dctv

COLLABORATORS			
	TITLE : dctv		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		April 16, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

dctv

1.1 dctv.guide

```
                AllocDCTVCvt()

                AllocDCTVCvtTagList()

                AllocDCTVCvtTags()

                CvtDCTVLine()

                DCTVCvtLine()

                DCTVFormat()

                FormatDCTV()

                FreeDCTVCvt()

                InitDCTVCvt()

                ReadDCTVPixel()

                SetDCTVColorTable()

                TestDCTVSignature()
```

1.2 dctv.library/AllocDCTVCvt

```
                NAME
  AllocDCTVCvt -- Old form of AllocDCTVCvtTagList().

SYNOPSIS
  struct DCTVCvtHandle *AllocDCTVCvt (struct BitMap *BitMap,
                  a0

      ULONG Width, ULONG Height, ULONG Flags)
```

```
            d0     d1    d2

FUNCTION
  This is the V1 compatible form of
               AllocDCTVCvtTagList()
               .  It offers
  only a subset of the features of AllocDCTVCvtTagList():
  . RGB->DCTV conversion only.
  . No extended error code.

  This function is equivalent to:

  AllocDCTVCvtTags ( BitMap,
         DCTVCVTA_Width,  Width,
         DCTVCVTA_Height, Height,
         DCTVCVTA_Flags,  Flags,
         TAG_END );

  This function is considered obsolete, but is still present in the
  library for backwards compatibility.

INPUTS
  BitMap - BitMap arg for AllocDCTVCvtTagList().

  Width,
  Height,
  Flags - Values for DCTVCVTA_Width, DCTVCVTA_Height, and DCTVCVTA_Flags
  respectively.

RESULTS
  Pointer to initialized DCTVCvtHandle for use with other conversion
  functions.  NULL on failure.

SEE ALSO

               AllocDCTVCvtTagList()
               ,
               FreeDCTVCvt()
               , libraries/dctv.h
```

1.3 dctv.library/AllocDCTVCvtTagList

```
               NAME
  AllocDCTVCvtTagList -- Allocate a conversion environment. (V3)

SYNOPSIS
  struct DCTVCvtHandle *AllocDCTVCvtTagList
      (struct BitMap *BitMap, struct TagItem *TagList)
          a0         a1

FUNCTION
  Allocates and initializes a DCTV conversion environment.  The following
  types of conversions are supported:
  . 24-bit RGB to DCTV BitMap
  . DCTV BitMap to 24-bit RGB
```

```
In both cases the 24-bit RGB data is expected to be in 70ns (HIRES)
pixels, interlace or non-interlace.  The RGB and DCTV pixels correspond
  1:1.  No image scaling is performed.

The supplied taglist desribes the conversion type, image dimensions,
and a variety of other parameters.

There are two methods of handling DCTV BitMaps:  full-screen (default)
and 1-line.  A full-screen BitMap contains the entire DCTV display
image (either as destination or source depending on conversion type).
Successive calls to
              CvtDCTVLine()
               process successive lines in the
BitMap.

A 1-line BitMap (DCTVCVTF_1LineBM set) contains only 1 line of the
DCTV BitMap instead of the full image.  This sort of BitMap is not
displayable, but is useful as a file buffer.  It also requires less
memory than a full-screen BitMap.  In this case, each call to

              CvtDCTVLine()
               uses the same line in the BitMap.

RGB data is handled in the form of a 24-bit RGB line buffer in the
form of 3 UBYTE arrays in the conversion environment: one for each
component (Red, Green, and Blue).  The RGB data is in chunk pixel
format instead of a BitMap.  These arrays are used to buffer 1 line
of RGB data and must be read or written to (depending on conversion
type) between calls to CvtDCTVLine().

Normally AllocDCTVCvtTagList() allocates the 24-bit RGB line buffer.
When DCTVCVTF_CustomRGBBuf is set, the client is allowed to supply
the line buffer instead.  This is useful if your RGB buffer is
taller than 1 line since this mode allows changing the RGB buffer
pointers between calls to
              CvtDCTVLine()
              .  This also eliminates copying
to and from the conversion 24-bit line buffer.

Note that the buffer pointers in the DCTVCvtHandle may only be changed
if DCTVCVTF_CustomRGBBuf is set.  These pointers are read-only when
this bit isn't set.

A special case exists in which DCTVCVTF_CustomRGBBuf may be specified
and the buffers left NULL.  This is legal only when using
DCTVCVTT_DCTVtoRGB conversion mode and only calling ReadDCTVPixel()
  (not
              CvtDCTVLine()
              ) because
              ReadDCTVPixel()
               does not access the RGB
line buffers.

The actual conversion process consists of calling CvtDCTVLine()
for each line to be converted.
```

```
  The line number values SrcLineNum, DstLineNum and NDelayLines are used
  to determine which line is to be input and output by the next call to

                CvtDCTVLine()
                .  NDelayLines indicates how many lines of source data
  are required to be buffered in order to generate the first line of
  output data.  The actual value of NDelayLines depends on the
  conversion type and flags and may change from one library version to
  another.

  The following relationship is always true:
  SrcLineNum - DstLineNum == NDelayLines.

  Usually
                CvtDCTVLine()
                 must be called a total of Height + NDelayLines
  times to completely generate the output image.

  A conversion environment can only be used for the type of conversion
  specified in DCTVCVTA_Type.  A single conversion environment can be
  used to process multiple frames sequentially by calling InitDCTVCvt()
  between frames.  The only restriction is that each frame must use the
  same BitMap and be the same dimensions.

  Call
                FreeDCTVCvt()
                 to dispose of conversion environment.

INPUTS
  BitMap - Pointer to an initialized BitMap structure with the following
  dimension limits:

      Width >= 256
      Height >= 2 for non-lace, 3 for lace
      Depth >= 3

  The Planes[] data need not be in chip ram for conversion (but
  obviously do if you expect to display the BitMap).

  Both interleaved and non-interleaved bitmaps are supported.

  For full-screen mode, the BitMap must be large enough to hold the
  entire conversion as specified by DCTVCVTA_Width and
  DCTVCVTA_Height.  For 1-line mode, the BitMap needs to have at
  least one line of data wide enough to hold an entire line as
  specified by DCTVCVTA_Width.

  The supplied BitMap structure and Planes[] data must remain valid
  until
                FreeDCTVCvt()
                 is called.

  TagList - Pointer to TagItem array with Tags described below.

TAGS
  DCTVCVTA_Type     (UWORD)
  DCTVCVTT_RGBtoDCTV - Convert 24-bit RGB chunky pixels to DCTV a
```

