Flix Engine Linux

8.0.17.2

Generated by Doxygen 1.5.5

Tue Nov 2 16:52:53 2010

Contents

1 Flix Engine Documentation **1**

1.1 Page Contents 1

1.2 Introduction 1

1.3 Starting Points 1

1.4 New Codecs & Container Formats 2

1.5 Support Options & FAQ 2

2 About Flix Engine **5**

2.1 Version Information 6

2.2 Third Parties 6

3 Language Binding API Deviations **9**

3.1 Flix Engine Handle Creation 10

3.2 Codec Handle Creation 11

3.3 Filter Handle Creation 11

3.4 Constants 11

3.5 Functions 12

4 FE2_VP6_CONCURRENCY Performance **15**

4.1 Intel Pentium 4 540 16

4.2 Intel Xeon X5355 20

5 CHANGELOG **25**

6 flixd man page **39**

7 Samples **43**

8 Overview **45**

9 Command Line **49**

10 Command Line **53**

11 Command Line **57**

12 CGI **61**

12.1 flix2_sample.cgi 62

12.2 process_sample.cgi 98

13 Command Line **103**

14 CGI **107**

14.1 flix2_sample.php 108

14.2 process_sample.php 144

15 Command Line **149**

16 CGI **153**

16.1 flix2_sample.py 154

16.2 process_sample.py 190

17 Deprecated List **195**

18 Module Index **209**

18.1 Modules 209

19 File Index **211**

19.1 File List 211

20 Module Documentation **213**

20.1 Audio Encoding Options 213

20.2 Deprecated 215

20.3 Codecs 223

20.4 Common Codec Parameters 224

20.5 Video Codecs 226

20.6 Audio Codecs 227

20.7 AAC 228

20.8 AAC+ 230

20.9 AMR_NB - FFmpeg 232

20.10 H263 - FFmpeg 234

20.11 H264 237

20.12 MP3 - LAME 241

20.13 libvorbis - FFmpeg . . . 244
20.14 VP6 . . . 245
20.15 VP6 with Alpha . . . 255
20.16 VP8 . . . 264
20.17 Encoding Statistics . . . 273
20.18 Filters . . . 278
20.19 Video Filters . . . 279
20.20 Audio Filters . . . 280
20.21 Deinterlace . . . 281
20.22 Brightness/Contrast/Hue/Saturation . . . 284
20.23 Blur . . . 295
20.24 Crop . . . 298
20.25 Cut . . . 303
20.26 Denoise . . . 309
20.27 Frame Rate . . . 310
20.28 Highpass . . . 316
20.29 Lowpass . . . 318
20.30 Mirror . . . 320
20.31 Overlay (Watermark) . . . 322
20.32 PNG Image Export (Thumbnail) . . . 336
20.33 Resample . . . 344
20.34 Rotate . . . 345
20.35 Scale . . . 346
20.36 Sharpen . . . 353
20.37 Flix Engine API . . . 354
20.38 Main Engine Interface . . . 355
20.39 Flix Engine Core Library . . . 371
20.40 Filter Manipulation . . . 375
20.41 Codec Manipulation . . . 379
20.42 Muxer Manipulation . . . 383
20.43 Frame Server API . . . 387
20.44 Deprecated . . . 393
20.45 Muxers . . . 394
20.46 FLV . . . 395
20.47 FXM . . . 400
20.48 3GPP - FFmpeg . . . 403

20.49 3GPP2 - FFmpeg 405
20.50 MOV - FFmpeg 407
20.51 MP4 - FFmpeg 409
20.52 SWF 411
20.53 WebM - FFmpeg 444
20.54 Base Types 445
20.55 Video Encoding Options 450
20.56 Deprecated 455

21 File Documentation **469**

21.1 about.dox File Reference 469
21.2 binding_deviations.dox File Reference 470
21.3 flixengine2/audio_options.h File Reference 471
21.4 flixengine2/codec_constants.h File Reference 473
21.5 flixengine2/codecs/aac.h File Reference 474
21.6 flixengine2/codecs/amr.h File Reference 475
21.7 flixengine2/codecs/codec_common.h File Reference 476
21.8 flixengine2/codecs/h263.h File Reference 477
21.9 flixengine2/codecs/h264.h File Reference 478
21.10 flixengine2/codecs/lame.h File Reference 479
21.11 flixengine2/codecs/vorbis.h File Reference 480
21.12 flixengine2/codecs/vp6.h File Reference 481
21.13 flixengine2/codecs/vp6_alpha.h File Reference 483
21.14 flixengine2/codecs/vp8.h File Reference 485
21.15 flixengine2/encoding_status.h File Reference 487
21.16 flixengine2/filter_constants.h File Reference 488
21.17 flixengine2/filters/adaptive_deinterlace.h File Reference 489
21.18 flixengine2/filters/bchs.h File Reference 490
21.19 flixengine2/filters/blur.h File Reference 491
21.20 flixengine2/filters/crop.h File Reference 492
21.21 flixengine2/filters/cut.h File Reference 493
21.22 flixengine2/filters/denoise.h File Reference 494
21.23 flixengine2/filters/framerate.h File Reference 495
21.24 flixengine2/filters/highpass.h File Reference 496
21.25 flixengine2/filters/lowpass.h File Reference 497
21.26 flixengine2/filters/mirror.h File Reference 498
21.27 flixengine2/filters/overlay.h File Reference 499

21.28 flixengine2/filters/png_export.h File Reference 501
21.29 flixengine2/filters/resample.h File Reference 503
21.30 flixengine2/filters/rotate.h File Reference 504
21.31 flixengine2/filters/scale.h File Reference 505
21.32 flixengine2/filters/sharpen.h File Reference 506
21.33 flixengine2/flixengine2.h File Reference 507
21.34 flixengine2/fs_options.h File Reference 513
21.35 flixengine2/media_editor_options.h File Reference 514
21.36 flixengine2/muxer_constants.h File Reference 516
21.37 flixengine2/muxers/flv.h File Reference 517
21.38 flixengine2/muxers/fxm.h File Reference 519
21.39 flixengine2/muxers/isomedia.h File Reference 520
21.40 flixengine2/muxers/swf.h File Reference 522
21.41 flixengine2/muxers/webm.h File Reference 524
21.42 flixengine2/on2types.h File Reference 525
21.43 flixengine2/overlay_options.h File Reference 527
21.44 flixengine2/swf_options.h File Reference 528
21.45 flixengine2/video_options.h File Reference 531
21.46 mainpage.dox File Reference 535
21.47 samples.dox File Reference 536

Chapter 1

Flix Engine Documentation

1.1 Page Contents

- Introduction
- Starting Points
- New Codecs & Container Formats
- Support Options & FAQ

1.2 Introduction

Thank you for licensing the On2 Flix Engine SDK. This SDK provides a framework for embedding the acclaimed On2 Flix multimedia engine in your server-based applications. Flix Engine is able to transcode nearly any type of multimedia input to high-quality Adobe® Flash Video (FLV and SWF) and other popular formats, including 3GPP, 3GPP2, MOV, and MP4.

Also available for Windows! Flix Engine for Windows provides full COM automation support for access from ASP/ASP.NET, C/C++/C#, ColdFusion, Delphi, Java (via JNI/COM Bridge), PHP, Python, Visual Basic/VB.NET, VBScript, and any other COM-enabled container application or programming language. On2 Flix Engine for Windows is available as a dynamically linked library (DLL).

1.3 Starting Points

- Consult the change log for a complete list of improvements and fixes in this and prior Flix Engine releases.

- Reading and running the sample code gives a strong overview of what you can achieve with Flix Engine.
- Flix Engine API reference
- Also note language binding differences from the main API.

1.4 New Codecs & Container Formats

Flix Engine 8.0.17.0 supports the html5 ready WebM container using VP8 and Vorbis.

Flix Engine 8.0.10.0 now supports **MPEG-4 Part 10 standard**, commonly known as H.264. Specifically, Flix Engine can now encode video using the H.264 codec and audio using AAC / AAC+, in addition to the already available encoders: On2 VP6 and H.263 video, and MP3 and AMR-NB audio.

Accordingly, Flix Engine is now able to produce the common MPEG-4 container formats, MP4, MOV and 3GPP2.

Another new capability of Flix Engine 8.0.10.0 is the **VP6_S encoding profile**, which is optimized for encoding higher resolution material destined for playback in low-resource environments, such as older PCs. The standard VP6 encoding profile (VP6_E) is the default. See FE2_VP6_PROFILE in the API reference.

1.4.1 Flash Player Support and AVC Licensing

Please note carefully that, at this writing, no Flash Player release supports MPEG-4 streams. We have added these new codecs and capabilities to Flix Engine in anticipation of our users' future needs, and to expand choice.

Also note that the distribution of AVC/H.264-encoded media is not royalty-free in many cases. Be sure to comply with the H.264 licensing terms available at `http://www.mpegla.com` and `http://www.vialicensing.com`.

A **longer discussion** `of technical and licensing issues` is available at On2.com.

1.5 Support Options & FAQ

Before contacting support, please consult our **Frequently Asked Questions** list at On2.com, which provides fast answers to the questions that Flix Engine users ask most.

Any further questions can be directed to `flixsupport@on2.com`. Make sure you add *flixd* or *flix engine linux* to your *Subject* line. Also include the version of Flix Engine you are using, as well as a detailed description of the problem and any pertinent system information, in the message body.

The use or implementation of Flix Engine may require copyright and/or patent licenses from parties other than On2 Technologies. The user of this software is solely responsible for obtaining any and all of such licenses. In addition, supply of this Flix product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial, satellite, cable and/or other distribution channels), streaming applications (via Internet, intranets and/or other networks), other content distribution systems (pay-audio or audio-on-demand applications and the like) or on physical media (compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details regarding some of the foregoing, please visit `http://mp3licensing.com`, `http://www.mpegla.com`, `http://www.vialicensing.com` and `http://www.codingtechnologies.com`.

THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL AND NONCOMMERCIAL USE OF A CONSUMER TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO")

AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL AND NON-COMMERCIAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM.

Chapter 2

About Flix Engine

2.1 Version Information

Version: 8.0.17.2

2.2 Third Parties

- This Flix product includes H.264 files licensed by Intel Corporation.

  Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

- This Flix product includes aacPlus developed by Coding Technologies.

  http://www.codingtechnologies.com

- This Flix product uses libavcodec, libavformat and libavutil which are part of FFmpeg (ffmpeg.mplayerhq.hu)

  ```
  Copyright (c) 2001 Fabrice Bellard
  ```

- This Flix product uses LAME (www.mp3dev.org)

- This Flix product uses MEncoder (http://www.mplayerhq.hu/).

  ```
  MEncoder is free software distributed under the GNU General Public
  License (GPL). MEncoder comes WITHOUT ANY WARRANTY; without even the
  implied warranty of MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE,
  or NON-INFRINGEMENT. You are entitled to receive a copy of the original
  MEncoder source code along with the modifications made to it.
  You can find a copy along with patch and build instructions here:
  http://support.on2.com/gpl/mplayer
  ```

- This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)

  ```
  Copyright (c) 1998-2007 The OpenSSL Project.  All rights reserved.

  Redistribution and use in source and binary forms, with or without
  modification, are permitted provided that the following conditions
  are met:

  1. Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.

  2. Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in
     the documentation and/or other materials provided with the
     distribution.

  3. All advertising materials mentioning features or use of this
     software must display the following acknowledgment:
     "This product includes software developed by the OpenSSL Project
     for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

  4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
     endorse or promote products derived from this software without
     prior written permission. For written permission, please contact
     openssl-core@openssl.org.
  ```

```
5. Products derived from this software may not be called "OpenSSL"
   nor may "OpenSSL" appear in their names without prior written
   permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following
   acknowledgment:
   "This product includes software developed by the OpenSSL Project
   for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
====================================================================

This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com).  This product includes software written by Tim
Hudson (tjh@cryptsoft.com).

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written
by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as
the following conditions are aheared to.  The following conditions
apply to all code found in this distribution, be it the RC4, RSA,
lhash, DES, etc., code; not just the SSL code.  The SSL documentation
included with this distribution is covered by the same copyright terms
except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in
the code are not to be removed.
If this package is used in a product, Eric Young should be given attribution
as the author of the parts of the library used.
This can be in the form of a textual message at program startup or
in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
   must display the following acknowledgement:
   "This product includes cryptographic software written by
    Eric Young (eay@cryptsoft.com)"
   The word 'cryptographic' can be left out if the rouines from the library
   being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from
   the apps directory (application code) you must include an acknowledgement:
   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
```

```
THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The licence and distribution terms for any publically available version or
derivative of this code cannot be changed.  i.e. this code cannot simply be
copied and put under another distribution licence
[including the GNU Public Licence.]
```

- libpng (http://www.libpng.org/)

```
libpng version 1.2.37 - June 4, 2009
Copyright (c) 1998-2009 Glenn Randers-Pehrson
(Version 0.96 Copyright (c) 1996, 1997 Andreas Dilger)
(Version 0.88 Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.)
```

- libogg (http://xiph.org/)

```
Copyright (c) 2002, Xiph.org Foundation

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.org Foundation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE FOUNDATION
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

- zlib general compression library (http://zlib.net/)

```
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler
```

Chapter 3

Language Binding API Deviations

The Flix Engine language bindings endeavor to remain as true as possible to the Flix Engine API. In certain cases, however, the C implementation does not transfer directly. This guide will provide an overview of the changes, both necessary and those done for the sake of convenience. Currently Flix Engine has support for Java, Perl, PHP and Python.

The main differences include:

- how Flix Engine handles are created
- how Codec handles are created
- how Filter handles are created
- how constants are referenced
- how functions that return values via a result parameter accomplish this

A word on Java:

For better consistency with Java function call semantics, the Java bindings consist of three main wrapper classes, FlixEngine2, Codec and Filter. In addition to this, each enumeration becomes its own class. *Javadoc* documentation is provided to highlight these differences. If you chose to install the Java bindings, the *Javadoc* will be installed by default in:
`/usr/local/src/flixmodules/flixjava/doc`

Attention:

The language bindings need only be rebuilt should the CHANGELOG note changes to the interface, e.g., the addition of constants/functions. Without rebuilding, these new features will be unavailable and depending on the bindings being used will either generate a compile error or, in the case of constants, will return ON2_NOT_SUPP should they be used.

3.1 Flix Engine Handle Creation

When creating a handle to the Flix Engine one calls Flix2_Create() or Flix2_CreateEx() with a pointer to a FLIX2HANDLE, allowing the value of the handle to be stored back to this location. So for example:

```
FLIX2HANDLE flix;
Flix2_Create(&flix);
/*...upon success flix holds the address of the handle...*/
```

When dealing with various interpreted languages there is no way to directly accomplish this. For this reason a helper function is introduced to first create a storage location for the handle. Note that though the following is in Perl, the helper functions carry over to all of the language bindings.

```
$flixptr= new_flix2handlep();
```

We can now use this storage location to create a new handle to the engine:

```
Flix2_Create($flixptr);
```

Before we use the new handle in the remaining API functions we have to retrieve its actual value. Another helper function is introduced for this purpose:

```
$flix= flix2handlep_value($flixptr);
```

`$flix` is now the handle to use in the remaining API calls, for example:

```
Flix2_SetInputFile($flix,"/path/to/input.video");
```

When you're finished with the handle you can clean up any resources allocated to it by calling:

```
delete_flix2handlep($flixptr);
```

3.2 Codec Handle Creation

The Codec Manipulation functions use the same interface to create a handle as the Filters. See below for a detailed example.

3.3 Filter Handle Creation

The Filter Manipulation functions present an issue similar to creating a Flix Engine handle. The solution is the same, to introduce a few helper functions, as demonstrated by the following PHP code:

```
##create a storage location for the filter handle
$filterptr= new_flix2plgnhandlep();
Flix2_AddFilter($filterptr,$flix,FE2_FILTER_PNGEX);

##retrieve the value of the handle for use in the remaining filter functions
$filter= flix2plgnhandlep_value($filterptr);
Flix2_FilterSetParam($filter,FE2_PNGEX_AUTO_EXPORT_COUNT,10);

## ...encode...
##cleanup
delete_flix2plgnhandlep($filterptr);
```

The above adds the PNG Image Export (Thumbnail) to the filter chain and sets it to auto export 10 images. You will notice that the string constants are referenced by name.

Note:

> You may reuse `$filterptr` in subsequent calls to Flix2_AddFilter(). Note that you must retrieve the filter handle with `flix2plgnhandlep_value()` each time.

3.4 Constants

The string constants defined by the Filters, as well as any other defines and enumeration members can all be referenced by name in the language bindings. The method for referencing them varies from language to language:

- Java

```
String a_string_constant= FlixEngine2.FE2_FILTER_PNGEX;
try {
   codec.setParam(FlixEngine2.FE2_VP6_BITRATE,750.0)
} catch (FlixException e) {}
```

- Perl

```
##using the flix engine namespace
package On2::flixengine2;
$a_string_constant= $FE2_FILTER_PNGEX;
$sc= Flix2_CodecSetParam($codec,$FE2_VP6_BITRATE,750);
if($sc != $ON2_OK) {}

##or explicitly
$a_string_constant= $On2::flixengine2::FE2_FILTER_PNGEX;
$sc= Flix2_CodecSetParam($codec,$On2::flixengine2::FE2_VP6_BITRATE,750);
if($sc != $On2::flixengine2::ON2_OK) {}
```

- PHP

```
$a_string_constant= FE2_FILTER_PNGEX;
$sc= Flix2_CodecSetParam($codec,FE2_VP6_BITRATE,750);
if($sc != ON2_OK) {}
```

- Python

```
a_string_constant= flixengine2.FE2_FILTER_PNGEX
sc= flixengine2.Flix2_CodecSetParam(codec,flixengine2.FE2_VP6_BITRATE,750)
if(sc != flixengine2.ON2_OK):
```

Note:

Constants should ALWAYS be referenced by name as their values are subject to change.

3.5 Functions

In the same vein as the Flix Engine and Filter handle issue is the issue with functions that return values via a result parameter. In this case the value is returned in a list instead of creating a storage location or modifying a pointer.

Some examples will help clarify this point:

- Java

```
try {
   boolean b = flix.IsEncoderRunning();
   int w = flix.video_options_GetSourceWidth();
   double d = flix.video_options_GetVideoFramerateAsDouble();
} catch (FlixException e) {}
```

- Perl

```
##in separate scalars
#boolean
($sc,$b)= Flix2_IsEncoderRunning($flix);
#integer
($sc,$w)= video_options_GetSourceWidth($flix);
#double
($sc,$fps)= video_options_GetVideoFramerateAsDouble($flix);

##or in one array, sc is $res[0], value is $res[1]
#boolean
@res= Flix2_IsEncoderRunning($flix);
#integer
@res= video_options_GetSourceWidth($flix);
#double
@res= video_options_GetVideoFramerateAsDouble($flix);
```

- PHP

```
##sc is $res[0], value is $res[1]
#boolean
$res= Flix2_IsEncoderRunning($flix);
#integer
$res= video_options_GetSourceWidth($flix);
#double
$res= video_options_GetVideoFramerateAsDouble($flix);
```

- Python

