

**Open Design Specification for .dwg files**  
**Version 5.1**

**Open Design Alliance**

**[www.opendesign.com](http://www.opendesign.com)**

Copyright © 1998-2010 Open Design Alliance, Inc. All rights reserved.

Information in these materials is furnished for informational use only, is subject to change without notice and does not represent a commitment on the part of Open Design Alliance. Open Design Alliance assumes no responsibility or liability for any errors or inaccuracies that may appear in these materials. Use these materials at your own risk.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, OPEN DESIGN ALLIANCE AND ITS SUPPLIERS DISCLAIM ANY AND ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT, AND THOSE ARISING OUT OF USAGE OF TRADE OR COURSE OF DEALING, CONCERNING THESE MATERIALS. THESE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL OPEN DESIGN ALLIANCE OR ITS SUPPLIERS (OR THEIR RESPECTIVE AGENTS, DIRECTORS, EMPLOYEES OR REPRESENTATIVES) BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, ECONOMIC, PUNITIVE OR SIMILAR DAMAGES, OR DAMAGES FOR LOSS OF BUSINESS PROFITS, LOSS OF GOODWILL, BUSINESS INTERRUPTION, COMPUTER FAILURE OR MALFUNCTION, LOSS OF BUSINESS INFORMATION OR ANY AND ALL OTHER COMMERCIAL OR PECUNIARY DAMAGES OR LOSSES) ARISING OUT OF THE USE OF THESE MATERIALS, HOWEVER CAUSED AND ON ANY LEGAL THEORY OF LIABILITY (WHETHER IN TORT, CONTRACT OR OTHERWISE), EVEN IF OPEN DESIGN ALLIANCE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

Teigha™ is a trademark of Open Design Alliance in the United States and/or other countries. All other trademarks, trade names or company names referenced herein are used for identification only and are the property of their respective owners.

US Government Restricted Rights: These materials are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. The contractor/manufacturer is Open Design Alliance, 5025 N Central Ave #602, Phoenix, AZ, USA.

Printed in USA.

\* DWG is the native and proprietary file format for AutoCAD® and a trademark of Autodesk, Inc. The Open Design Alliance is not associated with Autodesk.

## Open Design Specification for .dwg files

### Table of Contents

1	Introduction.....	5
2	BIT CODES AND DATA DEFINITIONS.....	6
3	R13-R15 DWG FILE FORMAT ORGANIZATION .....	19
4	R2004 DWG FILE FORMAT ORGANIZATION .....	21
5	R2007 DWG FILE FORMAT ORGANIZATION .....	33
6	R2010 DWG FILE FORMAT ORGANIZATION .....	64
7	Data section AcDb:Header (HEADER VARIABLES) .....	65
8	Data section AcDb:Classes .....	79
9	PADDING (R13C3 AND LATER) .....	81
<b>10</b>	Data section: “” .....	82
	Data section AcDb:SummaryInfo Section .....	83
12	Data section AcDb:Preview .....	85
<b>13</b>	Data section AcDb:VBAProject Section .....	87
14	Data section AcDb:AppInfo.....	88
15	Data section AcDb:FileDepList .....	90
16	Data section AcDb:RevHistory .....	92
17	Data section AcDb:Security .....	93
18	Data section AcDb:AcDbObjects .....	95
19	Data section AcDb:ObjFreeSpace.....	209
20	Data section AcDb:Handles (OBJECT MAP) .....	210
21	UNKNOWN SECTION.....	211
22	SECOND FILE HEADER (R13-R15).....	212

23	Data section: AcDb:AuxHeader (Auxiliary file header).....	215
24	Extended Entity Data (Extended Object Data) .....	217
25	PROXY ENTITY GRAPHICS .....	218

# 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 2010. 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)

BS : bitshort

BL : bitlong

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

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 BITSHORT:

### 1<sup>st</sup> 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:

```
0000000001000000011011010000111110
```

It would be parsed like this:

```
00 00000001 00000001 (short 257)
10                (0)
11                (256)
01 00001111      (15)
10                (0)
```

## 2.2 BITLONG:

### 1<sup>st</sup> 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:

00000000010000000100000000000000000000100100001111110

It would be parsed like this:

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

## 2.3 BITDOUBLE:

### 1<sup>st</sup> 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.

## 2.4 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.

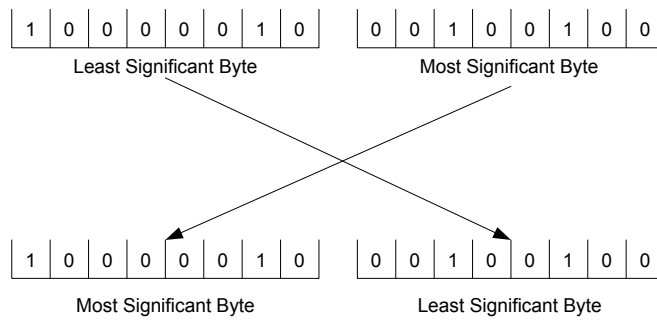
In each byte, the high bit is a flag; when set, it indicates that another byte follows. The concept is not difficult to understand, but is a little difficult to explain. Let's look at an example.

Assume the next two bytes in the file are:

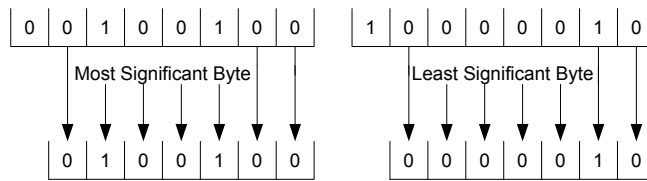
10000010 00100100

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.

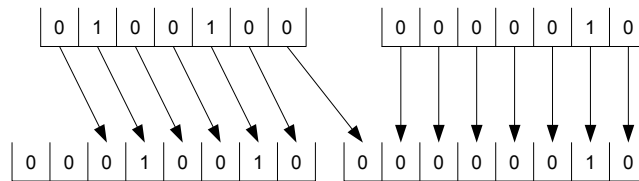




Now we drop the high order flag bits:



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

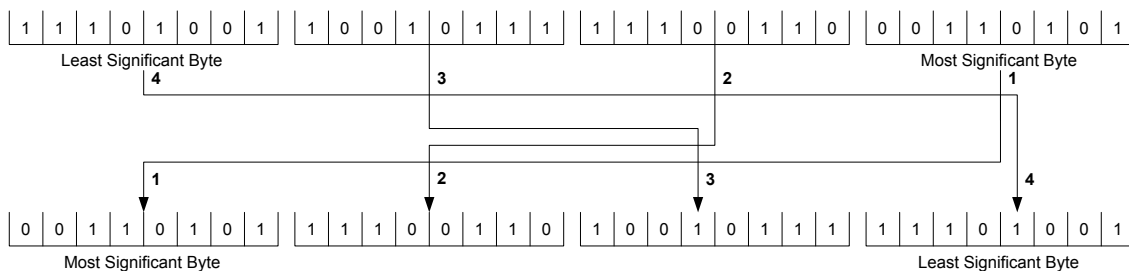


**Result** = 2 + 18\*256 = 4610

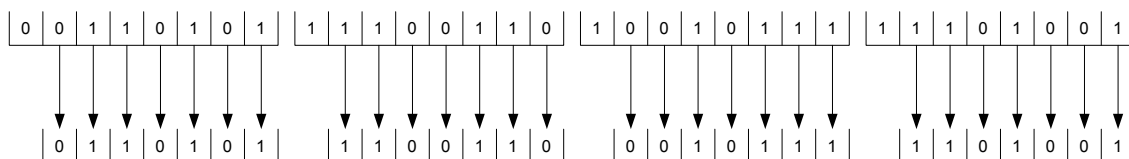
Here's another example using the basic form F1101001 F0010111 F1100110 00110101:

11101001 10010111 11100110 00110101

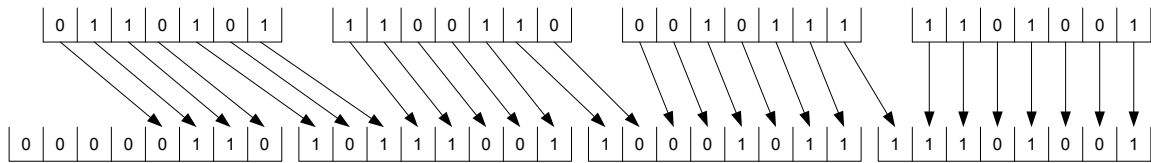
We read bytes until we reach a byte with a high bit of 0. Obviously the fourth byte meets that criterion. Since we are reading from least significant to most significant, let's reverse the order of the bytes so that they read MSB to LSB from left to right.



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:



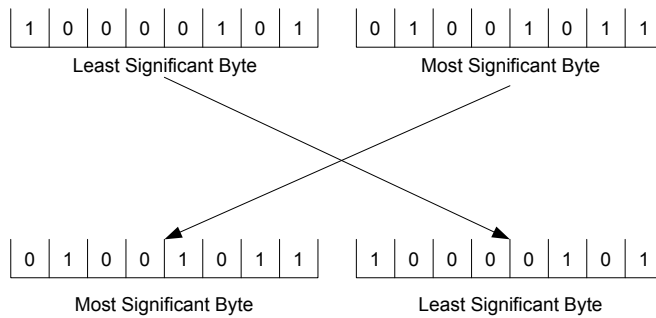
-----

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

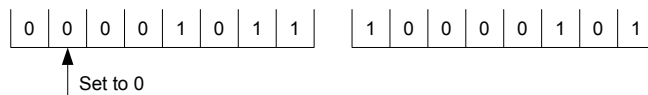
This process is further complicated by the fact that if the final byte (high bit 0) also has the 64 bit (0x40) set, this means to negate the number.

This is a negative number: 10000101 01001011

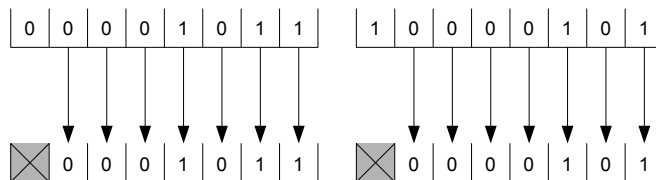
Since we are reading from least significant to most significant, let's reverse the order of the bytes so that they read MSB to LSB from left to right.



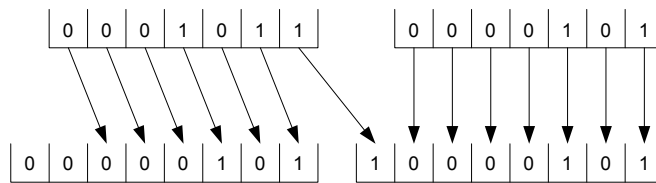
We then clear the bit that was used to represent the negative number, and note that the result must be negated:



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:



:  $133+5*256=1413$ , 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.

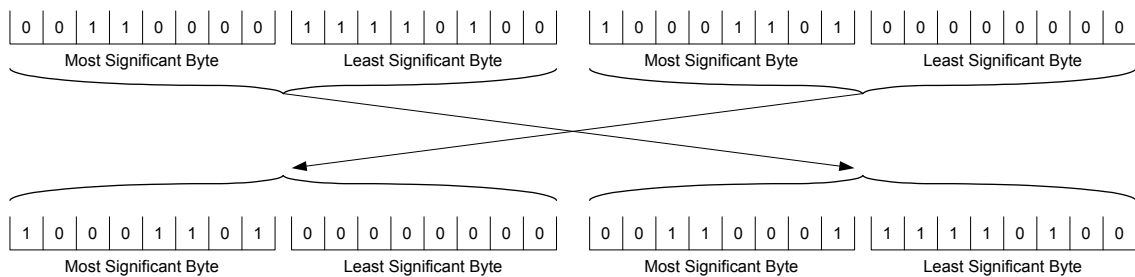
## 2.5 MODULAR SHORTS

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

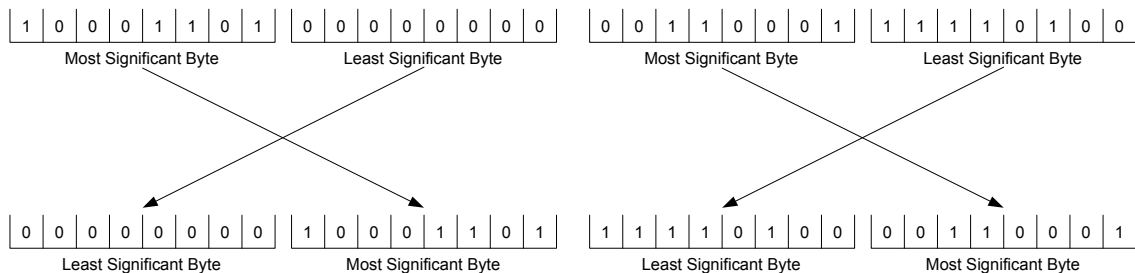
There are only two cases to worry about here (from a practical point of view), because, in the case of shorts, two modules make a long, and since these are used only to indicate object sizes, a maximum object size of 1 GB is probably correct.

00110001 11110100 10001101 00000000.