display BitMap. This is the default conversion type.

DCTVCVTT_DCTVtoRGB - Convert a DCTV display BitMap to 24-bit RGB chunky pixels.

DCTVCVTA_Width (WORD)
Pixel width of image to convert in 70ns (HIRES) pixels. Must be multiple of 16 in the range of 256 .. BitMap width. The practical maximum width is 736 (the maximum display width). Non-multiples of 16 are truncated.

The default value is derived from the BitMap.

It's safest to always include this tag when using a 1-line BitMap since there is some ambiguity as to what BitMap->BytesPerRow actually means for a 1-line BitMap.

DCTVCVTA_Height (WORD)
Pixel height of image to convert.

For full-screen mode, must be in the range of 2 for non-lace (3 for lace) .. BitMap height. For 1-line mode, must be at least 2 for non-lace, 3 for lace.

The default value is derived from the BitMap.

It's safest to always include this tag when using a 1-line BitMap since there is some ambiguity as to what BitMap->Rows actually means for a 1-line BitMap.

DCTVCVTA_Flags (UWORD)
DCTVCVTF_Lace - Interlaced DCTV display BitMap.

DCTVCVTF_Filter - Apply RGB filtering. Only observed for DCTVCTVT_RGBtoDCTV.

DCTVCVTF_1LineBM - Treat BitMap as 1-line instead of full-screen.

DCTVCVTF_CustomRGBBuf (V3) - Allow client to supply a custom RGB line buffer. Normally this function allocates the RGB line buffer.

Unknown flags are ignored.

DCTVCVTA_ColorTable (UWORD *)
Initial palette associated with BitMap for DCTVCVTT_DCTVtoRGB conversion. Must contain 1 << BitMap->Depth entries. This array must remain valid until it is replaced with SetDCTVColorTAble() or the conversion environment is freed.