```
##sc is res[0], value is res[1]
#boolean
res= flixengine2.Flix2_IsEncoderRunning(flix)
#integer
res= flixengine2.video_options_GetSourceWidth(flix)
#double
res= flixengine2.video_options_GetVideoFramerateAsDouble(flix)
```

Attention:

Functions that return a string, e.g., Flix2_GetInputFile(), are NOT currently supported by the language bindings.

Note:

As the Java bindings have been modified to throw an exception instead of directly returning a status code, the return value is generally a single value or an array where appropriate. See the *Javadoc* for more information.

Chapter 4

FE2_VP6_CONCURRENCY Performance

What follows are graphs of the mean encode time over 5 successive encodes for 4 relatively short videos using v8.0.9.0. In addition the same encodes were run against v8.0.8.2, less the FE2_VP6_CONCURRENCY parameter, as some modest gains were made outside the scope of this parameter. Note that exceeding the per-processor input buffer maximum (60) will reduce the overall gain as the encode process will become serialized until all frames are buffered.

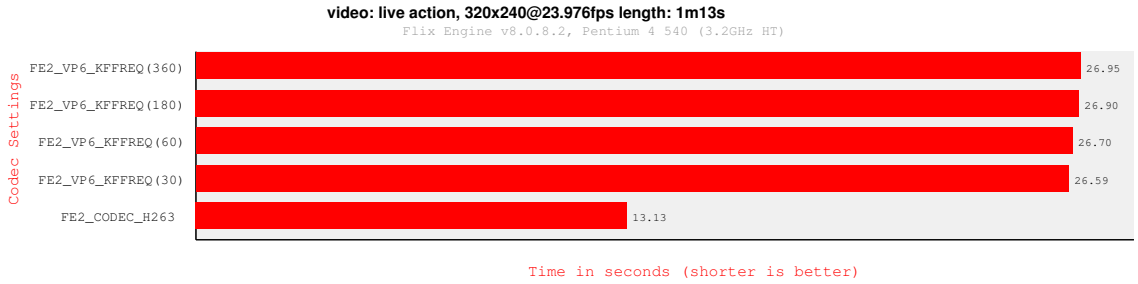
The graphs are split into 2 main sections grouped by the processor on the host machine:

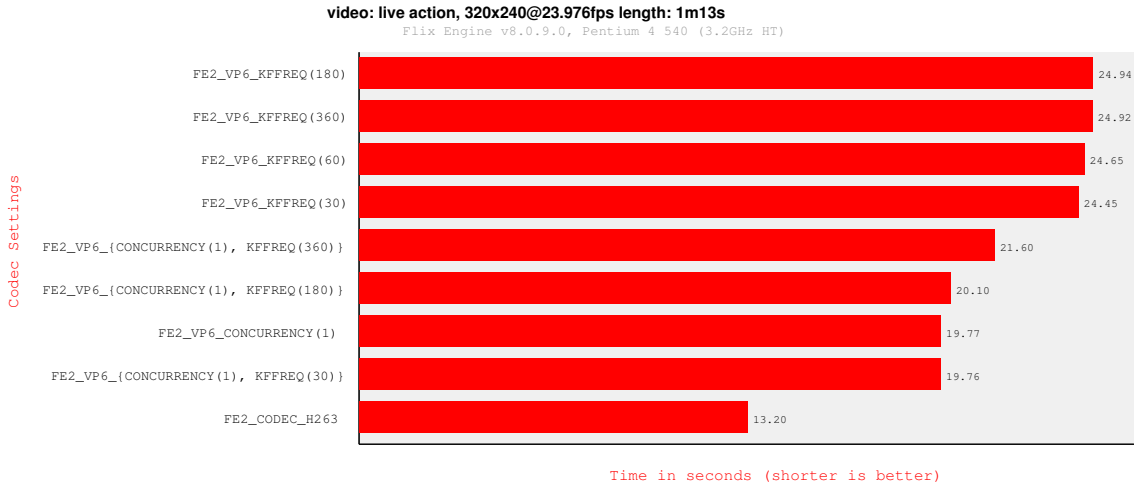
- Intel Pentium 4 540
    1. 320x240 (QVGA)
    2. 640x480 (VGA)
    3. 1024x464
    4. 1280x720 (720p)
- Intel Xeon X5355
    1. 320x240 (QVGA)
    2. 640x480 (VGA)
    3. 1024x464
    4. 1280x720 (720p)

4.1 Intel Pentium 4 540

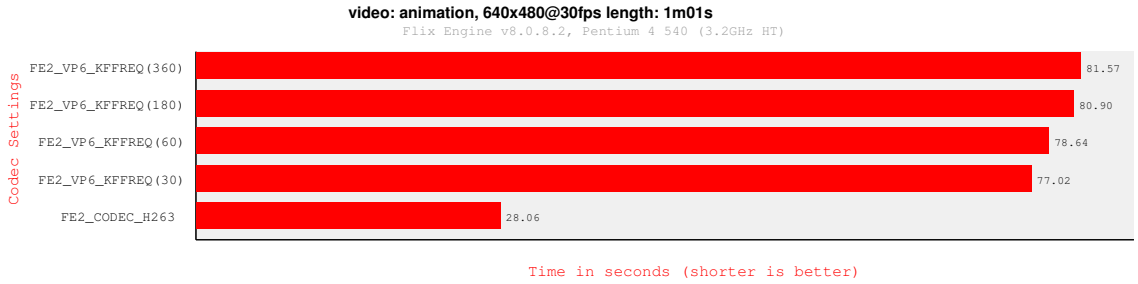
System Specifications: Slackware 10.2 (i686), Pentium 4 540 (3.2GHz HT)

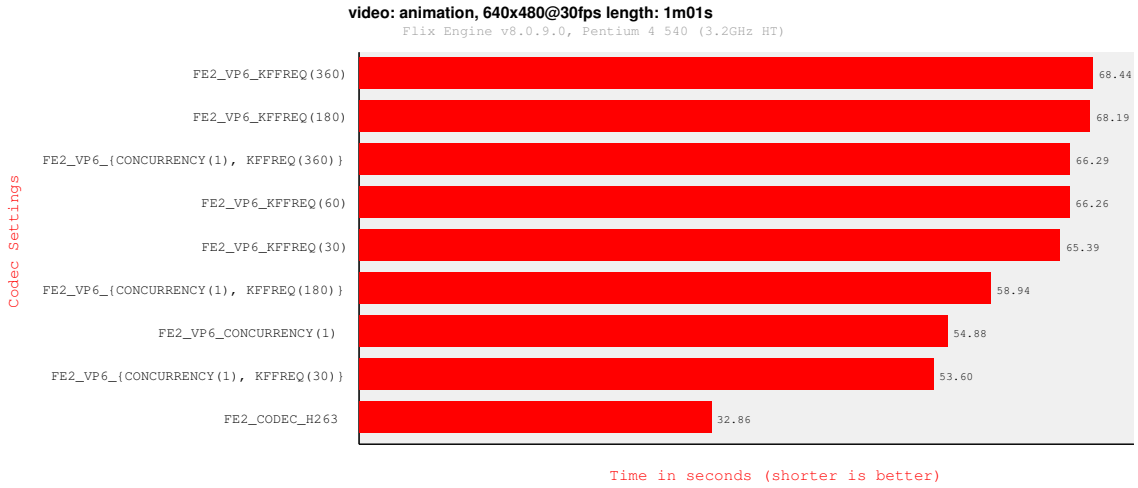
4.1.1 320x240 (QVGA)



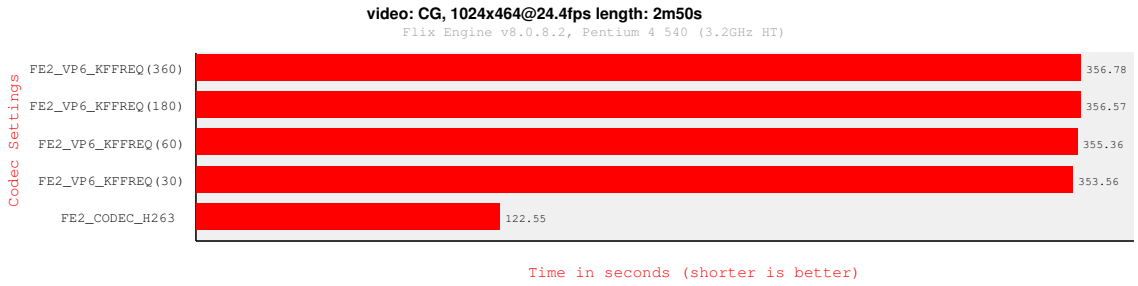


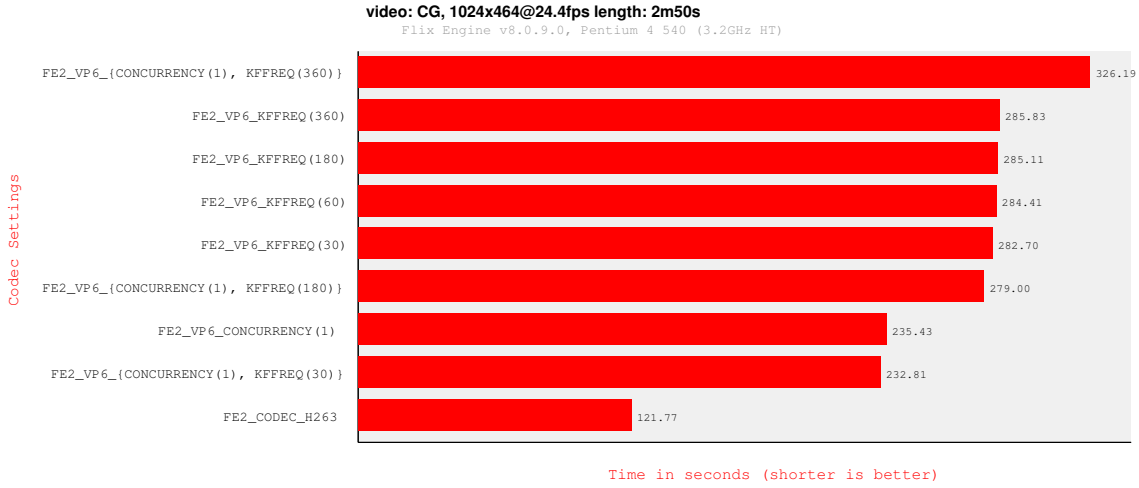
4.1.2 640x480 (VGA)



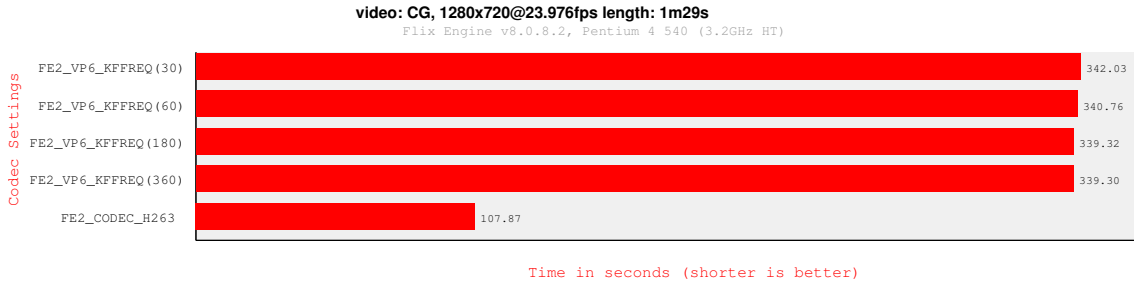


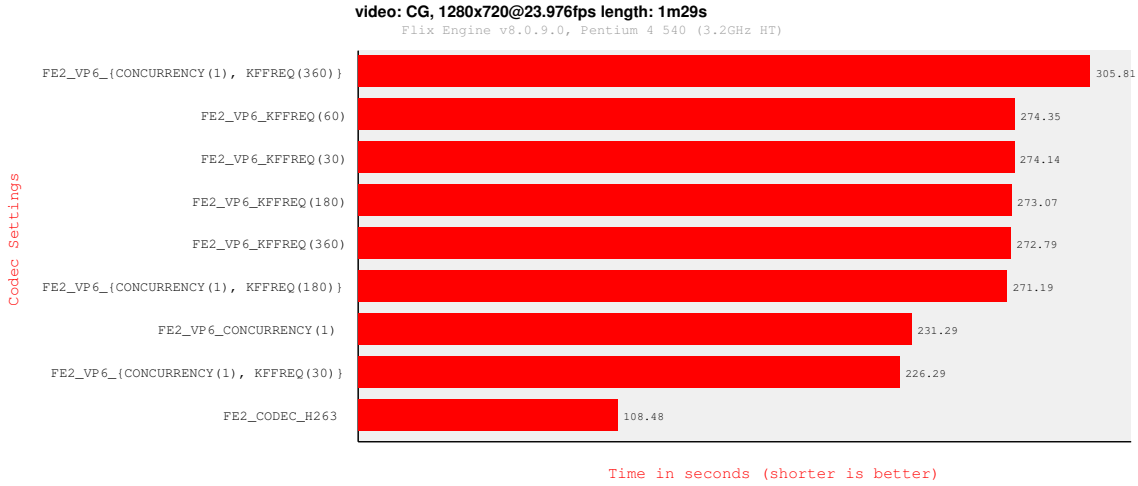
4.1.3 1024x464





4.1.4 1280x720 (720p)

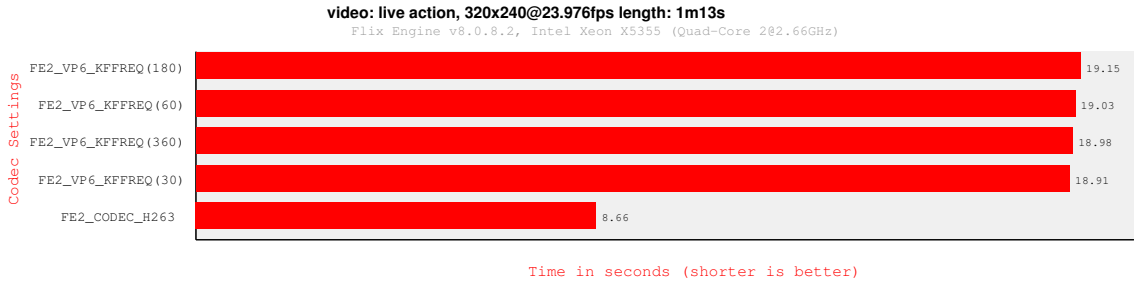


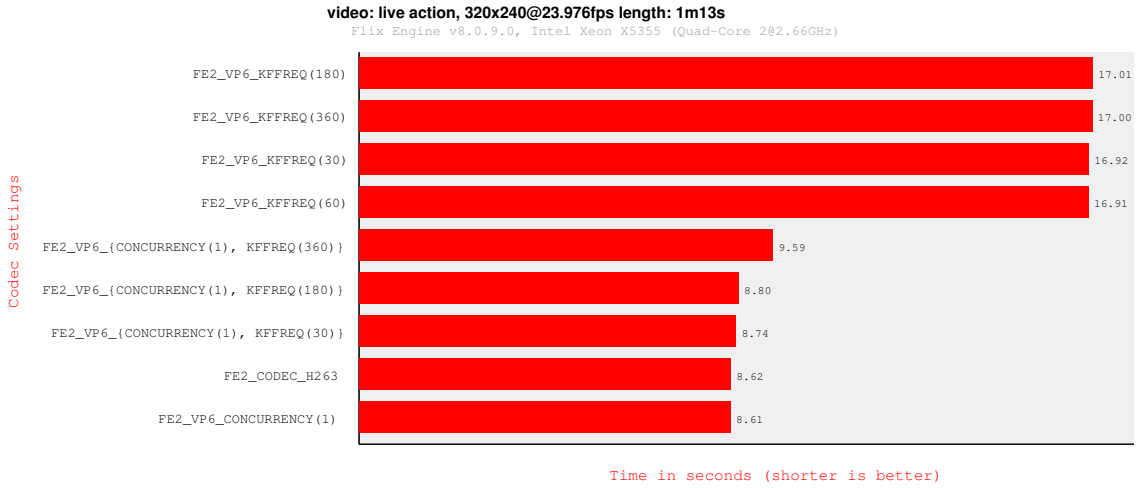


4.2 Intel Xeon X5355

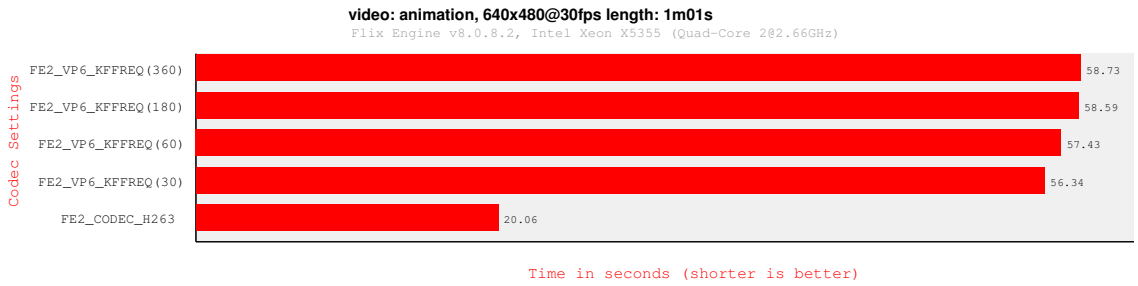
System Specifications: openSUSE 10.2 (x86_64), Intel Xeon X5355 (Quad-Core 2@2.66GHz)

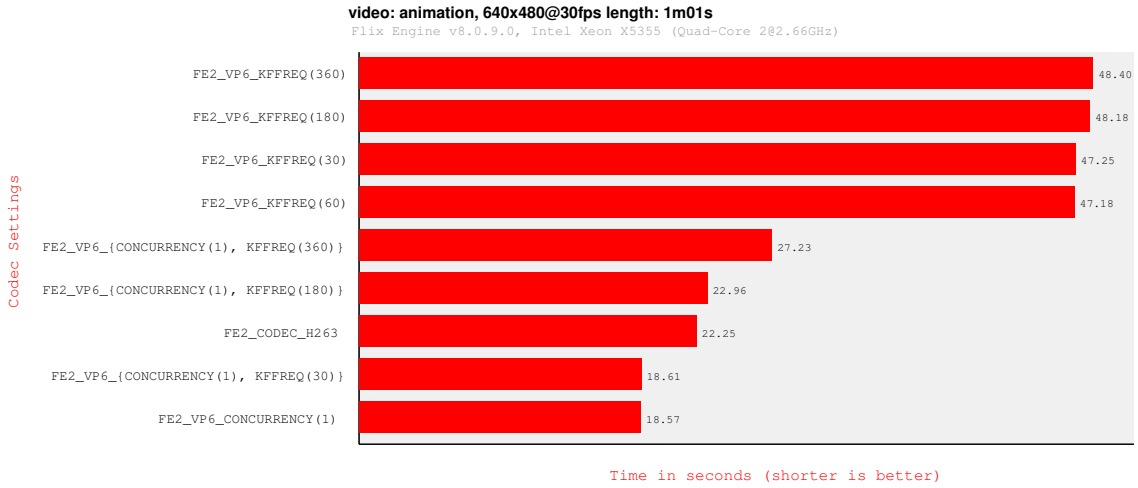
4.2.1 320x240 (QVGA)



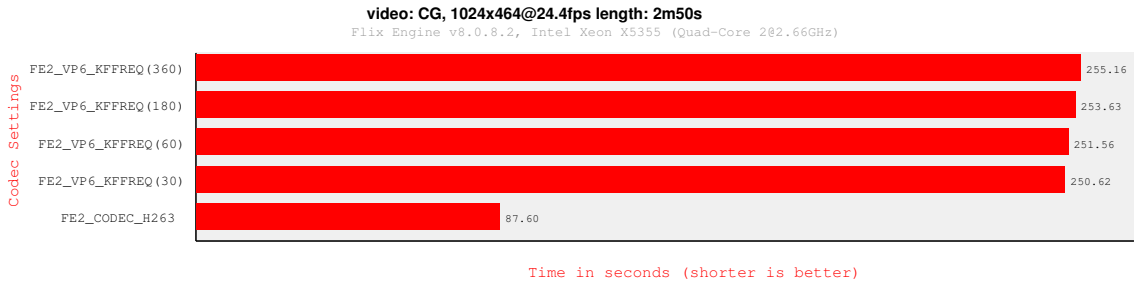


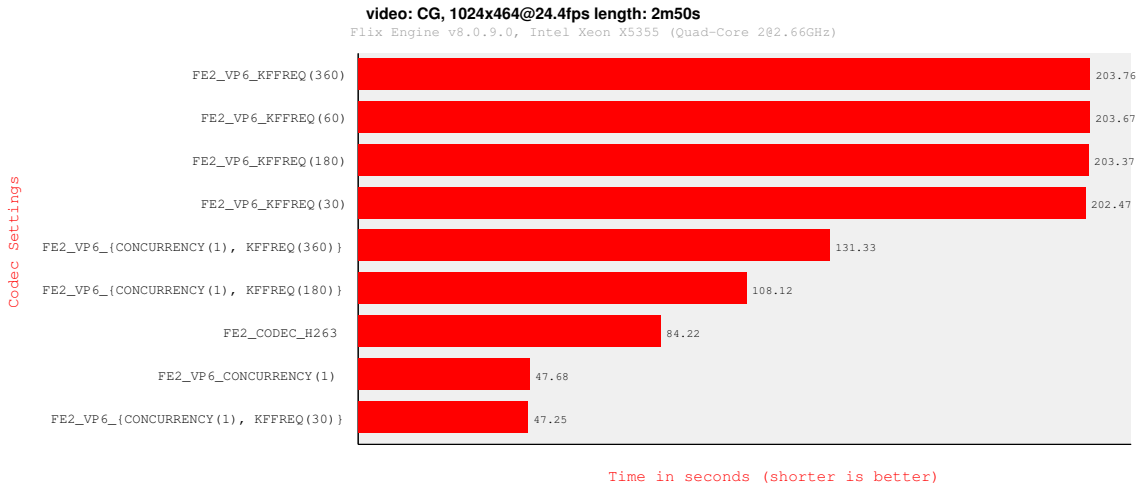
4.2.2 640x480 (VGA)



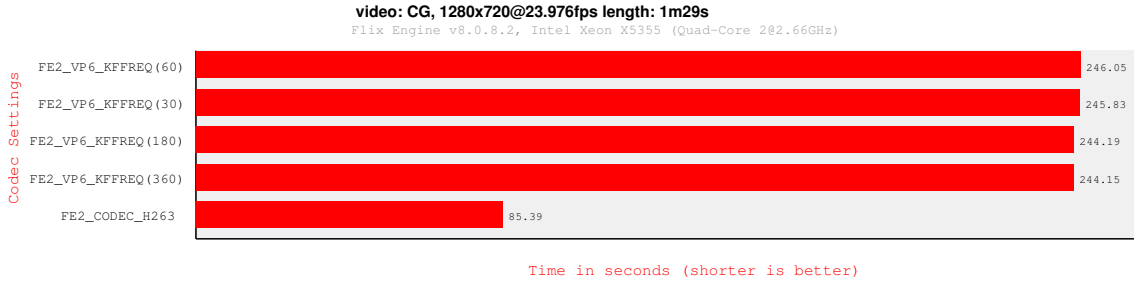


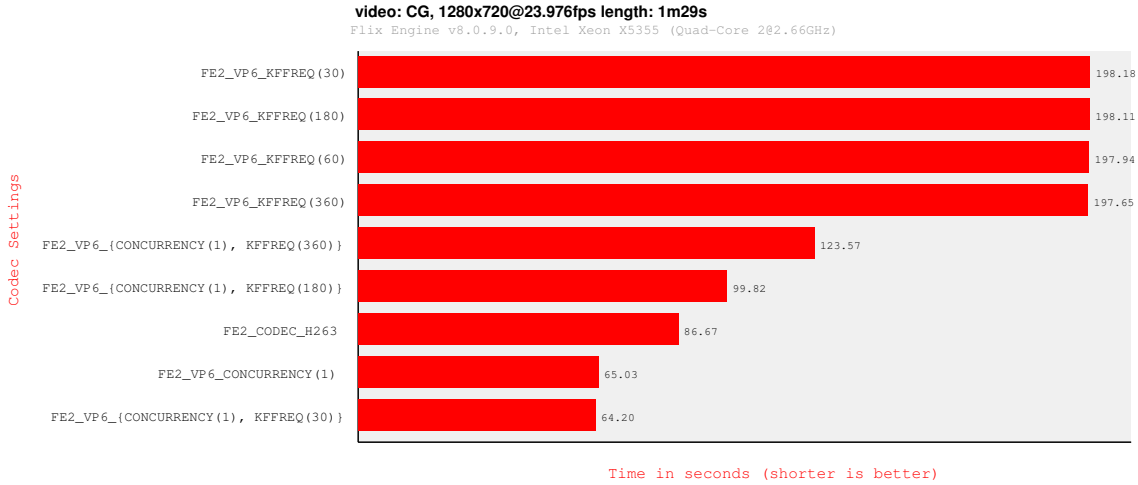
4.2.3 1024x464





4.2.4 1280x720 (720p)





Chapter 5

CHANGELOG

```
2010-11-02 v8.0.17.2 Linux
2010-11-02 v8.0.17.2_DEMO Linux
2010-11-02 v8.0.17.2 Windows COM v0.0.9.0
2010-11-02 v8.0.17.2_DEMO Windows COM v0.0.9.0
Bug Fixes:
- FE2_CODEC_VP8, update libvpx to Aylesbury
- FE2_CODEC_H264, fix a potential crash due to a bad buffer reallocation
- FE2_FILTER_FRAMERATE, fix bug causing unnecessary frame drops when using the
_FPS parameter. This would result in jitter in material even when using a
value matching a fixed-framerate input.
+--------------------------+
2010-07-20 v8.0.17.1 Linux
2010-07-20 v8.0.17.1_DEMO Linux
2010-07-20 v8.0.17.1 Windows COM v0.0.9.0
2010-07-20 v8.0.17.1_DEMO Windows COM v0.0.9.0
Bug Fixes:
- FE2_CODEC_H264, fix bug causing grey dots/grid appearing on scene change
  to a mostly white frame when using BASE_H264PROFILE
+--------------------------+
2010-05-17 v8.0.17.0 Linux
2010-05-17 v8.0.17.0_DEMO Linux
2010-05-17 v8.0.17.0 Windows COM v0.0.9.0
2010-05-17 v8.0.17.0_DEMO Windows COM v0.0.9.0
Features:
- (Linux) NOTE:
http://support.on2.com/flixengine/linux/api/lb_deviation.html#update
applies to this release
- (Windows) NOTE:
http://support.on2.com/flixengine/windows/api/group__flixengine__com.html#warn_dualinterface
applies to this release
- add FE2_MUXER_WEBM
    New header file: muxers/webm.h
- add FE2_CODEC_VP8/FE2_CODEC_VORBIS
    New header files: codecs/{vp8,vorbis}.h
- (Linux) mencoder, add WebM support
 - new patches:
   o libvpxdec.diff, vp8_webm-decode_plumbing.diff
+--------------------------+
2010-01-26 v8.0.16.1 Linux
2010-01-26 v8.0.16.1_DEMO Linux
2010-01-26 v8.0.16.1 Windows COM v0.0.8.0
2010-01-26 v8.0.16.1_DEMO Windows COM v0.0.8.0
Bug Fixes:
- FE2_FILTER_PNGEX, after v8.0.16.0, allow FE2_PNGEX_WIDTH/FE2_PNGEX_HEIGHT to
be queried
- (Linux) mencoder, update snapshot to r30380
Adds further QCELP support and the ability to transcode FLV5 (VP6 + alpha)
source.
 - new patches:
   o demux_asf-flush_buffered_asf_packet.diff - flush any pending packets at
     eof, avoids audio/video truncation in certain cases
 - updated patches:
   o mencoder-proto_ext_file_conf__UPSTREAM.diff - relocated common components
     to mpcommon.c and marked patch for its viability upstream based on
     comments from mplayer-dev-eng
   o mencoder_02_audio_only_hack.diff - exit immediately when no audio or
     video stream is found
   o codecs_conf-workarounds.diff - for compatibility w/prior releases revert
     r30265 and avoid untested ffwmapro
 - removed patches:
   o asf-correct_movielength.diff, avi_check_idxflags.diff,
     demux_lavf-probe_small_files.diff, mpegvideo-revert_r18381.diff: rendered
     unnecessary
+--------------------------+
2009-10-08 v8.0.16.0 Linux
2009-10-08 v8.0.16.0_DEMO Linux
2009-10-08 v8.0.16.0 Windows COM v0.0.8.0
```

```
2009-10-08 v8.0.16.0_DEMO Windows COM v0.0.8.0
Features:
- FE2_FILTER_PNGEX, FE2_PNGEX_WIDTH/FE2_PNGEX_HEIGHT allow use of reserved
values in the same manner as FE2_FILTER_SCALE allowing for simpler aspect
ratio preservation for thumbnails
Bug Fixes:
- (Linux) mencoder,
 - new patches:
   o demux_lavf-probe_small_files.diff - allow files smaller than 32kB to be
   inspected, providing behavior similar to r27474 snapshot.
 - updated patches:
   o libavformat_mov-avoid_reporting_empty_av_streams.diff,
     mov_06_tkhd_matrix_scale.diff - fix bug causing rotation of 180 to go
     unreported
- (Linux) flixd, return glibc requirement to 2.3.2. glibc-2.3.4 dependencies
were unintentionally added in v8.0.15.3.
- (Windows) FE2_FILTER_RESAMPLE, after v8.0.15.3, fix over reading of 8
bits/sample input creating potential for a crash
- (Windows) flvsplit.dll v2.7.0.4, fix bug causing crash in duplicate
timestamp handling
- (Windows) On2QTSource.dll v2.7.0.4, fix bug causing encode timeout/potential
crash w/Apple Pro Res files
+---------------------------+
2009-08-07 v8.0.15.3 Linux
2009-08-07 v8.0.15.3_DEMO Linux
2009-08-07 v8.0.15.3 Windows COM v0.0.8.0
2009-08-07 v8.0.15.3_DEMO Windows COM v0.0.8.0
Bug Fixes:
- Upgraded to libpng-1.2.37. Addresses an uninitialized-memory-read bug that
may have security implications.
For more information, see:
 http://www.libpng.org/pub/png/libpng.html
- Automatically add FE2_FILTER_RESAMPLE when input channels exceed 2
- (Linux) Automatically add FE2_FILTER_ROTATE for clips that specify a
presentation rotation angle. Most visibly improves support for iPhone 3GS
'tallscreen' video. Original behavior may be restored by adding
FE2_FILTER_ROTATE w/FE2_ROTATE_ANGLE=0.
- (Linux) mencoder, update snapshot to r29460
 - oggpcm.c, leave audio frame segmentation to libogg allowing non-zero start
   times to be properly transferred to the output
 - new patches:
   o libavformat_mov-track_rotation_metadata.diff,
   demux_lavf-ID_VIDEO_ROTATION.diff - allows reporting of presentation
   rotation angle present in e.g., iPhone 3GS 'tallscreen' videos
   o libavformat_mov-avoid_reporting_empty_av_streams.diff - more restricted
   version of removed demux_lavf-skip_empty_audio_streams.diff
   o mencoder_10_correct_pts.diff - mplayer derived correct-pts option used in
   conjunction w/muxer_ogg.
 - updated patches:
   o mov_06_tkhd_matrix_scale.diff - demux_mov analog to
   demux_lavf-ID_VIDEO_ROTATION.diff
 - removed patches:
   o demux_lavf-skip_empty_audio_streams.diff - Had the potential to miss
   detect empty audio streams w/FLV.
   o extension-revert_r29181.diff, demux_lavf-remove_mov_preferred.diff -
   Favor is now given to LAVF ISO demuxer over demux_mov as it addresses many
   issues including problems w/various AAC/QCELP containing clips along w/some
   sync issues.
- (Windows) fix bug causing 1 frame video input & input for which duration
could not be retrieved to fail
+---------------------------+
2009-05-14 v8.0.15.2 Linux
2009-05-14 v8.0.15.2_DEMO Linux
2009-05-14 v8.0.15.2 Windows COM v0.0.8.0
2009-05-14 v8.0.15.2_DEMO Windows COM v0.0.8.0
Bug Fixes:
- (Linux) mencoder, update snapshot to r29308
```

```
 - new patches:
   o demux_lavf-skip_empty_audio_streams.diff - do not report empty audio
   streams as this affects default output stream selection
   o extension-revert_r29181.diff - temporary, currently lavf/mov demuxer
   does not successfully demux all internal test clips
   o mencoder-ext_file_conf.diff - from mplayer.c, add support for extension
   profiles & file specific conf files
   o mp3lib-validate_winarray_index.diff - avoid potential crash when
   accessing fixed size array w/out of bounds index
   o mpegvideo-revert_r18381.diff - temporary, changes at this revision crash
   select internal test clips
   o stream_file-clear_eof_on_seek.diff - addresses a regression in the
   removal of mov_06_invalid_audio_size.diff. Files w/truncated audio data,
   but complete indexes, would result in video truncation.
   o vd_qtvideo-validate_ImageDesc_size.diff - avoid crash should this decoder
   be paired w/the lavf demuxer
 - updated patches:
   o demux_lavf-add_dv_mts_preferred.diff - in addition to DV files, allow
   libavformat to demux MTS/M2TS files, often the container for AVCHD
   o reduce_spurious_logging.diff - compact duplicate error messages
 - removed patches:
   o demux_increase_buffer.diff, flv-r16254_backport.diff,
   mplayer_demux_real.patch, workaround_libswscale_bgr15_confusion.diff:
   rendered unnecessary
- (Linux) Similar to v8.0.15.1 another case was identified on open that could
cause a hang. This has been addressed and a timeout added to avoid further
issues related to the change.
- (Windows) If the input file's aspect ratio is available on open it will now
be made available for use in FE2_FILTER_SCALE. This will not address those
that require a frame decode to provide this information (e.g., some WMVs).
+--------------------------+
2009-04-16 v8.0.15.1 Linux
2009-04-16 v8.0.15.1_DEMO Linux
Bug Fixes:
- (Linux) Fix hang introduced in change made for encode completion stability
for v8.0.15.0. This would occur if mencoder exited prior to producing any
data.
+--------------------------+
2009-04-02 v8.0.15.0 Linux
2009-04-02 v8.0.15.0_DEMO Linux
2009-04-02 v8.0.15.0 Windows COM v0.0.8.0
2009-04-02 v8.0.15.0_DEMO Windows COM v0.0.8.0
Features:
- (Linux) NOTE:
http://support.on2.com/flixengine/linux/api/lb_deviation.html#update
applies to this release
- (Windows) NOTE:
http://support.on2.com/flixengine/windows/api/group__flixengine__com.html#warn_dualinterface
applies to this release
- FE2_MUXER_FLV/FE2_MUXER_FXM, add
FE2_FLV_METADATA_ENABLE/FE2_FLV_METADATA_DISABLE and
FE2_FXM_METADATA_ENABLE/FE2_FXM_METADATA_DISABLE parameters for use in
controlling metadata output. Also add output support for
lastkeyframetimestamp, lastkeyframelocation and the keyframes object,
w/defaults being compatible w/prior releases. See the API documentation for
available entries.
- FE2_CODEC_H264, add FE2_H264_SPEED parameter. Default changed from 0 to 1
improving quality on some clips which exhibited extreme degradation.
- FE2_FILTER_SCALE, add further reserved values allowing resolution to be
constrained to multiples of 2,4,8,etc. See the API documentation for further
details regarding their use.
Bug Fixes:
- FE2_CODEC_AMR_NB, correct output timestamp calculation avoiding potential
encode failure due to memory constraints
- Upgraded to libpng-1.2.35. Addresses CVE-2009-0040.
For more information, see:
 http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0040
```

```
 http://www.libpng.org/pub/png/libpng.html
- (Linux) Modify encode completion procedure improving stability for those
seeing crashes at that point. Originally reported under RHEL 4.
+---------------------------+
2009-02-04 v8.0.14.1 Linux
2009-02-04 v8.0.14.1_DEMO Linux
2009-02-04 v8.0.14.1 Windows COM v0.0.7.0
2009-02-04 v8.0.14.1_DEMO Windows COM v0.0.7.0
Bug Fixes:
- FE2_MUXER_3GP/FE2_MUXER_3G2/FE2_MUXER_MOV/FE2_MUXER_MP4, write edit list
containing empty edit for streams w/non-zero start time correcting sync
- FE2_CODEC_AAC/FE2_CODEC_AACPLUS/FE2_CODEC_LAME, correct timestamp
calculation. Fixes drift on long form content, esp. visible w/FE2_CODEC_AAC.
- FE2_MUXER_FLV/FE2_MUXER_MP4 and their relatives, adjust non-monotone
timestamps avoiding flvcheck/potential encode failure
- (Linux) fix potential hang should mencoder exit unexpectedly during open
- (Linux) mencoder,
 - new patches:
   o flv-r16254_backport.diff - bug fixes/feature additions from upstream,
   namely: speex and >2GB file support and correction of timestamp reporting
   o workaround_libswscale_bgr15_confusion.diff - temporary workaround, as
   libswscale has undergone extensive changes upstream, for issues related to
   codecs producing BGR15 output. These would have red and blue swapped in
   prior releases.
 - removed patches:
   o mov_06_invalid_audio_size.diff - patch was rendered unnecessary after
   v8.0.13.0 and in certain cases would result in disabling otherwise valid
   audio tracks. Remaining mov patches renamed to reflect reduction.
+---------------------------+
2008-12-02 v8.0.14.0 Linux
2008-12-02 v8.0.14.0_DEMO Linux
2008-12-02 v8.0.14.0 Windows COM v0.0.7.0
2008-12-02 v8.0.14.0_DEMO Windows COM v0.0.7.0
Features:
- (Linux) NOTE:
http://support.on2.com/flixengine/linux/api/lb_deviation.html#update
applies to this release
- (Windows) NOTE:
http://support.on2.com/flixengine/windows/api/group__flixengine__com.html#warn_dualinterface
applies to this release
- add FE2_MUXER_FXM
    New header file: muxers/fxm.h
Bug Fixes:
- silently add FE2_FILTER_RESAMPLE if necessary to ensure
FE2_CODEC_AAC/FE2_CODEC_AACPLUS configuration success, as is done
w/FE2_MUXER_FLV
- (Linux) mencoder,
 - updated patches
   o mov_03_moof_fragments.diff: Do not rebuild sample index when no fragments
   are present. This avoids large memory consumption and potential failure
   w/long constant sample size clips.
- (Linux) FE2_FILTER_CUT, ensure proper stream interleaving when discarding
frames due to FE2_CUT_START_SEC and FE2_CUT_USE_SEEK=0. Prevents encode failure
w/certain clips due to memory constraints (flixerrno=-2).
- (Windows) flvsplit.dll v1.6.6.0, fix bug causing FLV files w/PCM to fail
- (Windows) update image eos check added in v8.0.10.2 to avoid failure on 1
frame files and the addition of 1 frame w/animated GIFs
+---------------------------+
2008-10-06 v8.0.13.0 Linux
2008-10-06 v8.0.13.0_DEMO Linux
2008-10-06 v8.0.13.0 Windows COM v0.0.6.0
2008-10-06 v8.0.13.0_DEMO Windows COM v0.0.6.0
Features:
- (Linux) NOTE:
http://support.on2.com/flixengine/linux/api/lb_deviation.html#update
applies to this release
- (Windows) NOTE:
```

```
http://support.on2.com/flixengine/windows/api/group__flixengine__com.html#warn_dualinterface
applies to this release
- FE2_FILTER_ADAPTIVE_DEINTERLACE, add deintmode_t enum for consistency
w/other filters & codecs
- add FE2_FILTER_DENOISE
    New header file: filters/denoise.h
- add FE2_FILTER_BLUR, FE2_FILTER_SHARPEN
    New header file: filters/{blur,sharpen}.h
- add FE2_FILTER_MIRROR, FE2_FILTER_ROTATE
    New header file: filters/{mirror,rotate}.h
- (Linux) mencoder, update snapshot to r27474
   Additional differences may be obtained from the MPlayer
   changelog/subversion logs (svn://svn.mplayerhq.hu/mplayer/trunk), aside
   from overall stability and decoder/demuxer updates, notably addresses/adds:
   o Nellymoser, Musepack SV8 audio decode
 - new patches:
   o ad_faad-restrict_aac_probe_adts.diff: restrict data probe prior to
   libfaad init allowing additional AAC streams to successfully decode
   o demux_lavf_02_remove_mov_preferred.diff: set legacy 3GP/3G2/MP4/MOV
   demuxer (libmpdemux) to default for backward compatibility until lavf
   demuxer's performance can be assessed
   o reduce_spurious_logging.diff: combine vo_fonts_warn_msg_level.diff &
   vd_incompatible_vo_warn_msg_level.diff, add ffmpeg logging initialization
   to mencoder.c
   o mplayer_demux_real.patch: addresses security advisories
   oCERT-2008-013 / CVE-2008-3827, the potential for heap overflow
   within the real demuxer.
   For more information, see:
   http://www.ocert.org/advisories/ocert-2008-013.html
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3827
 - updated patches:
   o mencoder_03_joint_av_eof.diff: combine
   mencoder_03_audio_eof.diff/mencoder_09_joint_audio_video_eof.diff
   o mencoder_06_asf_pts_passing.diff, mencoder_07_demux_update_pts.diff,
   mencoder_08_decoded_before_dups.diff, mov_08_tkhd_matrix_scale.diff,
   mov_09_increase_sample_pts_size.diff: rename based on patch reduction
   o mov-ignore_reference_trak.diff: ignore tracks that reference external
   files, avoiding potential decode errors caused by non-existent data
 - removed patches:
   o flv_02_r12184_LE_codec.diff, flv_handle_bad_header.diff,
   mencoder_06_conf.diff, mov_07_missing_stsd.diff: rendered unnecessary
Bug Fixes:
- FE2_FILTER_CROP, do not require both right and bottom to be set, consistent
w/documentation.
- FE2_FILTER_FRAMERATE/FE2_FILTER_PNGEX, fix bug causing duration to be
applied to incorrect frame resulting in incorrect transitions. seen esp.
w/text frames.
- (Linux) mencoder-muxer_ogg.c, discard frames w/neg. timestamps. allows
transcoding of certain WMV files to mov,mp4,etc., though in some cases
sync issues may still remain.
- (Linux) mencoder-muxer_ogg.c, treat 0 byte audio frame as indication of eos
avoiding transcode failure due to ErrFileIO w/certain clips.
NOTE: in some cases this is caused by multi-trak audio (MOV), the trailing
audio will be missing as this is currently unsupported.
- (Linux) FE2_CODEC_VP6, fix chroma problem caused by input w/an odd width
- (Windows) flvsplit.dll (v1.6.2.0) expose FFDShow compatible Nellymoser
output pin allowing audio to be transcoded
+--------------------------+
2008-07-16 v8.0.12.0 Linux
2008-07-16 v8.0.12.0_DEMO Linux
2008-07-16 v8.0.12.0 Windows COM v0.0.5.0
2008-07-16 v8.0.12.0_DEMO Windows COM v0.0.5.0
Features:
- FE2_MUXER_3GP/FE2_MUXER_3G2/FE2_MUXER_MOV/FE2_MUXER_MP4, add
FE2_ISOMEDIA_FASTSTART parameter allowing 'moov' box placement to be
influenced, enabling progressive download.
- FE2_MUXER_FLV, add support for FE2_CODEC_AAC/FE2_CODEC_AACPLUS.
```