Reverse the order of the shorts:



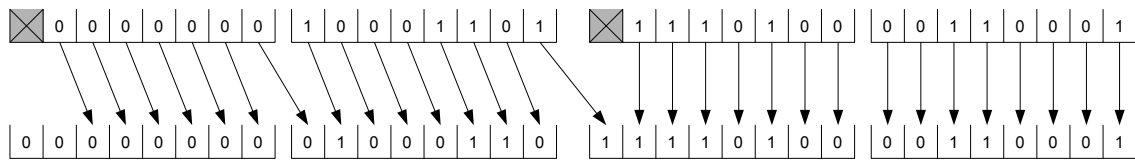
Reverse the order of the bytes in each short:



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:



-----  
 : 62513+70\*65536=4650033

## 2.6 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.

## 2.7 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.

## 2.8 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.

## 2.9 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).

---

## 2.10 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.

## 2.11 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	Description
2	
3	
4	
5	

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)

```

## 2.12 CRCS:

### 2.12.1 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,
0xCC01,0x0CC0,0x0D80,0xCD41,0x0F00,0xCF41,0xCE81,0x0E40,
0x0A00,0xCAC1,0xCB81,0x0B40,0xC901,0x09C0,0x0880,0xC841,
0xD801,0x18C0,0x1980,0xD941,0x1B00,0xDB41,0xDA81,0x1A40,
0x1E00,0xDE41,0xDF81,0x1F40,0xDD01,0x1DC0,0x1C80,0xDC41,
0x1400,0xD4C1,0xD581,0x1540,0xD701,0x17C0,0x1680,0xD641,
0xD201,0x12C0,0x1380,0xD341,0x1100,0xD141,0xD081,0x1040,
0xF001,0x30C0,0x3180,0xF141,0x3300,0xF341,0xF281,0x3240,
0x3600,0xF6C1,0xF781,0x3740,0xF501,0x35C0,0x3480,0xF441,
0x3C00,0xFCC1,0xFD81,0x3D40,0xFF01,0x3FC0,0x3E80,0xFE41,
0xFA01,0x3AC0,0x3B80,0xFB41,0x3900,0xF941,0xF881,0x3840,
0x2800,0xE8C1,0xE981,0x2940,0xEB01,0x2BC0,0x2A80,0xEA41,
0xEE01,0x2EC0,0x2F80,0xEF41,0x2D00,0xED41,0xEC81,0x2C40,
0xE401,0x24C0,0x2580,0xE541,0x2700,0xE741,0xE681,0x2640,
0x2200,0xE2C1,0xE381,0x2340,0xE101,0x21C0,0x2080,0xE041,
0xA001,0x60C0,0x6180,0xA141,0x6300,0xA341,0xA281,0x6240,
0x6600,0xA6C1,0xA781,0x6740,0xA501,0x65C0,0x6480,0xA441,
0x6C00,0xACC1,0xAD81,0x6D40,0xAF01,0x6FC0,0x6E80,0xAE41,
0xAA01,0x6AC0,0x6B80,0xAB41,0x6900,0xA941,0xA881,0x6840,
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.

## 2.12.2 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,
    0x1dad47d, 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, 0xbf061116,
    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
  PADDING (R13C3 AND LATER)
  IMAGE DATA (PRE-R13C3)
  OBJECT DATA
    All entities, table entries, dictionary entries, etc. go in this
    section.
  OBJECT MAP
  UNKNOWN SECTION (R13C3 AND LATER)
  SECOND HEADER
  IMAGE DATA (R13C3 AND LATER)
```

### 3.2 FILE HEADER

#### 3.2.1 VERSION ID:

The first 6 bytes are "AC1012" for R13, "AC1014" for R14, and "AC1015" for R2000.

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 UNKNOWN SECTION:

Bytes at 0x11 and 0x12 are still unknown; usually 0.

#### 3.2.4 DWGCODEPAGE:

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

### 3.2.5 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 R14, points to a location where there may be data stored. Currently we have
  seen only the MEASUREMENT variable stored here.

  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 Dwg 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)

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 %= 0xFFFF1;
        sum2 %= 0xFFFF1;
    }
    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 include Section 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)
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
0x08	8	Start offset for this page (OdUInt64)

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	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**

**compOffset**

**litCount**

**0x00 – 0x0F**

**0x10:**

- 
- 
- 

**0x11 :**

**0x12– 0x1F :**

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

**0x20 :**

- $\text{compressedBytes}$  is read as the next Long Compression Offset (see format below) + 0x21.
- $\text{compOffset}$  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 :**

- $\text{compressedBytes} = \text{opcode1} - 0x1E$ .
- 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 :**

- $\text{compressedBytes} = ((\text{opcode1} \& 0xF0) \gg 4) - 1$
- Read the next byte (call it opcode2):
- $\text{compOffset} = (\text{opcode2} \ll 2) \mid ((\text{opcode1} \& 0x0C) \gg 2)$
- 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****Two Byte Offset**

## **Long Compression Offset**



## 5 R2007 DWG FILE FORMAT ORGANIZATION

### 5.1 Sections and pages overview

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	The data page map
Page map 2	0x400	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	Unknown
0x0C	1	Byte 0x00, 0x01, or 0x03
0x0D	4	Preview address (long)
0x11	1	DwgVer
0x12	1	MaintReleaseVer
0x13	2	Codepage
0x15	3	Unknown
0x18	4	SecurityType (long)
0x1C	4	Unknown long
0x20	4	DWG Property Addr
0x24	4	VBA Project Addr
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
0x30	8	Pages map2Id
0x38	8	PagesMapOffset
0x40	8	PagesMapId

0x48	8	Header2offset (relative to page map 1 address)
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



	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
	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,	hashcode	0x6c4205ca

	or fonts used by STYLE).	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 that wrote the .dwg file (encrypted = 2).	hashcode	0x3fa0043e
		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	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***

#### **5.2.1.1.2 *CrcSeed***

#### **5.2.1.1.3 *SectionsMapCrcSeed***

#### **5.2.1.1.4 *PagesMapCrcSeed***

#### **5.2.1.1.5 *Check data***

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

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(buffer[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;
}
```

```
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***

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

#### **5.2.1.3 Compress and calculate 64-bit CRC (compressed)**

#### **5.2.1.4 Create checking sequence and 64-bit CRC**

### 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.

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 the `UpdateSeed1` 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.

## 5.4 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 pseudocode 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 %= 0xffff1;
sum2 %= 0xffff1;
}
}

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

## 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 : Maximum 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:

- *length*
- 

### 5.10.1 Copying a compressed chunk

*length*

```

UInt32 ReadLiteralLength(byte[] buffer) {
    UInt32 length = opCode + 8;
    if (length == 0x17) {
        UInt32 n = buffer[sourceIndex++];
        length += n;
    }
}

```





```
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;
    }
}
```

```
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

```
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;
    }
}

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;
}

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;
}
```

## 5.1264-bit CRC calculation

---