Mid-BitMap palette changes, if any, can be set with

```
SetDCTVColorTable()
.
```

This tag is required for DCTVCVTT_DCTVtoRGB in order to be able to decode the bits in the BitMap. Ignored for DCTVCVTT_RGBtoDCTV.

```
  DCTVCVTA_ErrorCode  (ULONG *)
  Pointer to a ULONG where an extended error code is placed if
  AllocDCTVCvtTagList() fails.  The buffer pointed to by this
  tag is modified only on an error.

  Errors codes:

  DCTVCVTERR_NoMem - Not enough memory.

  DCTVCVTERR_BadBitMap - Supplied BitMap is not valid.  Could be
      invalid dimensions or depth.

  DCTVCVTERR_UnknownType - Unknown conversion type passed to
      DCTVCVTA_Type.

  DCTVCVTERR_BadParams - Invalid parameters provided for
      conversion.

RESULTS
  Pointer to initialized DCTVCvtHandle for use with other conversion
  functions.  NULL on failure.  Can fail if out of memory or supplied
  parameters are not legal.

  On success the DCTVCvtHandle structure is initialized as follows:
    (see libraries/dctv.h for a more complete explanation of each field)

  Red,
  Green,
  Blue - If DCTVCVTF_CustomRGBBuf isn't set, these point to allocated
      24-bit RGB line buffer, Width bytes wide.  The buffers are
      initialized to zero.

      If DCTVCVTF_CustomRGBBuf is set, these are set to NULL and
      the client is expected to set these pointers prior to calling

                CvtDCTVLine()
                .

  BitMap - BitMap pointer passed to AllocDCTVCvtTagList().

  Width,
  Height - Values from DCTVCVTA_Width and DCTVCVTA_Height or derived
      from BitMap.

  ImageBounds - Computed from Width, Height and Modes.

  ColorTable,
  NColors - Palette information for BitMap Depth.  For
      DCTVCVTT_RGBtoDCTV this is the palette supplied by dctv.library
      for the output BitMap.  For DCTVCVTT_DCTVtoRGB, this is the
      array supplied by DCTVCVTA_ColorTable.

  NDelayLines - Initialized depending on type and flags.

  SrcLineNum - Initialized to 0 (as set by
                InitDCTVCvt()
```

```
                ).

  DstLineNum - Initialized to -NDelayLines (as set by
                InitDCTVCvt()
                ).

NOTE
  V1
                AllocDCTVCvt()
                 was documented as tolerating a BitMap with the
  BitMap->Planes[] array not initialized.  This practice is not
  recommended and will probably break in the future.

SEE ALSO

                AllocDCTVCvtTags()
                ,
                FreeDCTVCvt()
                ,
                CvtDCTVLine()
                , libraries/dctv.h
```

1.4 dctv.library/AllocDCTVCvtTags

```
                NAME
  AllocDCTVCvtTags -- Varargs version of
                AllocDCTVCvtTagList()
                . (V3)

SYNOPSIS
  struct DCTVCvtHandle *AllocDCTVCvtTagList
      (struct BitMap *BitMap, Tag tag1, ...)

FUNCTION
  Varargs version of AllocCvtTagList().  See
                AllocDCTVCvtTagList()
                 for
  complete description.

INPUTS
  BitMap - BitMap arg for AllocDCTVCvtTagList().
  Tag1... - TagItem array on the stack to be passed to

                AllocDCTVCvtTagList()
                .

RESULTS
  Pointer to initialized DCTVCvtHandle for use with other conversion
  functions.  NULL on failure.

NOTE
    16-bit integer users:  be careful to cast integer tag data to 32-bits.
  For example:

  AllocDCTVCvtTags ( bm,
```

```
    DCTVCVTA_Type,   (ULONG)DCVTCVTT_RGBtoDCTV,
    DCTVCVTA_Width,  (ULONG)width,
    DCTVCVTA_Height, (ULONG)height,
    DCTVCVTA_Flags,  (ULONG)DCTVCVTF_Filter | DCTVCVTF_Lace,
    TAG_END );

SEE ALSO

                AllocDCTVCvtTagList()
                ,
                FreeDCTVCvt()
                ,
                CvtDCTVLine()
                , libraries/dctv.h
```