```
See API documentation for limitations.
Bug Fixes:
- FE2_MUXER_3GP/FE2_MUXER_3G2/FE2_MUXER_MOV/FE2_MUXER_MP4, avoid using fixed
framerate as video timebase. avoids failure should observed disagree
w/reported, esp. w/variable framerate input.
- FE2_CODEC_H263/FE2_CODEC_H263_BASELINE, should configuration of 2nd pass
fail attempt a single pass avoiding outright encode failure.
- FE2_FILTER_OVERLAY, fix crash when using gray scale PNG w/bit depth < 8
- (Linux) mencoder,
 - new patches:
   o asfhdr_correct_movielength.diff: correct duration calculation by using
   double precision as well as correct units for pre-roll (ms).
 - updated files/patches:
   o muxer_ogg.c,oggpcm.c: re-base timestamps guaranteeing one stream starting
   at 0. avoids unnecessary startup delays and at times A/V sync issues esp.
   w/ASF/WMV.
- (Linux) upgraded to SWIG-1.3.35 for language binding generation. Fixes issue
seen w/PHP language bindings, from swig-1.3.35/CHANGES:
        [PHP5] If ZTS is enabled, release <module>_globals_id in MSHUTDOWN
        to avoid PHP interpreter crash on shutdown.  This solution was
        suggested here: http://bugs.php.net/bug.php?id=40985
Other language bindings are unaffected.
- (Linux) flixd, all components respect TMPDIR env. var. if set
- (Windows) flixengine_com.jar, resolve issue resulting in leaked interface
handle as well as causing input file to remain locked should encode not be
called.
- (Windows) flvdecvp6.dll (v1.4.0.0) fix incorrect -- top rather than bottom
-- crop w/height non-multiple of 16
- (Windows) On2QTSource.dll (v1.3.0.2),
  o properly handle clips w/odd widths avoiding output stride
  problems. ('vertically skewed' output)
  o allow muxed MPEG-1 movies to be rendered by traditional filters
  duplicating behavior of <=8.0.10.2
+---------------------------+
2008-05-16 v8.0.11.0 Linux
2008-05-16 v8.0.11.0_DEMO Linux
2008-05-16 v8.0.11.0 Windows COM v0.0.4.0
2008-05-16 v8.0.11.0_DEMO Windows COM v0.0.4.0
Features:
- FE2_CODEC_VP6, updated encoder. provides increased quality across profiles,
esp. in high motion sequences.
Bug Fixes:
- upgraded to libpng-1.2.27.
  Along w/various bug fixes addresses CVE-2008-1382.
  For more information, see:
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1382
  http://www.libpng.org/pub/png/libpng.html
- (Linux) mencoder,
 - new patches:
   o mov_06_invalid_audio_size.diff - disable audio tracks w/less data than
   reported in track header. avoids encoding timeout (Flix2_Errno -2)
   o mov_08_tkhd_matrix_scale.diff - set aspect ratio based on track's
   transformation matrix
   o mov_09_increase_sample_pts_size.diff - avoid timestamp rollover on long
   material or material in which timestamps increase w/large scale.
 - updated patches:
   o codecs_conf_custom.diff - prefer FFmpeg DV decoder over qdv for DVSD
   contained w/in AVI files.
   o mencoder_08_demux_update_pts.diff - restrict change to ASF/WMV files
   which it is intended to fix. avoids potential sync issues w/other media.
   o mov_01_edl_kf_search.diff
- (Windows) On2QTSource.dll (v1.2.7.1), improved support for:
  o multi-track video files
  o audio streams w/either >2 channels or samplerate >48kHz
+---------------------------+
2008-04-17 v8.0.10.2 Linux
2008-04-17 v8.0.10.2_DEMO Linux
```

```
2008-04-17 v8.0.10.2 Windows COM v0.0.4.0
2008-04-17 v8.0.10.2_DEMO Windows COM v0.0.4.0
Bug Fixes:
- FE2_MUXER_FLV, fix bug introduced in v8.0.10.1, in to changes made to pad
clips w/truncated audio, resulting in canSeekToEnd metadata field being set to
0 in most cases.
- adjust default encode timeout added in v8.0.9.0 (Windows)/v8.0.10.0 (Linux)
to 60000ms from 15000ms. helps avoid spurious failures b/w passes while under
heavy load. the FLIX_HUNG_ENCODE_TIMEOUT env. var may be used for further
adjustment. value is in ms, 0=no timeout.
- FE2_CODEC_VP6ALPHA, add pixel normalize to help avoid loss of transparency
seen when re-encoding alpha content.
- FE2_CODEC_VP6, fix codec bug causing HD clips at high bitrates (>~5000KiB)
under VBR_2PASSControl to greatly overshoot the data rate.
- correct appl. of forced FE2_FILTER_RESAMPLE due to sample rate/channel
restrictions in output. prior versions would fail should the user specify one,
but not the other (required) param.
- (Linux) in addition to encode timeout modifications above, modify
mencoder std(out|err) output handling to avoid spurious timeouts caused by
heavy logging on either.
- (Linux) libogg, check all memory allocation return values returning an error
to caller on failure. this coupled w/FLIX_OGG_PHYMEM_PCTMAX further increases
flixd stability w/malformed input.
- (Linux) libogg, allow audio timestamp to be forwarded to the application
- (Linux) mencoder,
 - new patches:
   o asfhdr_use_best_stream.diff - selects best (highest bitrate) available
   stream for decode. fixes poor output quality in certain ASF/WMV files.
   o avi_check_idxflags.diff - better AVIINDEXENTRY flag support, fixing clip
   truncation (AVI) in certain cases.
   o codecs_conf_custom.diff - combines local codecs.conf changes to one patch
   o demux_lavf_add_dv_preferred.diff - improved DV file support
   o flv_handle_bad_header.diff - upon discovery enables stream even if header
   does not report it. addresses missing audio/video in clips of this nature.
   o flv_02_r12184_LE_codec.diff - back port of PCM fix which correctly
   identifies endianness fixing static output
   o mp3lib_increase_maxframesize.diff - improves success rate of clips
   containing MP3 streams (avi,mp3,etc.). previous versions failing
   w/Flix2_Errno(-2,2) or, w/mp3 files, no indication at all.
   o mencoder_07_asf_pts_passing.diff, mencoder_08_demux_update_pts.diff,
   demux_asf_use_avg_fps.diff - collectively these patches address sync issues
   w/certain ASF/WMV files by forwarding the decoded timestamps.
   o mencoder_09_joint_audio_video_eof.diff - holds encoding completion until
   both streams have finished. fixes clip truncation in certain cases.
   o mencoder_10_decoded_before_dups.diff - waits to discard leading duplicate
   frames until a successful decode occurs. addresses video streams reporting
   dimensions of 0x0 in certain cases.
   o mov_07_missing_stsd.diff - skip malformed stream descriptions allowing
   transcode to complete successfully w/MOV files of this type.
 - updated patches:
   o mov_03_moof_fragments.diff - changes for file format conformance,
   improving success rate of select MOV files.
 - removed patches:
   o codecs_ffwmav2_ffsvq3_buggy.diff, hdv1_use_libmpeg2.diff,
   vp6[01]_use_vfw.diff, wmv_remove_yv12.diff - replaced by
   codecs_conf_custom.diff
   o dec_audio_dmo_pts_passing.diff - replaced
   w/mencoder_08_demux_update_pts.diff
- (Linux) upgraded to libpng-1.2.23. issues referenced under Windows apply.
- (Windows) modify end of stream check w/animated image files (e.g., GIF)
avoiding hang.
NB: current implementation ignores any loop count information stored w/in the
file and encodes supplied frames only once.
- (Windows) fix timeout when attempting to encode certain WMV files containing
'MSS2' video, created by Microsoft Office LiveMeeting 2005 among others.
- (Windows) on license failure, when running as a service, a message will be
written to the Application Event Log. avoids hang when attempting to display a
```

```
message box to a non-visible window station.
NB: the above assumes the user account the service is run as has write access
to same. see: http://support.microsoft.com/kb/842209
- (Windows) upgraded to libpng-1.2.25
This addresses a crash bug (v1.2.21) when reading the ICC-profile chunk, iCCP
(CVE-2007-5267). In addition, versions 1.2.20 and earlier have a number of
potential crash-bugs due to out-of-bounds reads in certain chunk-handlers;
MITRE has collectively assigned them the identifiers CVE-2007-5266,
CVE-2007-5268 and CVE-2007-5269.
+--------------------------+
2007-12-14 v8.0.10.1 Linux
2007-12-14 v8.0.10.1_DEMO Linux
2007-12-14 v8.0.10.1 Windows COM v0.0.4.0
2007-12-14 v8.0.10.1_DEMO Windows COM v0.0.4.0
Bug Fixes:
- FE2_MUXER_FLV, pad clips w/truncated audio stream allowing playback to fully
  complete, avoiding potential hang in many third party players.
- FE2_CODEC_H264, add preliminary iPhone support. see
  group__codec__h264.html#h264_appledevices in the API documentation for further
  details.
- FE2_FILTER_PNGEX, fix regression causing lost thumbnails under 2 pass,
  w/clips misreporting their duration. observed duration is now used.
- FE2_FILTER_ADAPTIVE_DEINTERLACE, correct placement in filter chain fixing a
  case where FE2_FILTER_SCALE could precede it depending on _AddFilter order,
  resulting in less than optimal output.
- FE2_CODEC_AMR_NB, correct bitrate adjustment to avoid possible codec open
  failure due to an unsupported value, evidenced by Flix2_Errno==-7 post
  encode.
- (linux) mencoder, new patches:
   o codecs_ffwmav2_ffsvq3_buggy.diff - prefer wma dmo to avoid audio
     corruption (static,etc.) across transitions in asf/wmv files. evident
     w/clips made w/Windows Movie Maker among others.
   o dec_audio_dmo_pts_passing.diff - addresses sync issues w/various asf/wmv
     files.
   o mencoder_05_mov_lavf_pts_var_fps.diff, mov_05_ffmpeg_bitrate.diff -
     addresses sync issues w/various mov files.
   o mencoder_06_conf.diff - allows use of a system-wide mencoder.conf as
     described in the mencoder man page under 'CONFIGURATION FILES'.
- (linux) mencoder, modified mov_01_edl_kf_search.diff to use both the audio
  and video edit lists if available. addresses various sync issues seen
  w/mov files.
- (linux) mencoder, enable hdv1 support
- (windows) qtsource.dll (v1.2.4.0) now preserves the alpha channel when
  encoding using FE2_CODEC_VP6ALPHA.
- (windows) fix some DirectShow related memory leaks
- (windows) flvsplit.dll (v1.5.5.2), give preference to sampling rate
  contained w/in mp3 data itself rather than relying on flv tags. addresses
  data timeout (ErrFileIO) w/same.
+--------------------------+
2007-10-08 v8.0.10.0 Linux
2007-10-08 v8.0.10.0_DEMO Linux
2007-10-08 v8.0.10.0 Windows COM v0.0.4.0
2007-10-08 v8.0.10.0_DEMO Windows COM v0.0.4.0
Features:
 - add FE2_MUXER_3G2
   NOTE: Only available if this feature has been added to your license.
         Contact sales@on2.com for further details.
 - add FE2_MUXER_MOV, FE2_MUXER_MP4
     Renamed header file: muxers/3gp.h -> muxer/isomedia.h
 - add FE2_CODEC_AAC, FE2_CODEC_AACPLUS, FE2_CODEC_H264
     New header files: codecs/{aac,h264}.h
 - add FE2_VP6_PROFILE, allows selection of VP6-S in addition to (default)
   VP6-E. See API documentation for further details.
 - add FE2_LAME_RC_MODE, allows selection of ABR/VBR in addition to
   legacy (default) CBR.
 - (linux) update mencoder snapshot to r24143.
   A full list of differences can be obtained via the subversion logs
```

```
   (svn://svn.mplayerhq.hu/mplayer/trunk), notably addresses:
     o AC3 setup regression introduced prior to r22906 causing loss of audio
       in output
     o some MJPEG related decode errors
     new patches:
     o mencoder_03_audio_eof.diff - avoids truncated (video) output due to
       missing audio track data
     o mencoder_05_lavf_var_fps.diff - addresses sync/duration issues seen
       w/variable framerate flv files.
     o ffmpeg_correct_dimensions.diff - addresses cropped output in
       MPG/MP4/MOV files
     o mov_01_edl_kf_search.diff - updated to avoid video frame duplication
       (entire video track sans audio appearing after clip completion)
     o demux_mpg_short_video_fix.diff - correctly reports duration on
       extremely short clips (<1s)
 - (windows) add ColdFusion CGI sample
Bug Fixes:
 - FE2_CODEC_VP6, avoid output artifacts w/files encoded using
   FE2_VP6_CONCURRENCY and FE2_VP6_TEMPORAL_RESAMPLING caused by keyframes not
   being produced at FE2_VP6_KFFREQ.
 - attempt to close input file at encode completion. avoids potential hang
   when dealing w/corrupt input (often seen under windows, esp. files decoded
   by QuickTime). additionally, under windows, obviates forced garbage
   collection upon successful completion, previously necessary to guarantee
   timely destruction of the COM object thereby closing the input. note that
   should the encode fail due to a hung filter at close the open file handle
   will remain.
 - (linux) add input timeout similar to windows (8.0.9.0) to avoid corrupt
   files from hanging encode thread.
 - (windows) update flvdecvp6.dll, addresses decoder memory leak
 - (windows) compile flixengine_com.jar w/-source 1.5 option allowing for use
   under older versions of Java w/o a recompile.
   Note: com4j, used to generate flixengine_com.jar, utilizes 1.5 features
   therefore further compatibility changes are unlikely.
+---------------------------+
2007-07-27 v8.0.9.1 Linux
2007-07-27 v8.0.9.1_DEMO Linux
2007-07-27 v8.0.9.1 Windows COM v0.0.3.0
2007-07-27 v8.0.9.1_DEMO Windows COM v0.0.3.0
Bug Fixes:
 - FE2_CODEC_VP6, should initialization of FE2_VP6_CONCURRENCY=1 fail due to
   memory constraints, fall back to 0.
 - FE2_FILTER_PNGEX, allow multiple instances as was possible w/8.0.8.2.
 - FE2_FILTER_PNGEX, avoid generating (many) additional PNGs when used in
   conjunction w/FE2_FILTER_FRAMERATE.
 - (linux) flixd, enable --interface to accept numeric IPv4 address or device
   name
 - (windows) update qtsource.dll to v1.0.2.2, fixes out of sync transcode due
   to clips w/variable framerate and/or non-zero audio start time.
 - (windows) fix bug causing audio decode failure in WMV files (observed under
   win2k3)
+---------------------------+
2007-06-21 v8.0.9.0 Linux
2007-06-21 v8.0.9.0_DEMO Linux
2007-06-21 v8.0.9.0 Windows COM v0.0.3.0
2007-06-21 v8.0.9.0_DEMO Windows COM v0.0.3.0
Features:
 - FE2_CODEC_VP6, add FE2_VP6_CONCURRENCY parameter. Allows the encode
   process to take advantage of multiple cores/processors yielding a
   potentially significant gain in performance. See the API documentation for
   further details.
 - FE2_FILTER_SCALE, add support for reserved values [-3,0] to allow aspect
   ratio to be maintained. See the API documentation for further details
   regarding their use.
 - (windows) add Java command line sample. See the API documentation for usage
   notes.
Bug fixes:
```

```
 - FE2_MUXER_SWF, allow audio only SWF creation.
 - encoding_status_PercentComplete, avoid reporting >50% on first pass should
   duration be misreported.
 - upgraded to libpng-1.2.18
   This addresses a NULL-pointer-dereference vulnerability involving palette
   images with a malformed tRNS chunk (i.e., one with a bad CRC value). This
   vulnerability is given the identifiers VU#684664 and CVE-2007-2445 by CERT
   and MITRE, respectively.
   For more information, see:
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2445
 - FE2_FILTER_PNGEX, should end-of-stream be encountered and not all requested
   PNGs have been written (due to misreported duration) output 1 final PNG.
 - FE2_FILTER_PNGEX, when using FE2_PNGEX_AUTO_EXPORT_COUNT and clip duration
   cannot be determined output the first FE2_PNGEX_AUTO_EXPORT_COUNT
   frame(s), as opposed to none at all.
 - (linux) mencoder, add preliminary support for movie fragments contained in
   file formats based on ISO-14496 Base File Format, most notably 3GPP2 (.3g2)
   files. addresses clip truncation.
 - (linux) mencoder, add patch (avi_forceidx_fallback.diff) forcing index
   regeneration should the reported number of frames not match the number of
   index entries. addresses various duration issues for this format.
 - (linux) mencoder, update muxer_ogg to allow delayed stream setup. addresses
   missing video in certain mpeg based streams when info is unavailable prior
   to decode.
 - (linux) remove scale=0:0 mencoder filter as prescaling interlaced material
   results in output artifacts. FE2_FILTER_SCALE can now be used to achieve
   similar results.
 - (windows) fix color convert bug causing video encode to fail (ErrEncodeV)
   should the source have an odd width.
 - (windows) add decode timeout to avoid hung encode session due to corrupt
   input file. most visibly addresses hangs caused by wmv decoders supplied
   with windows media player 9/10, the latter being the release currently
   available under win2k3. Flix2_Errno will be set to -2 in this case. this
   change only adds a timeout for data production by DirectShow and will not
   correct hangs related to destroying the decode graph (currently seen with
   certain QuickTime files).
 - (windows) FE2_FILTER_PNGEX, enable support for '/' delimited paths. fixes
   incorrect PNG name generation resulting in misplacement or outright failure
   should FE2_PNGEX_(OUTPUT_DIRECTORY|FILENAME_PREFIX) be left unset and the
   output filename be '/' delimited.
+---------------------------+
2007-04-13 v8.0.8.2 Linux
2007-04-13 v8.0.8.2_DEMO Linux
2007-04-13 v8.0.8.2 Windows COM v0.0.2.0
2007-04-13 v8.0.8.2_DEMO Windows COM v0.0.2.0
Features:
 - add optimized scaling code to FE2_FILTER_SCALE improving encode time 25-35%
   depending on the input. greatest gains are seen when scaling large input to
   comparatively smaller size, e.g., 720x480->256x192.
 - (linux) updated mencoder snapshot to r22906.
   A full list of differences can be obtained via the subversion logs
   (svn://svn.mplayerhq.hu/mplayer/trunk), notably addresses:
   o buffer overflow in DirectShow loader.
     For more information, see:
     http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1246
     http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1387
   o artifacts when transcoding flvs w/height a non-multiple of 16
   o framerate issues (>actual causing fast playback) w/certain input flvs
   o inability to load drvc.so and/or atrc.so.6.0 (Real binary codecs) necessary
     for certain Real input clips. This build now selectively statically links
     prerequisite libraries leaving base system libraries as dynamic.
     requirements are similar to flixd so should not affect the binary's
     ability to load. should this become an issue and a fully static binary is
     required, the sources are available via: http://www.on2.com/gpl/mplayer
Bug fixes:
 - fix crash in FE2_FILTER_RESAMPLE caused by the demuxer producing 0 byte
   audio frames. this was only observed under windows but had the potential of
```

```
   causing a crash on either platform.
 - create key frame when navigation cue point(s) are set with FE2_FLV_CUEPT_NAV
 - fix encoder hang when specifying an explicit framerate
   w/FE2_FILTER_FRAMERATE. in addition to the fix made in v8.0.8.1 there was
   another case when using either FE2_FILTER_ADAPTIVE_DEINTERLACE or
   FE2_FILTER_CROP, or should the input frame need to be extended or
   color converted (windows only), that could cause a hang to occur.
 - (linux) revert LAME library to 3.96.1. w/certain input files 3.97 currently
   produces encode errors.
 - (linux,mencoder) allow mean framerate patch to only increase output
   framerate. avoids artifacts when decoding certain h264 clips (mov, etc).
 - (windows) update flvsplit.dll to v1.5.3.0, ignores flv header
   information indicating presence of audio, manually inspecting the data,
   allowing for audio extraction from malformed files.
 - (windows) update qtsource.dll to v1.0.2.0, addresses incorrect framerate
   calculation causing file to be transcoded at low framerate.
 - (windows) forward base timestamp from the demuxer through the codec. allows
   for material w/non-zero start times to remain in sync after transcode.
 - (windows) fix color conversion bug producing green bar in videos
   having an input width a non-multiple of 4.
+--------------------------+
2007-03-09 v8.0.8.1 Windows COM v0.0.2.0
2007-03-09 v8.0.8.1_DEMO Windows COM v0.0.2.0
2007-03-09 v8.0.8.1 Linux
2007-03-09 v8.0.8.1_DEMO Linux
Bug fixes:
 - fix crash when removing unnecessary (i.e., desired==input)
   FE2_FILTER_RESAMPLE
 - fix encoder hang when specifying an explicit framerate > source with
   FE2_FILTER_FRAMERATE
 - (windows) fix bug causing loss of audio at input end. amt. varied based on
             the clip w/as much as 8s loss possible.
 - (linux) updated 64bit libflixengine2.so to allow for use on platforms whose
           binutils and/or dynamic linker/loader do not have .gnu.hash
           support, as evidenced by output similar to:
           '/usr/local/lib64/libflixengine2.so: file not recognized: File
           format not recognized'
+--------------------------+
2007-02-26 v8.0.8.0 Windows COM v0.0.2.0
2007-02-26 v8.0.8.0_DEMO Windows COM v0.0.2.0
2007-02-26 v8.0.8.0 Linux
2007-02-26 v8.0.8.0_DEMO Linux
Features:
 - upgrade to lame-3.97
 - add FE2_LAME_QUALITY parameter to FE2_CODEC_LAME
 - added muxer interface
     New API calls: Flix2_MuxerGetParam/AsStr, Flix2_MuxerSetParam/AsStr
                    Flix2_AddMuxer/Flix2_RemoveMuxer
     New header files: muxer_constants.h, muxers/3gp.h,
                       muxers/flv.h, muxers/swf.h
     NOTE: This update has deprecated many of the muxer related functions in
           (swf|video)_options.h. Consult the deprecated list in the API
           documentation as they will be removed in a future release.
 - add FE2_MUXER_3GP. See the API documentation for limitations.
     NOTE: Only available if this feature has been added to your license.
           Contact sales@on2.com for further details.
 - add FE2_CODEC_H263_BASELINE for use in conjunction with the 3gpp muxer.
 - add FE2_CODEC_AMR_NB for use in conjunction with the 3gpp muxer.
 - (windows) add ASP, ASP.NET and VB(6|.NET) samples
 - (linux) add libtheora-1.0alpha7 (http://www.theora.org/) to mencoder build
 - (linux) add support for FLIX_OGG_PHYMEM_PCTMAX environment variable used to
           control individual ogg demuxer's max memory usage to avoid a
           segmentation fault due to unchecked memory allocations within
           libogg. The default set in flixd's start script is 70. See the
           flixd man page for further details.
Bug fixes:
 - modify FE2_FILTER_RESAMPLE to use FFmpeg based filter. addresses sync
```

```
   issues when resampling longer content (esp. non-integer conversions, e.g.,
   48->44.1kHz) as well as the issue of giving precedence to the left channel
   when converting stereo to mono.
 - fix bug causing audio and video to be improperly interleaved in certain
   cases, mostly when dealing with clips that were variable framerate (wmv). a
   video frame with a timestamp much greater than the current audio would be
   inserted followed by a large amount of (earlier timestamped) audio causing
   playback and FMS issues.
 - fix bug in FE2_CODEC_LAME causing the frame prior to the final block to be
   duplicated. depending on the content this may have produced an audible
   distortion in the output. in addition, break final mp3 block into individual
   frames to avoid storing one (potentially large) block at the end of the file.
 - fix bug, most evident in the demo version, that could cause a crash
   should the crop filter be used to adjust the width of the input.
 - remove video height restriction from overlay filter. prior versions
   required the height to be divisible by 2.
 - (windows) On2QTSource.dll,v1.0.1.3
              o updated to handle the mov transformation matrix allowing for
                the correct output size/viewport.
              o avoid potential hang on shutdown, seen most often when using
                FE2_CUT_STOP_SEC or explicitly disabling stream.
 - (windows) add support for 8bit audio input (converted to 16 as linux)
 - (windows) fix bug related to shared 1st pass file w/VP6 2pass resulting in
             incorrect output when running concurrent encodes
 - (windows) fix bug causing the alpha channel to appear inverted when using
             CODEC_VP6ALPHA.
 - (linux) avoid flixd hang due to unconstrained mencoder error output on open
+---------------------------+
2006-12-13 v8.0.7.1 Windows COM v0.0.1.0
2006-12-13 v8.0.7.1_DEMO Windows COM v0.0.1.0
2006-12-13 v8.0.7.1_DEMO Linux
2006-12-13 v8.0.7.1 Linux
 - fix bug in cut filter when setting FE2_CUT_STOP_SEC and using
   FE2_CUT_USE_SEEK along with a non-zero FE2_CUT_START_SEC causing the output
   to have an incorrect duration. The stop time is now adjusted accordingly
   should the seek succeed.
 - (linux) add 64bit version of libflixengine2.so, the client-side rpc lib, to
           allow for 64bit language bindings to be built.
           NOTE: Support has yet to be added to the installer.
           libflixengine2.so must be extracted and installed by hand.
           From a running installer the file is found under:
           $HOME/flix-engine-installer.<pid>/.flix-engine-installation-files/testing/lib64
           Install the desired language binding sources but 'skip' the build
           during install to avoid errors.
           Tested under FC6 running on an amd64x2. As this is currently under
           development, support will not be provided for installing this
           library.
 - fix bug in resample filter (upsampling) causing clicks/pops in the output.
   rather than out ranging, in this case the problem stemmed from over reading.
 - (windows) add FE2_PNGEX_(WIDTH|HEIGHT) to the COM interface
 - upgraded to libpng-1.2.14. This fixes a bug where a specially crafted PNG
   file could cause a crash. For more information, see:
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5793
+---------------------------+
2006-11-30 v8.0.7.0 Windows COM v0.0.0.1
2006-11-30 v8.0.7.0_DEMO Windows COM v0.0.0.1
 - (windows) correct errant reference count within On2QTSource.dll (QuickTime
             source filter) allowing the filter to be properly unloaded,
             releasing references to the input file
2006-11-29 v8.0.7.0_DEMO Linux
2006-11-29 v8.0.7.0 Linux
 - (linux) add mencoder patch to calculate mean framerate w/in mov and related
           files. Addresses issue w/3gp files reporting incorrect (and often
           low) values.
 - add FE2_PNGEX_(WIDTH|HEIGHT) parameters to PNG exporter allowing the image
   to be scaled
 - add FE2_PNGEX_COMPRESSION_LEVEL parameter to PNG exporter
```

```
 - add support for frame rate filter, FE2_FILTER_FRAMERATE, via the filter api.
     New header file: filters/framerate.h
     NOTE: This has deprecated:
           video_options_(G|S)et(VideoFramerateAsDouble|
                                 DecimateValue|UseSourceFramerate)
 - (windows) fix object creation to allow for instances of IFlix to be created
             across multiple threads. close references to flixengine.lic when
             they are unneeded, addresses file descriptor leak.
 - correct frame rate calculated by frame rate filter when using decimate.
   previous versions would correctly decimate the input, but report the source
   frame rate, adversely affecting video quality.
 - (windows) correct PNG export filter to accept '\' as the path name
             separator. fixes a potential crash when either
             FE2_PNGEX_DIRECTORY or FE2_PNGEX_FILENAME_PREFIX were left unset.
 - add FE2_CUT_USE_SEEK parameter. If set and FE2_CUT_START_SEC is non-zero
   attempts to seek the input instead of decoding/discarding all leading data,
   improving processing time. This is now the default. Set to 0 for legacy
   behavior.
 - fix bug in resample filter causing 'clicks'/'pops' in the output due to
   out ranging of the samples
 - fix bug in resample filter causing distorted output
   when the number of input and output channels differed
+---------------------------+
2006-10-24 v8.0.6.0_DEMO Windows
2006-10-20 v8.0.6.0_DEMO Linux
2006-10-20 v8.0.6.0 Linux
 - (linux) mencoder
   o snapshot updated to r20146, adds (most notably):
      o VP6 flv import
      o QCELP support
   o add amr_[nw]b support
   o patch to support mov files w/audio samplerate of 5512Hz
     (produced by various cellphones)
   o correct mencoder identify patch to provide more correct clip info
     if available. addresses incorrect/unknown stream duration being
     reported with some mpeg files.
 - (linux) when installing java bindings, symlink javadoc to the doc install
           directory, default: /usr/local/share/doc/on2
 - (linux) sync ffmpeg snapshot to that used under windows (r5990)
+---------------------------+
2006-09-29 v8.0.6.0 Windows COM v0.0.0.0
 - fix possible encoder crash when transcoding low framerate clips
 - upgraded to libpng 1.2.12
 - fix bad chroma in png's generated from input of odd width/height
 - exposed FE2_CODEC_PCM to complete audio_options_SetFlvAudioFormat deprecation
 - add support for brightness/contrast/hue/saturation filter, FE2_FILTER_BCHS,
   via the filter api
     New header file: filters/bchs.h
     NOTE: This has deprecated:
           editor_options_(G|S)et(Brightness|Contrast|Hue|Saturation) and
           editor_options_(G|S)etUse(Brightness|Contrast|Hue|Saturation)
 - add resample filter, FE2_FILTER_RESAMPLE
     New header file: filters/resample.h
     NOTE: This has deprecated: audio_options_(G|S)et(Samplingrate|Stereo)
 - fix misleading 100 returned from encoding_status_PercentComplete before
   encoding starts
 - fix small memory leak when using FE2_PNGEX_EXPORT_TIME_STRING
 - (windows) initial release of COM interface
```

Chapter 6

flixd man page

```
FLIXD(8)                                                    FLIXD(8)


NAME
       flixd - Flix encoding daemon

SYNOPSIS
       flixd  [ -d ] [ -p port ] [ -i interface ] [ -r ] [ --max-sessions snct
       ] [ --authdir path ] [ --pidfile file ] [ --logfile file ] [ --loglevel
       level ]

       flixd [ -h ] [ -v ]

DESCRIPTION
       flixd  is  an  rpc-based flix encoding server. As such, portmap MUST be
       running for it to operate. The port and interface to listen on  can  be
       specified  using  the  --port and --interface options. By default flixd
       will allow the rpc service to bind to any available port and interface.
       A port is usually specified to bypass the necessity for a connection to
       portmap by a client.

       By  default  flixd  will  allow  an  unlimited  (except  by   available
       resources)  number  of encoding sessions. This behavior can be modified
       by specifying the --max-sessions option with a count >0.  Any  sessions
       that start an encode after the count is reached will be queued.

       flixd  will  look for authentication files (flixengine.lic, provided by
       On2 at the time of install) in the user's home directory unless a  path
       is specified with the --authdir option. The default install location is
       $HOME/.on2.

       flixd can be stopped by sending it the TERM  signal.  The  pid  can  be
       stored for this purpose in a file by giving the --pidfile option.

       By default, when running as a daemon, flixd will log using syslog(3). A
       log file can be specified by using the --logfile option.  The amount of
       logging can be adjusted using the --loglevel option.


OPTIONS
       -h,--help
              Prints a summary of available command line options and exits.

       -v,--version
              Prints version information and exits.