```
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

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

```
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,  
0x167ff3eacba2af1, 0x548f120162451c62, 0x939e303d987b47d7, 0xd16ed1d631917144,  
0x5f4c95afc5edc62e, 0x1dbc74446c07f0bd, 0xdaad56789639ab08, 0x985db7933fd39d9b,  
0x84193f60d72af34f, 0xc6e9de8b7ec0c5dc, 0x01f8fcb784fe9e69, 0x43081d5c2d14a8fa,  
0xcd2a5925d9681f90, 0x8fdab8ce70822903, 0x48cb9af28abc72b6, 0x0a3b7b1923564425,  
0x70428b155b4eaf1e, 0x32b26afef2a4998d, 0xf5a348c2089ac238, 0xb753a929a170f4ab,  
0x3971ed50550c43c1, 0x7b810cbbf6ce67552, 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, 0xf652blad0167feea, 0xb4a25046a88dc879,
0xa8e6d8b54074a6ad, 0xea16395ee99e903e, 0x2d071b6213a0cb8b, 0x6ff7fa89ba4afd18,
0xe1d5bef04e364a72, 0xa3255f1be7dc7ce1, 0x64347d271de22754, 0x26c49cccb40811c7,
0x5cbd6cc0cc10fafc, 0x1e4d8d2b65facc6f, 0xd95caf179fc497da, 0x9bac4efc362ea149,
0x158e0a85c2521623, 0x577eeb6e6bb820b0, 0x906fc95291867b05, 0xd29f28b9386c4d96,
0xcdba04ad0952342, 0x8c2b41a1797f15d1, 0x4b3a639d83414e64, 0x09ca82762aab78f7,
0x87e8c60fdded7cf9d, 0xc51827e4773df90e, 0x020905d88d03a2bb, 0x40f9e43324e99428,
0x2cffe7d5975e55e2, 0xe0f063e3eb46371, 0xa91e2402c48a38c4, 0xebeec5e96d600e57,
0x65cc8190991cb93d, 0x273c607b30f68fae, 0xe02d4247cac8d41b, 0xa2dda3ac6322e288,
0xbe992b5f8b8bc5c, 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, 0xf7021977f9ccea4, 0xb5f2f89c5026dc37,
0x3bd0bce5a45a6b5d, 0x79205d0e0db05dce, 0xbe317f32f78e067b, 0xfcc19ed95e6430e8,
0x86b86ed5267cddb3, 0xc4488f3e8f96ed40, 0x0359ad0275a8b6f5, 0x41a94ce9dc428066,
0xcf8b0890283e370c, 0x8d7be97b81d4019f, 0x4a6acb477bea5a2a, 0x089a2aacd2006cb9,
0x14dea25f3af9026d, 0x562e43b4931334fe, 0x913f6188692d6f4b, 0xd3cf8063c0c759d8,
0x5dedc41a34bbeeb2, 0x1fd25f19d51d821, 0xd80c07cd676f8394, 0x9afce626ce85b507

```

## 5.12.2 Mirrored CRC

```

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

```

```

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, 0xd1ab4dea28ed965,  
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, 0x6debd121b863991, 0x1733afda2bb3b0e8,  
0xf34b37458bb86399, 0x8993478dbb8deae0, 0x06fbd6d5ebd3716b, 0x7c23a61ddb6e6f812,  
0x3373d23613f9d516, 0x49aba2fe23cc5c6f, 0xc6c333a67392c7e4, 0xbc1b436e43a74e9d,  
0x95ac9329ac4bc9b5, 0xef74e3e19c7e40cc, 0x601c72b9cc20db47, 0x1ac40271fc15523e,  
0x5594765a340a7f3a, 0x2f4c0692043ff643, 0xa02497ca54616dc8, 0xdafce7026454e4b1,  
0x3e847f9dc45f37c0, 0x445c0f55f46abeb9, 0xcb349e0da4342532, 0xb1ecec59401ac4b,  
0xfebc9aee5c1e814f, 0x8464ea266c2b0836, 0x0b0c7b7e3c7593bd, 0x71d40bb60c401ac4,  
0xe8a46c1224f5a634, 0x927c1cda14c02f4d, 0x1d148d82449eb4c6, 0x67ccfd4a74ab3dbf,  
0x289c8961bcb410bb, 0x5244f9a98c8199c2, 0xdd2c68f1dcd0249, 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, 0xd476a1c3461bbbed2, 0xaaead10b762e37ab,  
0x37deb6af5e9b8b5b, 0x4d06c6676eae0222, 0xc26e573f3ef099a9, 0xb8b627f70ec510d0,

0xf7e653dcc6da3dd4, 0x8d3e2314f6efb4ad, 0x0256b24ca6b12f26, 0x788ec2849684a65f,  
0x9cf65a1b368f752e, 0xe62e2ad306bafc57, 0x6946bb8b56e467dc, 0x139ecb4366d1eea5,  
0x5cceb6f68aecec3a1, 0x2616cfa09efb4ad8, 0xa97e5ef8cea5d153, 0xd3a62e30fe90582a,  
0xb0c7b7e3c7593bd8, 0xca1fc72bf76cb2a1, 0x45775673a732292a, 0x3faf26bb9707a053,  
0x70ff52905f188d57, 0x0a2722586f2d042e, 0x854fb3003f739fa5, 0xff97c3c80f4616dc,  
0x1bef5b57af4dc5ad, 0x61372b9f9f784cd4, 0xee5fbac7cf26d75f, 0x9487ca0fff135e26,  
0xdbd7be24370c7322, 0xa10fceed0739fa5b, 0x2e675fb4576761d0, 0x54bf2f7c6752e8a9,  
0xcdcf48d84fe75459, 0xb71738107fd2dd20, 0x387fa9482f8c46ab, 0x42a7d9801fb9cfd2,  
0x0df7adabd7a6e2d6, 0x772fdd63e7936baf, 0xf8474c3bb7cdf024, 0x829f3cf387f8795d,  
0x66e7a46c27f3aa2c, 0x1c3fd4a417c62355, 0x935745fc4798b8de, 0xe98f353477ad31a7,  
0xa6df411fbfb21ca3, 0xdc0731d78f8795da, 0x536fa08dfd90e51, 0x29b7d047efec8728

## 5.13 Reed-Solomon encoding

---

•

•

### **5.13.1 Non-interleaved**

### **5.13.2 Interleaved**

## 6 R2010 DWG FILE FORMAT ORGANIZATION



## 7 Data section AcDb:Header (HEADER VARIABLES)

Beginning sentinel  
 Size of the section (a 4 byte long)  
 R2010: 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

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

TYPE	DESCRIPTION
------	-------------

R2007 Only:

RL : Size in bits

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 : TREDEPTH

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).

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 : DIMXTDIRECTION
        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_MLINESSTYLE) (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)
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 : HIDE TEXT
RC : XCLIPFRAME
RC : DIMASSOC
RC : HALOGAP
BS : OBSCURED COLOR
BS : INTERSECTION COLOR
RC : OBSCURED LTYPE
RC : INTERSECTION DISPLAY
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.

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

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 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 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
    
```



```
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 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 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
```

## 8 Data section AcDb:Classes

### 8.1 R13-R15

---

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

---



---

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.

---



---

RS : CRC

---

0x72,0x5L,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.

### 8.2 R18+

---

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

RL : size of class data area.

