be ignored.

For each object stored in the data segment there has to be one item in the sorted index table (just start numbering at 0 and increase by one for every next object). Also in the 2D index array one array has to be present containing one entry for every object stored in the data segment. The object handle is stored, together with the index that was also used in the sorted index table. When the schema search data is not created, AutoCAD will ignore the entity.

Version	Field type	DXF group code	Description
	UInt32		Schema name index
	UInt64		Sorted index count
			Begin sorted index count
	UInt64		Sorted index
			End sorted index count
			For clarity: below a 2D jagged array is described of an ID entry object, containing itself a handle and an array of indexes.
	UInt32		ID indexes count (for the set of ID indexes).
			If ID indexes count > 0
	UInt32		Unknown (0)
			Begin repeat ID indexes count
	UInt32		ID index count
			Begin repeat ID index count (in this loop the ID entry object is serialized)
	UInt64		Handle of the object present in the data segment (see paragraph 24.2.2.3).
	UInt64		Index count
			Begin repeat index count
	UInt64		Index (same as Sorted index value above). The ODA only writes one index per handle.
			End repeat index count
			End repeat ID index count
			End repeat ID indexes count
			End If ID indexes count > 0

24.3 Default contents

Below is a dump of the default contents (i.e. when there is no data in the Data Storage section).

```
schemas {
  item {
    index: 0
    name: "AcDb3DSolid_ASM_Data"
    indexes {
      item: 0,
      item: 1,
    }
    propertyDescriptors {
    }
    properties {
      item {
        flags: 0
        nameIndex: 4294967295
        type: 10
        customTypeSize: 0
        typeSize: 8
        unknown1: 0
        unknown2: 0
        propertyValues {
          item: {02, 00, 00, 00, 00, 00, 00, 00},
          item: {03, 00, 00, 00, 00, 00, 00, 00},
        }
        name: "AcDbDs::ID"
      },
      item {
        flags: 0
        nameIndex: 4294967295
        type: 15
        customTypeSize: 0
        typeSize: 0
        unknown1: 0
        unknown2: 0
        propertyValues {
        }
        name: "ASM_Data"
      },
    }
  },
  item {
    index: 1
    name: "AcDbDs::TreatedAsObjectDataSchema"
    indexes {
    }
    propertyDescriptors {
    }
    properties {
      item {
        flags: 0
        nameIndex: 4294967295
        type: 1
        customTypeSize: 0
        typeSize: 0
        unknown1: 0
        unknown2: 0
        propertyValues {
        }
        name: "AcDbDs::TreatedAsObjectData"
      },
    }
  },
  item {
    index: 2
    name: "AcDbDs::LegacySchema"
    indexes {
    }
    propertyDescriptors {
    }
    properties {
      item {
        flags: 0
        nameIndex: 4294967295
        type: 1
        customTypeSize: 0
        typeSize: 0
        unknown1: 0
        unknown2: 0
        propertyValues {
        }
        name: "AcDbDs::Legacy"
      },
    }
  },
  item {
    index: 3
    name: "AcDbDs::IndexedPropertySchema"
    indexes {
    }
    propertyDescriptors {
    }
    properties {
      item {
        flags: 0
        nameIndex: 4294967295
        type: 1
        customTypeSize: 0
        typeSize: 0
        unknown1: 0
        unknown2: 0
        propertyValues {
        }
        name: "AcDs:Indexable"
      },
    }
  },
  item {
    index: 4
    name: "AcDbDs::HandleAttributeSchema"
    indexes {
    }
    propertyDescriptors {
    }
    properties {
      item {
        flags: 8
        nameIndex: 4294967295
        type: 7
        customTypeSize: 0
        typeSize: 1
        unknown1: 0
        unknown2: 1
        propertyValues {
          item: {00},
        }
        name: "AcDbDs::HandleAttribute"
      },
    }
  },
}
schemaUnknownProperties {
  item {
    dataSize: 8
    unknownFlags: 1
  },
  item {
    dataSize: 8
    unknownFlags: 1
  },
  item {
    dataSize: 8
    unknownFlags: 1
  },
  item {
    dataSize: 8
    unknownFlags: 0
  },
}
schemaSearchDataList {
  item {
    schemaNameIndex: 0
    sortedIndexes {
    }
    idIndexesSet {
    item[] {
    }
    }
  },
}
handleToDataRecord {
}
```

25 UNKNOWN SECTION

This section is largely unknown. The total size of this section is 53. We simply patch in "known to be valid" data. We first write a 0L, then the number of entries in the objmap +3, as a long. Then 45 bytes of "known to be valid data". Then we poke in the start address for objects at offset 16.