       -d     debug mode; do not daemonize

       -p,--port port
              port to listen for rpc connections on (default: any)

       -i,--interface interface
              interface  to bind rpc socket to. Argument may be a numeric IPv4
              address or device name; --port MUST be specified when using this
              option (default: any available)

       -r,--reuseaddr
              Set SO_REUSEADDR socket option on listen socket. Only meaningful
              when used in conjunction with the --port option

       --max-sessions snct
              max number of concurrent encoding sessions (default: -1,  unlim-
              ited)

       --authdir path
              path  to  search  for flixd authorization files (default: user's
```

```
              home directory)

       --pidfile file
              write pid to file; e.g., /var/run/flixd.pid

       --logfile file
              file to log messages to (default:  debug  mode:  stderr,  daemon
              mode: use syslog(3))

       --loglevel level
              logging verbosity (0-4, 0: none, 1: errors, 2: status, 3: debug,
              4: heavy), the levels are cumulative. (default: 2, errors+infor-
              mational)


ENVIRONMENT
       flixd's behavior is affected by the following environment variables.

       FLIXD_MENCODER
              This  variable  specifies the path to mencoder that flixd should
              use during transcoding. By default flixd allows  the  system  to
              locate  mencoder  via  the  PATH environment variable. Therefore
              this variable is useful if you have installed mencoder to a non-
              standard  location  or  have  multiple  versions  of  mencoder
              installed. For  example,  if  you  have  installed  mencoder  to
              /opt/on2/flixengine/bin    you    would    set    FLIXD_MEN-
              CODER=/opt/on2/flixengine/bin/mencoder to ensure this version is
              used during transcoding.

       FLIX_OGG_PHYMEM_PCTMAX
              This variable controls the maximum amount of physical memory any
              one encode instance's ogg demuxer will use. This is to prevent a
              possible  segmentation fault due to unchecked memory allocations
              within libogg. This value is given as an integer percent,  e.g.,
              FLIX_OGG_PHYMEM_PCTMAX=50, will limit the demuxer's memory usage
              to 50% of available physical memory. If left unset or set  to  0
              the  only  upper  bound is that imposed by the system itself.  A
              reasonable target is 400-500MB.

       HOME   When --authdir is omitted HOME  is  used  as  the  authorization
              directory.

       TMPDIR Temporary file location. If unset /tmp is used.


COPYRIGHT
       2008 On2 Technologies, Inc.


ACKNOWLEDGEMENTS
       On2 Technologies flixd uses:

       o    FFmpeg (ffmpeg.mplayerhq.hu)

       o    LAME mp3 encoder (lame.sourceforge.net)

       o    libogg (xiph.org/ogg/)

       o    libpng (www.libpng.org)

       o    OpenSSL Toolkit (www.openssl.org)

       o    MEncoder (mplayerhq.hu)

       o    zlib (zlib.net)

       o      See  the  Flix  Engine  API  documentation  for  complete  list-
```

```
          ing/description

BUGS
       Email bug reports to flixsupport@on2.com.  Make  sure  to  include  the
       word(s)  'flixd' or 'flix engine linux' in the subject field. If possi-
       ble  attach  a  copy  of  the  install   log,   'flix-engine-installer-
       linux-<version>.bin.log',  or  include  the Flix Engine version and any
       pertinent system information in the message body.

SEE ALSO
       portmap(8)


flixd                              June 2008                        FLIXD(8)
```

Chapter 7

Samples

Sample usage of the Flix Engine API. The easiest way to test your installation is to run the C command line sample or the command line sample of your language of choice. By default samples are installed in */usr/local/share/on2/flixsamples*.

NOTE: To run the other samples you will have to have installed the appropriate language bindings.

- Overview
- C
  - Command Line
- Java
  - Command Line
- Perl
  - Command Line
  - CGI
- PHP
  - Command Line
  - CGI
- Python
  - Command Line
  - CGI

Chapter 8

Overview

```
==============================================================================
  Copyright (c) On2 Technologies Inc. All Rights Reserved.
------------------------------------------------------------------------------
  File:        $Workfile$
               $Revision$

  Last Update: $DateUTC$
------------------------------------------------------------------------------
Flix Engine Samples README

==============================================================================
[Overview]
 The flix engine consists of two main parts:
  - the rpc server/encoding daemon (flixd)
  - the rpc client library (libflixengine2.so).
 The samples use libflixengine2.so to start encoding sessions on the server.
 Language extensions are used to enable support for java, php, perl and python.
These consist of a wrapper .so that calls into libflixengine2.so.

==============================================================================
[Tree structure]
flixsamples/
 c/
 java/
 perl/
 php/
 python/

==============================================================================
[Sample overview]
c/
 cli_encode.c - c command line example
                usage: cli_encode <infile> <outfile>
                One of the simplest examples for testing your installation.
                See c/README for build instructions.
java/
 cli_encode.java - java command line example
                   usage: java cli_encode <infile> <outfile>
                   NOTE: You must build and install the java bindings
                   before building and running this example. After install the
                   necessary files should be in your classpath. Compilation can
                   be accomplished by running 'javac cli_encode.java'
perl/
 NOTE: You must build and install the perl module before running these
       examples.
       The CGI examples require the CGI perl module be installed. You can run:
       $ perl -e 'use CGI;'
       to verify you have this module. If you do not, it can be downloaded
       from CPAN (http://cpan.org).
 cli_encode.pl - perl command line example
                 usage: cli_encode.pl <infile> <outfile>
 flix2_sample.cgi
 process_sample.cgi - slightly more involved cgi example that browses for
                      uploaded files and allows the user to set most of the
                      engine's options with the corresponding API call. These
                      are separated into sections that map to the engine's
                      include files. To use this example navigate to
                      flix2_sample.cgi in your web browser, select a file from
                      the list (by default the script looks for files in
                      /var/www/cgi-bin/flixmedia/in), set any of the desired
                      options and click submit. The selected options are
                      submitted to process_sample.cgi, the default location for
                      the output file is /var/www/cgi-bin/flixmedia/out.
php/
 NOTE: You must build and install the php extension before running
       these examples
 cli_encode.php - php command line example similar to the java and perl
```

```
                  examples described above.
 flix2_sample.php
 process_sample.php - php cgi example similar to the perl example.
python/
 NOTE: You must build and install the python extension before running
       these examples
 cli_encode.py - python command line example similar to the java and perl
                 examples described above
 flix2_sample.py
 process_sample.py - python cgi example similar to the perl example.

==============================================================================
[Building the language bindings]
 If you did not have the proper prerequisites or chose not to build the
extensions at install time, you can attempt to build and install them by
running the compile scripts that are installed by default in
/usr/local/flixmodules. Depending on your distribution/current setup you may
have to download and install the proper development packages before the
language bindings will compile. See your distribution's documentation for more
information.
```

Chapter 9

Command Line

```
/*
//==========================================================================
//
//  Copyright (c) On2 Technologies Inc. All Rights Reserved.
//
//--------------------------------------------------------------------------
//
//  File:        $Workfile$
//               $Revision$
//
//  Last Update: $DateUTC$
//
//--------------------------------------------------------------------------
*/
#include <stdio.h>  //printf, etc
#include <stdlib.h> //exit
#include <unistd.h> //sleep
#include <flixengine2/flixengine2.h>
#include "example_common.h"

int main(int argc, char** argv)
{
    const char* rpchost = "localhost";
    const int timeout_s = 0;
    FLIX2HANDLE flix;
    on2bool ier;
    int32_t srcw, srch, srcduration;
    on2sc sc;

    printf("Flix Engine client library v%s\n",Flix2_Version());
    printf("%s\n\n",Flix2_Copyright());

    if (argc < 3) {
        printf("usage: cli_encode <infile> <outfile>\n\n");
        printf("NOTE cli_encode uses libflixengine2.so which is a client\n"
               "NOTE side rpc library. All paths must be accessible to the\n"
               "NOTE server side, i.e., flixd, thus relative paths will most\n"
               "NOTE likely give undesired results. The same can be said\n"
               "NOTE for clients running on different machines.\n");
        return -1;
    }

    /*contact flixd on rpchost w/timeout of timeout_s.
      A port may be specified by giving rpchost in the form 'server:port'
      to avoid making a connection to portmap before contacting flixd.
      This is only useful if flixd is being run with the --port option.
      If timeout_s is 0 rpc's default timeout will be used (typically 25s)*/
    CHECKSC( Flix2_CreateEx(&flix, rpchost, timeout_s) )

    printf("Input File  : %s\n",argv[1]);
    if(*argv[1] != '/') printf("WARNING: path to input file is not absolute\n");
    CHECKSC( Flix2_SetInputFile(flix, argv[1]) )

    /*input file information*/
    CHECKSC( video_options_GetSourceWidth(flix,&srcw) )
    CHECKSC( video_options_GetSourceHeight(flix,&srch) )
    CHECKSC( Flix2_GetSourceDuration(flix,&srcduration) )
    printf("              Width:    %d\n"
           "              Height:   %d\n"
           "              Duration: %dms\n",
           srcw,srch,srcduration);

    printf("Output File : %s\n",argv[2]);
    if(*argv[2] != '/') printf("WARNING: path to output file is not absolute\n");
    CHECKSC( Flix2_SetOutputFile(flix, argv[2]) )

    /*
```

```
       Options may be set and codecs/filters/muxers may be added prior to Flix2_Encode()
    */

    /*Add the scale filter
    {
    FLIX2PLGNHANDLE filter;
    CHECKSC( Flix2_AddFilter(&filter,flix,FE2_FILTER_SCALE) );

    CHECKSC( Flix2_FilterSetParam(filter,FE2_SCALE_WIDTH,240) );
    CHECKSC( Flix2_FilterSetParam(filter,FE2_SCALE_HEIGHT,160) );
    }*/

    /*Add the vp6 codec. Though it is the default, you must add it in order
      to modify its settings
    {
    FLIX2PLGNHANDLE codec;
    CHECKSC( Flix2_AddCodec(&codec,flix,FE2_CODEC_VP6) );

    CHECKSC( Flix2_CodecSetParam(codec,FE2_VP6_RC_MODE,VBR_1PASSControl) );
    }*/

    /*Use the FLV muxer (default)
    {
    FLIX2PLGNHANDLE muxer;
    CHECKSC( Flix2_AddMuxer(&muxer,flix,FE2_MUXER_FLV) );
    }*/

    CHECKSC( Flix2_Encode(flix) )

    printf("\n");
    do {
        int32_t p;
        sleep(1);

        sc = Flix2_IsEncoderRunning(flix,&ier);

        CHECKSC( encoding_status_PercentComplete(flix,&p) )
        printf("\rEncoding...%d%%  ",p); fflush(stdout);
    } while(!sc && ier);

    printf("Done!\n");
    /*from example_common.h*/
    print_encoder_status(flix);

    /*cleanup*/
    CHECKSC( Flix2_Destroy(flix) )

    return 0;
}
```

Chapter 10

Command Line

```
/*
//==========================================================================
//
//  Copyright (c) On2 Technologies Inc. All Rights Reserved.
//
//--------------------------------------------------------------------------
//
//  File:        $Workfile$
//               $Revision$
//
//  Last Update: $DateUTC$
//
//--------------------------------------------------------------------------
*/
import com.on2.flix.*;
import java.io.*;

public class cli_encode {
    public static void main(String argv[])
    {
        FlixEngine2 flix;

        System.out.println("Using library path: "+
                           System.getProperty("java.library.path"));
        System.out.println("\nFlix Engine client library v"+
                           FlixEngine2.Version());
        System.out.println(FlixEngine2.Copyright()+"\n");

        if(argv.length < 2) {
            System.out.println("usage: java cli_encode <infile> <outfile>\n");
            System.out.println(
               "NOTE cli_encode uses libflixengine2.so which is a client\n"+
               "NOTE side rpc library. All paths must be accessible to the\n"+
               "NOTE server side, i.e., flixd, thus relative paths will most\n"+
               "NOTE likely give undesired results. The same can be said\n"+
               "NOTE for clients running on different machines.");
            System.exit(-1);
        }

        System.out.println("Connecting to Flix...");
        final int timeout_s = 0; //rpc timeout in seconds,
                                 //0=use default (25s)
        flix = new FlixEngine2("localhost", timeout_s);
        try {
            flix.Connect();

            File f = new File(argv[0]);
            System.out.println("Input file  : "+argv[0]);
            if(!f.isAbsolute())
                System.out.println("WARNING: path to input file is not absolute");
            flix.SetInputFile(argv[0]);

            //input file information
            System.out.println(
                "              Width:    "+flix.video_options_GetSourceWidth()+"\n"+
                "              Height:   "+flix.video_options_GetSourceHeight()+"\n"+
                "              Duration: "+flix.GetSourceDuration());

            f = new File(argv[1]);
            System.out.println("Output file : "+argv[1]);
            if(!f.isAbsolute())
                System.out.println("WARNING: path to output file is not absolute");
            flix.SetOutputFile(argv[1]);

            /*
               Options may be set and codecs/filters/muxers may be added prior to Encode()
            */
```

```
            /*Add the scale filter
            Filter filter = new Filter(flix,flix.FE2_FILTER_SCALE);
            filter.add();

            filter.setParam(flix.FE2_SCALE_WIDTH,240.0);
            filter.setParam(flix.FE2_SCALE_HEIGHT,160.0);
            */

            /*Add the vp6 codec. Though it is the default, you must add it in order
              to modify its settings
            Codec codec = new Codec(flix,flix.FE2_CODEC_VP6);
            codec.add();

            codec.setParam(flix.FE2_VP6_RC_MODE,
                FE2_VideoBitrateControls.VBR_1PASSControl.swigValue());
            */

            /*Use the FLV muxer (default)
            Muxer muxer = new Muxer(flix,flix.FE2_MUXER_FLV);
            muxer.add();*/

            System.out.println();
            flix.Encode();

            boolean ier;
            do {
                ier = flix.IsEncoderRunning();
                System.out.print("\rEncoding..." +
                    flix.encoding_status_PercentComplete() + "%  ");
                try {Thread.currentThread().sleep(1000);}
                catch(InterruptedException e) {}
            } while(ier);
            System.out.println("Done!");
            printEncoderStatus(flix);

            flix.Destroy();
        } catch (FlixException e) {
            System.out.println("Flix call failed: "+e);
            e.printStackTrace();

            //if e == ON2_NET_ERROR Flix2_Errno will return the specific
            //rpc error encountered as flixerrno along with the client lib's errno value
            try {
                long[] flixerr = flix.Errno();
                System.out.println("\tFlixEngine2.Errno: "+
                    (e.equals(on2sc.ON2_NET_ERROR)?
                     "rpcerr":"flixerrno")+": "+flixerr[0]+
                    " syserrno:"+flixerr[1]);
            } catch (FlixException ex) {}
        }
    }

    private static void printEncoderStatus(final FlixEngine2 flix)
    {
        try {
            System.out.println("\nEncoder Status");
            System.out.println(" FlixEngine2.GetEncoderState:"+
                flix.GetEncoderState());
            long[] flixerr = flix.Errno();
            System.out.println(" FlixEngine2.Errno: flixerrno:"+
                flixerr[0]+" syserrno:"+flixerr[1]);
        } catch (FlixException e) {}
    }
}
```

Chapter 11

Command Line

```
#!/usr/bin/perl -w
##==========================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##--------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##--------------------------------------------------------------------------
##
use On2::flixengine2;
use File::Spec::Functions;
package On2::flixengine2; #use flixengine2's namespace

#checks the return value of an API function printing error information on
#failure. usage checksc(funcname,sc)
sub checksc($$)
{
    my $sc = $_[1];
    if ($sc != $ON2_OK) {
        my $esc;
        select STDERR;
        print "$_[0] failed: sc= $sc\n";

        ##if sc == ON2_NET_ERROR Flix2_Errno will return the specific rpc error
        ##encountered as flixerrno along with the client lib's errno value
        @_ = Flix2_Errno($flix); $esc= shift;
        printf(" Flix2_Errno: sc:%d %s:%d syserrno:%d\n",
            $esc,($sc==$ON2_NET_ERROR)?"rpcerr":"flixerrno",$_[0],$_[1]);
        exit 1;
    }
}

sub print_encoder_status()
{
    print "\nEncoder Status\n";

    my $state = Flix2_GetEncoderState($flix);
    print " Flix2_GetEncoderState: $state\n";

    my ($sc,@err) = Flix2_Errno($flix);
    printf(" Flix2_Errno: sc:%d flixerrno:%d syserrno:%d\n",$sc,$err[0],$err[1]);
}

my $rpchost = "localhost";
my $timeout_s = 0; #rpc timeout in seconds, 0=use default (25s)

print "Flix Engine client library v".Flix2_Version()."\n";
print Flix2_Copyright()."\n\n";

die "usage: ./cli_encode.pl <infile> <outfile>\n\n".
    "NOTE cli_encode.pl uses libflixengine2.so which is a client\n".
    "NOTE side rpc library. All paths must be accessible to the\n".
    "NOTE server side, i.e., flixd, thus relative paths will most\n".
    "NOTE likely give undesired results. The same can be said\n".
    "NOTE for clients running on different machines.\n" if @ARGV < 2;

#create a new flix2handle ptr
my $flixptr = new_flix2handlep();

## contact flixd on rpchost w/timeout of timeout_s.
## A port may be specified by giving rpchost in the form 'server:port'
## to avoid making a connection to portmap before contacting flixd.
```

```
## This is only useful if flixd is being run with the --port option.
## If timeout_s is 0 rpc's default timeout will be used (typically 25s)
my $sc = Flix2_CreateEx($flixptr, $rpchost, $timeout_s);

#recover the actual handle value to be used in remaining flixengine calls
$flix = flix2handlep_value($flixptr);
  checksc('Flix2_CreateEx',$sc);

print "Input File  : $ARGV[0]\n";
print "WARNING: path to input file is not absolute\n"
    unless File::Spec::Functions::file_name_is_absolute($ARGV[0]);
$sc = Flix2_SetInputFile($flix, $ARGV[0]);
  checksc('Flix2_SetInputFile', $sc);

##input file information
my ($srcduration,$srcw,$srch);
($sc,$srcduration)= Flix2_GetSourceDuration($flix);
  checksc('Flix2_GetSourceDuration', $sc);
($sc,$srcw)= video_options_GetSourceWidth($flix);
  checksc('video_options_GetSourceWidth', $sc);
($sc,$srch)= video_options_GetSourceHeight($flix);
  checksc('video_options_GetSourceHeight', $sc);

print <<EOT;
              Width:     $srcw
              Height:    $srch
              Duration:  ${srcduration}ms
EOT

print "Output File : $ARGV[1]\n";
print "WARNING: path to output file is not absolute\n"
    unless File::Spec::Functions::file_name_is_absolute($ARGV[1]);
$sc = Flix2_SetOutputFile($flix, $ARGV[1]);
  checksc('Flix2_SetOutputFile', $sc);

##
## Options may be set and codecs/filters/muxers may be added prior to Flix2_Encode()
##

##Add the scale filter
##create a storage location for the filter handle
#my $filterptr= new_flix2plgnhandlep();
#$sc = Flix2_AddFilter($filterptr,$flix,$FE2_FILTER_SCALE);
#  checksc('Flix2_AddFilter($FE2_FILTER_SCALE)',$sc);
#
##retrieve the value of the handle for use in the remaining filter functions
#my $filter= flix2plgnhandlep_value($filterptr);
#$sc = Flix2_FilterSetParam($filter,$FE2_SCALE_WIDTH,240);
#  checksc('Flix2_FilterSetParam($FE2_SCALE_WIDTH,240)',$sc);
#$sc = Flix2_FilterSetParam($filter,$FE2_SCALE_HEIGHT,160);
#  checksc('Flix2_FilterSetParam($FE2_SCALE_HEIGHT,160)',$sc);
#
##cleanup
#delete_flix2plgnhandlep($filterptr); $filterptr= undef;
#
##Add the vp6 codec. Though it is the default, you must add it in order
##to modify its settings
#my $codecptr= new_flix2plgnhandlep();
#$sc = Flix2_AddCodec($codecptr,$flix,$FE2_CODEC_VP6);
#  checksc('Flix2_AddCodec($FE2_CODEC_VP6)',$sc);
#
##retrieve the value of the handle for use in the remaining codec functions
#my $codec= flix2plgnhandlep_value($codecptr);
#
#$sc = Flix2_CodecSetParam($codec,$FE2_VP6_RC_MODE,$VBR_1PASSControl);
#  checksc('Flix2_CodecSetParam($FE2_VP6_RC_MODE,$VBR_1PASSControl)',$sc);
#
```

```
##cleanup
#delete_flix2plgnhandlep($codecptr); $codecptr= undef;
#
##Use the FLV muxer (default)
#my $muxerptr= new_flix2plgnhandlep();
#$sc = Flix2_AddMuxer($muxerptr,$flix,$FE2_MUXER_FLV);
#  checksc('Flix2_AddMuxer($FE2_MUXER_FLV)',$sc);
#
##retrieve the value of the handle for use in the remaining muxer functions
#my $muxer= flix2plgnhandlep_value($muxerptr);
#
##cleanup
#delete_flix2plgnhandlep($muxerptr); $muxerptr= undef;

$sc = Flix2_Encode($flix); checksc('Flix2_Encode',$sc);

print "\n";
$| = 1; #auto flush
my ($ier,$pcnt);
do {
    ($sc,$ier)= Flix2_IsEncoderRunning($flix);
      checksc('Flix2_IsEncoderRunning',$sc);

    ($sc,$pcnt)= encoding_status_PercentComplete($flix);
      checksc('encoding_status_PercentComplete',$sc);
    print "\rEncoding...$pcnt%  ";

    sleep(1);
} while($ier);

print "Done!\n";
print_encoder_status();

#cleanup
$sc = Flix2_Destroy($flix); checksc('Flix2_Destroy',$sc);
delete_flix2handlep($flixptr);
```

Chapter 12

CGI

This example consists of 2 parts: flix2_sample.cgi and process_sample.cgi. flix2_sample.cgi searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's include files.

To use this example navigate to flix2_sample.cgi in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to process_sample.cgi which runs the encode loop.

Default file locations used by the scripts:

Input : */var/www/cgi-bin/flixmedia/in* (`$indir` in flix2_sample.cgi)
Overlay : */var/www/cgi-bin/flixmedia/overlay* (`$overlaydir` in flix2_sample.cgi)
Output : */var/www/cgi-bin/flixmedia/out* (`$outdir` in process_sample.cgi)

12.1 flix2_sample.cgi

```
#!/usr/bin/perl -w
##==============================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##------------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##------------------------------------------------------------------------------
##

#flix2_sample.cgi
#expose API functions available in the perl bindings
#submits a form to process_sample.cgi which makes calls to libflixengine2.so
#through the bindings

use CGI qw(:standard -no_xhtml);
use On2::flixengine2 ();  #ensure Flix Engine module existence prior to process

$| = 1; #auto flush

my $prefix     = "/var/www/cgi-bin/";
my $indir      = $prefix."flixmedia/in/";      #src file directory
my $overlaydir = $prefix."flixmedia/overlay/"; #overlay image file directory

print header;
print <<HTML_END;

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head>
 <title>Flix CGI Sample - Perl</title>
 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

 <script type="text/javascript">
 function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
        i++;
    }
```

```
}

function toggle_expand(_this)
{
    _this.parentNode.className= (_this.parentNode.className=='expanded') ?
                                'collapsed' : 'expanded';
   document.getElementById('showall').checked=false;
}

function set_table_visible(ctable,visible)
{
   var table= document.getElementById(ctable);

   if(table) {
      if(visible && table.className == 'hidden') {
          table.className= '';
      } else if(!visible && table.className == '') {
          table.className= 'hidden';
      }
   }
}

function hide_tablelist(vistablename,tablelist)
{
  var table= document.getElementById(vistablename);

  if(table&&table.className=='hidden') {
    table.className= '';

    var i=0;
    while(tablelist[i]) {
      if(tablelist[i]!=vistablename) {
        var hiddentable= document.getElementById(tablelist[i]);

        set_table_visible(tablelist[i],false);
        /*clear down any values set in the hidden table to avoid posting
          unnecessary values*/
        clear_value(hiddentable.getElementsByTagName('input'));
        clear_value(hiddentable.getElementsByTagName('select'));
      }
      i++;
    }
  }
}

function set_acodec_visible(ctable)
{
  var acodecs= new Array('aactable','aacplustable','amrnbtable','lametable','pcmtable','vorbistable');
  hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
  var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8table');
  hide_tablelist(ctable,vcodecs);
}

function set_muxer_visible(mtable)
{
  var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','tg2table','tgptable','web
  hide_tablelist(mtable,muxers);
}

function clear_value(list)
{
   var i=0;
   while(list[i]) {
```

```
      if(list[i].type=='checkbox') { list[i++].checked=false; }
      else { list[i++].value=''; }
    }
}

function toggle_ftable(ftable,enabled)
{
  var table= document.getElementById(ftable);

  if(table) {
    table.className = enabled ? '' : 'disabled';
    if(!enabled) {
      clear_value(table.getElementsByTagName('input'));
      clear_value(table.getElementsByTagName('select'));
    }
  }
}

function reset_tables()
{
   var table_list= document.getElementsByTagName('table');
   var i=0;
   while(table_list[i]) {
     if (table_list[i].id.length > 7 &&
       table_list[i].id.substring(0,7) == 'filter_') {
       table_list[i].className= 'disabled';
     } else if (table_list[i].id.indexOf('table') != -1) {
       table_list[i].className= 'hidden';
     }
     i++;
   }
}

</script>

<style type="text/css">
   <!--
   html {
       font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
       font-size: 100%;
       color: rgb(56,56,56);
       background-color: rgb(236,236,236);
       border-style: solid;
       border-color: rgb(236,236,236);
   }

   body {
       text-align: center;
       margin: 0 auto;
   }

   div.content {
       color: rgb(56,56,56);
       background-color: rgb(246,246,246);
       text-align: left;
       margin: 0 auto;
       width: 80%;
       min-width: 768px;
       max-width: 932px;
       border-width: 0 1px;
       border-color: rgb(144,144,144);
       border-style: solid;
   }

   div.content:after {
       content: "";
       color: inherit;
```

```
    background-color: rgb(250,250,250);
    border-top: 1px solid rgb(144,144,144);
    height: 20px;
    width: 100%;
    display: block;
}

fieldset table, fieldset {display: none;}
fieldset.expanded, fieldset.collapsed {display: block;}

/*first is fallback for IE*/
fieldset.expanded table {display: block;}
fieldset.expanded table {display: table;}

h1 {
    font-family: sans-serif;
    font-size: 150%;
    font-weight: normal;
    text-align: left;
    letter-spacing: -1px;
    color: rgb(74,74,74);
    background-color: inherit;
    margin: 0;
}

a {
    color: rgb(74,74,74);
    background-color: transparent;
}

label {
    font-size: 75%;
}

fieldset {
    font-size: 75%;
    line-height: 130%;
    padding: 0;
    margin: 20px;
    border: none;
}

fieldset.expanded {
    color: inherit;
    background-color: rgb(252,252,252);
    border-style: solid;
    border-width: 1px;
    border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
}

legend {
    padding: 0 5px;
    border-left: 10px solid rgb(217,217,217);
    cursor: pointer;
}
legend:hover {text-decoration: underline;}
fieldset.expanded legend {
    font-size: 150%;
    font-weight: bold;
    letter-spacing: -1px;
    background: transparent;
    margin-left: 12px;
    border-right: 10px solid rgb(217,217,217);
    display: block;
}
table {
    font-size: 100%;
```

```
        border-spacing: 0;
        /*border-collapse: collapse;*/
        width: 100%;
    }

    th:before { display: none; }
    th, td {
        width: 50%;
        vertical-align: top;
        padding: 2px 3px;
        border-width: 1px 0;
        border-style: solid;
        border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
    }
    tr:first-child>* {border-top-color: rgb(252,252,252);}
    th {
        font-weight: normal;
        text-align: left;
        padding: 2px 2px 2px 5px;
    }
    input[type] {
        font-family: monospace;
        font-size: 100%;
        color: rgb(56,56,56);
        background-color: inherit;
    }
    [type="text"], [type="number"] {
        margin-right: 13px;
        width: 222px;
    }

    [type="checkbox"].filter {
        margin-left: 4px
    }
    [type="checkbox"] {
        margin-left: 13px
    }
    [type="button"], [type="submit"] {
        font-size: 1em;
        margin: 0 2px 0 13px;
    }

    fieldset {
        font-size: 75%;
        margin: 20px 10px;
    }
    fieldset input[type="text"], fieldset input[type="number"] {
        width: 95% !important;
        margin: 0;
        display: block;
    }
    fieldset.expanded table.hidden {
        display: none;
    }

    table.disabled {
        color: rgb(176,176,176);
    }
    -->
 </style>
</head>

<body>
 <div class='content'>
 <noscript>
  <p>This page requires javascript be enabled.</p>
 </noscript>
```

```
<hr>
<h1>Flix CGI Sample</h1>



<p><small>flix2_sample.cgi version 1.9</small></p>
<h4>Instructions</h4>
<ul>
 <li>In this sample you must choose a source file and an output file.<br>
 If you leave all the other options blank then the sample will not call
 the corresponding Flix Engine function and the default will be used.<br>
 When done please press the "Start Encode" button at the bottom of the page.<br>
 <li>Mouse over a function name to see its default, if applicable.
 <li>Current source file directory: $indir
</ul>

<p>
 <label><input type="checkbox" id='showall' onclick='showall(this.checked)'>Show all</label>
</p>

<form action="process_sample.cgi" method="post">

<!-- ##SOURCE FILE############################################################ -->
<hr>
<fieldset class='expanded' id="srcfile">
<legend onclick='toggle_expand(this)'>Source File</legend>
<table>

<tr>
 <td>
HTML_END

if(opendir DIR,$indir) {
    #search the input directory for potential files to encode
    #populating a list box in the process
    @a = sort grep { -f $indir.$_ } readdir DIR;
    foreach (@a) {
        $a{$indir.$_} = $_;
        $_ = $indir.$_;
    }
    print scrolling_list(-name=>'Flix2_SetInputFile',
                         -values=>\@a,
                         -labels=>\%a,
                         -default=>$a[0],
                         -size=>scalar(@a)<15?scalar(@a):15);
    closedir DIR;
} else {
    print "WARNING couldn't open $indir: $!",br;
}

print <<HTML_END;

 </td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE################################################################ -->
<hr>
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)'>Output File</legend>
<table>

<tr>
 <td>
```

```
  <input type="text" name="Flix2_SetOutputFile" value="cgi-pl-out.flv">
 </td>
</tr>

</table>
</fieldset>

<!-- ##MAIN OPTIONS################################################################ -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)'>Main Options</legend>
<table>

<tr>
 <th><abbr title="Default: FALSE">Flix2_SetOverwriteExistingFiles</abbr></th>
 <td>
 <select name="Flix2_SetOverwriteExistingFiles">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: TRUE">Flix2_SetExportAudio</abbr></th>
 <td>
 <select name="Flix2_SetExportAudio">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: TRUE">Flix2_SetExportVideo</abbr></th>
 <td>
 <select name="Flix2_SetExportVideo">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

</table>
</fieldset>

<!-- ##CODECS########################################################################## -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this)'>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
 <td>
 <label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
         onfocus="set_vcodec_visible('vp8table')">
  <abbr title="For use with WebM">FE2_CODEC_VP8</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
```

```
        onfocus="set_vcodec_visible('vp6table')">
 <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
        onfocus="set_vcodec_visible('vp6atable')">
 <abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H263'
        onfocus="set_vcodec_visible('h263table')">
 <abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr> 
</label>
<br>

<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
        onfocus="set_vcodec_visible('h263table')">
 <abbr title="For use with 3GP">FE2_CODEC_H263_BASELINE</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H264'
        onfocus="set_vcodec_visible('h264table')">
 <abbr title="For use with 3GP/3G2/MOV/MP4">FE2_CODEC_H264</abbr> 
</label>

