Devices

COLLABORATORS			
	TITLE : Devices		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 23, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Devices

1.1 A / Additional Documents

```
            Intro to IFF Amiga ILBM Files and Amiga Viewmodes

The IFF (Interchange File Format) for graphic images on the Amiga is
called FORM ILBM (InterLeaved BitMap).  It follows a standard parsable IFF
format.

 Sample hex dump of beginning of an ILBM
 Interpreting ILBMs
 Body Compression
 Interpreting the Scan Line Data
 How Amiga HAM mode works
 How Amiga HALFBRITE mode works
 Other Notes
```

1.2 Additional Info / Sample hex dump of beginning of an ILBM

```
Important note!  You can NOT ever depend on any particular ILBM chunk
being at any particular offset into the file!  IFF files are composed, in
their simplest form, of chunks within a FORM.  Each chunk starts starts
with a 4-letter chunkID, followed by a 32-bit length of the rest of the
chunk.  You PARSE IFF files, skipping past unneeded or unknown chunks by
seeking their length (+1 if odd length) to the next 4-letter chunkID.

0000: 464F524D 00016418 494C424D 424D4844    FORM..d.ILBMBMHD
0010: 00000014 01400190 00000000 06000100    .....@..........
0020: 00000A0B 01400190 43414D47 00000004    .....@..CAMG....
0030: 00000804 434D4150 00000030 001100EE    ....CMAP...0....
0040: EEEE0000 22000055 33333355 55550033    .... ..P000PPP.0
0050: 99885544 77777711 66EE2266 EE6688DD    ..P@ppp.`. `.`..
0060: AAAAAAAA 99EECCCC CCDDAAEE 424F4459    ............BODY
0070: 000163AC F8000F80 148A5544 2ABDEFFF    ..c.......UD*...  etc.

Interpretation:

      'F O R M' length  'I L B M''B M H D'<-start of BitMapHeader chunk
```

```
0000: 464F524D 00016418 494C424D 424D4844    FORM..d.ILBMBMHD

       length  WideHigh XorgYorg PlMkCoPd <- Planes Mask Compression Pad
0010: 00000014 01400190 00000000 06000100    .....@..........

                                             start of C-AMiGa
      TranAspt PagwPagh 'C A M G' length  <- View modes chunk
0020: 00000A0B 01400190 43414D47 00000004    .....@..CAMG....

      Viewmode 'C M A P' length  R g b R  <- Viewmode 800=HAM | 4=LACE
0030: 00000804 434D4150 00000030 001100EE    ....CMAP...0....

      g b R g  b R g b  R g b R  g b R g  <- Rgb's are for reg0 thru regN
0040: EEEE0000 22000055 33333355 55550033    .... ..P000PPP.0

      b R g b  R g b R  g b R g  b R g b
0050: 99885544 77777711 66EE2266 EE6688DD    ..P@ppp.`. `.`..

      R g b R  g b R g  b R g b 'B O D Y'
0060: AAAAAAAA 99EECCCC CCDDAAEE 424F4459    ............BODY

                                             Compacted
       length   start of body data        <- (Compression=1 above)
0070: 000163AC F8000F80 148A5544 2ABDEFFF    ..c.......UD*...
0080: FFBFF800 0F7FF7FC FF04F85A 77AD5DFE    ...........Zw.].  etc.

Notes on CAMG Viewmodes:  HIRES=0x8000  LACE=0x4  HAM=0x800  HALFBRITE=0x80
```

1.3 Additional Info / Interpreting ILBMs

```
ILBM is a fairly simple IFF FORM.  All you really need to deal with to
extract the image are the following chunks:

(Note - Also watch for AUTH Author chunks and (c) Copyright chunks
 and preserve any copyright information if you rewrite the ILBM)

   BMHD - info about the size, depth, compaction method
          (See interpreted hex dump above)

   CAMG - optional Amiga viewmodes chunk
          Most HAM and HALFBRITE ILBMs should have this chunk.  If no
          CAMG chunk is present, and image is 6 planes deep, assume
          HAM and you'll probably be right.  Some Amiga viewmodes
          flags are HIRES=0x8000, LACE=0x4, HAM=0x800, HALFBRITE=0x80.
    Note that new Amiga 2.0 ILBMs may have more complex 32-bit
    numbers (modeid) stored in the CAMG.  However, the bits
    described above should get you a compatible old viewmode.

   CMAP - RGB values for color registers 0 to n
          (each component left justified in a byte)
    If a deep ILBM (like 12 or 24 planes), there should be no CMAP
    and instead the BODY planes are interpreted as the bits of RGB
    in the order R0...Rn G0...Gn B0...Bn

   BODY - The pixel data, stored in an interleaved fashion as follows:
```

```
(each line individually compacted if BMHD Compression = 1)
   plane 0 scan line 0
   plane 1 scan line 0
   plane 2 scan line 0
   ...
   plane n scan line 0
   plane 0 scan line 1
   plane 1 scan line 1
   etc.
```

1.4 Additional Info / Body Compression

The BODY contains pixel data for the image. Width, Height, and depth (Planes) is specified in the BMHD.