The 45 bytes of known to be valid data are:

```
0xA7,0x62,0x25,0x00,0xF6,0xAF,0x25,0x02,
0x3B,0x04,0x00,0x00,0x04,0x32,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x64,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x02,0x00,
0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,
0xFF,0x00,0x00,0x00,0x00
```

26 SECOND FILE HEADER (R13-R15)

26.1 Beginning sentinel

```
{0xD4,0x7B,0x21,0xCE,0x28,0x93,0x9F,0xBF,0x53,0x24,0x40,0x09,0x12,0x3C,0xAA,0x01 };
```

```
RL : size of this section
 L : Location of this header (long, loc of start of sentinel).
RC : "AC1012" or "AC1014" for R13 or R14 respectively
RC : 6 0's
 B : 4 bits of 0
RC : 0x18,0x78,0x01,0x04 for R13, 0x18,0x78,0x01,0x05 for R14

RC : 0
 L : header address
 L : header size
RC : 1
 L : class address
 L : class data size
RC : 2
 L : Object map address (natural table)
 L : Object map size
RC : 3
 L : Address of unknown section 3
 L : size of that section

 S : 14 (# of handle records following)

RC : size of (valid chars in) handseed
RC : 0
RC : "size" characters of the handle

RC : size of (valid chars in) block control objhandle
RC : 1
RC : "size" characters of the handle

RC : size of (valid chars in) layer control objhandle
RC : 2
RC : "size" characters of the handle
RC : size of (valid chars in) shapefile control objhandle
RC : 3
RC : "size" characters of the handle

RC : size of (valid chars in) linetype control objhandle
RC : 4
RC : "size" characters of the handle

RC : size of (valid chars in) view control objhandle
RC : 5
RC : "size" characters of the handle

RC : size of (valid chars in) ucs control objhandle
RC : 6
RC : "size" characters of the handle

RC : size of (valid chars in) vport control objhandle
RC : 7
RC : "size" characters of the handle

RC : size of (valid chars in) reg app control objhandle
RC : 8
RC : "size" characters of the handle

RC : size of (valid chars in) dimstyle control objhandle
RC : 9
RC : "size" characters of the handle

RC : size of (valid chars in) viewport entity header objhandle
RC : 10
RC : "size" characters of the handle

RC : size of (valid chars in) dictionary objhandle
RC : 11
RC : "size" characters of the handle

RC : size of (valid chars in) default multi-line style objhandle
RC : 12
RC : "size" characters of the handle
```

```
  RC : size of (valid chars in) group dictionary objhandle
  RC : 13
  RC : "size" characters of the handle


 CRC


  RC : 8 bytes of junk (R14 only).  Note that the junk is counted in the size of this
       section at the start.
```

```
Ending sentinel
{0x2B,0x84,0xDE,0x31,0xD7,0x6C,0x60,0x40,0xAC,0xDB,0xBF,0xF6,0xED,0xC3,0x55,0xFE}
```

27 Data section: AcDb:AuxHeader (Auxiliary file header)

The auxiliary file header contains mostly redundant information and was introduced in R15.

```
RC  :  0xff 0x77 0x01

RS  :  DWG version:
       AC1010 = 17,
       AC1011 = 18,
       AC1012 = 19,
       AC1013 = 20,
       AC1014 = 21,
       AC1015 (beta) = 22,
       AC1015 = 23,
       AC1018 (beta) = 24,
       AC1018 = 25,
       AC1021 (beta) = 26,
       AC1021 = 27,
       AC1024 (beta) = 28,
       AC1024 = 29
       AC1027 (beta) = 30,
       AC1027 = 31,
       AC1032 (beta) = 32,
       AC1032 = 33

RS  :  Maintenance version

RL  :  Number of saves (starts at 1)

RL  :  -1

RS  :  Number of saves part 1 ( = Number of saves – number of saves part 2)

RS  :  Number of saves part 2 ( = Number of saves – 0x7fff if Number of saves > 0x7fff,
       otherwise 0)

RL  :  0

RS  :  DWG version string

RS  :  Maintenance version

RS  :  DWG version string

RS  :  Maintenance version

RS  :  0x0005

RS  :  0x0893

RS  :  0x0005

RS  :  0x0893

RS  :  0x0000

RS  :  0x0001

RL  :  0x0000

RL  :  0x0000

RL  :  0x0000
```

```
          RL : 0x0000
          RL : 0x0000
          TD : TDCREATE (creation datetime)
          TD : TDUPDATE (update datetime)
          RL : HANDSEED (Handle seed) if < 0x7fffffff, otherwise -1.
          RL : Educational plot stamp (default value is 0)
          RS : 0
          RS : Number of saves part 1 – number of saves part 2
          RL : 0
          RL : 0
          RL : 0
          RL : Number of saves
          RL : 0
          RL : 0
          RL : 0
          RL : 0
R2018+
          RS : 0
          RS : 0
          RS : 0
```

28 Extended Entity Data (Extended Object Data)

EED directly follows the entity handle.