<!-- VP6 codec parameters -->
<table id='vp6table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_VP6_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP6_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP6_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
   <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6_CXMODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_CXMODE">
   <option value=""></option>
```

```
   <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
   <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: VP6_E">FE2_VP6_PROFILE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_PROFILE">
   <option value=""></option>
   <option value="VP6_E">VP6_E</option>
   <option value="VP6_S">VP6_S</option>
  </select>
  </td>
 </tr>

 <tr><th><b>Advanced Settings:</b></th></tr>

 <tr>
  <th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_CONCURRENCY'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_UNDERSHOOT_PCT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MIN_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_MIN_Q'></td>
 </tr>

 <tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MAX_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_MAX_Q'></td>
 </tr>

 <tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_SHARPNESS</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_SHARPNESS'></td>
 </tr>

 <tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_NOISE_REDUCTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_NOISE_REDUCTION'></td>
 </tr>

 <tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEMPORAL_RESAMPLING</abbr></
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
 </tr>

 <tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEMPORAL_DOWN_WATERMARK</abb
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_PREBUFFER'></td>
```

```
 </tr>

 <tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_MAX_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_2PASS_MIN_SECTION'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_2PASS_MAX_SECTION'></td>
 </tr>

</table>
<!-- END - VP6 codec parameters -->

<!-- VP6A codec parameters -->
<table id='vp6atable' class='hidden'>
 <tr>
  <th><abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 68kbps (15% of default 448kbps)">FE2_VP6A_ALPHA_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6A_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP6A_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6A_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
   <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
  </td>
 </tr>

 <tr>
```

```
 <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP6A_CXMODE">
  <option value=""></option>
  <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
  <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
 </select>
 </td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
 <th><abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_RESAMPLING</abbr><
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_DOWN_WATERMARK</ab
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'></td>
</tr>
```

```
 <tr>
  <th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_PREBUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_2PASS_MIN_SECTION'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_2PASS_MAX_SECTION'></td>
 </tr>

</table>
<!-- END - VP6A codec parameters -->

<!-- H263 codec parameters -->
<table id='h263table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H263_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_H263_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H263_RC_MODE">
    <option value=""></option>
    <option value="VBR_2PASSControl">VBR_2PASSControl</option>
    <option value="CBR_2PASSControl">CBR_2PASSControl</option>
    <option value="VBR_1PASSControl">VBR_1PASSControl</option>
    <option value="CBR_1PASSControl">CBR_1PASSControl</option>
   </select>
  </td>
```

```
 </tr>

 <tr>
  <th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_MAX_Q'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_MIN_Q'></td>
 </tr>

</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_H264_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H264_RC_MODE">
    <option value=""></option>
    <option value="VBR_1PASSControl">VBR_1PASSControl</option>
    <option value="CBR_1PASSControl">CBR_1PASSControl</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H264_PROFILE">
    <option value=""></option>
    <option value="BASE_H264PROFILE">BASE_H264PROFILE</option>
    <option value="MAIN_H264PROFILE">MAIN_H264PROFILE</option>
    <option value="HIGH_H264PROFILE">HIGH_H264PROFILE</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_B_FRAME_RATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: Dependent on profile selection, see API docs. Valid Range [0,5]">FE2_H264_SPE
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_SPEED'></td>
 </tr>

</table>
<!-- END - H264 codec parameters -->

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
```

```
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_BITRATE'></td>
</tr>

<tr>
 <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP8_KFINTTYPE">
  <option value=""></option>
  <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
  <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP8_KFFREQ</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_KFFREQ'></td>
</tr>

<tr>
 <th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP8_RC_MODE">
  <option value=""></option>
  <option value="VBR_2PASSControl">VBR_2PASSControl</option>
  <option value="CBR_2PASSControl">CBR_2PASSControl</option>
  <option value="VBR_1PASSControl">VBR_1PASSControl</option>
  <option value="CBR_1PASSControl">CBR_1PASSControl</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP8_CXMODE">
  <option value=""></option>
  <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
  <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_THREADS'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_PROFILE'></td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_LAG'></td>
</tr>

<tr>
 <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_UNDERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
```

```
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_OVERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
 <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_ALTREF'></td>
</tr>

<tr>
 <th><abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_MAX_FRAMES'></td>
</tr>

<tr>
 <th><abbr title="">FE2_VP8_AR_TYPE</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_TYPE'></td>
</tr>
```

```
  <tr>
   <th><abbr title="">FE2_VP8_AR_STRENGTH</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_STRENGTH'></td>
  </tr>

  <tr>
   <th><abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MB_STATIC_THRESHOLD'></td>
  </tr>

  <tr>
   <th><abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_TOKEN_PARTITIONS'></td>
  </tr>

 </table>
 <!-- END - VP8 codec parameters -->

 </td>
</tr> <!-- END - video codecs -->

<tr><th><b>Audio Codecs</b></th></tr>
<tr>
 <td>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_VORBIS'
         onfocus="set_acodec_visible('vorbistable')">
  <abbr title="For use with WebM">FE2_CODEC_VORBIS</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AAC'
         onfocus="set_acodec_visible('aactable')">
  <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AAC</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AACPLUS'
         onfocus="set_acodec_visible('aacplustable')">
  <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AACPLUS</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_LAME'
         onfocus="set_acodec_visible('lametable') ">
  <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_LAME</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_PCM'
         onfocus="set_acodec_visible('pcmtable')">
  <abbr title="For use with FLV/SWF">FE2_CODEC_PCM</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AMR_NB'
         onfocus="set_acodec_visible('amrnbtable')">
  <abbr title="For use with 3GP">FE2_CODEC_AMR_NB</abbr> 
 </label>

 <!-- AMR_NB codec parameters -->
 <table id='amrnbtable' class='hidden'>
  <tr>
   <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_AMR_BITRATE'></td>
  </tr>
```

```
</table>
<!-- END - AMR_NB codec parameters -->

<!-- AAC codec parameters -->
<table id='aactable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_AAC_BITRATE'></td>
 </tr>

</table>
<!-- END - AAC codec parameters -->

<!-- AACPLUS codec parameters -->
<table id='aacplustable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_AACPLUS_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: disabled (aacPlus v1)">FE2_AACPLUS_PARAMETRIC_STEREO</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_AACPLUS_PARAMETRIC_STEREO">
   <option value=""></option>
   <option value="0">disable (aacPlus v1)</option>
   <option value="1">enable (aacPlus v2)</option>
   </select>
  </td>
 </tr>
</table>
<!-- END - AACPLUS codec parameters -->

<!-- LAME codec parameters -->
<table id='lametable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_LAME_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_LAME_QUALITY'></td>
 </tr>

 <tr>
  <th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_LAME_RC_MODE">
   <option value=""></option>
   <option value="LAME_CBR">LAME_CBR</option>
   <option value="LAME_ABR">LAME_ABR</option>
   <option value="LAME_VBR_rh">LAME_VBR_rh</option>
   <option value="LAME_VBR_mtrh">LAME_VBR_mtrh</option>
   </select>
  </td>
 </tr>
</table>
<!-- END - LAME codec parameters -->

<!-- PCM codec parameters -->
<table id='pcmtable' class='hidden'>
 <tr>
  <th>(FE2_CODEC_PCM defines no parameters)</th>
 </tr>
</table>
<!-- END - PCM codec parameters -->
```

```
 <!-- VORBIS codec parameters -->
 <table id='vorbistable' class='hidden'>
  <tr>
   <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VORBIS_BITRATE'></td>
  </tr>
 </table>
 <!-- END - VORBIS codec parameters -->

 </td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS######################################################################## -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this)'>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_CUT' value='avfilter:'
          onchange="toggle_ftable('filter_cut',this.checked)">
   FE2_FILTER_CUT
  </label>

  <table id='filter_cut' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CUT_START_SEC'></td>
   </tr>

   <tr>
    <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CUT_STOP_SEC'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_CUT_USE_SEEK">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - CUT filter parameters -->

<tr><th><b>Video Filters</b></th></tr>

<tr>
 <td>
  <label>
```

```
   <input type='checkbox' class='filter'
          name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:'
          onchange="toggle_ftable('filter_adaptive_deinterlace',this.checked)">
   FE2_FILTER_ADAPTIVE_DEINTERLACE
  </label>

  <table id='filter_adaptive_deinterlace' class='disabled'>
   <tr>
    <th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_ADAPTIVE_DEINTERLACE_MODE">
     <option value=""></option>
     <option value="DEINTERLACE_NONE">DEINTERLACE_NONE</option>
     <option value="DEINTERLACE_1_2_1_BLUR">DEINTERLACE_1_2_1_BLUR</option>
     <option value="DEINTERLACE_DROP_FIELD">DEINTERLACE_DROP_FIELD</option>
     <option value="DEINTERLACE_ADAPTIVE">DEINTERLACE_ADAPTIVE</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - ADAPTIVE DEINTERLACE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_BCHS' value='vfilter:'
          onchange="toggle_ftable('filter_bchs',this.checked)">
   FE2_FILTER_BCHS
  </label>

  <table id='filter_bchs' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_BRIGHTNESS'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_CONTRAST'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_HUE'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_SATURATION'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - BCHS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_BLUR' value='vfilter:'
          onchange="toggle_ftable('filter_blur',this.checked)">
   FE2_FILTER_BLUR
  </label>

  <table id='filter_blur' class='disabled'>
```

```
   <tr>
    <th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_BLUR_FILTER">
     <option value=""></option>
     <option value="BLUR_LOWPASS">BLUR_LOWPASS</option>
     <option value="BLUR_GAUSS">BLUR_GAUSS</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_BLUR_MASKSIZE">
     <option value=""></option>
     <option value="MASK_3x3">MASK_3x3</option>
     <option value="MASK_5x5">MASK_5x5</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_CROP' value='vfilter:'
          onchange="toggle_ftable('filter_crop',this.checked)">
   FE2_FILTER_CROP
  </label>

  <table id='filter_crop' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_TOP'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_BOTTOM'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_LEFT'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_RIGHT'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - CROP filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_DENOISE' value='vfilter:'
```

```
          onchange="toggle_ftable('filter_denoise',this.checked)">
   FE2_FILTER_DENOISE
  </label>

  <table id='filter_denoise' class='disabled'>
   <tr>
    <th><abbr title="Default: 0. Range: [0.0,1.0)">FE2_DENOISE_NOISE_LEVEL</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_DENOISE_NOISE_LEVEL'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_FRAMERATE' value='vfilter:'
          onchange="toggle_ftable('filter_framerate',this.checked)">
   FE2_FILTER_FRAMERATE
  </label>

  <table id='filter_framerate' class='disabled'>
   <tr>
    <th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAMERATE_DECIMATE</abbr></th
    <td><input type='text' name='Flix2_FilterSetParam:FE2_FRAMERATE_DECIMATE'></td>
   </tr>

   <tr>
    <th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FRAMERATE_FPS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_FRAMERATE_FPS'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_MIRROR' value='vfilter:'
          onchange="toggle_ftable('filter_mirror',this.checked)">
   FE2_FILTER_MIRROR
  </label>

  <table id='filter_mirror' class='disabled'>
   <tr>
    <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_MIRROR_HORIZONTAL">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_MIRROR_VERTICAL">
     <option value=""></option>
     <option value="on2false">FALSE</option>
```

```
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_OVERLAY' value='vfilter:'
          onchange="toggle_ftable('filter_overlay',this.checked)">
   FE2_FILTER_OVERLAY
  </label>

  <table id='filter_overlay' class='disabled'>
   <tr>
    <th>
     <abbr title="Currently searching $overlaydir for overlay images">FE2_OVERLAY_FILE</abbr>
    </th>
    <td>
HTML_END

if(opendir DIR,$overlaydir) {
    #populate the list box with files from the overlay directory
    @a = sort grep { -f $overlaydir.$_ } readdir DIR;
    foreach (@a) {
        $a{$overlaydir.$_} = $_;
        $_ = $overlaydir.$_;
    }
    unshift(@a,"");
    print scrolling_list(-name=>'Flix2_FilterSetParamAsStr:FE2_OVERLAY_FILE',
                         -values=>\@a,
                         -labels=>\%a,
                         -size=>scalar(@a)),br;
    print "default: None, must be set to the absolute path of the overlay".
          " input file, e.g. '/path/to/my/overlay.png'";
    closedir DIR;
} else {
    print "WARNING couldn't open $overlaydir: $!",br;
}

print <<HTML_END;

    </td>
   </tr>

   <tr>
    <th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_MASK_XY">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_X'></td>
   </tr>
   <tr>
```

```
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_Y'></td>
   </tr>

   <tr>
    <th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_MASK_RGB">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_R'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_G'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_B'></td>
   </tr>

   <tr>
    <th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_POS">
      <option value=""></option>
      <option value="FE2_OVERLAY_POS_MODE_TOPLEFT">FE2_OVERLAY_POS_MODE_TOPLEFT</option>
      <option value="FE2_OVERLAY_POS_MODE_BOTLEFT">FE2_OVERLAY_POS_MODE_BOTLEFT</option>
      <option value="FE2_OVERLAY_POS_MODE_CENTER">FE2_OVERLAY_POS_MODE_CENTER</option>
      <option value="FE2_OVERLAY_POS_MODE_TOPRIGHT">FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
      <option value="FE2_OVERLAY_POS_MODE_BOTRIGHT">FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
      <option value="FE2_OVERLAY_POS_MODE_XY">FE2_OVERLAY_POS_MODE_XY</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_POS_X'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_POS_Y'></td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_PNGEX' value='vfilter:'
          onchange="toggle_ftable('filter_pngex',this.checked)">
   FE2_FILTER_PNGEX
  </label>
```

```
<table id='filter_pngex' class='disabled'>
 <tr>
  <th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'></td>
 </tr>

 <tr>
  <th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_FILENAME_PREFIX'></td>
 </tr>

 <tr>
  <th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'></td>
 </tr>

 <tr>
  <th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_WIDTH'></td>
 </tr>

 <tr>
  <th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_HEIGHT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>
  <td>
   <select name="Flix2_FilterSetParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">
   <option value=""></option>
   <option value="on2false">FALSE</option>
   <option value="on2true">TRUE</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
  <td>
   <select name="Flix2_FilterSetParam:FE2_PNGEX_ENABLE_ALPHA">
   <option value=""></option>
   <option value="on2false">FALSE</option>
   <option value="on2true">TRUE</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="interval in ms; Default: disabled">FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_EXPORT_INTERVAL'></td>
 </tr>

 <tr>
  <th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_EXPORT_TIME_STRING'></td>
 </tr>

 <tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
  <td>
   <select name="Flix2_FilterSetParam:FE2_PNGEX_EXPORT_CUE_POINTS">
   <option value=""></option>
   <option value="FE2_PNGEX_CP_ALL">All cue points (FE2_PNGEX_CP_ALL)</option>
   <option value="FE2_PNGEX_CP_NAV">Only navigation cue points (FE2_PNGEX_CP_NAV)</option>
   <option value="FE2_PNGEX_CP_EVENT">Only event cue points (FE2_PNGEX_CP_EVENT)</option>
   </select>
```

```
    </td>
   </tr>

   <tr>
    <th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_COMPRESSION_LEVEL'></td>
   </tr>

   <tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>

   <tr>
    <th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></td>
   </tr>

   <tr>
    <th><abbr title="start time in ms; Default: 0">FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_START_TIME'></td>
   </tr>

   <tr>
    <th><abbr title="stop time in ms; Default: <clip length>">FE2_PNGEX_AUTO_EXPORT_END_TIME</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_END_TIME'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD'></td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - PNGEX filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_ROTATE' value='vfilter:'
          onchange="toggle_ftable('filter_rotate',this.checked)">
   FE2_FILTER_ROTATE
  </label>

  <table id='filter_rotate' class='disabled'>
   <tr>
    <th><abbr title="Default: 0. valid: {0,90,180,270}">FE2_ROTATE_ANGLE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_ROTATE_ANGLE'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - ROTATE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_SCALE' value='vfilter:'
          onchange="toggle_ftable('filter_scale',this.checked)">
   FE2_FILTER_SCALE
  </label>

  <table id='filter_scale' class='disabled'>
   <tr>
```

```
    <th><abbr title="Default: input image width">FE2_SCALE_WIDTH</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_SCALE_WIDTH'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image height">FE2_SCALE_HEIGHT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_SCALE_HEIGHT'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - SCALE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_SHARPEN' value='vfilter:'
          onchange="toggle_ftable('filter_sharpen',this.checked)">
   FE2_FILTER_SHARPEN
  </label>

  <table id='filter_sharpen' class='disabled'>
   <tr>
    <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
   </tr>
  </table>

 </td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_HIGHPASS' value='afilter:'
          onchange="toggle_ftable('filter_highpass',this.checked)">
   FE2_FILTER_HIGHPASS
  </label>

  <table id='filter_highpass' class='disabled'>
   <tr>
    <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_HIGHPASS_Q'></td>
   </tr>

   <tr>
    <th>FE2_HIGHPASS_CUTOFF</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_HIGHPASS_CUTOFF'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_LOWPASS' value='afilter:'
          onchange="toggle_ftable('filter_lowpass',this.checked)">
   FE2_FILTER_LOWPASS
```

```
  </label>

  <table id='filter_lowpass' class='disabled'>
   <tr>
    <th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_LOWPASS_Q'></td>
   </tr>

   <tr>
    <th>FE2_LOWPASS_CUTOFF</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_LOWPASS_CUTOFF'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_RESAMPLE' value='afilter:'
          onchange="toggle_ftable('filter_resample',this.checked)">
   FE2_FILTER_RESAMPLE
  </label>

  <table id='filter_resample' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_RATE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_RESAMPLE_RATE'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_CHANNELS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_RESAMPLE_CHANNELS'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - RESAMPLE filter parameters -->

</table>
</fieldset>

<!-- ##MUXERS########################################################################## -->
<hr>
<fieldset class='collapsed' id="muxers">
<legend onclick='toggle_expand(this)'>Muxers</legend>
<table>

<tr>
 <td>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3GP'
         onfocus="set_muxer_visible('tgptable')">
  FE2_MUXER_3GP 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3G2'
         onfocus="set_muxer_visible('tg2table')">
  FE2_MUXER_3G2 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MOV'
         onfocus="set_muxer_visible('movtable')">
```

```
  FE2_MUXER_MOV 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MP4'
         onfocus="set_muxer_visible('mp4table')">
  FE2_MUXER_MP4 
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FLV'
         onfocus="set_muxer_visible('flvtable',true)">
  FE2_MUXER_FLV 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_SWF'
         onfocus="set_muxer_visible('swftable',true)">
  FE2_MUXER_SWF
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
         onfocus="set_muxer_visible('fxmtable',true)">
  FE2_MUXER_FXM 
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
         onfocus="set_muxer_visible('webmtable',true)">
  FE2_MUXER_WEBM 
 </label>
 </td>
</tr>

<tr>
 <td>
 <!-- 3GP muxer parameters -->
 <table id='tgptable' class='hidden'>
   <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
     <select name="Flix2_MuxerSetParam:FE2_3GP_FASTSTART">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
 </table>
 <!-- END - 3GP muxer parameters -->

 <!-- 3G2 muxer parameters -->
 <table id='tg2table' class='hidden'>
   <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
     <select name="Flix2_MuxerSetParam:FE2_3G2_FASTSTART">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
 </table>
 <!-- END - 3G2 muxer parameters -->
```

```
<!-- FLV muxer parameters -->
<table id='flvtable' class='hidden'>
 <tr>
  <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'">
      FE2_FLV_CUEPT_EVENT</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
 </tr>

 <tr>
     <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'">
      FE2_FLV_CUEPT_NAV</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_NAV'></td>
 </tr>

 <tr>
  <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'">
      FE2_FLV_CUEPT_PARAM</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
 </tr>

 <tr>
   <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
             FE2_FLV_METADATA_ENABLE</abbr></th>
   <td>
     <table id='flv_metadata_enable' class=''>
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Ena
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' name='Flix
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Ena
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' name='Flix
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' name=
      </tr>

      <tr>
```

```
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr title="Default: Enable
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' name='Flix2_M
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr title="Default: Enabl
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' name='Flix2_
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abbr title="Default: En
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' name='Fli
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default:
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' name='
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title=
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAM
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION'><abbr title="
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Di
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' name='Fli
     </tr>
    </table>
  </td>
</tr>

<tr>
  <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
            FE2_FLV_METADATA_DISABLE</abbr></th>
  <td>
    <table id='flv_metadata_disable' class=''>
     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abbr title="Default: En
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' name='Fli
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: En
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' name='Fli
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><abbr title="Default:
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' name='F
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default:
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' name='F
     </tr>
```

```
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabl
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' name='Flix2_
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enab
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' name='Flix2
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTA
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELOCATIO
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES' name='Fl
      </tr>
     </table>
   </td>
 </tr>
</table>
<!-- END - FLV muxer parameters -->

<!-- FXM muxer parameters -->
<table id='fxmtable' class='hidden'>
```

```
<tr>
 <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'">
     FE2_FXM_CUEPT_EVENT</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_EVENT'></td>
</tr>

<tr>
    <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'">
     FE2_FXM_CUEPT_NAV</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_NAV'></td>
</tr>

<tr>
 <th><abbr title="e.g. 'cuept_name&amp;amp;n0=v0&amp;amp;n1=v1...'">
     FE2_FXM_CUEPT_PARAM</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_PARAM'></td>
</tr>

<tr>
  <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
            FE2_FXM_METADATA_ENABLE</abbr></th>
  <td>
    <table id='fxm_metadata_enable' class=''>
     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Ena
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DURATION' name='Flix
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Ena
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' name='Flix
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' name='Fl
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' name='Fl
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr title="Default: Enable
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' name='Flix2_M
     </tr>
```

```
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr title="Default: Enabl
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' name='Flix2_
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' name='
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAM
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION'><abbr title="
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Di
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' name='Fli
      </tr>
     </table>
   </td>
 </tr>

 <tr>
   <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
             FE2_FXM_METADATA_DISABLE</abbr></th>
   <td>
     <table id='fxm_metadata_disable' class=''>
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' name='F
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' name='F
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE' nam
```

```
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabl
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' name='Flix2_
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enab
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' name='Flix2
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTA
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATIO
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: D
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' name='Fl
      </tr>
     </table>
    </td>
  </tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>
  <tr>
   <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
   <td>
```

```
    <select name="Flix2_MuxerSetParam:FE2_MOV_FASTSTART">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
   </td>
  </tr>
</table>
<!-- END - MOV muxer parameters -->

<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
  <tr>
   <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
   <td>
    <select name="Flix2_MuxerSetParam:FE2_MP4_FASTSTART">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
   </td>
  </tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
 <tr>
  <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_WIDTH'></td>
 </tr>

 <tr>
  <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_HEIGHT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_FRAMERATE'></td>
 </tr>

 <tr>
  <th>FE2_SWF_LOOP_COUNT</th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_LOOP_COUNT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
 </tr>

 <tr>
  <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
  <td>
   <select name='Flix2_MuxerSetParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>
    <option value=""></option>
    <option value="_self">_self</option>
    <option value="_blank">_blank</option>
    <option value="_parent">_parent</option>
    <option value="_top">_top</option>
   </select>
   </td>
 </tr>

 <tr>
  <th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</abbr></th>
```

```
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_EMBEDDED_URL_TYPE'>
   <option value=""></option>
   <option value="EmbeddedUrlIsGetUrl">EmbeddedUrlIsGetUrl</option>
   <option value="EmbeddedUrlIsLoadMovie">EmbeddedUrlIsLoadMovie</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="e.g. n0=v0&amp;n1=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
</tr>

<tr><th><b>Preloader Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_PRELOAD_TYPE'>
   <option value=""></option>
   <option value="SwfPreloaderNone">SwfPreloaderNone</option>
   <option value="SwfFixedPreloader">SwfFixedPreloader</option>
   <option value="SwfAdaptivePreloader">SwfAdaptivePreloader</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTION</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_ON_START_OPTION'>
   <option value=""></option>
   <option value="SwfOnMovieStartAutomatically">SwfOnMovieStartAutomatically</option>
   <option value="SwfOnMovieStartOnClick">SwfOnMovieStartOnClick</option>
   <option value="SwfOnMovieStartWait">SwfOnMovieStartWait</option>
   <option value="SwfOnMovieStartEmbedSTOP">SwfOnMovieStartEmbedSTOP</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_START_BLANK_FRAME'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_START_WAIT_SEC'></td>
</tr>

<tr><th><b>End Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
```

```
   <td>
    <select name='Flix2_MuxerSetParam:FE2_SWF_ON_END_OPTION'>
     <option value=""></option>
     <option value="SwfOnMovieEndNothing">SwfOnMovieEndNothing</option>
     <option value="SwfOnMovieEndSTOP">SwfOnMovieEndSTOP</option>
     <option value="SwfOnMovieEndLoop">SwfOnMovieEndLoop</option>
     <option value="SwfOnMovieEndUnload">SwfOnMovieEndUnload</option>
     <option value="SwfOnMovieEndLoadMovie">SwfOnMovieEndLoadMovie</option>
    </select>
   </td>
  </tr>

  <tr>
   <th>FE2_SWF_ON_END_URL</th>
   <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_ON_END_URL'></td>
  </tr>
 </table>
 <!-- END - SWF muxer parameters -->

 <!-- WEBM muxer parameters -->
 <table id='webmtable' class='hidden'>
   <tr>
    <th>(FE2_MUXER_WEBM defines no parameters)</th>
   </tr>
 </table>
 <!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM########################################################## -->
<hr>
<p>
 <input type="submit" value="Start Encode">
 <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>
HTML_END
```

12.2 process_sample.cgi

```
#!/usr/bin/perl -w
##==========================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##--------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##--------------------------------------------------------------------------
##

#process_sample.cgi
#receive a form via post from flix2_sample.cgi, treating each name=value pair
#as a function/param pair.
#these map to the flixengine2 API and each function that has a valid param
#will be called.
#once setup is complete, call encode to produce an output file located in outdir
```

```
use CGI qw(:standard);
use On2::flixengine2;

sub print_encoder_status()
{
    package On2::flixengine2;
    print "<br>Encoder Status:<br>";
    print " Flix2_GetEncoderState: ".Flix2_GetEncoderState($flix)."<br>";

    my ($sc,@err) = Flix2_Errno($flix);
    printf(" Flix2_Errno: sc:%d flixerrno:%d syserrno:%d<br>",$sc,@err);
}

sub process_sc($$)
{
    my ($funcname,$sc) = @_;
    package On2::flixengine2;

    print "<td align=\"center\">$sc</td>";

    if($sc != $ON2_OK) {
        my $esc;
        @_ = Flix2_Errno($flix); $esc= shift;
        printf("<td>Flix2_Errno: sc:%d %s:%d syserrno:%d</td>",
            $esc,($sc==$ON2_NET_ERROR)?"rpcerr":"flixerrno",@_);
        die "Error calling $funcname, sc= $sc;".
            "esc= $esc Flix2_Errno( $_[0], $_[1] )\n";
    }

    print "</tr>";
}

sub leadin()
{
    package On2::flixengine2;

    print "<tr><td>Flix2_CreateEx()</td>";

    $flixptr = new_flix2handlep();
    my $sc = Flix2_CreateEx($flixptr,$::rpchost,0);

    #extract the handle value returned from _Create. $flix will be used in
    #every Flix API call that follows
    $flix = flix2handlep_value($flixptr);

    ::process_sc('Flix2_CreateEx',$sc);
}

sub setfunc($$)
{
    my ($funcname,$funcparam) = @_;
    package On2::flixengine2;
    print "<tr><td>$funcname( $funcparam )</td>";
    my $sc;
    if($funcname eq 'Flix2_SetInputFile' || $funcname eq 'Flix2_SetOutputFile') {
        $sc = &$funcname($flix,$funcparam);
    } else {
        $sc = &$funcname($flix,${$funcparam});
    }
    ::process_sc($funcname,$sc);
}

sub init_codec($)
{
    my $name= $_[0];
    package On2::flixengine2;
```

```
    ##if name is a codec name, e.g. FE2_CODEC_VP6, add an instance
    ##we'll assume all Flix2_CodecSetParam's relate to this codec until we
    ##hit the next codec name
    if($codecptr) { delete_flix2plgnhandlep($codecptr); }

    print "<tr><td>Flix2_AddCodec( $name )</td>";
    $codecptr = new_flix2plgnhandlep();
    $sc = Flix2_AddCodec($codecptr, $flix, ${$name});
    ::process_sc('Flix2_AddCodec',$sc);

    #retrieve the codec handle to be used in all Codec API function calls
    $codec = flix2plgnhandlep_value($codecptr);
}

sub codec_interface($$$)
{
    my ($funcname,$name,$value) = @_;
    package On2::flixengine2;
    print "<tr><td>$funcname( $name, $value )</td>";

    $sc = &$funcname($codec,${$name},($funcname=~/AsStr/ or $value!~/[[:alpha:]_]/)?$value:${$value});
    ::process_sc($funcname,$sc);
}

sub init_filter($)
{
    my $name= $_[0];
    package On2::flixengine2;

    ##if name is a filter name, e.g. FE2_FILTER_CUT, add an instance
    ##we'll assume all Flix2_FilterSetParam's relate to this filter until we
    ##hit the next filter name
    if($filterptr) { delete_flix2plgnhandlep($filterptr); }

    print "<tr><td>Flix2_AddFilter( $name )</td>";
    $filterptr = new_flix2plgnhandlep();
    $sc = Flix2_AddFilter($filterptr, $flix, ${$name});
    ::process_sc('Flix2_AddFilter',$sc);

    #retrieve the codec handle to be used in all Codec API function calls
    $filter = flix2plgnhandlep_value($filterptr);
}

sub filter_interface($$$)
{
    my ($funcname,$name,$value) = @_;
    package On2::flixengine2;
    print "<tr><td>$funcname( $name, $value )</td>";

    $sc = &$funcname($filter,${$name},($funcname=~/AsStr/ or $value!~/[[:alpha:]_]/)?$value:${$value});
    ::process_sc($funcname,$sc);
}

sub init_muxer($)
{
    my $name= $_[0];
    package On2::flixengine2;

    ##if name is a muxer name, e.g. FE2_MUXER_FLV, add an instance
    ##we'll assume all Flix2_MuxerSetParam's relate to this muxer until we
    ##hit the next muxer name
    if($muxerptr) { delete_flix2plgnhandlep($muxerptr); }

    print "<tr><td>Flix2_AddMuxer( $name )</td>";
    $muxerptr = new_flix2plgnhandlep();
    $sc = Flix2_AddMuxer($muxerptr, $flix, ${$name});
```

```
    ::process_sc('Flix2_AddMuxer',$sc);

    #retrieve the muxer handle to be used in all Muxer API function calls
    $muxer = flix2plgnhandlep_value($muxerptr);
}

sub muxer_interface($$$)
{
    my ($funcname,$name,$value) = @_;
    package On2::flixengine2;
    print "<tr><td>$funcname( $name, $value )</td>";