1.5 dctv.library/CvtDCTVLine

```
                NAME
  CvtDCTVLine -- Convert a line.

SYNOPSIS
  void CvtDCTVLine (struct DCTVCvtHandle *Handle)
           a0

FUNCTION
  Converts a line as described below.

  NOTE: All references to SrcLineNum and DstLineNum values below are
  the values before calling CvtDCTVLine().

  DCTVCVTT_RGBtoDCTV - Converts the RGB data in the Handle's RGB line
  buffers to DCTV.  Assumes that the buffers contain data for
  SrcLineNum and will generate the BitMap line indexed by DstLineNum.

  The source RGB line buffer is only read when SrcLineNum is in the
  vertical range of ImageBounds.  Otherwise it is ignored.  Pixels
  outside the range of ImageBounds are ignored.

  Destination BitMap data is only emitted for values of DstLineNum
  in the range of 0..Height-1.  The BitMap is not affected for
  DstLineNum values outside this range.

  The contents of the RGB line buffer are not affected by this
  function.

  DCTVCVTT_DCTVtoRGB - Converts the DCTV display BitMap line indexed by
  SrcLineNum to the RGB line buffer.  The resulting RGB data is for
  DstLineNum.

  Source BitMap data is only read for SrcLineNum values in the
  vertical range of ImageBounds.

  Destination RGB data is only emitted for DstLineNum values in the
  range of 0..Height-1.  Pixels outside the ImageBounds rectangle
  are emitted as black.
```

```
  The BitMap is not affected by this function.

  SrcLineNum and DstLineNum are incremented after processing the line.

  CvtDCTVLine() should be called until DstLineNum >= Height in order
  to convert the entire image (see example below) when NDelayLines
  is non-zero.  Be careful to avoid overindexing your buffers.

INPUTS
  Handle - DCTVCvtHandle returned by
               AllocDCTVCvtTagList()
               .  Note to
  DCTVCVTF_CustomRGBBuf users:  the RGB line buffer pointers must
  point to a valid buffer by the time of this call (i.e. non-NULL)
  for line number ranges that would access the RGB line buffer
  (see above for when this happens).

RESULTS
  Outputs BitMap or RGB data (depending on conversion type) for line
  DstLineNum when DstLineNum is in the range of 0 <= DstLineNum < Height
  prior to call.

  Increments SrcLineNum and DstLineNum.

EXAMPLE
  while (cvt->DstLineNum < cvt->Height) {
  renderline (cvt);           /* Render a line of the gradient to
              DCTVCvtHandle RGB line buffer.
              This function must be smart enough
              to deal w/ SrcLineNum >= Height. */

  CvtDCTVLine (cvt);          /* Convert the line in the line buffer
              to DCTV display data. Results in
              rendering a line of the BitMap. */
    }

SEE ALSO

               AllocDCTVCvtTagList()
               ,
               InitDCTVCvt()
               , libraries/dctv.h
```

1.6 dctv.library/DCTVCvtLine

```
               NAME
  DCTVCvtLine -- Old name for CvtDCTVLine().

SYNOPSIS
  void DCTVCvtLine (struct DCTVCvtHandle *Handle)
           a0

FUNCTION
  V1 name for
```

```
                CvtDCTVLine()
                .  This is the same library vector with a new
  name so it is completely run-time compatible with old software and V1
  of dctv.library.

  There is a #define in clib/dctv_protos.h to translate the old name to
  the new name.  Also dctv.lib contains glue functions and LVOs for both
  names.

  Use
                CvtDCTVLine()
                 from now on.

SEE ALSO
  CvtDCTVLine()
```

1.7 dctv.library/DCTVFormat

```
                NAME
  DCTVFormat -- Old name for FormatDCTV().

SYNOPSIS
  void DCTVFormat (struct DCTVCvtHandle *Handle)
           a0

FUNCTION
  V1 name for
                FormatDCTV()
                .  This is the same library vector with a new
  name so it is completely run-time compatible with old software and V1
  of dctv.library.

  There is a #define in clib/dctv_protos.h to translate the old name to
  the new name.  Also dctv.lib contains glue functions and LVOs for both
  names.

  Use
                FormatDCTV()
                 from now on.

SEE ALSO
  FormatDCTV()
```

1.8 dctv.library/FormatDCTV

```
NAME
  FormatDCTV -- Format BitMap for DCTV display.

SYNOPSIS
  void FormatDCTV (struct DCTVCvtHandle *Handle)
           a0
```

```
FUNCTION
  Formats the Handle's BitMap for a black DCTV display.  This is useful
  if you wish to display the conversion process while in progress.

  Only use with a full-screen BitMap in a DCTVCVTT_RGBtoDCTV conversion
  environment.

INPUTS
  Handle - DCTVCvtHandle returned by AllocDCTVCvtTagList().

RESULTS
  Formatted BitMap.

SEE ALSO
  AllocDCTVCvtTagList()
```

1.9 dctv.library/FreeDCTVCvt

```
NAME
  FreeDCTVCvt -- Free conversion environment.

SYNOPSIS
  void FreeDCTVCvt (struct DCTVCvtHandle *Handle)
           a0

FUNCTION
  Frees a DCTVCvtHandle allocated by AllocDCTVCvtTagList().

INPUTS
  Handle - DCTVCvtHandle returned by AllocDCTVCvtTagList().

RESULTS
  None.

SEE ALSO
  AllocDCTVCvtTagList()
```