R2010+

RL : unknown, possibly the high 32 bits of a 64-bit size?

R2004+

BS : Maximum class number

RC : 0x00

RC : 0x00

B : true

---

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)

---

RS : CRC

---

0x72,0x5L,0x3B,0x47,0x3B,0x56,0x07,0x3A,0x3F,0x23,0x0B,0xA0,0x18,0x30,0x49,0x75



## 9 PADDING (R13C3 AND LATER)





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

## 12 Data section AcDb:Preview

### 12.1 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.

### 12.2 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 };
```

---

## 13 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, 0x00, 0x1c, 0x00, 0x00, 0x19, 0x00, 0x00, 0x00
byte	n	The VBA project data
Int32	4	0

## 14 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 DWG file format version in the file header. So a R18 .dwg file might have an R21 AppInfo section.

### 14.1 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".

### 14.2 R21

In R21 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".

## 15 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

## 16 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.

### 16.1R18

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.

### 16.2R21

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.

## 17 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	Cryptographic 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.

## 18 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).

Objects have the following general format:

---

```

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).
Common:
OT : Object type
R2000-R2007 Only:
RL : Size of object data in bits (number of bits before the handles), or the “endbit”
      of the pre-handles section.
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. Only entities have this
      flag

```

```

        if (graphicimageflag is 1) {
            RL : Size of graphic image in bytes
            X : The graphic image
        }

```

R13-R14 Only:

RL : Size of object data in bits

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:

X : Handles associated with this object (begins at offset specified by number of bits before handles above)

RS : CRC

---

The CRC includes the size bytes.

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).

Common:

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) {
            RL: Size of graphic image in bytes

```



```
        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
    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.

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

---

TEXT	1	XLINE	0x29
ATTRIB	2	DICTIONARY	0x2A
ATTDEF	3		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

---

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

```

DICTIONARYVAR
DICTIONARYWDFLT
HATCH
IDBUFFER
IMAGE
IMAGEDEF
IMAGEDEFREACTOR
LAYER_INDEX
LAYOUT
LWPLINE
OLE2FRAME
PLACEHOLDER
RASTERVARIABLES
SORTENTSTABLE
SPATIAL_FILTER
SPATIAL_INDEX
VBA_PROJECT
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 field determines the type of the object.

See the sections on EED a description of that areas.

## 18.1 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.

### 18.1.1 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.
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 = continuous, 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	

### 18.1.2 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).

### 18.1.3 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 subsequent 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	---
-----	---	-----

**18.1.3.1 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 [e..... 0101 1011 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
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

```

**18.1.4 ATTRIB (2)**

Common Entity Data

R13-R14 Only:

Elevation	BD	---
Ins pt	2RD	10
Align pt	2RD	11
Extrusion	3BD	210
Thickness	BD	39
Oblique ang	BD	51
Rotation ang	BD	50
Height	BD	40
Width factor	BD	41

Value	TV	1	
Generation	BS	71	
Horiz align.	BS	72	
Vert align.	BS	74	(It's 73 in TEXT.)
R2000+ Only:			
DataFlags	RC		Used to determine presence of subsequent 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	74	present if !(DataFlags & 0x80)
R2010+:			
Version	RC	?	
Common:			
Tag	TV	2	
Field length	BS	73	unused
Flags	RC	70	NOT bit-pair-coded.
R2007+:			
Lock position flag	B	280	
Common:			
Common Entity Handle Data			
	H	7	STYLE (hard pointer)
CRC	X	---	

#### 18.1.4.1 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 e.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 [e..... 0101 1011 0100 0000 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 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 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 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 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 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
```

### 18.1.5 ATTDEF (3)

Common Entity Data

R13-R14 Only:

Elevation	BD	---	
Ins pt	2RD	10	
Align pt	2RD	11	
Extrusion	3BD	210	
Thickness	BD	39	
Oblique ang	BD	51	
Rotation ang	BD	50	
Height	BD	40	
Width factor	BD	41	
Default value	TV	1	
Generation	BS	71	
Horiz align.	BS	72	
Vert align.	BS	74	(It's 73 in TEXT.)

R2000+ Only:

DataFlags	RC		Used to determine presence of subsequent 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)
Default value	TV	1	
Generation	BS	71	present if !(DataFlags & 0x20)
Horiz align.	BS	72	present if !(DataFlags & 0x40)
Vert align.	BS	74	present if !(DataFlags & 0x80)

Common:

Tag	TV	2	
Field length	BS	73	unused
Flags	RC	70	NOT bit-pair-coded.

R2007+:

Lock position flag	B	280	
--------------------	---	-----	--

R2010+:

```

Version          RC      ?
Common:
Prompt          TV      3

Common Entity Handle Data
                H      7      STYLE (hard pointer)
CRC            X      ---

```

---

### 18.1.5.1 **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 [e..... 0101 1011 0100 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 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

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

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

```

### 18.1.6 BLOCK (4)

```

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

```

---

#### 18.1.6.1 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 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.

### 18.1.7 ENDBLK (5)

Common Entity Data

Common Entity Handle Data

CRC X ---

#### 18.1.7.1 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 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

```

### 18.1.8 SEQEND (6)

Common Entity Data

Common Entity Handle Data

CRC X ---

#### 18.1.8.1 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 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

```

### 18.1.9 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 &amp; 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	---	

**18.1.9.1 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 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
    
```

**18.1.10 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	---	

**18.1.10.1 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 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
    
```

**18.1.11 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	---	

---

**18.1.11.1 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 0001

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

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

**18.1.12 VERTEX (3D) (11)**

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

---

**18.1.12.1 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 0001

00D7E 5B 10 00 00 00 00 00 00  [..... 0101 1011 0001 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 1111

00D8E 00 80          ..          0000 0000 1000 0000

00D90 8C 03          crc

```

### 18.1.13 VERTEX (MESH) (12)

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

#### 18.1.13.1 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 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

```

### 18.1.14 VERTEX (PFACE) (13)

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

#### 18.1.14.1 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 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

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

```

### 18.1.15 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	---
-----	---	-----

**18.1.15.1 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 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

```

**18.1.16 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	---
-----	---	-----

---

**18.1.16.1 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 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

```



**18.1.17 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	---	

**18.1.17.1 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 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

```

**18.1.18 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 ---

---

**18.1.18.1 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  DèY$....  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

```

**18.1.19 CIRCLE (18)**

## Common Entity Data

```

Center          3BD    10
Radius          BD     40
Thickness       BT     39
Extrusion       BE    210

```

## Common Entity Handle Data

---

CRC X ---

---

**18.1.19.1 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

```

**18.1.20 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	---	

**18.1.20.1 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

```

**18.1.21 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.

### 18.1.22 DIMENSION (ORDINATE) (20)

Common Entity Data				
Common Dimension Data			See paragraph 18.1.21.	
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	---	

---

**18.1.22.1 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 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

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

```

**18.1.23 DIMENSION (LINEAR) (21)**

## Common Entity Data

## Common Dimension Data

See paragraph 18.1.21.

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	---	

---

**18.1.23.1 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 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

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

```

### 18.1.24 DIMENSION (ALIGNED) (22)

Common Entity Data

Common Dimension Data

See paragraph 18.1.21.