    $sc = &$funcname($muxer,${$name},($funcname=~/AsStr/ or $value!~/[[:alpha:]_]/)?$value:${$value});
    ::process_sc($funcname,$sc);
}

sub encode()
{
    package On2::flixengine2;

    print "<tr><td>Flix2_Encode()</td>";
    my ($sc) = Flix2_Encode($flix);
    ::process_sc('Flix2_Encode',$sc);
    print "</table><br>\n";

    print CGI::br,"Encoding...(video frames encoded, percent complete). ",
                  "Total frames will reset when doing 2pass.",CGI::br;
    my $ier;
    do {
        $ier= Flix2_IsEncoderRunning($flix);
        my ($sc, $tf)= encoding_status_GetTotalFrames($flix);
        my $p= encoding_status_PercentComplete($flix);
        print "($tf, $p%)<br>" if ($sc==$ON2_OK);
        sleep 1;
    } while($ier);
    print "<br>Done!";
    ::print_encoder_status();

    Flix2_Destroy($flix);
    delete_flix2plgnhandlep($filterptr) if($filterptr);
    delete_flix2plgnhandlep($codecptr) if($codecptr);
    delete_flix2plgnhandlep($muxerptr) if($muxerptr);
    delete_flix2handlep($flixptr);
}

$| = 1; #auto flush

print header;
print start_html("Flix CGI Process Sample - Perl");

$rpchost = "localhost";
my $prefix = "/var/www/cgi-bin/";
my $outdir = $prefix."flixmedia/out/";

print "<hr>";
print "<p>process_sample.cgi version 1.8<br>";
print "Flix Engine client library v".
      On2::flixengine2::Flix2_Version()."<br>";
($_= On2::flixengine2::Flix2_Copyright()) =~ s/\n/<br>/g;
print "$_</p>";

##verify outdir's presence and accessibility
if ($rpchost eq "localhost" && !(-d $outdir && -w $outdir)) {
  my ($i,$tt)=("<i>'$outdir'</i>",'<tt>$outdir</tt>');
  print "<p>****************************************<br>\n".
        "<b>WARNING</b>: $i MUST exist and be writeable by <i>flixd</i>.\n".
        "<br>Please make $i accessible or modify the $tt".
```

```
        " value defined in '$0'.<br>\n";
  my $g=`grep -m 1 -H -n '^my \$outdir' '$0'`;
  if($g) {
    print "The definition of $tt can be found here:<br>\n$g<br>\n";
  }
  print "****************************************</p>\n";
}

print "<table border='1' cellpadding='5'>";
print "<caption>Flix Function Calls</caption>\n";
print "<tr><th>Function Name</th><th>Return Value</th></tr>\n";

leadin();

foreach my $p (param()) {
    if(param($p) ne "") {
        if ($p eq 'Flix2_SetOutputFile') {
            $outfile = $outdir.param($p);
            setfunc($p,$outfile)
        } elsif ($p =~ /(?:a|v)codec:/) {
            init_codec(param($p));
        } elsif (param($p) =~ /(?:a|v)filter:/) {
            init_filter($p);
        } elsif ($p =~ /muxer:/) {
            init_muxer(param($p));
        } elsif ($p =~ /CodecSetParam/) {
            my @temp = split(/:/,$p);
            codec_interface($temp[0],$temp[1],param($p));
        } elsif ($p =~ /FilterSetParam/) {
            my @temp = split(/:/,$p);
            filter_interface($temp[0],$temp[1],param($p));
        } elsif ($p =~ /MuxerSetParam/) {
            my @temp = split(/:/,$p);
            muxer_interface($temp[0],$temp[1],param($p));
        } else {
            #call an api function of the form FUNC(flixhandle,param)
            setfunc($p,param($p));
        }
    }
}

encode();
print end_html;

# vim:ts=4:expandtab
```

Chapter 13

Command Line

```
#!/usr/bin/env php
<?php
##=========================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##-------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##-------------------------------------------------------------------------
##

#checks the return value of an API function printing error information on
#failure. usage checksc(funcname,sc)
function checksc($func, $sc) {
    global $flix;
    if ($sc != ON2_OK) {
        echo "$func failed: sc= $sc\n";

        ##if sc == ON2_NET_ERROR Flix2_Errno will return the specific rpc error
        ##encountered as flixerrno along with the client lib's errno value
        $res = Flix2_Errno($flix);
        printf(" Flix2_Errno: sc:%d %s:%d syserrno:%d\n",
            $res[0],($sc==ON2_NET_ERROR)?"rpcerr":"flixerrno",$res[1],$res[2]);
        die;
    }
}

function print_encoder_status() {
    global $flix;

    echo "\nEncoder Status\n";

    $res = Flix2_GetEncoderState($flix);
    echo " Flix2_GetEncoderState: $res[1]\n";

    $res = Flix2_Errno($flix);
    printf(" Flix2_Errno: sc:%d flixerrno:%d syserrno:%d\n",
        $res[0],$res[1],$res[2]);
}

# Load the FlixEngine module
$flixphp = 'flixengine2.php';
echo 'Loading flix: ' . $flixphp . "\n";
# If this include fails you may need to edit you include_path variable in
# your php.ini file.
# See the accompanying Flix Linux Engine documentation for further details.
# add extension_dir/.. to 'include_path' as this is often missing from php.ini
set_include_path(get_include_path().PATH_SEPARATOR.ini_get("extension_dir")."/..");
include ($flixphp);

printf("Flix Engine client library v%s\n",Flix2_Version());
printf("%s\n\n",Flix2_Copyright());

if (count($argv) < 3) {
    die("usage: cli_encode.php <infile> <outfile>\n\n".
        "NOTE cli_encode.php uses libflixengine2.so which is a client\n".
        "NOTE side rpc library. All paths must be accessible to the\n".
        "NOTE server side, i.e., flixd, thus relative paths will most\n".
        "NOTE likely give undesired results. The same can be said\n".
        "NOTE for clients running on different machines.\n");
}
```

```
$timeout_s = 0; #rpc timeout in seconds, 0=use default (25s)

$flixptr = new_flix2handlep();
$sc = Flix2_CreateEx($flixptr, "localhost", $timeout_s);

# retrieve the actual flix handle for use in the remaining API calls
$flix = flix2handlep_value($flixptr);
  checksc('Flix2_CreateEx',$sc);

echo "Input File  : $argv[1]\n";
if(!ereg("^/",$argv[1])) { echo "WARNING: path to input file is not absolute\n"; }
$sc = Flix2_SetInputFile($flix, $argv[1]);
  checksc('Flix2_SetInputFile',$sc);

##input file information
$srcduration = Flix2_GetSourceDuration($flix);
  checksc('Flix2_GetSourceDuration',$srcduration[0]);
$srcw = video_options_GetSourceWidth($flix);
  checksc('video_options_GetSourceWidth',$srcw[0]);
$srch = video_options_GetSourceHeight($flix);
  checksc('video_options_GetSourceHeight',$srch[0]);

echo <<<EOT
             Width:     $srcw[1]
             Height:    $srch[1]
             Duration:  $srcduration[1]ms

EOT;

echo "Output File : $argv[2]\n";
if(!ereg("^/",$argv[2])) { echo "WARNING: path to output file is not absolute\n"; }
$sc = Flix2_SetOutputFile($flix, $argv[2]);
  checksc('Flix2_SetOutputFile',$sc);

##
## Options may be set and codecs/filters/muxers may be added prior to Flix2_Encode()
##

##Add the scale filter
##create a storage location for the filter handle
#$filterptr= new_flix2plgnhandlep();
#$sc = Flix2_AddFilter($filterptr,$flix,FE2_FILTER_SCALE);
#  checksc('Flix2_AddFilter(FE2_FILTER_SCALE)',$sc);
#
##retrieve the value of the handle for use in the remaining filter functions
#$filter= flix2plgnhandlep_value($filterptr);
#$sc = Flix2_FilterSetParam($filter,FE2_SCALE_WIDTH,240);
#  checksc('Flix2_FilterSetParam(FE2_SCALE_WIDTH,240)',$sc);
#$sc = Flix2_FilterSetParam($filter,FE2_SCALE_HEIGHT,160);
#  checksc('Flix2_FilterSetParam(FE2_SCALE_HEIGHT,160)',$sc);
#
##cleanup
#delete_flix2plgnhandlep($filterptr); unset($filterptr);
#
##Add the vp6 codec. Though it is the default, you must add it in order
##to modify its settings
#$codecptr= new_flix2plgnhandlep();
#$sc = Flix2_AddCodec($codecptr,$flix,FE2_CODEC_VP6);
#  checksc('Flix2_AddCodec(FE2_CODEC_VP6)',$sc);
#
##retrieve the value of the handle for use in the remaining codec functions
#$codec= flix2plgnhandlep_value($codecptr);
#
#$sc = Flix2_CodecSetParam($codec,FE2_VP6_RC_MODE,VBR_1PASSControl);
#  checksc('Flix2_CodecSetParam(FE2_VP6_RC_MODE,VBR_1PASSControl)',$sc);
#
##cleanup
```

```
#delete_flix2plgnhandlep($codecptr); unset($codecptr);
#
##Use the FLV muxer (default)
#$muxerptr= new_flix2plgnhandlep();
#$sc = Flix2_AddMuxer($muxerptr,$flix,FE2_MUXER_FLV);
#  checksc('Flix2_AddMuxer(FE2_MUXER_FLV)',$sc);
#
##retrieve the value of the handle for use in the remaining muxer functions
#$muxer= flix2plgnhandlep_value($muxerptr);
#
##cleanup
#delete_flix2plgnhandlep($muxerptr); unset($muxerptr);

$sc = Flix2_Encode($flix); checksc('Flix2_Encode',$sc);

echo "\n";
do {
    $ier = Flix2_IsEncoderRunning($flix);
      checksc('Flix2_IsEncoderRunning',$ier[0]);

    $pcnt = encoding_status_PercentComplete($flix);
      checksc('encoding_status_PercentComplete',$pcnt[0]);
    echo "\rEncoding...$pcnt[1]%  ";

    sleep(1);
} while(($sc == ON2_OK) && ($ier[1] != on2false));

echo "Done!\n";
print_encoder_status();

#cleanup
$sc = Flix2_Destroy($flix); checksc('Flix2_Destroy',$sc);
delete_flix2handlep($flixptr);
?>
```

Chapter 14

CGI

This example consists of 2 parts: flix2_sample.php and process_sample.php. flix2_sample.php searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's include files.

To use this example navigate to flix2_sample.php in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to process_sample.php which runs the encode loop.

Default file locations used by the scripts:

Input : */var/www/cgi-bin/flixmedia/in* (`$indir` in flix2_sample.php)
Overlay : */var/www/cgi-bin/flixmedia/overlay* (`$overlaydir` in flix2_sample.php)
Output : */var/www/cgi-bin/flixmedia/out* (`$outdir` in process_sample.php)

14.1 flix2_sample.php

```
#!/usr/bin/env php
<?php
##==============================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##------------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##------------------------------------------------------------------------------
##

## add extension_dir/.. to 'include_path' as this is often missing from php.ini
set_include_path(get_include_path().PATH_SEPARATOR.ini_get("extension_dir")."/..");

$incret     = include ('flixengine2.php');
$prefix     = "/var/www/cgi-bin/";
$indir      = $prefix."flixmedia/in/";
$overlaydir = $prefix."flixmedia/overlay/";

echo <<<HTML_END

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head>
 <title>Flix CGI Sample - PHP</title>
 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

 <script type="text/javascript">
 function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
        i++;
    }
 }

 function toggle_expand(_this)
 {
     _this.parentNode.className= (_this.parentNode.className=='expanded') ?
                                 'collapsed' : 'expanded';
```

```
   document.getElementById('showall').checked=false;
}

function set_table_visible(ctable,visible)
{
   var table= document.getElementById(ctable);

   if(table) {
      if(visible && table.className == 'hidden') {
          table.className= '';
      } else if(!visible && table.className == '') {
          table.className= 'hidden';
      }
   }
}

function hide_tablelist(vistablename,tablelist)
{
  var table= document.getElementById(vistablename);

  if(table&&table.className=='hidden') {
    table.className= '';

    var i=0;
    while(tablelist[i]) {
      if(tablelist[i]!=vistablename) {
        var hiddentable= document.getElementById(tablelist[i]);

        set_table_visible(tablelist[i],false);
        /*clear down any values set in the hidden table to avoid posting
          unnecessary values*/
        clear_value(hiddentable.getElementsByTagName('input'));
        clear_value(hiddentable.getElementsByTagName('select'));
      }
      i++;
    }
  }
}

function set_acodec_visible(ctable)
{
  var acodecs= new Array('aactable','aacplustable','amrnbtable','lametable','pcmtable','vorbistable');
  hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
  var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8table');
  hide_tablelist(ctable,vcodecs);
}

function set_muxer_visible(mtable)
{
  var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','tg2table','tgptable','web
  hide_tablelist(mtable,muxers);
}

function clear_value(list)
{
   var i=0;
   while(list[i]) {
     if(list[i].type=='checkbox') { list[i++].checked=false; }
     else { list[i++].value=''; }
   }
}

function toggle_ftable(ftable,enabled)
```

```
{
  var table= document.getElementById(ftable);

  if(table) {
    table.className = enabled ? '' : 'disabled';
    if(!enabled) {
      clear_value(table.getElementsByTagName('input'));
      clear_value(table.getElementsByTagName('select'));
    }
  }
}

function reset_tables()
{
   var table_list= document.getElementsByTagName('table');
   var i=0;
   while(table_list[i]) {
     if (table_list[i].id.length > 7 &&
       table_list[i].id.substring(0,7) == 'filter_') {
       table_list[i].className= 'disabled';
     } else if (table_list[i].id.indexOf('table') != -1) {
       table_list[i].className= 'hidden';
     }
     i++;
   }
}

</script>

<style type="text/css">
   <!--
   html {
       font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
       font-size: 100%;
       color: rgb(56,56,56);
       background-color: rgb(236,236,236);
       border-style: solid;
       border-color: rgb(236,236,236);
   }

   body {
       text-align: center;
       margin: 0 auto;
   }

   div.content {
       color: rgb(56,56,56);
       background-color: rgb(246,246,246);
       text-align: left;
       margin: 0 auto;
       width: 80%;
       min-width: 768px;
       max-width: 932px;
       border-width: 0 1px;
       border-color: rgb(144,144,144);
       border-style: solid;
   }

   div.content:after {
       content: "";
       color: inherit;
       background-color: rgb(250,250,250);
       border-top: 1px solid rgb(144,144,144);
       height: 20px;
       width: 100%;
       display: block;
   }
```

```
fieldset table, fieldset {display: none;}
fieldset.expanded, fieldset.collapsed {display: block;}

/*first is fallback for IE*/
fieldset.expanded table {display: block;}
fieldset.expanded table {display: table;}

h1 {
    font-family: sans-serif;
    font-size: 150%;
    font-weight: normal;
    text-align: left;
    letter-spacing: -1px;
    color: rgb(74,74,74);
    background-color: inherit;
    margin: 0;
}

a {
    color: rgb(74,74,74);
    background-color: transparent;
}

label {
    font-size: 75%;
}

fieldset {
    font-size: 75%;
    line-height: 130%;
    padding: 0;
    margin: 20px;
    border: none;
}

fieldset.expanded {
    color: inherit;
    background-color: rgb(252,252,252);
    border-style: solid;
    border-width: 1px;
    border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
}

legend {
    padding: 0 5px;
    border-left: 10px solid rgb(217,217,217);
    cursor: pointer;
}
legend:hover {text-decoration: underline;}
fieldset.expanded legend {
    font-size: 150%;
    font-weight: bold;
    letter-spacing: -1px;
    background: transparent;
    margin-left: 12px;
    border-right: 10px solid rgb(217,217,217);
    display: block;
}
table {
    font-size: 100%;
    border-spacing: 0;
    /*border-collapse: collapse;*/
    width: 100%;
}

th:before { display: none; }
```

```
    th, td {
        width: 50%;
        vertical-align: top;
        padding: 2px 3px;
        border-width: 1px 0;
        border-style: solid;
        border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
    }
    tr:first-child>* {border-top-color: rgb(252,252,252);}
    th {
        font-weight: normal;
        text-align: left;
        padding: 2px 2px 2px 5px;
    }
    input[type] {
        font-family: monospace;
        font-size: 100%;
        color: rgb(56,56,56);
        background-color: inherit;
    }
    [type="text"], [type="number"] {
        margin-right: 13px;
        width: 222px;
    }

    [type="checkbox"].filter {
        margin-left: 4px
    }
    [type="checkbox"] {
        margin-left: 13px
    }
    [type="button"], [type="submit"] {
        font-size: 1em;
        margin: 0 2px 0 13px;
    }

    fieldset {
        font-size: 75%;
        margin: 20px 10px;
    }
    fieldset input[type="text"], fieldset input[type="number"] {
        width: 95% !important;
        margin: 0;
        display: block;
    }
    fieldset.expanded table.hidden {
        display: none;
    }

    table.disabled {
        color: rgb(176,176,176);
    }
    -->
 </style>
</head>

<body>
 <div class='content'>
 <noscript>
  <p>This page requires javascript be enabled.</p>
 </noscript>

<hr>
<h1>Flix CGI Sample</h1>

HTML_END;
if($incret == FALSE) {
```

```
    echo "<p><b>".
         "ERROR: include('flixengine2.php') failed.<br>".
         "ERROR: This include file loads the PHP bindings for the Flix Engine.<br>".
         "ERROR: Without them this sample will be unable to contact the encoding daemon.<br>".
         "ERROR: See the server error log for more information.".
         "</b></p>";
    echo "</body></html>";
    error_log("[flix2_sample.php] include('flix2_engine.php') failed, see 'PHP Notice's' above");
    die;
}
echo <<<HTML_END


<p><small>flix2_sample.cgi version 1.9</small></p>
<h4>Instructions</h4>
<ul>
 <li>In this sample you must choose a source file and an output file.<br>
 If you leave all the other options blank then the sample will not call
 the corresponding Flix Engine function and the default will be used.<br>
 When done please press the "Start Encode" button at the bottom of the page.<br>
 <li>Mouse over a function name to see its default, if applicable.
 <li>Current source file directory: $indir
</ul>

<p>
 <label><input type="checkbox" id='showall' onclick='showall(this.checked)'>Show all</label>
</p>

<form action="process_sample.php" method="post">

<!-- ##SOURCE FILE############################################################ -->
<hr>
<fieldset class='expanded' id="srcfile">
<legend onclick='toggle_expand(this)'>Source File</legend>
<table>

<tr>
 <td>
HTML_END;

if(is_dir($indir)) {
    $first= TRUE;
    #let glob sort the directory contents for us
    $files= glob($indir.'*');
    echo "<select name='Flix2_SetInputFile' size='".(count($files)-1<15?count($files)-1:15)."'>\n";
    foreach($files as $f) {
        if(!is_dir($f)) {
            echo "<option ";
            if($first) { echo "selected "; $first=FALSE; };
            echo "value=\"$f\">".str_replace($indir,'',$f)."</option>\n";
        }
    }
    echo "</select>";
} else {
    echo "WARNING couldn't open $indir: is_dir() returned FALSE<br>\n";
}

echo <<<HTML_END

 </td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE################################################################ -->
<hr>
```

```
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)'>Output File</legend>
<table>

<tr>
 <td>
  <input type="text" name="Flix2_SetOutputFile" value="cgi-php-out.flv">
 </td>
</tr>

</table>
</fieldset>

<!-- ##MAIN OPTIONS################################################################ -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)'>Main Options</legend>
<table>

<tr>
 <th><abbr title="Default: FALSE">Flix2_SetOverwriteExistingFiles</abbr></th>
 <td>
 <select name="Flix2_SetOverwriteExistingFiles">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: TRUE">Flix2_SetExportAudio</abbr></th>
 <td>
 <select name="Flix2_SetExportAudio">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: TRUE">Flix2_SetExportVideo</abbr></th>
 <td>
 <select name="Flix2_SetExportVideo">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

</table>
</fieldset>

<!-- ##CODECS#######################################################################  -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this)'>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
 <td>
 <label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
         onfocus="set_vcodec_visible('vp8table')">
```

```
 <abbr title="For use with WebM">FE2_CODEC_VP8</abbr> 
</label>
<br>

<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
        onfocus="set_vcodec_visible('vp6table')">
 <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
        onfocus="set_vcodec_visible('vp6atable')">
 <abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H263'
        onfocus="set_vcodec_visible('h263table')">
 <abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr> 
</label>
<br>

<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
        onfocus="set_vcodec_visible('h263table')">
 <abbr title="For use with 3GP">FE2_CODEC_H263_BASELINE</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H264'
        onfocus="set_vcodec_visible('h264table')">
 <abbr title="For use with 3GP/3G2/MOV/MP4">FE2_CODEC_H264</abbr> 
</label>

<!-- VP6 codec parameters -->
<table id='vp6table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_VP6_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP6_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP6_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
   <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
  </td>
 </tr>
```

```
<tr>
 <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6_CXMODE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP6_CXMODE">
  <option value=""></option>
  <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
  <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: VP6_E">FE2_VP6_PROFILE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP6_PROFILE">
  <option value=""></option>
  <option value="VP6_E">VP6_E</option>
  <option value="VP6_S">VP6_S</option>
 </select>
 </td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_CONCURRENCY'></td>
</tr>

<tr>
 <th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_UNDERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEMPORAL_RESAMPLING</abbr></
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEMPORAL_DOWN_WATERMARK</abb
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
 <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
```

```
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_PREBUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_MAX_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_2PASS_MIN_SECTION'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_2PASS_MAX_SECTION'></td>
 </tr>

</table>
<!-- END - VP6 codec parameters -->

<!-- VP6A codec parameters -->
<table id='vp6atable' class='hidden'>
 <tr>
  <th><abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 68kbps (15% of default 448kbps)">FE2_VP6A_ALPHA_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6A_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP6A_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6A_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
```

```
  <option value="CBR_1PASSControl">CBR_1PASSControl</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP6A_CXMODE">
  <option value=""></option>
  <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
  <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
 </select>
 </td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
 <th><abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_RESAMPLING</abbr><
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
</tr>
```

```
 <tr>
  <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_DOWN_WATERMARK</ab
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_PREBUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_2PASS_MIN_SECTION'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_2PASS_MAX_SECTION'></td>
 </tr>

</table>
<!-- END - VP6A codec parameters -->

<!-- H263 codec parameters -->
<table id='h263table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H263_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_H263_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H263_RC_MODE">
    <option value=""></option>
```

```
    <option value="VBR_2PASSControl">VBR_2PASSControl</option>
    <option value="CBR_2PASSControl">CBR_2PASSControl</option>
    <option value="VBR_1PASSControl">VBR_1PASSControl</option>
    <option value="CBR_1PASSControl">CBR_1PASSControl</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_MAX_Q'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_MIN_Q'></td>
 </tr>

</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_H264_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H264_RC_MODE">
    <option value=""></option>
    <option value="VBR_1PASSControl">VBR_1PASSControl</option>
    <option value="CBR_1PASSControl">CBR_1PASSControl</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H264_PROFILE">
    <option value=""></option>
    <option value="BASE_H264PROFILE">BASE_H264PROFILE</option>
    <option value="MAIN_H264PROFILE">MAIN_H264PROFILE</option>
    <option value="HIGH_H264PROFILE">HIGH_H264PROFILE</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_B_FRAME_RATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: Dependent on profile selection, see API docs. Valid Range [0,5]">FE2_H264_SPE
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_SPEED'></td>
 </tr>

</table>
```

```
<!-- END - H264 codec parameters -->

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP8_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP8_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP8_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
   <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP8_CXMODE">
   <option value=""></option>
   <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
   <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_THREADS'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_PROFILE'></td>
 </tr>

 <tr><th><b>Advanced Settings:</b></th></tr>

 <tr>
  <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_LAG'></td>
 </tr>

 <tr>
```

```
 <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_UNDERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_OVERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
 <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_ALTREF'></td>
</tr>

<tr>
 <th><abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_MAX_FRAMES'></td>
```

```
  </tr>

  <tr>
   <th><abbr title="">FE2_VP8_AR_TYPE</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_TYPE'></td>
  </tr>

  <tr>
   <th><abbr title="">FE2_VP8_AR_STRENGTH</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_STRENGTH'></td>
  </tr>

  <tr>
   <th><abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MB_STATIC_THRESHOLD'></td>
  </tr>

  <tr>
   <th><abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_TOKEN_PARTITIONS'></td>
  </tr>

 </table>
 <!-- END - VP8 codec parameters -->

 </td>
</tr> <!-- END - video codecs -->

<tr><th><b>Audio Codecs</b></th></tr>
<tr>
 <td>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_VORBIS'
         onfocus="set_acodec_visible('vorbistable')">
  <abbr title="For use with WebM">FE2_CODEC_VORBIS</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AAC'
         onfocus="set_acodec_visible('aactable')">
  <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AAC</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AACPLUS'
         onfocus="set_acodec_visible('aacplustable')">
  <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AACPLUS</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_LAME'
         onfocus="set_acodec_visible('lametable') ">
  <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_LAME</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_PCM'
         onfocus="set_acodec_visible('pcmtable')">
  <abbr title="For use with FLV/SWF">FE2_CODEC_PCM</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AMR_NB'
         onfocus="set_acodec_visible('amrnbtable')">
  <abbr title="For use with 3GP">FE2_CODEC_AMR_NB</abbr> 
 </label>

 <!-- AMR_NB codec parameters -->
```

```
<table id='amrnbtable' class='hidden'>
 <tr>
  <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_AMR_BITRATE'></td>
 </tr>

</table>
<!-- END - AMR_NB codec parameters -->

<!-- AAC codec parameters -->
<table id='aactable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_AAC_BITRATE'></td>
 </tr>

</table>
<!-- END - AAC codec parameters -->

<!-- AACPLUS codec parameters -->
<table id='aacplustable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_AACPLUS_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: disabled (aacPlus v1)">FE2_AACPLUS_PARAMETRIC_STEREO</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_AACPLUS_PARAMETRIC_STEREO">
   <option value=""></option>
   <option value="0">disable (aacPlus v1)</option>
   <option value="1">enable (aacPlus v2)</option>
   </select>
  </td>
 </tr>
</table>
<!-- END - AACPLUS codec parameters -->

<!-- LAME codec parameters -->
<table id='lametable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_LAME_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_LAME_QUALITY'></td>
 </tr>

 <tr>
  <th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_LAME_RC_MODE">
   <option value=""></option>
   <option value="LAME_CBR">LAME_CBR</option>
   <option value="LAME_ABR">LAME_ABR</option>
   <option value="LAME_VBR_rh">LAME_VBR_rh</option>
   <option value="LAME_VBR_mtrh">LAME_VBR_mtrh</option>
   </select>
  </td>
 </tr>
</table>
<!-- END - LAME codec parameters -->

<!-- PCM codec parameters -->
```

```
 <table id='pcmtable' class='hidden'>
  <tr>
   <th>(FE2_CODEC_PCM defines no parameters)</th>
  </tr>
 </table>
 <!-- END - PCM codec parameters -->

 <!-- VORBIS codec parameters -->
 <table id='vorbistable' class='hidden'>
  <tr>
   <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VORBIS_BITRATE'></td>
  </tr>
 </table>
 <!-- END - VORBIS codec parameters -->

 </td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS######################################################################### -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this)'>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_CUT' value='avfilter:'
          onchange="toggle_ftable('filter_cut',this.checked)">
   FE2_FILTER_CUT
  </label>

  <table id='filter_cut' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CUT_START_SEC'></td>
   </tr>

   <tr>
    <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CUT_STOP_SEC'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_CUT_USE_SEEK">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - CUT filter parameters -->
```

```
<tr><th><b>Video Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:'
          onchange="toggle_ftable('filter_adaptive_deinterlace',this.checked)">
   FE2_FILTER_ADAPTIVE_DEINTERLACE
  </label>

  <table id='filter_adaptive_deinterlace' class='disabled'>
   <tr>
    <th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_ADAPTIVE_DEINTERLACE_MODE">
     <option value=""></option>
     <option value="DEINTERLACE_NONE">DEINTERLACE_NONE</option>
     <option value="DEINTERLACE_1_2_1_BLUR">DEINTERLACE_1_2_1_BLUR</option>
     <option value="DEINTERLACE_DROP_FIELD">DEINTERLACE_DROP_FIELD</option>
     <option value="DEINTERLACE_ADAPTIVE">DEINTERLACE_ADAPTIVE</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - ADAPTIVE DEINTERLACE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_BCHS' value='vfilter:'
          onchange="toggle_ftable('filter_bchs',this.checked)">
   FE2_FILTER_BCHS
  </label>

  <table id='filter_bchs' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_BRIGHTNESS'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_CONTRAST'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_HUE'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_SATURATION'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - BCHS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
```

```
          name='FE2_FILTER_BLUR' value='vfilter:'
          onchange="toggle_ftable('filter_blur',this.checked)">
   FE2_FILTER_BLUR
  </label>

  <table id='filter_blur' class='disabled'>
   <tr>
    <th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_BLUR_FILTER">
     <option value=""></option>
     <option value="BLUR_LOWPASS">BLUR_LOWPASS</option>
     <option value="BLUR_GAUSS">BLUR_GAUSS</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_BLUR_MASKSIZE">
     <option value=""></option>
     <option value="MASK_3x3">MASK_3x3</option>
     <option value="MASK_5x5">MASK_5x5</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_CROP' value='vfilter:'
          onchange="toggle_ftable('filter_crop',this.checked)">
   FE2_FILTER_CROP
  </label>

  <table id='filter_crop' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_TOP'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_BOTTOM'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_LEFT'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_RIGHT'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - CROP filter parameters -->
```

```
<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_DENOISE' value='vfilter:'
          onchange="toggle_ftable('filter_denoise',this.checked)">
   FE2_FILTER_DENOISE
  </label>

  <table id='filter_denoise' class='disabled'>
   <tr>
    <th><abbr title="Default: 0. Range: [0.0,1.0)">FE2_DENOISE_NOISE_LEVEL</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_DENOISE_NOISE_LEVEL'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_FRAMERATE' value='vfilter:'
          onchange="toggle_ftable('filter_framerate',this.checked)">
   FE2_FILTER_FRAMERATE
  </label>

  <table id='filter_framerate' class='disabled'>
   <tr>
    <th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAMERATE_DECIMATE</abbr></th
    <td><input type='text' name='Flix2_FilterSetParam:FE2_FRAMERATE_DECIMATE'></td>
   </tr>

   <tr>
    <th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FRAMERATE_FPS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_FRAMERATE_FPS'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_MIRROR' value='vfilter:'
          onchange="toggle_ftable('filter_mirror',this.checked)">
   FE2_FILTER_MIRROR
  </label>

  <table id='filter_mirror' class='disabled'>
   <tr>
    <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_MIRROR_HORIZONTAL">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
```

```
   <tr>
    <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_MIRROR_VERTICAL">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_OVERLAY' value='vfilter:'
          onchange="toggle_ftable('filter_overlay',this.checked)">
   FE2_FILTER_OVERLAY
  </label>

  <table id='filter_overlay' class='disabled'>
   <tr>
    <th>
     <abbr title="Currently searching $overlaydir for overlay images">FE2_OVERLAY_FILE</abbr>
    </th>
    <td>
HTML_END;

if(is_dir($overlaydir)) {
    echo "<select name=\"Flix2_FilterSetParamAsStr:FE2_OVERLAY_FILE\">\n";
    echo "<option value=\"\"></option>\n";
    foreach(glob($overlaydir.'*') as $f) {
        if ((eregi("\.bmp",$f) || eregi("\.png",$f)) && !is_dir($f)) {
            echo "<option value=\"$f\">".
                 str_replace($overlaydir,'',$f)."</option>\n";
        }
    }
    echo "</select>\n";
    echo "<br>default: None, must be set to the absolute path of the overlay";
    echo " input file, e.g. '/path/to/my/overlay.png'";
} else {
    echo "WARNING couldn't open $overlaydir: is_dir() returned FALSE<br>\n";
}

echo <<<HTML_END

    </td>
   </tr>

   <tr>
    <th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_MASK_XY">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
```