Each application's data is structured as follows:

```
    |Length|Application handle|Data items|
```

Length is a bitshort indicating the length of the data for an app, not including itself, the bit-pair, or the app table handle. The above format repeats until a length of zero is found.

The application handle is a standard table handle reference: 0101|4-bit length|handle bytes|

Each data item has a 1-byte code (DXF group code minus 1000) followed by the value. It looks like there's no bit-pair coding within the data; that would throw off the length value (it would need to count bits, too). The form of the value is listed below for each type:

```
    0 (1000)   String.

               R13-R2004:  1st byte of value is the length N; this is followed by a 2-byte short
        indicating the codepage, followed by N single-byte characters.

               R2007+: 2-byte length N, followed by N Unicode characters (2 bytes each).

    1 (1001)   This one seems to be invalid; can't even use as a string inside braces.  This
would be a registered application that this data relates to, but we've already had that above, so
it would be redundant or irrelevant here.

    2 (1002)   A '{' or '}'; 1 byte; ASCII 0 means '{', ASCII 1 means '}'

    3 (1003)   A layer table reference.  The value is the handle of the layer; it's 8 bytes --
even if the leading ones are 0.  It's not a string; read it as hex, as usual for handles.
(There's no length specifier this time.) Even layer 0 is referred to by handle here.

    4 (1004)   Binary chunk.  The first byte of the value is a char giving the length; the bytes
follow.

    5 (1005)   An entity handle reference.  The value is given as 8 bytes -- even if the leading
ones are 0. It's not a string; read it as hex, as usual for handles.  (There's no length
specifier this time.)

    10 - 13 (1010 - 1013)
               Points; 24 bytes (XYZ) -- 3 doubles

    40 - 42 (1040 - 1042)
               Reals; 8 bytes (double)

    70 (1070)  A short int; 2 bytes
    71 (1071)  A long  int; 4 bytes
```

29 PROXY ENTITY GRAPHICS

Proxy entities (zombies prior to R14) can have associated graphics data. The presence or absence of this data is indicated by the single bit which we call the "graphic present flag", which mostly occurs on entity-type proxies, and very few other entities. Entity type proxies are proxies where the related class's **itemclassid** field is equal to 0x1F2.

If that bit is 1, then following it, and preceding the RL which indicates the number of bits in the object, is an RL which indicates the number of bytes of proxy entity graphic data to follow.

Graphics data is padded to 4 byte boundaries! So, for instance, strings which are too short are padded out to the next 4 byte boundary. Similarly for lists of shorts.

In addition to the data definitions from chapter 2 there are a few additional data types:

PS : Padded string. This is a string, terminated with a zero byte. The file's text encoding (code page) is used to encode/decode the bytes into a string.

PUS : Padded Unicode string. The bytes are encoded using Unicode encoding. The bytes consist of byte pairs and the string is terminated by 2 zero bytes.

We use the following defines to discriminate sub-item presence:

```
#define adHasPrimTraits(a)         (a & 0xFFFFL)
#define adPrimsHaveColors(a)       (a & 0x0001L)
#define adPrimsHaveLayers(a)       (a & 0x0002L)
#define adPrimsHaveLinetypes(a)    (a & 0x0004L)
#define adPrimsHaveMarkers(a)      (a & 0x0020L)
#define adPrimsHaveVisibilities(a) (a & 0x0040L)
#define adPrimsHaveNormals(a)      (a & 0x0080L)
#define adPrimsHaveOrientation(a)  (a & 0x0400L)
```

The graphics data comes in chunks with the following format:

```
         RL  :  size
         RL  :  type
type-specific data


valid types are:
```

Extents 1

```
       3 RD  :  extext min
       3 RD  :  extent max
```

CIRCLE 2

```
       3 RD  :  center of circle
```

```
  RD  :  radius
3 RD  :  normal
```

CIRCLE3PT 3 (3 point circle)

```
3 RD  :  first point
3 RD  :  second point
3 RD  :  third point
```

CIRCULARARC 4

```
3 RD  :  center
  RD  :  radius
3 RD  :  normal
3 RD  :  start vector direction
  RD  :  sweep angle
  RL  :  arc type
```

CIRCULARARC3PT 5

```
3 RD  :  first point
3 RD  :  second point
3 RD  :  third point
  RL  :  arc type
```

