

DWG file format

An attempt to specify the DWG (R12) file format using the BFF grammar for binary files.

Acknowledgements

I would like to thank Reini Urban <rurban@sbox.tu-graz.ac.at> for his contributions.

Definition of the elementary elements

```
typedef word word :=
  byte : b1, byte : b2
  return (word)f | ((word)s << 8).
typedef longword longp :=
  byte : b1, byte : b2, byte : b3, byte : b4
  return (longword)b1 | ((longword)b2 << 8)
  | ((longword)b3 << 16) | ((longword)b4 << 24).
typedef longword longword :=
  byte : b1, byte : b2, byte : b3, byte : b4
  return (longword)b1 | ((longword)b2 << 8)
  | ((longword)b3 << 16) | ((longword)b4 << 24).
```

Definition of the whole file

```
root dwg_file :=
[begin : end](
  char[12] : version,
  byte, word, word, word, byte,
  longp : p_entities, longp : p_entend,
  longp : p_blocksec, byte[4], longp : p_bsend, byte[4],
  tablepos : block_table,
  tablepos : layer_table,
  tablepos : style_table,
  tablepos : ltype_table,
  tablepos : view_table,
  header, [cur : 0x3EF]byte*,
  tablepos : ucs_table, [cur : 0x500]byte*,
  tablepos : vport_table, byte[8],
  tablepos : appid_table, byte[6],
  tablepos : dimstyle_table, [cur : 0x69F]byte*,
  tablepos : p13_table, bytes[38],
  [p_entities : p_entend]entities : ents, byte[19],
  [block_table.start : ]blocks : block_table,
  [layer_table.start : ]layers : layer_table,
  [style_table.start : ]styles : style_table,
  [ltype_table.start : ]ltypes : ltype_table,
  [view_table.start : ]table : view_table,
  [ucs_table.start : ]table : ucs_table,
  [vport_table.start : vport_table.end]table : vport_table,
  [appid_table.start : ]appids : appid_table,
  [dimstyle_table.start : ]table : dimstyle_table,
  [p13_table.start : ]table : p13_table,
  [p_blocksec : p_bsend]entities : blocks, bytes[36],
  longp = p_entities, longp = p_entend,
  longp = blocksec, longp = bsend,
  bytes[12],
  bytes[6],
  longp = block_table.start, bytes[6],
  longp = layer_table.start, bytes[6],
  longp = style_table.start, bytes[6],
  longp = ltype_table.start, bytes[6],
  longp = view_table.start, bytes[6],
  longp = ucs_table.start, bytes[6],
  longp = vport_table.start, bytes[6],
  longp = appid_table.start, bytes[6],
  longp = dimstyle_table.start, bytes[6],
  longp = p13_table.start, bytes[6],
  longp bytes*,
).
```

A table position

```
tablepos :=
  word : size,
  long : nr,
  long : start,
```

The header

```
header :=
  word,
  point(TRUE) : inbase,
  point(TRUE) : extmin,
  point(TRUE) : extmax,
  point(FALSE) : limmin,
  point(FALSE) : limmax,
  double[4],
  byte[2],
  double[2],
  byte[56],
  double[3],
  byte[18],
  double .
```

The block table

```
blocks :=
  ( [size](
    byte : flag,
    char[32] : name,
    word : used,
    byte, word, byte, word,
    check_2
  )
  ) [nr] : blocks_info,
  check_32.
```

```
check_2 := byte[2].
check_32 := byte[32].
```

The layer table

```
layers :=
  ( [size](
    byte : flag,
    char[32] : name,
    word : used,
    word : color,
    word : style,
    check_2
  )
  ) [nr] : layer_info,
  check_32.
```

The style table

```
styles :=
  ( [size](
    byte : flag,
    char[32] : name,
    word, double[3], byte, double, char[128],
    check_2
  )
  ) [nr] : style_info,
  check_32.
```

The line-type table

```
ltypes :=
  ( [size](
    byte : flag,
    char[32] : name,
    word, char[48], byte,
    byte, double[13],
  )
```

```

        check_2
    )
) [nr] : ltype_info,
check_32 .

```

The application identifier table

```

appids :=
( [size](
    byte : flag,
    char[32] : name,
    word,
    check_2
)
) [nr] : appid_info,
check_32 .

```

The other tables

```

table :=
( [size](
    byte : flag,
    [size - 3]byte*,
    check_2
)
) [nr],
check_32 .

```

The entities

(Experimental)

```

entities :=
( byte : kind,
  byte : flag,
  word : length,
  [length - 4](
    word : layer,
    word : opts,
    if (flag & 1) then byte : color else color = 0 fi,
    if (flag & 0x40) then byte : extra else extra = 0 fi,
    if (extra & 2) then xdata fi,
    if (flag & 2) then word : type fi,
    if (flag & 4 && kind > 2 && kind != 22) then double : z fi,
    if (flag & 8) then double : th fi,
    if (flag & 0x20) then handle fi,
    if (extra & 4) then word : paper fi,
    switch (kind)
    case 1: /* LINE */
      point(!(flag & 4)) : l10,
      point(!(flag & 4)) : l11,
      if (opts & 1) then point(TRUE) : l210 fi,
      if (opts & 2) then double : l38 fi,
    case 2: /* POINT */
      point(!(flag & 4)) : l10,
      if (opts & 1) then point(TRUE) : l210 fi,
      if (opts & 2) then double : l38 fi,
    case 3: /* CIRCLE */
      point(FALSE) : l10,
      double : l40,
      if (opts & 1) then point(TRUE) : l210 fi,
      if (opts & 2) then double : l38 fi,
    case 4: /* SHAPE */
      point(FALSE) : l10,
      word : l2,
      if (opts & 1) then point(TRUE) : l210 fi,
      if (opts & 2) then double : l38 fi,
    case 7: /* TEXT */
      point(FALSE) : l10,
      double : l40,
      string : l1,
      if (opts & 1) then double : l50 fi,
      if (opts & 2) then double : l41 fi,