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	---	

#### 18.1.24.1 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 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

```

```

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 00 2A 41 59 .....*AY 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

```

### 18.1.25 DIMENSION (ANGULAR, 3-PT) (23)

Common Entity Data

Common Dimension Data

See paragraph 18.1.21.

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	---	

#### 18.1.25.1 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 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

```





```

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 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

```

### 18.1.28 DIMENSION (DIAMETER) (26)

Common Entity Data

Common Dimension Data See paragraph 18.1.21.

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	---	

#### 18.1.28.1 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

```

### 18.1.29 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	---
-----	---	-----

#### 18.1.29.1 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 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

```

### 18.1.30 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		
1 <sup>st</sup> corner x	RD	10	
1 <sup>st</sup> corner y	RD	20	
1 <sup>st</sup> corner z	RD	30	Present only if "Z is zero bit" is 0.
2 <sup>nd</sup> corner	3DD	11	Use 10 value as default point
3 <sup>rd</sup> corner	3DD	12	Use 11 value as default point
4 <sup>th</sup> 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	---	

### 18.1.30.1 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 0100
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 . j . 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

```

### 18.1.31 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			



**18.1.32.1 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 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

```

**18.1.33 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      ---

```

**18.1.33.1 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 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

```

**18.1.34 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	---
-----	---	-----

**18.1.34.1 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 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.....e 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  ..&!. 1 1010 0010 0000 0101 0010 0110 0000 1010 0010 0001 1111 1000 0010 0000 0110 1100

01941 1A          .          0001 1010

01942 7E C2          crc

```

**18.1.35 SHAPE (33)**

Common Entity Data

Ins pt	3BD	10	
Scale	BD	40	
Rotation	BD	50	
Width factor	BD	41	
Oblique	BD	51	
Thickness	BD	39	
Shapeno	BS	2	(string in DXF)
Extrusion	3BD	210	

## Common Entity Handle Data

	H	SHAPEFILE (hard pointer)
CRC	X	--

---

**18.1.35.1 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  H8}gH...  0100 1000 0100 0000 0111 1101 0110 0111 0100 1000 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

```

**18.1.36 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)

### 18.1.36.1 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 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 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

### 18.1.37 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      ---

```

---

#### 18.1.37.1 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

```

### 18.1.38 SPLINE (36)

```

Common Entity Data
Scenario              BS          a flag which is 2 for fitpts only, 1 for
                    ctrlpts/knots
Degree                BS          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          BS      74      Stored as a LONG, defined as a short.  See # of
                                   knots below.

        Fit points          3BD      11      As many as specified by the 74.
}

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, fitpts, etc.
    Numctrlpts              BL         73      Number of 10's (and 41's, if weighted) that follow.
                                   Same, stored as LONG, defined as 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          ---

```

**18.1.38.1 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

**18.1.39 REGION (37) 3DSOLID (38) BODY (39)**

Common Entity Data

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	---	

**18.1.39.1 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 .e...@.. 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 .e...@.e 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 @...@...e 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 ..e...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 @...@...e 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 Tve...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 @...@...e 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  e.e.e.e@  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 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
```

```
01FOC 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<F. 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 .....fhh 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 1110 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.sgg.. 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 .O..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 .....#e= 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

```

### 18.1.40 RAY (40)

Common Entity Data

Point                    3BD     10

Vector                   3BD     11

Common Entity Handle Data

CRC                      X     ---

#### 18.1.40.1 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

```

### 18.1.41 XLINE (41)

#### 18.1.41.1 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

```

**18.1.42 DICTIONARY (42)**

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 dictionary 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)

**18.1.42.1 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 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 USE..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

**18.1.43 DICTIONARYWDFLT**

H	7	Default entry (hard pointer)
---	---	------------------------------

**18.1.44 MTEXT (44)**

Common Entity 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 (although 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.

R2000+:

Linespacing Style	BS	73	
-------------------	----	----	--

Linespacing Factor	BD	44	
--------------------	----	----	--

Unknown bit	B		
-------------	---	--	--

R2004+:

Background flags	BL	0 = no background, 1 = background fill, 2 = background fill with drawing fill color.
Background scale factor	BL	Present if background flags = 1, default = 1.5
Background color	CMC	Present if background flags = 1
Background transparency	BL	Present if background flags = 1

Common:

Common Entity Handle Data		
	H	7 STYLE (hard pointer)
CRC	X	---

**18.1.44.1 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 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 0000 0000

00564 6F F0          crc

```

**18.1.45 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.

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.
Extrusion	3BD	210	
x direction	3BD	211	
offsettoblockinspt	3BD	212	Used when the BLOCK option is used. Seems to be an unused feature.
R14+:			
Unknown	3BD		
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	---	

**18.1.45.1 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

```

### 18.1.46 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)
---	-------------------------

#### 18.1.46.1 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

```

### 18.1.47 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 {
    numseparms BS
    do numseparms times {
      sepparm      BD      segment parameter
    }
    numareafillparms BS
    do num area fill parms times {
      areafillparm  BD      area fill parameter
    }
  }
}

```

## Common Entity Handle Data

---

H mline style object handle (hard pointer)

---

**18.1.47.1 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 e..... 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
```

### 18.1.48 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	---	

#### 18.1.48.1 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.ed... 0100 1100 0000 0000 0100 0000 0110 0100 1000 0000 0000 0000 0000 0000 1001
0046E 08 40 30 21 93 21 9F 21 .e0!...! 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. 1111 0100 0010 0010 0000 0001 0001 0011 0011 0001 0001 1001 0011 0001 0001 0110
00486 C1 3A            crc
```

### 18.1.49 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	---

**18.1.49.1 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 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
    
```

**18.1.50 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	---	

**18.1.50.1 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 1001

024BB 02 40 30 21 0F 21 99  .@!..!  0000 0010 0100 0000 0011 0000 0010 0001 0000 1111 0010 0001 1001 1001

024C2 C3 1D          crc

```

**18.1.51 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:

		6	linetype (hard pointer)
CRC	X	---	

**18.1.51.1 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 1001

02F9B 09 44 45 46 50 4F 49 4E  .DEFFOIN 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.θ..  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

```

**18.1.52 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	---	

### 18.1.52.1 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 1001

024CE 02 40 30 21 10 21 F3  .@!..!.. 0000 0010 0100 0000 0011 0000 0010 0001 0001 0000 0010 0001 1111 0011

024D5 33 8B          crc

```

### 18.1.53 SHAPEFILE (53)

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	---	

### 18.1.53.1 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 1001

02FBA 08 53 54 41 4E 44 41 52 .STANDAR 0000 1000 0101 0011 0101 0100 0100 0001 0100 1110 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

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

