opendivx

COLLABORATORS			
	TITLE : opendivx		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 31, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

opendivx

1.1 opendivx.doc

```
    DIVX_Decode68k()
    DIVX_DecodePPC()
    DIVX_Encode68k()
    DIVX_EncodePPC()
```

1.2 opendivx.library/DIVX_Decode68k

```
  NAME
DIVX_Decode68k -- Decode a OpenDivx using an 68k Function

  SYNOPSIS
result = DIVX_Decode68k(handle, dec_opt, param1, param2)
D0                      d0      D1       A0      A1



  FUNCTION
This function decodes a frame of a OpenDivx. It also can be used to
initialize or quit the replay, or to set the postprocessing level.
If called on a 68k machine, this function is 68k, if called on a
PowerPC machine it contextswitches over to the PowerPC, using WarpOS.

On the first call, dec_opt should be DEC_OPT_INIT, and param1 a
pointer to a correctly filled DEC_PARAM structure. On the second call
dec_opt should be DEC_OPT_SETPP, and param1 should be a
DEC_SET structure with postproc_level set to 0.

On a DEC_OPT_DECODE_RGB or a DEC_OPT_DECODE_YUV call param1 should be
a correctly initialized DEC_FRAME structure. bitstream is the input buffer
here, bmp the output buffer. In case of YUV-Decoding, the bmap parameter
needs to be a char **, with three elements: The pointer to the Y-array,
to the U-array, and the V-array.

A DEC_OPT_RELEASE operation with param1 and param2 both 0 terminates the
Decoding.
```

```
Once DEC_OPT_DECODE_RGB is implemented 16/24/32 Bit will be possible
(RGB16/RGB24/ARGB32 formats), but currently only YUV-Output is supported.

  INPUTS
handle - currently ignored, for future use, put 0 in there for now
dec_opt - the function you want to perform
param1 - the parameter structure, according to the function
param2 - for future use, currently set to 0

  RESULTS
rtnCode - DEC_OK if no error occurred.
  DEC_MEMORY indicates that not enough memory was present to perform
  the operation. DEC_BAD_FORMAT indicates that this was no OpenDivx file.
  DEC_OPERATION_UNIMPLEMENTED indicates that you try to replay with yuv2rgb
  conversion (DEC_OPT_DECODE_RGB), currently only DEC_OPT_DECODE_YUV is  ↩
     implemented.

  SEE ALSO
```

1.3 opendivx.library/DIVX_DecodePPC

```
  NAME
DIVX_DecodePPC -- Decode a OpenDivx using a PPC Function

  SYNOPSIS
result = DIVX_Decode68k(handle, dec_opt, param1, param2)
r3                      r3      r4       r5      r6



  FUNCTION
This function decodes a frame of a OpenDivx. It also can be used to
initialize or quit the replay, or to set the postprocessing level.
If called on a 68k machine, this function is PPC. It requires a PowerPC
to be called. It does not use any contextswitches, but is completely
PPC Native (WarpOS).

On the first call, dec_opt should be DEC_OPT_INIT, and param1 a
pointer to a correctly filled DEC_PARAM structure. On the second call
dec_opt should be DEC_OPT_SETPP, and param1 should be a
DEC_SET structure with postproc_level set to 0.

On a DEC_OPT_DECODE_RGB or a DEC_OPT_DECODE_YUV call param1 should be
a correctly initialized DEC_FRAME structure. bitstream is the input buffer
here, bmp the output buffer. In case of YUV-Decoding, the bmap parameter
needs to be a char **, with three elements: The pointer to the Y-array,
to the U-array, and the V-array.

A DEC_OPT_RELEASE operation with param1 and param2 both 0 terminates the
Decoding.

Once DEC_OPT_DECODE_RGB is implemented 16/24/32 Bit will be possible
(RGB16/RGB24/ARGB32 formats), but currently only YUV-Output is supported.
```

```
  INPUTS
handle - currently ignored, for future use, put 0 in there for now
dec_opt - the function you want to perform
param1 - the parameter structure, according to the function
param2 - for future use, currently set to 0

  RESULTS
rtnCode - DEC_OK if no error occurred.
  DEC_MEMORY indicates that not enough memory was present to perform
  the operation. DEC_BAD_FORMAT indicates that this was no OpenDivx file.
  DEC_OPERATION_UNIMPLEMENTED indicates that you try to replay with yuv2rgb
  conversion (DEC_OPT_DECODE_RGB), currently only DEC_OPT_DECODE_YUV is  ↩
     implemented.

  SEE ALSO
```

1.4 opendivx.library/DIVX_Encode68k

```
  NAME
DIVX_Encode68k -- Encode a OpenDivx using an 68k Function

  SYNOPSIS
result = DIVX_Encode68k(handle, dec_opt, param1, param2)
D0                      d0      D1       A0      A1



  FUNCTION
This function encodes a frame of a OpenDivx. It also can be used to
initialize or quit the Encoding. If called on a 68k machine, this
function is 68k, if called on a PowerPC machine it contextswitches
over to the PowerPC, using WarpOS.

On ENC_OPT_INIT (which requires to be the first call of the encoder)
you have to provide a correctly initialized ENC_PARAM structure,
on ENC_OPT_WRITE or ENC_OPT_KEY you can encode a frame, and need
to provide a ENC_FRAME structure, where bmp is a 24 Bit image,
and bitstream is the buffer for the encoded Bitstream. The structures
are passed into param1. In param2 you get a ENC_RESULT returned,
which tells you if the encoded frame was a keyframe or not (note,
that ENC_OPT_WRITE and ENC_OPT_KEY currently make no difference).

  INPUTS
handle - currently ignored, for future use, put 0 in there for now
dec_opt - the function you want to perform
param1 - the parameter structure, according to the function
param2 - for future use, currently set to 0

  RESULTS
rtnCode - DEC_OK if no error occurred.
  DEC_MEMORY indicates that not enough memory was present to perform
  the operation. DEC_BAD_FORMAT indicates a format error.

  SEE ALSO
```

1.5 opendivx.library/DIVX_EncodePPC

```
  NAME
DIVX_EncodePPC -- Encode a OpenDivx using an 68k Function

  SYNOPSIS
result = DIVX_EncodePPC(handle, dec_opt, param1, param2)
r3                      r3      r4       r5      r6



  FUNCTION
This function encodes a frame of a OpenDivx. It also can be used to
initialize or quit the Encoding. This function is PPC, using WarpOS.
It is a complete PPC Native function without contextswitches.

On ENC_OPT_INIT (which requires to be the first call of the encoder)
you have to provide a correctly initialized ENC_PARAM structure,
on ENC_OPT_WRITE or ENC_OPT_KEY you can encode a frame, and need
to provide a ENC_FRAME structure, where bmp is a 24 Bit image,
and bitstream is the buffer for the encoded Bitstream. The structures
are passed into param1. In param2 you get a ENC_RESULT returned,
which tells you if the encoded frame was a keyframe or not (note,
that ENC_OPT_WRITE and ENC_OPT_KEY currently make no difference).

  INPUTS
handle - currently ignored, for future use, put 0 in there for now
dec_opt - the function you want to perform
param1 - the parameter structure, according to the function
param2 - for future use, currently set to 0

  RESULTS
rtnCode - DEC_OK if no error occurred.
  DEC_MEMORY indicates that not enough memory was present to perform
  the operation. DEC_BAD_FORMAT indicates a format error.

  SEE ALSO
```