```

```

if (opts & 4) then double : 151 fi,           /*?*/
if (opts & 8) then byte : 17 fi,
if (opts & 0x10) then byte : 171 fi,
if (opts & 0x20) then byte : 172 fi,
if (opts & 0x40) then point(FALSE) : 111 fi,
if (opts & 0x100) then byte : 173 fi,
case 8: /* ARC */
point(FALSE) : 110,
double : 140,
double : 150,
double : 151,
if (opts & 1) then point(TRUE) : 1210 fi,
if (opts & 2) then double : 138 fi,
case 9: /* TRACE */
point(FALSE) : 110,
point(FALSE) : 111,
point(FALSE) : 112,
point(FALSE) : 113,
if (opts & 1) then point(TRUE) : 1210 fi,
if (opts & 2) then double : 138 fi,
case 11: /* SOLID */
point(FALSE) : 111,
point(FALSE) : 112,
point(FALSE) : 113,
point(FALSE) : 114,
if (opts & 1) then point(TRUE) : 1210 fi,
if (opts & 2) then double : 138 fi
case 12: /* BLOCK */
point(FALSE) : 110,           /*?*/
string : 11,                 /* if (opts & 1) then ? */
if (opts & 2) then string : 13 fi
case 13: /* ENDBLK */
case 14: /* INSERT */
word : 11,
point(FALSE) : 110,
if (opts & 1) then double : 141 fi,
if (opts & 2) then double : 142 fi,
if (opts & 4) then double : 143 fi,
if (opts & 8) then double : 150 fi,
if (opts & 0x10) then byte : 170 fi,           /*?*/
if (opts & 0x20) then byte : 171 fi,         /*?*/
if (opts & 0x40) then double : 144 fi,       /*?*/
if (opts & 0x80) then double : 145 fi        /*?*/
case 15: /* ATTDEF */
point(FALSE) : 110,
double : 140,
string : 11,
string : 13,
string : 12,
byte : 170,
if (opts & 1) then byte : 173 fi,           /*?*/
if (opts & 2) then double : 150 fi,         /*?*/
if (opts & 4) then double : 141 fi,
if (opts & 8) then double : 142 fi,
if (opts & 0x10) then byte : 17 fi,
if (opts & 0x20) then byte : 171 fi,
if (opts & 0x40) then byte : 172 fi,
if (opts & 0x80) then point(FALSE) : 111 fi, /*?*/
if (opts & 0x100) then point(TRUE) : 1210 fi,
if (opts & 0x200) then double : 138 fi      /*?*/
case 16: /* ATTRIB */
point(FALSE) : 110,
double : 140,
string : 11,
string : 12,
byte : 170,
if (opts & 1) then byte : 173 fi,           /*?*/
if (opts & 2) then double : 150 fi,         /*?*/
if (opts & 4) then double : 141 fi,
if (opts & 8) then double : 142 fi,
if (opts & 0x10) then byte : 17 fi,
if (opts & 0x20) then byte : 171 fi,
if (opts & 0x40) then byte : 172 fi,

```

```

    if (opts & 0x80) then point(FALSE) : 111 fi, /*?*/
    if (opts & 0x100) then point(TRUE) : 1210 fi,
    if (opts & 0x200) then double : 138 fi /*?*/
case 17: /* S/BEND */
    long
case 19: /* PLINE */
    if (opts & 1) then byte : 170 fi,
    if (opts & 2) then double : 140 fi, /*?*/
    if (opts & 4) then byte : 171 fi, /*?*/
    if (opts & 8) then byte : 172 fi, /*?*/
    if (opts & 0x10) then byte : 173 fi, /*?*/
    if (opts & 0x20) then byte : 174 fi, /*?*/
    if (opts & 0x40) then byte : 175 fi /*?*/
case 20: /* VERTEX */
    point(FALSE) : 110,
    if (opts & 1) then double : 140 fi, /*?*/
    if (opts & 2) then double : 141 fi, /*?*/
    if (opts & 4) then byte : 170 fi, /*?*/
    if (opts & 8) then double : 150 fi /*?*/
case 22: /* 3DFACE */
    point(!(flag & 4)) : 110,
    point(!(flag & 4)) : 111,
    point(!(flag & 4)) : 112,
    point(!(flag & 4)) : 113
case 23: /* DIM */
    word : 11,
    point(TRUE) : 110,
    point(FALSE) : 111, /*?*/
    if (opts & 2) then byte : 170 fi,
    if (opts & 1) then point(TRUE) : 112 fi, /*?*/
    if (opts & 4) then string : 11 fi,
    if (opts & 8) then point(TRUE) : 113 fi,
    if (opts & 0x10) then point(TRUE) : 114 fi,
    if (opts & 0x20) then point(TRUE) : 115 fi,
    if (opts & 0x40) then point(TRUE) : 116 fi,
    if (opts & 0x80) then double : 140 fi,
    if (opts & 0x100) then double : 150 fi,
    if (opts & 0x200) then double : 151 fi,
    if (opts & 0x400) then double : 152 fi,
    if (opts & 0x800) then double : 153 fi
case 24: /* VPORT */
    point(TRUE) : 110,
    double : 140,
    double : 141,
    word : 168
endswitch
check_2
)
)* : entities.

```

Still need to define xdata and handle. (to be continued...)

Last updated: Tuesday, 09-Jan-96 20:21:36 MET