```

### 18.1.54 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 ---

**18.1.54.1 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 1001
024E1 01 40 30 21 15 31 13 31 .@!.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
    
```

**18.1.55 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.
}			
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	---	
-----	---	-----	--

---

**18.1.56 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	---	

---

**18.1.56.1 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 0.A..... 0100 1111 0000 0000 0100 0001 1010 0100 1000 0000 0000 0000 0000 0000 1001

00A0E 01 40 30 21 3F      .@!??      0000 0001 0100 0000 0011 0000 0010 0001 0011 1111

00A13 E1 20          crc

```

**18.1.57 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	RC	?	Default value is 1
Type			
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	---	

**18.1.57.1 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 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

```

**18.1.58 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	---	

**18.1.58.1 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 0.A..... 0100 1111 1000 0000 0100 0001 1110 0100 1000 0000 0000 0000 0000 1001

03515 01 40 30 21 4C          .@!L          0000 0001 0100 0000 0011 0000 0010 0001 0100 1100

0351A A0 6F          crc

```

**18.1.59 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	---	
-----	---	-----	--

**18.1.59.1 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 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 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 FE 33 .....3 1100 1111 1000 1111 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 ....e 1111 1101 0000 0100 0001 1100 1100 0001 0100 0000

03EF8 BE 62          crc

```

### 18.1.60 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	---	

#### 18.1.60.1 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 1001

03526 04 40 30 20 21 4E 21 4F .@ !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

```

### 18.1.61 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	---	

**18.1.61.1 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 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
    
```

**18.1.62 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	---	
-----	---	-----	--

**18.1.62.1 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 1001

0353C 02 40 30 21 11 21 86  .@!..!  0000 0010 0100 0000 0011 0000 0010 0001 0001 0010 0001 1000 0110

03543 FA D9          crc

```

**18.1.63 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	---	

**18.1.63.1 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 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

```

**18.1.64 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 ---

**18.1.64.1 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 1001

0354F 03 40 30 21 1D 21 4D 20  .@!.!M  0000 0011 0100 0000 0011 0000 0010 0001 0001 1101 0010 0001 0100 1101 0010 0000

03557 BA 14          crc
    
```

**18.1.65 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	BD	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	?	DIMMZS	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 ---

### 18.1.65.1 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 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

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 0000 0000

04162 CC 33          crc

```

### 18.1.66 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	---	

### 18.1.66.1 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 1001

03563 06 40 30 21 51 21 52 21  .@!Q!R! 0000 0110 0100 0000 0011 0000 0010 0001 0101 0001 0010 0001 0101 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

```

### 18.1.67 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	---	

### 18.1.67.1 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 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

```

### 18.1.68 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)

### 18.1.68.1 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 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

```

### 18.1.69 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
Ltindex	BS	Linetype index (yes, index)

END REPEAT

Handle refs	H	parenthandle (soft pointer) [Reactors (soft pointer)] xdicobjhandle (hard owner)
-------------	---	--

**18.1.69.1 Example:**

OBJECT: mlinestyle (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 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

```



```

043B5 C0 00 00 00 00 0E 03 ..... 1100 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
043E5 00 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

```

## 500 EQUALS THE TYPE OF THIS OBJECT

### 18.1.70 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)

#### 18.1.70.1 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

```

**18.1.71 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 (pathypestatus==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 (pathypestatus==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
}
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      ---

```

---

### 18.1.71.1 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..68.( 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

```

### 18.1.72 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)

**18.1.72.1 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 >.Eb.... 0011 1110 1000 0000 0100 0000 0110 0010 1110 0110 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

```

**18.1.73 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 ---

```

---

**18.1.73.1 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 >@@[1'.. 0011 1110 0100 0000 0100 0000 0101 1011 0110 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 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

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

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 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

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 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

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 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

```

```

02DB8 00 00 00 00 06 B5 6E 62 .....nb 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
02DD0 00 00 00 00 06 D0 38 00 .....8. 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
02DF0 00 00 10 07 F4 00 00 00 ..... 0000 0000 0000 0000 0001 0000 0000 0111 1111 0100 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
02E00 00 00 00 10 07 F0 00 00 ..... 0000 0000 0000 0000 0000 0000 0001 0000 0000 0111 1111 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
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
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
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
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
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

```

### 18.1.74 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)

### 18.1.74.1 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  =.EZ. . . 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  `e..... 0110 0000 0100 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

```

### 18.1.75 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)

### 18.1.75.1 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

02E55 09 02 60 30          ..`0          0000 1001 0000 0010 0110 0000 0011 0000

02E59 A1 13          crc

```

### 18.1.76 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

### 18.1.76.1 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  B@d.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

```

### 18.1.77 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	BS	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)

### 18.1.78 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 far)
    }
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

```

**18.1.78.1 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 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 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 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 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 0000 0010 0000
03EBC 00 00 00 00 00 40 23 20  ....@# 0000 0000 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 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 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 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 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 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 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 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 0000 0000
03F04 00 00 00 00 00 00 1F 20  .... 0000 0000 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 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
    
```

### 18.1.79 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	---	

---

**18.1.79.1 <<No example>>****18.1.80 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	
Object Drawing Format	BL	95	
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.

---

**18.1.80.1 <<No example>>****18.1.81 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)

### 18.1.81.1 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

```

### 18.1.82 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)
Handle refs	H		parenthandle (soft pointer) [Reactors (soft pointer)] xdicobjhandle (hard owner) owner handle (soft pointer) handles of objects (numentries of these, soft pointer)

### 18.1.82.1 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 >.8ig... 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.U8 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.OA.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