1.10 dctv.library/InitDCTVCvt

```
                NAME
  InitDCTVCvt -- Init conversion environment for frame.

SYNOPSIS
  void InitDCTVCvt (struct DCTVCvtHandle *Handle)
           a0

FUNCTION
  Initializes line numbers for a new frame (image).  This is done
  automatically by
                AllocDCTVCvtTagList()
                .  Use it if you need to process
  multiple frames using the same DCTVCvtHandle.
```

```
  This function does not initialize the RGB line buffer data or BitMap
  data in any way.

  Also, there is no provision to change image dimensions, BitMap, or any
  other parameter of a conversion environment.

INPUTS
  Handle - DCTVCvtHandle returned by AllocDCTVCvtTagList().

RESULTS
  DCTVCvtHandle fields are initialized as follows:
  Handle->SrcLineNum = 0
  Handle->DstLineNum = -Handle->NDelayLines

SEE ALSO

                AllocDCTVCvtTagList()
                , FormatDCTV()
```

1.11 dctv.library/ReadDCTVPixel

```
                NAME
  ReadDCTVPixel -- Get 24-bit RGB value of a pixel from a DCTV display
         BitMap. (V3)

SYNOPSIS
  ULONG ReadDCTVPixel (struct DCTVCvtHandle *Handle, LONG x, LONG y)
              a0     d0  d1

FUNCTION
  Returns 24-bit RGB value for the specified pixel in a
  DCTVCVTT_DCTVtoRGB environment in the following format 32-bit format:

  xxxxxxxx rrrrrrrr gggggggg bbbbbbbb

  where the most significant byte value is undefined.

  Returns 0 for pixels outside of ImageBounds.

INPUTS
  Handle - DCTVCvtHandle returned by
                AllocDCTVCvtTagList()
                .  The RGB
  line buffers are not used by this function and may be NULL
  if DCTVCVTF_CustomRGBBuf was specified.
  x, y - Pixel to examine using the coordinate system of the BitMap.

RESULTS
    24-bit right-justifed RGB value or 0.

NOTE
  For the sake of speed, this function does less horizontal filtering
  than
                CvtDCTVLine()
```

```
                and therefore yields slightly different results.
  The most visible difference is that sharp vertical edges in the DCTV
  image get a bit "zippered" when converted to RGB using
  ReadDCTVPixel().
               CvtDCTVLine()
                typically makes smoother RGB images.

SEE ALSO

               AllocDCTVCvtTagList()
               ,
               CvtDCTVLine()
               , libraries/dctv.h
```

1.12 dctv.library/SetDCTVColorTable

```
               NAME
  SetDCTVColorTable -- Set mid-BitMap palette change for DCTV->RGB
       conversion. (V3)

SYNOPSIS
  void SetDCTVColorTable (struct DCTVCvtHandle *Handle,
            a0

           UWORD *ColorTable)
              a1

FUNCTION
  Replaces the Handle's ColorTable for DCTVCVTT_DCTVtoRGB conversions.
  This can be used to supply a mid-screen palette change for the source
  DCTV display BitMap.  This function is of limited usefulness, but is
  essential for a couple of things:
  . converting the DCTVPaint control panel to 24-bit RGB.
  . handling PCHG chunks in a DCTV display ILBM.

INPUTS
  Handle - DCTVCvtHandle returned by AllocDCTVCvtTagList().

  ColorTable - RGB4 UWORD array containing NColors (1<<BitMap->Depth)
  entries.  This array must remain valid until it is replaced with
  another call to SetDCTVColorTable() or the conversion environment
  is freed.

RESULTS
  Updates Handle->ColorTable.

NOTE
  This function is ignored for DCTVCVTT_RGBtoDCTV.

SEE ALSO

               AllocDCTVCvtTagList()
               , libraries/dctv.h
```

1.13 dctv.library/TestDCTVSignature

```
NAME
  TestDCTVSignature -- Check a BitMap for DCTV signature.

SYNOPSIS
  BOOL TestDCTVSignature (struct BitMap *BitMap)
                a0

FUNCTION
  Checks the top line of the supplied BitMap for the DCTV signature.
  This signature must appear at the top of all DCTV pictures.  This
  function provides a way of determining if a picture, possibly loaded
  from an ILBM, is a DCTV display picture.

INPUTS
  BitMap - BitMap to check.  Planes[] data does not need to be in
  chip memory for this function.

RESULTS
  TRUE (non-zero) if the top line of the BitMap contains a DCTV
  signature.  FALSE otherwise.

SEE ALSO
```