```
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_X'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_Y'></td>
   </tr>

   <tr>
    <th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_MASK_RGB">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_R'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_G'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_B'></td>
   </tr>

   <tr>
    <th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_POS">
      <option value=""></option>
      <option value="FE2_OVERLAY_POS_MODE_TOPLEFT">FE2_OVERLAY_POS_MODE_TOPLEFT</option>
      <option value="FE2_OVERLAY_POS_MODE_BOTLEFT">FE2_OVERLAY_POS_MODE_BOTLEFT</option>
      <option value="FE2_OVERLAY_POS_MODE_CENTER">FE2_OVERLAY_POS_MODE_CENTER</option>
      <option value="FE2_OVERLAY_POS_MODE_TOPRIGHT">FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
      <option value="FE2_OVERLAY_POS_MODE_BOTRIGHT">FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
      <option value="FE2_OVERLAY_POS_MODE_XY">FE2_OVERLAY_POS_MODE_XY</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_POS_X'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_POS_Y'></td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_PNGEX' value='vfilter:'
          onchange="toggle_ftable('filter_pngex',this.checked)">
```

```
 FE2_FILTER_PNGEX
</label>

<table id='filter_pngex' class='disabled'>
 <tr>
  <th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'></td>
 </tr>

 <tr>
  <th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_FILENAME_PREFIX'></td>
 </tr>

 <tr>
  <th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'></td>
 </tr>

 <tr>
  <th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_WIDTH'></td>
 </tr>

 <tr>
  <th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_HEIGHT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>
  <td>
   <select name="Flix2_FilterSetParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">
   <option value=""></option>
   <option value="on2false">FALSE</option>
   <option value="on2true">TRUE</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
  <td>
   <select name="Flix2_FilterSetParam:FE2_PNGEX_ENABLE_ALPHA">
   <option value=""></option>
   <option value="on2false">FALSE</option>
   <option value="on2true">TRUE</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="interval in ms; Default: disabled">FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_EXPORT_INTERVAL'></td>
 </tr>

 <tr>
  <th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
  <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_EXPORT_TIME_STRING'></td>
 </tr>

 <tr>
  <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
  <td>
   <select name="Flix2_FilterSetParam:FE2_PNGEX_EXPORT_CUE_POINTS">
   <option value=""></option>
   <option value="FE2_PNGEX_CP_ALL">All cue points (FE2_PNGEX_CP_ALL)</option>
```

```
     <option value="FE2_PNGEX_CP_NAV">Only navigation cue points (FE2_PNGEX_CP_NAV)</option>
     <option value="FE2_PNGEX_CP_EVENT">Only event cue points (FE2_PNGEX_CP_EVENT)</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_COMPRESSION_LEVEL'></td>
   </tr>

   <tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>

   <tr>
    <th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></td>
   </tr>

   <tr>
    <th><abbr title="start time in ms; Default: 0">FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_START_TIME'></td>
   </tr>

   <tr>
    <th><abbr title="stop time in ms; Default: <clip length>">FE2_PNGEX_AUTO_EXPORT_END_TIME</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_END_TIME'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD'></td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - PNGEX filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_ROTATE' value='vfilter:'
          onchange="toggle_ftable('filter_rotate',this.checked)">
   FE2_FILTER_ROTATE
  </label>

  <table id='filter_rotate' class='disabled'>
   <tr>
    <th><abbr title="Default: 0. valid: {0,90,180,270}">FE2_ROTATE_ANGLE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_ROTATE_ANGLE'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - ROTATE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_SCALE' value='vfilter:'
          onchange="toggle_ftable('filter_scale',this.checked)">
   FE2_FILTER_SCALE
  </label>
```

```
  <table id='filter_scale' class='disabled'>
   <tr>
    <th><abbr title="Default: input image width">FE2_SCALE_WIDTH</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_SCALE_WIDTH'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image height">FE2_SCALE_HEIGHT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_SCALE_HEIGHT'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - SCALE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_SHARPEN' value='vfilter:'
          onchange="toggle_ftable('filter_sharpen',this.checked)">
   FE2_FILTER_SHARPEN
  </label>

  <table id='filter_sharpen' class='disabled'>
   <tr>
    <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
   </tr>
  </table>

 </td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_HIGHPASS' value='afilter:'
          onchange="toggle_ftable('filter_highpass',this.checked)">
   FE2_FILTER_HIGHPASS
  </label>

  <table id='filter_highpass' class='disabled'>
   <tr>
    <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_HIGHPASS_Q'></td>
   </tr>

   <tr>
    <th>FE2_HIGHPASS_CUTOFF</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_HIGHPASS_CUTOFF'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
```

```
          name='FE2_FILTER_LOWPASS' value='afilter:'
          onchange="toggle_ftable('filter_lowpass',this.checked)">
   FE2_FILTER_LOWPASS
  </label>

  <table id='filter_lowpass' class='disabled'>
   <tr>
    <th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_LOWPASS_Q'></td>
   </tr>

   <tr>
    <th>FE2_LOWPASS_CUTOFF</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_LOWPASS_CUTOFF'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_RESAMPLE' value='afilter:'
          onchange="toggle_ftable('filter_resample',this.checked)">
   FE2_FILTER_RESAMPLE
  </label>

  <table id='filter_resample' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_RATE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_RESAMPLE_RATE'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_CHANNELS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_RESAMPLE_CHANNELS'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - RESAMPLE filter parameters -->

</table>
</fieldset>

<!-- ##MUXERS########################################################################## -->
<hr>
<fieldset class='collapsed' id="muxers">
<legend onclick='toggle_expand(this)'>Muxers</legend>
<table>

<tr>
 <td>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3GP'
         onfocus="set_muxer_visible('tgptable')">
  FE2_MUXER_3GP 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3G2'
         onfocus="set_muxer_visible('tg2table')">
  FE2_MUXER_3G2 
 </label>
```

```
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MOV'
         onfocus="set_muxer_visible('movtable')">
  FE2_MUXER_MOV 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MP4'
         onfocus="set_muxer_visible('mp4table')">
  FE2_MUXER_MP4 
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FLV'
         onfocus="set_muxer_visible('flvtable',true)">
  FE2_MUXER_FLV 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_SWF'
         onfocus="set_muxer_visible('swftable',true)">
  FE2_MUXER_SWF
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
         onfocus="set_muxer_visible('fxmtable',true)">
  FE2_MUXER_FXM 
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
         onfocus="set_muxer_visible('webmtable',true)">
  FE2_MUXER_WEBM 
 </label>
 </td>
</tr>

<tr>
 <td>
 <!-- 3GP muxer parameters -->
 <table id='tgptable' class='hidden'>
   <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
     <select name="Flix2_MuxerSetParam:FE2_3GP_FASTSTART">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
 </table>
 <!-- END - 3GP muxer parameters -->

 <!-- 3G2 muxer parameters -->
 <table id='tg2table' class='hidden'>
   <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
     <select name="Flix2_MuxerSetParam:FE2_3G2_FASTSTART">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
```

```
  </tr>
</table>
<!-- END - 3G2 muxer parameters -->

<!-- FLV muxer parameters -->
<table id='flvtable' class='hidden'>
 <tr>
  <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'">
      FE2_FLV_CUEPT_EVENT</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
 </tr>

 <tr>
     <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'">
      FE2_FLV_CUEPT_NAV</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_NAV'></td>
 </tr>

 <tr>
  <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'">
      FE2_FLV_CUEPT_PARAM</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
 </tr>

 <tr>
   <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
             FE2_FLV_METADATA_ENABLE</abbr></th>
   <td>
     <table id='flv_metadata_enable' class=''>
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Ena
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' name='Flix
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Ena
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' name='Flix
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' name=
```

```
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr title="Default: Enable
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' name='Flix2_M
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr title="Default: Enabl
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' name='Flix2_
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' name='
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAM
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION'><abbr title="
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Di
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' name='Fli
      </tr>
     </table>
   </td>
 </tr>

 <tr>
   <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
            FE2_FLV_METADATA_DISABLE</abbr></th>
   <td>
     <table id='flv_metadata_disable' class=''>
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' name='F
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default:
```

```
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' name='F
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Defau
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE' nam
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Defau
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE' nam
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabl
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' name='Flix2_
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enab
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' name='Flix2
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' name='Fl
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTA
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION'><abbr title=
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELOCATIO
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES' name='Fl
     </tr>
    </table>
   </td>
  </tr>
 </table>
 <!-- END - FLV muxer parameters -->
```

```
<!-- FXM muxer parameters -->
<table id='fxmtable' class='hidden'>
 <tr>
  <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'">
      FE2_FXM_CUEPT_EVENT</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_EVENT'></td>
 </tr>

 <tr>
     <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'">
      FE2_FXM_CUEPT_NAV</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_NAV'></td>
 </tr>

 <tr>
  <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'">
      FE2_FXM_CUEPT_PARAM</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_PARAM'></td>
 </tr>

 <tr>
   <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
             FE2_FXM_METADATA_ENABLE</abbr></th>
   <td>
     <table id='fxm_metadata_enable' class=''>
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Ena
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DURATION' name='Flix
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Ena
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' name='Flix
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' name=
      </tr>

      <tr>
```

```
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr title="Default: Enable
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' name='Flix2_M
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr title="Default: Enabl
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' name='Flix2_
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abbr title="Default: En
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' name='Fli
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default:
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' name='
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title=
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAM
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION'><abbr title="
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Di
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' name='Fli
     </tr>
    </table>
  </td>
</tr>

<tr>
  <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
            FE2_FXM_METADATA_DISABLE</abbr></th>
  <td>
    <table id='fxm_metadata_disable' class=''>
     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abbr title="Default: En
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' name='Fli
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: En
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' name='Fli
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><abbr title="Default:
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' name='F
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default:
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' name='F
     </tr>
```

```
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabl
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' name='Flix2_
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enab
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' name='Flix2
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTA
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATIO
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: D
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' name='Fl
      </tr>
     </table>
   </td>
 </tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>
```

```
  <tr>
   <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
   <td>
    <select name="Flix2_MuxerSetParam:FE2_MOV_FASTSTART">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
   </td>
  </tr>
</table>
<!-- END - MOV muxer parameters -->

<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
  <tr>
   <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
   <td>
    <select name="Flix2_MuxerSetParam:FE2_MP4_FASTSTART">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
   </td>
  </tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
 <tr>
  <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_WIDTH'></td>
 </tr>

 <tr>
  <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_HEIGHT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_FRAMERATE'></td>
 </tr>

 <tr>
  <th>FE2_SWF_LOOP_COUNT</th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_LOOP_COUNT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
 </tr>

 <tr>
  <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
  <td>
   <select name='Flix2_MuxerSetParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>
    <option value=""></option>
    <option value="_self">_self</option>
    <option value="_blank">_blank</option>
    <option value="_parent">_parent</option>
    <option value="_top">_top</option>
   </select>
   </td>
 </tr>
```

```
<tr>
 <th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_EMBEDDED_URL_TYPE'>
   <option value=""></option>
   <option value="EmbeddedUrlIsGetUrl">EmbeddedUrlIsGetUrl</option>
   <option value="EmbeddedUrlIsLoadMovie">EmbeddedUrlIsLoadMovie</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="e.g. n0=v0&amp;n1=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
</tr>

<tr><th><b>Preloader Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_PRELOAD_TYPE'>
   <option value=""></option>
   <option value="SwfPreloaderNone">SwfPreloaderNone</option>
   <option value="SwfFixedPreloader">SwfFixedPreloader</option>
   <option value="SwfAdaptivePreloader">SwfAdaptivePreloader</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTION</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_ON_START_OPTION'>
   <option value=""></option>
   <option value="SwfOnMovieStartAutomatically">SwfOnMovieStartAutomatically</option>
   <option value="SwfOnMovieStartOnClick">SwfOnMovieStartOnClick</option>
   <option value="SwfOnMovieStartWait">SwfOnMovieStartWait</option>
   <option value="SwfOnMovieStartEmbedSTOP">SwfOnMovieStartEmbedSTOP</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_START_BLANK_FRAME'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_START_WAIT_SEC'></td>
</tr>

<tr><th><b>End Settings:</b></th><td></td></tr>
```

```
  <tr>
   <th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
   <td>
    <select name='Flix2_MuxerSetParam:FE2_SWF_ON_END_OPTION'>
     <option value=""></option>
     <option value="SwfOnMovieEndNothing">SwfOnMovieEndNothing</option>
     <option value="SwfOnMovieEndSTOP">SwfOnMovieEndSTOP</option>
     <option value="SwfOnMovieEndLoop">SwfOnMovieEndLoop</option>
     <option value="SwfOnMovieEndUnload">SwfOnMovieEndUnload</option>
     <option value="SwfOnMovieEndLoadMovie">SwfOnMovieEndLoadMovie</option>
    </select>
   </td>
  </tr>

  <tr>
   <th>FE2_SWF_ON_END_URL</th>
   <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_ON_END_URL'></td>
  </tr>
 </table>
 <!-- END - SWF muxer parameters -->

 <!-- WEBM muxer parameters -->
 <table id='webmtable' class='hidden'>
   <tr>
    <th>(FE2_MUXER_WEBM defines no parameters)</th>
   </tr>
 </table>
 <!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM########################################################## -->
<hr>
<p>
 <input type="submit" value="Start Encode">
 <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>
HTML_END;
```

14.2 process_sample.php

```
#!/usr/bin/env php
<?php
##==========================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##--------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##--------------------------------------------------------------------------
##

function print_encoder_status() {
    global $flix;
```

```
    print "<br>Encoder Status<br>";

    $res= Flix2_GetEncoderState($flix);
    print " Flix2_GetEncoderState: $res[1]<br>";

    $res= Flix2_Errno($flix);
    printf(" Flix2_Errno: sc:%d flixerrno:%d syserrno:%d<br>",
        $res[0],$res[1],$res[2]);
}

function process_sc($func, $sc) {
    global $flix;
    echo "<td align=\"center\">$sc</td>";

    if($sc != ON2_OK) {
        $res = Flix2_Errno($flix);
        printf("<td>Flix2_Errno: sc:%d %s:%d syserrno:%d</td>",
            $res[0],($sc==ON2_NET_ERROR)?"rpcerr":"flixerrno",$res[1],$res[2]);
        die("Error calling $func, sc= $sc;".
            "esc= $res[0] Flix2_Errno( $res[1], $res[2] )\n");
    }

    echo "</tr>";
}

function SimpleGet($func) {
    global $flix;
    $res = call_user_func($func, $flix);
    $sc = $res[0];
    if($sc != ON2_OK) {die ($func . ' failed: sc=' . $sc);}
    return $res[1];
}

function SimpleSet($func, $arg) {
    global $flix;

    echo "<tr><td>$func( $arg )</td>";
    if(!strcmp($func,"Flix2_SetInputFile")||!strcmp($func,"Flix2_SetOutputFile")) {
        $sc = call_user_func($func, $flix, $arg);
    } else {
        $sc = call_user_func($func, $flix, constant("$arg"));
    }
    process_sc($func,$sc);
    return $sc;
}

function init_codec($name)
{
    global $flix, $codec, $codecptr;

    ##if name is a codec name, e.g. FE2_CODEC_VP6, add an instance
    ##we'll assume all Flix2_CodecSetParam's relate to this codec until we
    ##hit the next codec name
    if($codecptr) { delete_flix2plgnhandlep($codecptr); }

    echo "<tr><td>Flix2_AddCodec( ".$name." )</td>";
    $codecptr = new_flix2plgnhandlep();
    $sc = Flix2_AddCodec($codecptr, $flix, constant("$name"));
    process_sc('Flix2_AddCodec',$sc);

    #retrieve the codec handle to be used in all Codec API function calls
    $codec = flix2plgnhandlep_value($codecptr);
}

function codec_interface($funcname, $name, $value)
{
```

```
    global $codec;
    echo "<tr><td>$funcname( $name, $value )</td>";

    $sc = call_user_func($funcname,$codec,constant("$name"),
        (preg_match('/AsStr/',$funcname)||!preg_match('/[[:alpha:]_]/',$value))?$value:constant("$value"))
    process_sc($funcname,$sc);
}

function init_filter($name)
{
    global $flix, $filter, $filterptr;

    ##if name is a filter name, e.g. FE2_FILTER_CUT, add an instance
    ##we'll assume all Flix2_FilterSetParam's relate to this filter until we
    ##hit the next filter name
    if($filterptr) { delete_flix2plgnhandlep($filterptr); }

    echo "<tr><td>Flix2_AddFilter( $name )</td>";
    $filterptr = new_flix2plgnhandlep();
    $sc = Flix2_AddFilter($filterptr, $flix, constant("$name"));
    process_sc('Flix2_AddFilter',$sc);

    #retrieve the filter handle to be used in all Filter API function calls
    $filter = flix2plgnhandlep_value($filterptr);
}

function filter_interface($funcname, $name, $value)
{
    global $filter;
    echo "<tr><td>$funcname( $name, $value )</td>";

    $sc = call_user_func($funcname,$filter,constant("$name"),
        (preg_match('/AsStr/',$funcname)||!preg_match('/[[:alpha:]_]/',$value))?$value:constant("$value"))
    process_sc($funcname,$sc);
}

function init_muxer($name)
{
    global $flix, $muxer, $muxerptr;

    ##if name is a muxer name, e.g. FE2_MUXER_FLV, add an instance
    ##we'll assume all Flix2_MuxerSetParam's relate to this muxer until we
    ##hit the next muxer name
    if($muxerptr) { delete_flix2plgnhandlep($muxerptr); }

    echo "<tr><td>Flix2_AddMuxer( ".$name." )</td>";
    $muxerptr = new_flix2plgnhandlep();
    $sc = Flix2_AddMuxer($muxerptr, $flix, constant("$name"));
    process_sc('Flix2_AddMuxer',$sc);

    #retrieve the muxer handle to be used in all Muxer API function calls
    $muxer = flix2plgnhandlep_value($muxerptr);
}

function muxer_interface($funcname, $name, $value)
{
    global $muxer;
    echo "<tr><td>$funcname( $name, $value )</td>";

    $sc = call_user_func($funcname,$muxer,constant("$name"),
        (preg_match('/AsStr/',$funcname)||!preg_match('/[[:alpha:]_]/',$value))?$value:constant("$value"))
    process_sc($funcname,$sc);
}

function flush_output()
{
    flush(); if(ini_get("output_buffering")) ob_flush();
```

```
}

echo "\r\n\r\n<html><head><title>Flix CGI Process Sample - PHP</title></head><body>\n";
# output the function calls
echo "<hr><p>process_sample.php version 1.8<br>\n";

# Load the FlixEngine module
# add extension_dir/.. to 'include_path' as this is often missing from php.ini
set_include_path(get_include_path().PATH_SEPARATOR.ini_get("extension_dir")."/..");
$flixphp = 'flixengine2.php';
echo 'Loading flix: ' . $flixphp . "\n<br>";
include ($flixphp);

echo "Flix Engine client library v" . Flix2_Version() . "<br>";
echo str_replace("\n","<br>",Flix2_Copyright()) . "</p>";

$rpchost = "localhost";
$prefix  = "/var/www/cgi-bin/";
$outdir  = $prefix."flixmedia/out/";

##verify outdir's presence and accessibility
if ($rpchost == "localhost" && !(is_dir($outdir) && is_writeable($outdir))) {
  $t=array("<i>'$outdir'</i>",'<tt>$outdir</tt>'); $file=__FILE__;
  echo "<p>****************************************<br>\n".
       "<b>WARNING</b>: $t[0] MUST exist and be writeable by <i>flixd</i>.\n".
       "<br>Please make $t[0] accessible or modify the $t[1]".
       " value defined in '$file'.<br>\n";
  $g=`grep -m 1 -H -n '^\$outdir' $file`;
  if($g) {
    print "The definition of $t[1] can be found here:<br>\n$g<br>\n";
  }
  echo "****************************************</p>\n";
}

echo "<table border='1' cellpadding='5'>";
echo "<caption>Flix Function Calls</caption>\n";
echo "<tr><th>Function Name</th><th>Return Value</th></tr>\n";
echo "<tr><td>Flix2_CreateEx()</td>";
flush_output();

$flixptr= new_flix2handlep();
$sc = Flix2_CreateEx($flixptr, $rpchost, 0);

#extract the handle value returned from _Create. $flix will be used in
#every Flix API call that follows
$flix= flix2handlep_value($flixptr);

process_sc('Flix2_CreateEx',$sc);

foreach($_POST as $func=>$param) {
    if(strlen($param)) {
        if(!strcmp($func,"Flix2_SetOutputFile")) {
            SimpleSet($func, $outdir.$param);
        } else if(preg_match('/^(?:a|v)codec:/',$func)) {
            init_codec($param);
        } else if(preg_match('/^(?:a|v)filter:/',$param)) {
            init_filter($func);
        } else if(preg_match('/^muxer:$/',$func)) {
            init_muxer($param);
        } else if(strstr($func,"CodecSetParam")) {
            $temp = explode(":",$func);
            codec_interface($temp[0],$temp[1],$param);
        } else if(strstr($func, "FilterSetParam")) {
            $temp = explode(":",$func);
            filter_interface($temp[0],$temp[1],$param);
        } else if(strstr($func,"MuxerSetParam")) {
            $temp = explode(":",$func);
```

```
            muxer_interface($temp[0],$temp[1],$param);
        } else  {
            SimpleSet($func,$param);
        }
    }
}

echo "<tr><td>Flix2_Encode()</td>";
$sc = Flix2_Encode($flix);
echo "<td align=\"center\">$sc</td></tr>\n";
echo "</table>\n";
if($sc != ON2_OK) {die ('Flix2_Encode: sc=' . $sc ."\n");}

echo "<br>Encoding...(video frames encoded, percent complete). ".
     "Total frames will reset when doing 2pass.<br>";

do {
    $ier = SimpleGet('Flix2_IsEncoderRunning');
    $res = call_user_func('encoding_status_GetTotalFrames', $flix);
    $p = call_user_func('encoding_status_PercentComplete', $flix);
    if ($res[0] == ON2_OK) {
        echo "($res[1], $p[1]%)<br>\n";
        flush_output();
    }
    sleep(1);
} while(($sc == ON2_OK) && ($ier != on2false));

print "<br>Done!";
print_encoder_status();

$sc = Flix2_Destroy($flix);
if($sc != ON2_OK) {die("Flix2_Destroy: sc= $sc\n");}

if(isset($filterptr)) {delete_flix2plgnhandlep($filterptr);}
if(isset($codecptr)) {delete_flix2plgnhandlep($codecptr);}
if(isset($muxerptr)) {delete_flix2plgnhandlep($muxerptr);}
delete_flix2handlep($flixptr);
print "</body>\n</html>";
?>
```

Chapter 15

Command Line

```
#!/usr/bin/python -u
##==============================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##------------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##------------------------------------------------------------------------------
##
import flixengine2
import time,sys,os

#checks the return value of an API function printing error information on
#failure. usage checksc(funcname,sc)
def checksc(func, sc):
    if (sc != flixengine2.ON2_OK):
        print "%s failed: sc= %d" % (func,sc)

        #if sc == ON2_NET_ERROR Flix2_Errno will return the specific rpc error
        #encountered as flixerrno along with the client lib's errno value
        if (sc == flixengine2.ON2_NET_ERROR): str = "rpcerr";
        else: str = "flixerrno";

        res = flixengine2.Flix2_Errno(flix)
        print " Flix2_Errno: sc:%d %s:%d syserrno:%d\n" %\
            (res[0],str,res[1],res[2])
        raise Exception(func)

def print_encoder_status():
    print "\nEncoder Status"

    res = flixengine2.Flix2_GetEncoderState(flix)
    print " Flix2_GetEncoderState: %d" % res[1]

    res = flixengine2.Flix2_Errno(flix)
    print " Flix2_Errno: sc:%d flixerrno:%d syserrno:%d\n" %\
        (res[0],res[1],res[2])

print "Flix Engine client library v%s" % flixengine2.Flix2_Version()
print "%s\n" % flixengine2.Flix2_Copyright()

if (len(sys.argv) < 3):
    print "usage: cli_encode.py <infile> <outfile>\n\n"\
          "NOTE cli_encode.py uses libflixengine2.so which is a client\n"\
          "NOTE side rpc library. All paths must be accessible to the\n"\
          "NOTE server side, i.e., flixd, thus relative paths will most\n"\
          "NOTE likely give undesired results. The same can be said\n"\
          "NOTE for clients running on different machines.\n"
    raise Exception()

timeout_s = 0 #rpc timeout in seconds, 0=use default (25s)

flixptr = flixengine2.new_flix2handlep()
sc = flixengine2.Flix2_CreateEx(flixptr, "localhost", timeout_s)

# retrieve the actual flix handle for use in the remaining API calls
flix = flixengine2.flix2handlep_value(flixptr);\
  checksc('flixengine2.Flix2_CreateEx',sc)

print "Input File  : %s" % sys.argv[1]
if(os.path.isabs(sys.argv[1])!=True):
    print "WARNING: path to input file is not absolute"
```

```
sc = flixengine2.Flix2_SetInputFile(flix, sys.argv[1]);\
  checksc('flixengine2.Flix2_SetInputFile',sc)

##input file information
srcw = flixengine2.video_options_GetSourceWidth(flix);\
  checksc('flixengine2.video_options_GetSourceWidth',srcw[0])
srch = flixengine2.video_options_GetSourceHeight(flix);\
  checksc('flixengine2.video_options_GetSourceHeight',srch[0])
srcduration = flixengine2.Flix2_GetSourceDuration(flix);\
  checksc('flixengine2.Flix2_GetSourceDuration)',srcduration[0])
print "              Width:    %d\n" \
      "              Height:   %d\n" \
      "              Duration: %dms" % \
    (srcw[1],srch[1],srcduration[1])

print "Output File : %s" % sys.argv[2]
if(os.path.isabs(sys.argv[2])!=True):
    print "WARNING: path to output file is not absolute"
sc = flixengine2.Flix2_SetOutputFile(flix, sys.argv[2]);\
  checksc('flixengine2.Flix2_SetOutputFile',sc)

##
## Options may be set and codecs/filters/muxers may be added prior to Flix2_Encode()
##

##Add the scale filter
##create a storage location for the filter handle
#filterptr= flixengine2.new_flix2plgnhandlep()
#sc = flixengine2.Flix2_AddFilter(filterptr,
#        flix,flixengine2.FE2_FILTER_SCALE);\
#  checksc('Flix2_AddFilter(FE2_FILTER_SCALE)',sc)
#
##retrieve the value of the handle for use in the remaining filter functions
#filter= flixengine2.flix2plgnhandlep_value(filterptr)
#sc = flixengine2.Flix2_FilterSetParam(filter,
#         flixengine2.FE2_SCALE_WIDTH,240);\
#  checksc('Flix2_FilterSetParam(FE2_SCALE_WIDTH,240)',sc)
#sc = flixengine2.Flix2_FilterSetParam(filter,
#         flixengine2.FE2_SCALE_HEIGHT,160);\
#  checksc('Flix2_FilterSetParam(FE2_SCALE_HEIGHT,160)',sc)
#
##cleanup
#flixengine2.delete_flix2plgnhandlep(filterptr); filterptr= None
#
##Add the vp6 codec. Though it is the default, you must add it in order
##to modify its settings
#codecptr= flixengine2.new_flix2plgnhandlep()
#sc = flixengine2.Flix2_AddCodec(codecptr,flix,flixengine2.FE2_CODEC_VP6);\
#  checksc('Flix2_AddCodec(FE2_CODEC_VP6)',sc)
#
##retrieve the value of the handle for use in the remaining codec functions
#codec= flixengine2.flix2plgnhandlep_value(codecptr)
#
#sc = flixengine2.Flix2_CodecSetParam(codec,
#         flixengine2.FE2_VP6_RC_MODE,flixengine2.VBR_1PASSControl);\
#  checksc('Flix2_CodecSetParam(FE2_VP6_RC_MODE,VBR_1PASSControl)',sc)
#
##cleanup
#flixengine2.delete_flix2plgnhandlep(codecptr); codecptr= None
#
##Use the FLV muxer (default)
#muxerptr= flixengine2.new_flix2plgnhandlep()
#sc = flixengine2.Flix2_AddMuxer(muxerptr,flix,flixengine2.FE2_MUXER_FLV);\
#  checksc('Flix2_AddMuxer(FE2_MUXER_FLV)',sc)
#
##retrieve the value of the handle for use in the remaining muxer functions
#muxer= flixengine2.flix2plgnhandlep_value(muxerptr)
```

```
#
##cleanup
#flixengine2.delete_flix2plgnhandlep(muxerptr); muxerptr= None

sc = flixengine2.Flix2_Encode(flix); checksc('flixengine2.Flix2_Encode',sc)

ier = [0,1]
print
while(ier[1] != 0):
    ier = flixengine2.Flix2_IsEncoderRunning(flix);\
      checksc('flixengine2.Flix2_IsEncoderRunning',ier[0])

    pcnt = flixengine2.encoding_status_PercentComplete(flix);\
      checksc('flixengine2.encoding_status_PercentComplete',pcnt[0])
    print "\rEncoding...%d%%  " % pcnt[1],
    sys.stdout.flush();

    time.sleep(1)

print "Done!"
print_encoder_status()

#cleanup
sc = flixengine2.Flix2_Destroy(flix); checksc('flixengine2.Flix2_Destroy',sc)
flixengine2.delete_flix2handlep(flixptr)
```

Chapter 16

CGI

This example consists of 2 parts: flix2_sample.py and process_sample.py. flix2_sample.py searches for uploaded files to encode and allows the user to select one while giving the option to set values for most of the engine's functions. The engine options are separated into sections that map to the engine's include files.

To use this example navigate to flix2_sample.py in your web browser, select a file from the list, set any of the desired options and click the encode button. The selected options are submitted to process_sample.py which runs the encode loop.

Default file locations used by the scripts:

Input : */var/www/cgi-bin/flixmedia/in* (`$indir` in flix2_sample.py)
Overlay : */var/www/cgi-bin/flixmedia/overlay* (`$overlaydir` in flix2_sample.py)
Output : */var/www/cgi-bin/flixmedia/out* (`$outdir` in process_sample.py)

16.1 flix2_sample.py

```
#!/usr/bin/python
##==============================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##------------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##------------------------------------------------------------------------------
##
import os
import sys
import flixengine2

prefix     = "/var/www/cgi-bin/";
indir      = prefix + "flixmedia/in/"
overlaydir = prefix + "flixmedia/overlay/"

print """

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head>
 <title>Flix CGI Sample - Python</title>
 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

 <script type="text/javascript">
 function showall(show) {
    var legend_list = document.getElementsByTagName('legend');
    var i=0;
    while(legend_list[i]) {
        legend_list[i].parentNode.className=show?'expanded':'collapsed';
        i++;
    }
 }

 function toggle_expand(_this)
 {
     _this.parentNode.className= (_this.parentNode.className=='expanded') ?
                                 'collapsed' : 'expanded';
    document.getElementById('showall').checked=false;
 }
```