```

### 18.1.83 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)

### 18.1.83.1 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

```

### 18.1.84 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)

#### 18.1.84.1 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 ...1..... 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 1111 1110 0101 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 02 ."..... 0000 0000 0010 0010 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 0000 0000 0000 0000
0D3BA FE 50 00 00 FE 50 00 00 .P...P.. 1111 1110 0101 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
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
0D3D2 00 00 05 E0 01 00 22 00 .....". 0000 0000 0000 0000 0000 0101 1110 0000 0000 0001 0000 0000 0010 0010 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
0D3E2 FF 00 3F FE 8F 00 00 FE ..?..... 1111 1111 0000 0000 0011 1111 1111 1110 1000 1111 0000 0000 0000 0000 1111
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
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
0D3FA 00 FE 8F 00 00 01 08 00 ..... 0000 0000 1111 1110 1000 1111 0000 0000 0000 0000 0000 0001 0000 1000 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
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
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
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
0D422 02 00 00 02 01 09 00 3F .....? 0000 0010 0000 0000 0000 0000 0000 0010 0000 0001 0000 1001 0000 0000
0D42A FE 8F 00 70 FE C0 00 00 ...p.... 1111 1110 1000 1111 0000 0000 0111 0000 1111 1110 1100 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
0D43A 00 18 00 01 00 25 00 00 .....%. 0000 0000 0001 1000 0000 0000 0000 0001 0000 0000 0010 0101 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
0D44A 00 AF FF FF 00 00 FE 50 .....P 0000 0000 1010 1111 1111 1111 1111 1111 0000 0000 0000 0000 1111
0D452 0F 00 00 21 D3 01 00 01 ...!.... 0000 1111 0000 0000 0000 0000 0010 0001 1101 0011 0000 0001 0000
0D45A 00 01 00 01 00 01 00 29 .....) 0000 0000 0000 0001 0000 0000 0000 0001 0000 0000 0000 0001
0D462 00 76 00 01 00 01 00 01 .v..... 0000 0000 0111 0110 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
0D472 00 5A 00 D0 9A 00 00 02 .Z..... 0000 0000 0101 1010 0000 0000 1101 0000 1001 1010 0000 0000 0000
0D47A 01 0B 00 3F FE 8F 00 AF ...?.... 0000 0001 0000 1011 0000 0000 0011 1111 1111 1110 1000 1111
0D482 FF FF 00 00 FE 50 00 00 .....P.. 1111 1111 1111 1111 0000 0000 0000 1111 1110 0101 0000
0D48A 02 01 18 00 00 FE 8F 00 ..... 0000 0010 0000 0001 0001 1000 0000 0000 0000 0000 1111
0D492 70 FE C0 00 00 FE 50 01 p.....P. 0111 0000 1111 1110 1100 0000 0000 0000 0000 0000 1111
0D49A 00 00 04 54 00 00 00 00 ...T.... 0000 0000 0000 0000 0000 0100 0101 0100 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 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 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 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
0D622 08 3C BC 40 05 00 00 00 .<@.... 0000 1000 0011 1100 1011 1100 0100 0000 0000 0101 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
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
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
0D65A 14 16 AB 40 00 00 00 00 ...@.... 0001 0100 0001 0110 1010 1011 0100 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
    
```

**18.1.85 TABLE (varies)**

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	
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	
Cell flags	RC	172	
Cell merged value	B	173	
Autofit flag	B	174	
Merged width flag	BL	175	
Merged height flag	BL	176	
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	
Additional data flag	B		
If additional data 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:			
Additional data flag2	B		
If additional data flag2 == 1:			
Cell flag override	BL	177	
Virtual edge flag	RC	178	
Cell alignment	RS	170	Present only if bit 0x01 is set in cell flag override
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 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 linewt	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
Right grid color	CMC	65	Present only if bit 0x00080 is set in cell flag override
Right grid linewt	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
Bottom grid color	CMC	66	Present only if bit 0x00100 is set in cell flag override
Bottom grid linewt	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
Left grid color	CMC	68	Present only if bit 0x00200 is set in cell flag override
Left grid linewt	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
R2007+:			
Unknown	BL		
Flags	BL	93	Flags & 0x01 => type is kGeneral
Data type	BL	90	
Varies by type:			
0 - kLong	BL		
1 - kDouble	BD		

2 - kString	TV		
4 - kDate			BL data size N, followed by N bytes (Int64 value)
8 - kPoint			BL data size, followed by 2RD
16 - k3dPoint			BL data size, followed by 3RD
32 - kObjectId	H		Read from appropriate place in handles section.
64 - kBuffer			Unknown.
128 - kResBuf			Unknown.
256 - kGeneral			Unknown.
Unknown	BL	94	
Format String	TV	300	
Value String	TV	302	

Common:

End Cell Data (remaining data applies to entire table)

Table overrides pres. B

If table overrides present == 1:

Table flag override	BL	93	
Title suppressed	B	280	Present only if bit 0x0001 is set in table overrides flag
Header suppressed	--	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
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 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 table overrides present == 1

Border overrides pres. B

If border color overrides present == 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 border color overrides present == 1

Border lw. overrides B

If border linewidth overrides present == 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 border linewidth overrides present == 1

Border vis. overrides B

If border visibility overrides present == 1:

Overrides flag	BL	96	Border visibility overrides
Title hor. Top. lw.	BS		Present only if bit 0x01 is set in border visibility overrides flag
Title hor. ins. lw.	BS		Present only if bit 0x02 is set in border visibility overrides flag
Title hor. bot. lw.	BS		Present only if bit 0x04 is set in border visibility overrides flag
Title ver. left. lw.	BS		Present only if bit 0x08 is set in border visibility overrides flag

Title ver. ins. lw.	BS	Present only if bit 0x10 is set in border visibility overrides flag
Title ver. rt. lw.	BS	Present only if bit 0x20 is set in border visibility overrides flag
Header hor. Top. lw.	BS	Present only if bit 0x40 is set in border visibility overrides flag
Header hor. ins. lw.	BS	Present only if bit 0x80 is set in border visibility overrides flag
Header hor. bot. lw.	BS	Present only if bit 0x100 is set in border visibility overrides flag
Header ver. left. lw.	BS	Present only if bit 0x200 is set in border visibility overrides flag
Header ver. ins. lw.	BS	Present only if bit 0x400 is set in border visibility overrides flag
Header ver. rt. lw.	BS	Present only if bit 0x800 is set in border visibility overrides flag
Data hor. Top. lw.	BS	Present only if bit 0x1000 is set in border visibility overrides flag
Data hor. ins. lw.	BS	Present only if bit 0x2000 is set in border visibility overrides flag
Data hor. bot. lw.	BS	Present only if bit 0x4000 is set in border visibility overrides flag
Data ver. left. lw.	BS	Present only if bit 0x8000 is set in border visibility overrides flag
Data ver. ins. lw.	BS	Present only if bit 0x10000 is set in border visibility overrides flag
Data ver. rt. lw.	BS	Present only if bit 0x20000 is set in border visibility overrides flag

End If border visibility overrides present == 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

H Varies 344 for text cell, 340 for block cell

	H	331		Attr. Def. ID (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	---		

### 18.1.86 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

---

**18.1.86.1 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

```



## 19 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.

### 19.1 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.

## 20 Data section AcDb:Handles (OBJECT MAP)

### 20.1 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.

### 20.2 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).

## 21 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,  
0x3E, 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, 0x00
```

## 22 SECOND FILE HEADER (R13-R15)

### 22.1 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) viewport 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}

---

## 23 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
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
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



## 24 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: 01014-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
```

## 25 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 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.

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
  }
}
```

```

    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  
3 RD : x direction  
RL : ?  
PS : string, zero terminated, padded to 4 byte boundary name of font  
RL : ?

---

**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, 0xFFFFFFFFE 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 : Unknown

---

**POP\_CLIP 28**

---

empty

---

**PUSH\_MODELXFORM 29**

---

16 RD : transformation matrix

**PUSH\_MODELXFORM2 30**

---

16 RD : transformation matrix  
? : unknown data

**POP\_MODELXFORM 31**

---

empty



- END OF DOCUMENT -