POLYLINE 6

```
  RL  :  number of points
3 RD  :  a point (repeat "number of points" times)
```

POLYGON 7

```
  RL  :  number of points
3 RD  :  a point (repeat "number of points" times)
```

MESH 8

```
RL:number of rows
RL:number of columns
Repeat "rows" times:
  Repeat "cols" times
    3 RD: vertex
  Endrep
Endrep

RL:edge primitive flags
```

```
if (adHasPrimTraits(edgeprimflag)) {
  compute nummeshedges as (rows-1)*cols + (cols-1) * rows
  if (adPrimsHaveColors(edgeprimflag) {
    RL: color for each edge
  }
  if (adPrimsHaveLayers(edgeprimflag)) {
    RL: layer ids, 1 for each edge
  }
  if (adPrimsHaveLinetypes(edgeprimflag)) {
    RL: linetype ids, 1 for each edge
  }
  if (adPrimsHaveMarkers(edgeprimflag)) {
    RL: marker indices, 1 for each edge
  }
  if (adPrimsHaveVisibilities(edgeprimflag)) {
    RL: visibility indicator, 1 for each edge
  }
}

RL: face primitive flags
if (adHasPrimTraits(faceprimflag)) {
  compute nummeshfaces as (rows-1)*(cols-1)
  if (adPrimsHaveColors(faceprimflag) {
    RL: color for each face
  }
  if (adPrimsHaveLayers(faceprimflag)) {
    RL: layer ids, 1 for each face
  }
  if (adPrimsHaveMarkers(faceprimflag)) {
    RL: marker indices, 1 for each face
  }
  if (adPrimsHaveNormals(faceprimflag)) {
    3 RD: normal, 1 for each face
  }
  if (adPrimsHaveVisibilities(faceprimflag)) {
    RL: visibility indicator, 1 for each face
  }
}

RL: vertex primitive flags
```

```
if (adHasPrimTraits(vertprimflag)) {
  compute numvertices as rows * cols
  if (adPrimsHaveNormals(vertprimflag)) {
    3 RD: normal, 1 for each vertex
  }
  if (adPrimsHaveOrientation(vertprimflag)) {
    RL: orientation indicator, 1 ONLY
  }
}
```

SHELL 9

```
         RL  :  number of points
       3 RD  :  vertex, 1 set of 3 for each vertex
         RL  :  number of face entries
         RL  :  face entries, "number of face entries" of these indicates a face for the shell.
                negative entry indicates the number of entries to follow.  then follow the
                entries, which indicate the vertices, read above, that make up that face.  So for
                instance entries

                -3,2,3,4 would mean a 3 sided face of vertices 2,3 and 4.

                We scan this list and get the number of faces and edges.
```

```
RL: edge primitive flags
if (adHasPrimTraits(edgeprimflag)) {
  if (adPrimsHaveColors(edgeprimflag) {
    RL: color for each edge
  }
  if (adPrimsHaveLayers(edgeprimflag)) {
    RL: layer ids, 1 for each edge
  }
  if (adPrimsHaveLinetypes(edgeprimflag)) {
    RL: linetype ids, 1 for each edge
  }
  if (adPrimsHaveMarkers(edgeprimflag)) {
    RL: marker indices, 1 for each edge
  }
  if (adPrimsHaveVisibilities(edgeprimflag)) {
    RL: visibility indicator, 1 for each edge
  }
}

RL: face primitive flags
if (adHasPrimTraits(faceprimflag)) {
  if (adPrimsHaveColors(faceprimflag) {
    RL: color for each face
  }
  if (adPrimsHaveLayers(faceprimflag)) {
    RL: layer ids, 1 for each face
  }
  if (adPrimsHaveMarkers(faceprimflag)) {
    RL: marker indices, 1 for each face
  }
  if (adPrimsHaveNormals(faceprimflag)) {
    3 RD: normal, 1 for each face
  }
  if (adPrimsHaveVisibilities(faceprimflag)) {
    RL: visibility indicator, 1 for each face
  }
}

RL: vertex primitive flags
if (adHasPrimTraits(vertprimflag)) {
  compute numvertices as rows * cols
  if (adPrimsHaveNormals(vertprimflag)) {
    3 RD: normal, 1 for each vertex
  }
  if (adPrimsHaveOrientation(vertprimflag)) {
    RL: orientation indicator, 1 ONLY
  }
}
```

TEXT 10

```
3 RD  :  start point
3 RD  :  normal
3 RD  :  text direction
  RD  :  height
  RD  :  widthfactor
  RD  :  oblique angle
  PS  :  string, zero terminated and padded to 4 byte boundary
```