If the BMHD Compression byte is 0, then the scan line data is not compressed. If Compression=1, then each scan line is individually compressed as follows:

More than 2 bytes the same stored as BYTE code value n from -1 to -127
  followed by byte to be repeated (-n) + 1 times.
Varied bytes stored as BYTE code n from 0 to 127 followed by n+1 bytes of data.

The byte code -128 is a NOP.

1.5 Additional Info / Interpreting the Scan Line Data

If the ILBM is not HAM or HALFBRITE, then after parsing and uncompacting if necessary, you will have N planes of pixel data. Color register used for each pixel is specified by looking at each pixel thru the planes. I.e., if you have 5 planes, and the bit for a particular pixel is set in planes 0 and 3:

```
PLANE     4 3 2 1 0
PIXEL     0 1 0 0 1
```

then that pixel uses color register binary 01001 = 9

The RGB value for each color register is stored in the CMAP chunk of the ILBM, starting with register 0, with each register's RGB value stored as one byte of R, one byte G, and one byte of B, with each component scaled to 8-bits. (ie. 4-bit Amiga R, G, and B components are each stored in the high nibble of a byte. The low nibble may also contain valid data if the color was stored with 8-bit-per-gun color resolution).

BUT - if the picture is HAM or HALFBRITE, it is interpreted differently.
                        ===   =========

Hopefully, if the picture is HAM or HALFBRITE, the package that saved it properly saved a CAMG chunk (look at a hex dump of your file with ACSII

interpretation - you will see the chunks - they all start with a 4-ASCII-character chunk ID). If the picture is 6 planes deep and has no CAMG chunk, it is probably HAM. If you see a CAMG chunk, the "CAMG" is followed by the 32-bit chunk length, and then the 32-bit Amiga Viewmode flags.

HAM pics with a 16-bit CAMG will have the 0x800 bit set in CAMG ViewModes. HALBRITE pics will have the 0x80 bit set.

To transport a HAM or HALFBRITE picture to another machine, you must understand how HAM and HALFBRITE work on the Amiga.

1.6 Additional Info / How Amiga HAM mode works

Amiga HAM (Hold and Modify) mode lets the Amiga display all 4096 RGB values. In HAM mode, the bits in the two last planes describe an R G or B modification to the color of the previous pixel on the line to create the color of the current pixel. So a 6-plane HAM picture has 4 planes for specifying absolute color pixels giving up to 16 absolute colors which would be specified in the ILBM CMAP chunk. The bits in the last two planes are color modification bits which cause the Amiga, in HAM mode, to take the RGB value of the previous pixel (Hold and), substitute the 4 bits in planes 0-3 for the previous color's R G or B component (Modify) and display the result for the current pixel. If the first pixel of a scan line is a modification pixel, it modifies the RGB value of the border color (register 0). The color modification bits in the last two planes (planes 4 and 5) are interpreted as follows:

```
00 - no modification.  Use planes 0-3 as normal color register index
10 - hold previous, replacing Blue component with bits from planes 0-3
01 - hold previous, replacing Red component with bits from planes 0-3
11 - hold previous. replacing Green component with bits from planes 0-3
```

1.7 Additional Info / How Amiga HALFBRITE mode works

This one is simpler. In HALFBRITE mode, the Amiga interprets the bit in the last plane as HALFBRITE modification. The bits in the other planes are treated as normal color register numbers (RGB values for each color register is specified in the CMAP chunk). If the bit in the last plane is set (1), then that pixel is displayed at half brightness. This can provide up to 64 absolute colors.

1.8 Additional Info / Other Notes

Amiga ILBMs images must be a even number of bytes wide. Smaller images (such as brushes) are padded to an even byte width.

ILBMs created with Electronic Arts IBM and Amiga "DPaintII" packages are compatible (though you may have to use a '.lbm' filename extension on an IBM). The ILBM graphic files may be transferred between the machines (or

between the Amiga and IBM sides your Amiga if you have a CBM Bridgeboard
card installed) and loaded into either package.