TEXT2 11

```
3 RD  :  start point
3 RD  :  normal
3 RD  :  text direction
  PS  :  string, padded to 4 byte boundary
  RL  :  length of string, -1 if zero terminated
  RL  :  "raw"; 0 if raw, 1 if not.  raw means don't interpret %% stuff
  RD  :  height
  RD  :  widthfactor
  RD  :  oblique angle
  RD  :  Tracking percentage
  RL  :  Is backwards (0/1)
  RL  :  Is upside down (0/1)
  RL  :  Is vertical (0/1)
  RL  :  Is underlined (0/1)
  RL  :  Is overlined (0/1)
  PS  :  Font filename
  PS  :  Big font filename
```

XLINE 12

```
3 RD  :  a point on the construction line
```

```
       3 RD  :  another point
```

RAY 13

```
       3 RD  :  a point on the construction line
       3 RD  :  another point
```

These "SUBENT" items indicate changes for subsequently drawn items.

SUBENT_COLOR 14

```
         RL  :  color
```

SUBENT_LAYER 16

```
         RL  :  layer index
```

SUBENT_LINETYPE 18

```
         RL  :  linetype index, 0xFFFFFFFF for bylayer, 0xFFFFFFFE for byblock
```

SUBENT_MARKER 19

```
         RL  :  marker index
```

SUBENT_FILLON 20

```
         RL  :  fill on if 1, off if 0
```

SUBENT_TRUECOLOR 22

```
         RC  :  red
         RC  :  green
         RC  :  blue
```

SUBENT_LNWEIGHT 23

```
         RL  :  line weight
```

SUBENT_ LTSCALE 24

```
         RD  :  linetype scale
```

SUBENT_ THICKNESS 25

```
         RD  :  thickness
```

SUBENT_ PLSTNAME 26

```
         RL  :  type, BYLAYER == 0, BYBLOCK == 1, DICT_DEFAULT == 2, PLOTSTYLE_BY_ID == 3
         RL  :  plot style index
```

PUSH_CLIP 27

```
 3 RD : extrusion
 3 RD : clip boundary origin
   RL : number of points
 2 RD : 2D point, repeated number of points times
16 RD : clip boundary transformation matrix
16 RD : inverse block transformation matrix
   RL : front clip on
   RL : back clip on
   RD : front clip
   RD : back clip
   RL : draw boundary (0/1)
```

POP_CLIP 28

```
empty
```

PUSH_MODELXFORM 29

```
16 RD : transformation matrix
```

PUSH_MODELXFORM2 30

```
16 RD : transformation matrix
    ? : unknown data
```

POP_MODELXFORM 31

```
empty
```

Polyline with normal 32

```
   RL : number of points
 3 RD : a point (repeat "number of points" times)
   RD : normal vector
```

LWPOLYLINE 33

```
   RL : number of bytes containing the LWPOLYLINE entity data
    B : bytes containing the LWPOLYLINE entity data. This excludes the common entity data.
        More specifically: it starts at the LWPOLYLINE flags (BS), and ends with the width
        array (BD).
   RC : Unknown byte
   RC : Unknown byte
   RC : Unknown byte
```

Sub entity material 34

```
          H: Material handle
```

Sub entity mapper 35

```
         RL  :  dummy value 1
         RL  :  dummy value 2
         RL  :  projection
         RL  :  U-tiling
         RL  :  V-tiling
         RL  :  Auto transform
         RL  :  dummy value 3
```

Unicode text 36

Identical to text (10), but with the text string type being encoded in unicode.

Unknown 37

```
               Empty
```

Unicode text 2

Identical to text 2 (11), but with the text string type being encoded in unicode and some minor additions:

```
       3 RD  :  start point
       3 RD  :  normal
       3 RD  :  text direction
        PUS  :  string, padded to 4 byte boundary
         RL  :  length of string, -1 if zero terminated
         RL  :  "raw"; 0 if raw, 1 if not.  raw means don't interpret %% stuff
         RD  :  height
         RD  :  widthfactor
         RD  :  oblique angle
         RD  :  Tracking percentage
         RL  :  Is backwards (0/1)
         RL  :  Is upside down (0/1)
         RL  :  Is vertical (0/1)
         RL  :  Is underlined (0/1)
         RL  :  Is overlined (0/1)
True type font descriptor fields {
         RL  :  Is bold (0/1)
         RL  : Is italic (0/1)

         RL  : Charset (contains 1 byte)

         RL  : Pitch and family (contains 1 byte)

        PUS  : Type face

        PUS  : Font filename

}

        PUS  : Big font filename
```

- END OF DOCUMENT -