```
function set_table_visible(ctable,visible)
{
   var table= document.getElementById(ctable);

   if(table) {
      if(visible && table.className == 'hidden') {
          table.className= '';
      } else if(!visible && table.className == '') {
          table.className= 'hidden';
      }
   }
}

function hide_tablelist(vistablename,tablelist)
{
  var table= document.getElementById(vistablename);

  if(table&&table.className=='hidden') {
    table.className= '';

    var i=0;
    while(tablelist[i]) {
      if(tablelist[i]!=vistablename) {
        var hiddentable= document.getElementById(tablelist[i]);

        set_table_visible(tablelist[i],false);
        /*clear down any values set in the hidden table to avoid posting
          unnecessary values*/
        clear_value(hiddentable.getElementsByTagName('input'));
        clear_value(hiddentable.getElementsByTagName('select'));
      }
      i++;
    }
  }
}

function set_acodec_visible(ctable)
{
  var acodecs= new Array('aactable','aacplustable','amrnbtable','lametable','pcmtable','vorbistable');
  hide_tablelist(ctable,acodecs);
}

function set_vcodec_visible(ctable)
{
  var vcodecs= new Array('h263table','h264table','vp6atable','vp6table','vp8table');
  hide_tablelist(ctable,vcodecs);
}

function set_muxer_visible(mtable)
{
  var muxers= new Array('flvtable','fxmtable','movtable','mp4table','swftable','tg2table','tgptable','web
  hide_tablelist(mtable,muxers);
}

function clear_value(list)
{
   var i=0;
   while(list[i]) {
     if(list[i].type=='checkbox') { list[i++].checked=false; }
     else { list[i++].value=''; }
   }
}

function toggle_ftable(ftable,enabled)
{
  var table= document.getElementById(ftable);
```

```
  if(table) {
    table.className = enabled ? '' : 'disabled';
    if(!enabled) {
      clear_value(table.getElementsByTagName('input'));
      clear_value(table.getElementsByTagName('select'));
    }
  }
}

function reset_tables()
{
   var table_list= document.getElementsByTagName('table');
   var i=0;
   while(table_list[i]) {
     if (table_list[i].id.length > 7 &&
       table_list[i].id.substring(0,7) == 'filter_') {
       table_list[i].className= 'disabled';
     } else if (table_list[i].id.indexOf('table') != -1) {
       table_list[i].className= 'hidden';
     }
     i++;
   }
}

</script>

<style type="text/css">
   <!--
   html {
       font-family: Verdana, 'bitstream vera sans', Arial, sans-serif;
       font-size: 100%;
       color: rgb(56,56,56);
       background-color: rgb(236,236,236);
       border-style: solid;
       border-color: rgb(236,236,236);
   }

   body {
       text-align: center;
       margin: 0 auto;
   }

   div.content {
       color: rgb(56,56,56);
       background-color: rgb(246,246,246);
       text-align: left;
       margin: 0 auto;
       width: 80%;
       min-width: 768px;
       max-width: 932px;
       border-width: 0 1px;
       border-color: rgb(144,144,144);
       border-style: solid;
   }

   div.content:after {
       content: "";
       color: inherit;
       background-color: rgb(250,250,250);
       border-top: 1px solid rgb(144,144,144);
       height: 20px;
       width: 100%;
       display: block;
   }

   fieldset table, fieldset {display: none;}
```

```
fieldset.expanded, fieldset.collapsed {display: block;}

/*first is fallback for IE*/
fieldset.expanded table {display: block;}
fieldset.expanded table {display: table;}

h1 {
    font-family: sans-serif;
    font-size: 150%;
    font-weight: normal;
    text-align: left;
    letter-spacing: -1px;
    color: rgb(74,74,74);
    background-color: inherit;
    margin: 0;
}

a {
    color: rgb(74,74,74);
    background-color: transparent;
}

label {
    font-size: 75%;
}

fieldset {
    font-size: 75%;
    line-height: 130%;
    padding: 0;
    margin: 20px;
    border: none;
}

fieldset.expanded {
    color: inherit;
    background-color: rgb(252,252,252);
    border-style: solid;
    border-width: 1px;
    border-color: rgb(217,217,217) rgb(217,217,217) rgb(188,188,188);
}

legend {
    padding: 0 5px;
    border-left: 10px solid rgb(217,217,217);
    cursor: pointer;
}
legend:hover {text-decoration: underline;}
fieldset.expanded legend {
    font-size: 150%;
    font-weight: bold;
    letter-spacing: -1px;
    background: transparent;
    margin-left: 12px;
    border-right: 10px solid rgb(217,217,217);
    display: block;
}
table {
    font-size: 100%;
    border-spacing: 0;
    /*border-collapse: collapse;*/
    width: 100%;
}

th:before { display: none; }
th, td {
    width: 50%;
```

```
        vertical-align: top;
        padding: 2px 3px;
        border-width: 1px 0;
        border-style: solid;
        border-color: rgb(188,188,188) rgb(252,252,252) rgb(252,252,252);
    }
    tr:first-child>* {border-top-color: rgb(252,252,252);}
    th {
        font-weight: normal;
        text-align: left;
        padding: 2px 2px 2px 5px;
    }
    input[type] {
        font-family: monospace;
        font-size: 100%;
        color: rgb(56,56,56);
        background-color: inherit;
    }
    [type="text"], [type="number"] {
        margin-right: 13px;
        width: 222px;
    }

    [type="checkbox"].filter {
        margin-left: 4px
    }
    [type="checkbox"] {
        margin-left: 13px
    }
    [type="button"], [type="submit"] {
        font-size: 1em;
        margin: 0 2px 0 13px;
    }

    fieldset {
        font-size: 75%;
        margin: 20px 10px;
    }
    fieldset input[type="text"], fieldset input[type="number"] {
        width: 95% !important;
        margin: 0;
        display: block;
    }
    fieldset.expanded table.hidden {
        display: none;
    }

    table.disabled {
        color: rgb(176,176,176);
    }
    -->
 </style>
</head>

<body>
 <div class='content'>
 <noscript>
  <p>This page requires javascript be enabled.</p>
 </noscript>

<hr>
<h1>Flix CGI Sample</h1>



<p><small>flix2_sample.cgi version 1.9</small></p>
<h4>Instructions</h4>
```

```
<ul>
 <li>In this sample you must choose a source file and an output file.<br>
 If you leave all the other options blank then the sample will not call
 the corresponding Flix Engine function and the default will be used.<br>
 When done please press the "Start Encode" button at the bottom of the page.<br>
 <li>Mouse over a function name to see its default, if applicable.
 <li>Current source file directory: """
print indir
print """

</ul>

<p>
 <label><input type="checkbox" id='showall' onclick='showall(this.checked)'>Show all</label>
</p>

<form action="process_sample.py" method="post">

<!-- ##SOURCE FILE############################################################ -->
<hr>
<fieldset class='expanded' id="srcfile">
<legend onclick='toggle_expand(this)'>Source File</legend>
<table>

<tr>
 <td>
"""
if os.path.exists(indir):
    first = True
    files = os.listdir(indir); files.sort()

    print "<select name='Flix2_SetInputFile' size='",
    if len(files) < 15: print str(len(files))+"'>"
    else: print "15'>"

    for j in files:
        if os.path.isdir(os.path.join(indir,j))==False:
            print "<option ",
            if first: print "selected ",; first = False
            print "value=\"%s\">%s</option>" %\
                  (os.path.join(indir,j), j)
    print "</select>";
else:
    print "WARNING couldn't open " + indir + " os.path.exists() failed<br>"

print """

 </td>
</tr>

</table>
</fieldset>

<!-- ##DST FILE################################################################ -->
<hr>
<fieldset class='expanded' id="dstfile">
<legend onclick='toggle_expand(this)'>Output File</legend>
<table>

<tr>
 <td>
  <input type="text" name="Flix2_SetOutputFile" value="cgi-python-out.flv">
 </td>
</tr>

</table>
</fieldset>
```

```
<!-- ##MAIN OPTIONS################################################################ -->
<hr>
<fieldset class='collapsed' id="main_opts">
<legend onclick='toggle_expand(this)'>Main Options</legend>
<table>

<tr>
 <th><abbr title="Default: FALSE">Flix2_SetOverwriteExistingFiles</abbr></th>
 <td>
 <select name="Flix2_SetOverwriteExistingFiles">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: TRUE">Flix2_SetExportAudio</abbr></th>
 <td>
 <select name="Flix2_SetExportAudio">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: TRUE">Flix2_SetExportVideo</abbr></th>
 <td>
 <select name="Flix2_SetExportVideo">
 <option value=""></option>
 <option value="on2true">TRUE</option>
 <option value="on2false">FALSE</option>
 </select>
 </td>
</tr>

</table>
</fieldset>

<!-- ##CODECS########################################################################## -->
<hr>
<fieldset class='collapsed' id="codecs">
<legend onclick='toggle_expand(this)'>Codecs</legend>
<table>

<tr><th><b>Video Codecs</b></th></tr>
<tr>
 <td>
 <label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_VP8'
         onfocus="set_vcodec_visible('vp8table')">
  <abbr title="For use with WebM">FE2_CODEC_VP8</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_VP6'
         onfocus="set_vcodec_visible('vp6table')">
  <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_VP6</abbr> 
 </label>
 <label>
  <input type='radio' name='vcodec:' value='FE2_CODEC_VP6ALPHA'
         onfocus="set_vcodec_visible('vp6atable')">
```

```
 <abbr title="For use with FLV/SWF">FE2_CODEC_VP6ALPHA</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H263'
        onfocus="set_vcodec_visible('h263table')">
 <abbr title="For use with FLV/SWF">FE2_CODEC_H263</abbr> 
</label>
<br>

<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H263_BASELINE'
        onfocus="set_vcodec_visible('h263table')">
 <abbr title="For use with 3GP">FE2_CODEC_H263_BASELINE</abbr> 
</label>
<label>
 <input type='radio' name='vcodec:' value='FE2_CODEC_H264'
        onfocus="set_vcodec_visible('h264table')">
 <abbr title="For use with 3GP/3G2/MOV/MP4">FE2_CODEC_H264</abbr> 
</label>

<!-- VP6 codec parameters -->
<table id='vp6table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_VP6_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP6_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP6_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
   <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6_CXMODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6_CXMODE">
   <option value=""></option>
   <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
   <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
  </select>
  </td>
 </tr>
```

```
<tr>
 <th><abbr title="Default: VP6_E">FE2_VP6_PROFILE</abbr></th>
 <td>
  <select name="Flix2_CodecSetParam:FE2_VP6_PROFILE">
  <option value=""></option>
  <option value="VP6_E">VP6_E</option>
  <option value="VP6_S">VP6_S</option>
 </select>
 </td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP6_CONCURRENCY</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_CONCURRENCY'></td>
</tr>

<tr>
 <th><abbr title="Default: 90">FE2_VP6_UNDERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_UNDERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEMPORAL_RESAMPLING</abbr></
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6_TEMPORAL_DOWN_WATERMARK</abb
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
 <th><abbr title="Default: 100">FE2_VP6_STREAM_PEAK_BITRATE</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
 <th><abbr title="Default: 6 (CBR only)">FE2_VP6_STREAM_PREBUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_PREBUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 10 (CBR only)">FE2_VP6_STREAM_OPTIMAL_BUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_OPTIMAL_BUFFER'></td>
</tr>
```

```
 <tr>
  <th><abbr title="(CBR only)">FE2_VP6_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_STREAM_MAX_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 40">FE2_VP6_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_2PASS_MIN_SECTION'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 400">FE2_VP6_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6_2PASS_MAX_SECTION'></td>
 </tr>

</table>
<!-- END - VP6 codec parameters -->

<!-- VP6A codec parameters -->
<table id='vp6atable' class='hidden'>
 <tr>
  <th><abbr title="Default: 380kbps">FE2_VP6A_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 68kbps (15% of default 448kbps)">FE2_VP6A_ALPHA_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP6A_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6A_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP6A_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP6A_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6A_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
   <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP6A_CXMODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP6A_CXMODE">
   <option value=""></option>
   <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
   <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
```

```
 </select>
 </td>
</tr>

<tr><th><b>Advanced Settings:</b></th></tr>

<tr>
 <th><abbr title="Default: 90">FE2_VP6A_UNDERSHOOT_PCT</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_UNDERSHOOT_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_MIN_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_MIN_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_ALPHA_NOISE_REDUCTION</abbr
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_ALPHA_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_RESAMPLING</abbr><
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_TEMPORAL_RESAMPLING'></td>
</tr>

<tr>
 <th><abbr title="Default: bits per pixel dependent (see API docs)">FE2_VP6A_TEMPORAL_DOWN_WATERMARK</ab
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_TEMPORAL_DOWN_WATERMARK'></td>
</tr>

<tr>
 <th><abbr title="Default: 100">FE2_VP6A_STREAM_PEAK_BITRATE</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_PEAK_BITRATE'></td>
</tr>

<tr>
```

```
  <th><abbr title="Default: 6 (CBR only)">FE2_VP6A_STREAM_PREBUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_PREBUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 10 (CBR only)">FE2_VP6A_STREAM_OPTIMAL_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_OPTIMAL_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="(CBR only)">FE2_VP6A_STREAM_MAX_BUFFER</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_STREAM_MAX_BUFFER'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 40">FE2_VP6A_2PASS_MIN_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_2PASS_MIN_SECTION'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 400">FE2_VP6A_2PASS_MAX_SECTION</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP6A_2PASS_MAX_SECTION'></td>
 </tr>

</table>
<!-- END - VP6A codec parameters -->

<!-- H263 codec parameters -->
<table id='h263table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_H263_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_H263_KFINTTYPE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H263_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_H263_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_H263_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H263_RC_MODE">
    <option value=""></option>
    <option value="VBR_2PASSControl">VBR_2PASSControl</option>
    <option value="CBR_2PASSControl">CBR_2PASSControl</option>
    <option value="VBR_1PASSControl">VBR_1PASSControl</option>
    <option value="CBR_1PASSControl">CBR_1PASSControl</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 31">FE2_H263_MAX_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_MAX_Q'></td>
 </tr>
```

```
 <tr>
  <th><abbr title="Default: 2">FE2_H263_MIN_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H263_MIN_Q'></td>
 </tr>

</table>
<!-- END - H263 codec parameters -->

<!-- H264 codec parameters -->
<table id='h264table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_H264_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_H264_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_1PASSControl">FE2_H264_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H264_RC_MODE">
    <option value=""></option>
    <option value="VBR_1PASSControl">VBR_1PASSControl</option>
    <option value="CBR_1PASSControl">CBR_1PASSControl</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: MAIN_H264PROFILE">FE2_H264_PROFILE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_H264_PROFILE">
    <option value=""></option>
    <option value="BASE_H264PROFILE">BASE_H264PROFILE</option>
    <option value="MAIN_H264PROFILE">MAIN_H264PROFILE</option>
    <option value="HIGH_H264PROFILE">HIGH_H264PROFILE</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 0">FE2_H264_B_FRAME_RATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_B_FRAME_RATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: Dependent on profile selection, see API docs. Valid Range [0,5]">FE2_H264_SPE
  <td><input type='text' name='Flix2_CodecSetParam:FE2_H264_SPEED'></td>
 </tr>

</table>
<!-- END - H264 codec parameters -->

<!-- VP8 codec parameters -->
<table id='vp8table' class='hidden'>
 <tr>
  <th><abbr title="Default: 448kbps">FE2_VP8_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: MAX_KEYFRAMES">FE2_VP8_KFINTTYPE</abbr></th>
  <td>
```

```
   <select name="Flix2_CodecSetParam:FE2_VP8_KFINTTYPE">
   <option value=""></option>
   <option value="MAX_KEYFRAMES">MAX_KEYFRAMES</option>
   <option value="FIXED_KEYFRAMES">FIXED_KEYFRAMES</option>
   </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 12*fps or 360 frames if the framerate is unknown">FE2_VP8_KFFREQ</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_KFFREQ'></td>
 </tr>

 <tr>
  <th><abbr title="Default: VBR_2PASSControl">FE2_VP8_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP8_RC_MODE">
   <option value=""></option>
   <option value="VBR_2PASSControl">VBR_2PASSControl</option>
   <option value="CBR_2PASSControl">CBR_2PASSControl</option>
   <option value="VBR_1PASSControl">VBR_1PASSControl</option>
   <option value="CBR_1PASSControl">CBR_1PASSControl</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: COMPRESSMODE_GOOD">FE2_VP8_CXMODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_VP8_CXMODE">
   <option value=""></option>
   <option value="COMPRESSMODE_GOOD">COMPRESSMODE_GOOD</option>
   <option value="COMPRESSMODE_BEST">COMPRESSMODE_BEST</option>
  </select>
  </td>
 </tr>

 <tr>
  <th><abbr title="Default: 0">FE2_VP8_THREADS</abbr></th>
  <td>input type='text' name='Flix2_CodecSetParam:FE2_VP8_THREADS'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 0">FE2_VP8_PROFILE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_PROFILE'></td>
 </tr>

 <tr><th><b>Advanced Settings:</b></th></tr>

 <tr>
  <th><abbr title="Default: 0">FE2_VP8_LAG</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_LAG'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 95">FE2_VP8_UNDERSHOOT_PCT</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_UNDERSHOOT_PCT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 200">FE2_VP8_OVERSHOOT_PCT</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_OVERSHOOT_PCT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 4">FE2_VP8_MIN_Q</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MIN_Q'></td>
```

```
</tr>

<tr>
 <th><abbr title="Default: 63">FE2_VP8_MAX_Q</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MAX_Q'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_SHARPNESS</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_SHARPNESS'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_NOISE_REDUCTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_NOISE_REDUCTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_DROP_THRESH</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_DROP_THRESH'></td>
</tr>

<tr>
 <th><abbr title="Default: 4 (CBR only)">FE2_VP8_STREAM_PREBUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_PREBUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 5 (CBR only)">FE2_VP8_STREAM_OPTIMAL_BUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_OPTIMAL_BUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 6 (CBR only)">FE2_VP8_STREAM_MAX_BUFFER</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_STREAM_MAX_BUFFER'></td>
</tr>

<tr>
 <th><abbr title="Default: 40">FE2_VP8_2PASS_MIN_SECTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_2PASS_MIN_SECTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 400">FE2_VP8_2PASS_MAX_SECTION</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_2PASS_MAX_SECTION'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_VP8_ALTREF</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_ALTREF'></td>
</tr>

<tr>
 <th><abbr title="">FE2_VP8_AR_MAX_FRAMES</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_MAX_FRAMES'></td>
</tr>

<tr>
 <th><abbr title="">FE2_VP8_AR_TYPE</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_TYPE'></td>
</tr>

<tr>
 <th><abbr title="">FE2_VP8_AR_STRENGTH</abbr></th>
 <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_AR_STRENGTH'></td>
</tr>
```

```
  <tr>
   <th><abbr title="Default: 0">FE2_VP8_MB_STATIC_THRESHOLD</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_MB_STATIC_THRESHOLD'></td>
  </tr>

  <tr>
   <th><abbr title="Default: 1">FE2_VP8_TOKEN_PARTITIONS</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_VP8_TOKEN_PARTITIONS'></td>
  </tr>

 </table>
 <!-- END - VP8 codec parameters -->

 </td>
</tr> <!-- END - video codecs -->

<tr><th><b>Audio Codecs</b></th></tr>
<tr>
 <td>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_VORBIS'
         onfocus="set_acodec_visible('vorbistable')">
  <abbr title="For use with WebM">FE2_CODEC_VORBIS</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AAC'
         onfocus="set_acodec_visible('aactable')">
  <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AAC</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AACPLUS'
         onfocus="set_acodec_visible('aacplustable')">
  <abbr title="For use with FLV & 3GP/3G2/MOV/MP4">FE2_CODEC_AACPLUS</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_LAME'
         onfocus="set_acodec_visible('lametable') ">
  <abbr title="For use with FLV/FXM/SWF">FE2_CODEC_LAME</abbr> 
 </label>
 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_PCM'
         onfocus="set_acodec_visible('pcmtable')">
  <abbr title="For use with FLV/SWF">FE2_CODEC_PCM</abbr> 
 </label>
 <br>

 <label>
  <input type='radio' name='acodec:' value='FE2_CODEC_AMR_NB'
         onfocus="set_acodec_visible('amrnbtable')">
  <abbr title="For use with 3GP">FE2_CODEC_AMR_NB</abbr> 
 </label>

 <!-- AMR_NB codec parameters -->
 <table id='amrnbtable' class='hidden'>
  <tr>
   <th><abbr title="Default: 12.2kbps">FE2_AMR_BITRATE</abbr></th>
   <td><input type='text' name='Flix2_CodecSetParam:FE2_AMR_BITRATE'></td>
  </tr>

 </table>
 <!-- END - AMR_NB codec parameters -->

 <!-- AAC codec parameters -->
 <table id='aactable' class='hidden'>
  <tr>
```

```
  <th><abbr title="Default: 64kbps">FE2_AAC_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_AAC_BITRATE'></td>
 </tr>

</table>
<!-- END - AAC codec parameters -->

<!-- AACPLUS codec parameters -->
<table id='aacplustable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_AACPLUS_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_AACPLUS_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: disabled (aacPlus v1)">FE2_AACPLUS_PARAMETRIC_STEREO</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_AACPLUS_PARAMETRIC_STEREO">
   <option value=""></option>
   <option value="0">disable (aacPlus v1)</option>
   <option value="1">enable (aacPlus v2)</option>
   </select>
  </td>
 </tr>
</table>
<!-- END - AACPLUS codec parameters -->

<!-- LAME codec parameters -->
<table id='lametable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_LAME_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_LAME_BITRATE'></td>
 </tr>

 <tr>
  <th><abbr title="Default: 5">FE2_LAME_QUALITY</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_LAME_QUALITY'></td>
 </tr>

 <tr>
  <th><abbr title="Default: LAME_CBR">FE2_LAME_RC_MODE</abbr></th>
  <td>
   <select name="Flix2_CodecSetParam:FE2_LAME_RC_MODE">
   <option value=""></option>
   <option value="LAME_CBR">LAME_CBR</option>
   <option value="LAME_ABR">LAME_ABR</option>
   <option value="LAME_VBR_rh">LAME_VBR_rh</option>
   <option value="LAME_VBR_mtrh">LAME_VBR_mtrh</option>
   </select>
  </td>
 </tr>
</table>
<!-- END - LAME codec parameters -->

<!-- PCM codec parameters -->
<table id='pcmtable' class='hidden'>
 <tr>
  <th>(FE2_CODEC_PCM defines no parameters)</th>
 </tr>
</table>
<!-- END - PCM codec parameters -->

<!-- VORBIS codec parameters -->
<table id='vorbistable' class='hidden'>
 <tr>
  <th><abbr title="Default: 64kbps">FE2_VORBIS_BITRATE</abbr></th>
  <td><input type='text' name='Flix2_CodecSetParam:FE2_VORBIS_BITRATE'></td>
```

```
  </tr>
 </table>
 <!-- END - VORBIS codec parameters -->

 </td>
</tr> <!-- END - audio codecs -->

</table>

</fieldset>

<!-- ##FILTERS##################################################################### -->
<hr>
<fieldset class='collapsed' id="filters">
<legend onclick='toggle_expand(this)'>Filters</legend>
<table>

<tr><th><b>A/V Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_CUT' value='avfilter:'
          onchange="toggle_ftable('filter_cut',this.checked)">
   FE2_FILTER_CUT
  </label>

  <table id='filter_cut' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_CUT_START_SEC</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CUT_START_SEC'></td>
   </tr>

   <tr>
    <th><abbr title="Default: -1">FE2_CUT_STOP_SEC</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CUT_STOP_SEC'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 1">FE2_CUT_USE_SEEK</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_CUT_USE_SEEK">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - CUT filter parameters -->

<tr><th><b>Video Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_ADAPTIVE_DEINTERLACE' value='vfilter:'
          onchange="toggle_ftable('filter_adaptive_deinterlace',this.checked)">
   FE2_FILTER_ADAPTIVE_DEINTERLACE
  </label>
```

```
  <table id='filter_adaptive_deinterlace' class='disabled'>
   <tr>
    <th><abbr title="Default: DEINTERLACE_NONE">FE2_ADAPTIVE_DEINTERLACE_MODE</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_ADAPTIVE_DEINTERLACE_MODE">
     <option value=""></option>
     <option value="DEINTERLACE_NONE">DEINTERLACE_NONE</option>
     <option value="DEINTERLACE_1_2_1_BLUR">DEINTERLACE_1_2_1_BLUR</option>
     <option value="DEINTERLACE_DROP_FIELD">DEINTERLACE_DROP_FIELD</option>
     <option value="DEINTERLACE_ADAPTIVE">DEINTERLACE_ADAPTIVE</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - ADAPTIVE DEINTERLACE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_BCHS' value='vfilter:'
          onchange="toggle_ftable('filter_bchs',this.checked)">
   FE2_FILTER_BCHS
  </label>

  <table id='filter_bchs' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_BRIGHTNESS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_BRIGHTNESS'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_CONTRAST</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_CONTRAST'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_HUE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_HUE'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_BCHS_SATURATION</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_BCHS_SATURATION'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - BCHS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_BLUR' value='vfilter:'
          onchange="toggle_ftable('filter_blur',this.checked)">
   FE2_FILTER_BLUR
  </label>

  <table id='filter_blur' class='disabled'>
   <tr>
    <th><abbr title="Default: BLUR_GAUSS">FE2_BLUR_FILTER</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_BLUR_FILTER">
     <option value=""></option>
     <option value="BLUR_LOWPASS">BLUR_LOWPASS</option>
```

```
     <option value="BLUR_GAUSS">BLUR_GAUSS</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: MASK_3x3">FE2_BLUR_MASKSIZE</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_BLUR_MASKSIZE">
     <option value=""></option>
     <option value="MASK_3x3">MASK_3x3</option>
     <option value="MASK_5x5">MASK_5x5</option>
     </select>
    </td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - BLUR filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_CROP' value='vfilter:'
          onchange="toggle_ftable('filter_crop',this.checked)">
   FE2_FILTER_CROP
  </label>

  <table id='filter_crop' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_CROP_TOP</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_TOP'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image height">FE2_CROP_BOTTOM</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_BOTTOM'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_CROP_LEFT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_LEFT'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image width">FE2_CROP_RIGHT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_CROP_RIGHT'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - CROP filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_DENOISE' value='vfilter:'
          onchange="toggle_ftable('filter_denoise',this.checked)">
   FE2_FILTER_DENOISE
  </label>

  <table id='filter_denoise' class='disabled'>
   <tr>
```

```
    <th><abbr title="Default: 0. Range: [0.0,1.0)">FE2_DENOISE_NOISE_LEVEL</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_DENOISE_NOISE_LEVEL'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - DENOISE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_FRAMERATE' value='vfilter:'
          onchange="toggle_ftable('filter_framerate',this.checked)">
   FE2_FILTER_FRAMERATE
  </label>

  <table id='filter_framerate' class='disabled'>
   <tr>
    <th><abbr title="decimation interval, range: [1,] Default: disabled">FE2_FRAMERATE_DECIMATE</abbr></th
    <td><input type='text' name='Flix2_FilterSetParam:FE2_FRAMERATE_DECIMATE'></td>
   </tr>

   <tr>
    <th><abbr title="explicit frame rate, range: (0.0,] Default: disabled">FE2_FRAMERATE_FPS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_FRAMERATE_FPS'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - FRAMERATE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_MIRROR' value='vfilter:'
          onchange="toggle_ftable('filter_mirror',this.checked)">
   FE2_FILTER_MIRROR
  </label>

  <table id='filter_mirror' class='disabled'>
   <tr>
    <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_HORIZONTAL</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_MIRROR_HORIZONTAL">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0 (disabled)">FE2_MIRROR_VERTICAL</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_MIRROR_VERTICAL">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
  </table>
```

```
 </td>
</tr>
<!-- END - MIRROR filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_OVERLAY' value='vfilter:'
          onchange="toggle_ftable('filter_overlay',this.checked)">
   FE2_FILTER_OVERLAY
  </label>

  <table id='filter_overlay' class='disabled'>
   <tr>
    <th>
     <abbr title="Currently searching """
print overlaydir
print """
 for overlay images">FE2_OVERLAY_FILE</abbr>
    </th>
    <td>
"""
if os.path.exists(overlaydir):
    print """<select name="Flix2_FilterSetParamAsStr:FE2_OVERLAY_FILE">"""
    print "<option value=\"\"></option>\n"
    files = os.listdir(overlaydir); files.sort()
    for j in files:
        if os.path.isdir(os.path.join(overlaydir,j)):
            print "<option value=\"%s\">%s</option>" %\
                  (os.path.join(overlaydir,j), j)
    print "</select>"
    print """<br>default: None, must be set to the absolute path of the overlay
              input file, e.g. '/path/to/my/overlay.png'"""
else:
    print "WARNING couldn't open " + overlaydir + " os.path.exists() failed<br>"

print """

    </td>
   </tr>

   <tr>
    <th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_XY</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_MASK_XY">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_X</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_X'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_Y</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_Y'></td>
   </tr>

   <tr>
    <th><abbr title="Default: FALSE">FE2_OVERLAY_MASK_RGB</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_MASK_RGB">
     <option value=""></option>
```

```
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_R</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_R'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_G</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_G'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_MASK_B</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_MASK_B'></td>
   </tr>

   <tr>
    <th><abbr title="Default: TOP LEFT">FE2_OVERLAY_POS</abbr></th>
    <td>
     <select name="Flix2_FilterSetParam:FE2_OVERLAY_POS">
      <option value=""></option>
      <option value="FE2_OVERLAY_POS_MODE_TOPLEFT">FE2_OVERLAY_POS_MODE_TOPLEFT</option>
      <option value="FE2_OVERLAY_POS_MODE_BOTLEFT">FE2_OVERLAY_POS_MODE_BOTLEFT</option>
      <option value="FE2_OVERLAY_POS_MODE_CENTER">FE2_OVERLAY_POS_MODE_CENTER</option>
      <option value="FE2_OVERLAY_POS_MODE_TOPRIGHT">FE2_OVERLAY_POS_MODE_TOPRIGHT</option>
      <option value="FE2_OVERLAY_POS_MODE_BOTRIGHT">FE2_OVERLAY_POS_MODE_BOTRIGHT</option>
      <option value="FE2_OVERLAY_POS_MODE_XY">FE2_OVERLAY_POS_MODE_XY</option>
     </select>
    </td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_POS_X</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_POS_X'></td>
   </tr>
   <tr>
    <th><abbr title="Default: 0">FE2_OVERLAY_POS_Y</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_OVERLAY_POS_Y'></td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - OVERLAY filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_PNGEX' value='vfilter:'
          onchange="toggle_ftable('filter_pngex',this.checked)">
   FE2_FILTER_PNGEX
  </label>

  <table id='filter_pngex' class='disabled'>
   <tr>
    <th><abbr title="Default: output file directory">FE2_PNGEX_OUTPUT_DIRECTORY</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_OUTPUT_DIRECTORY'></td>
   </tr>

   <tr>
    <th><abbr title="Default: none">FE2_PNGEX_FILENAME_PREFIX</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_FILENAME_PREFIX'></td>
```

```
</tr>

<tr>
 <th><abbr title="Default: none">FE2_PNGEX_FILENAME_SUFFIX</abbr></th>
 <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_FILENAME_SUFFIX'></td>
</tr>

<tr>
 <th><abbr title="Default: input width">FE2_PNGEX_WIDTH</abbr></th>
 <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_WIDTH'></td>
</tr>

<tr>
 <th><abbr title="Default: input height">FE2_PNGEX_HEIGHT</abbr></th>
 <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_HEIGHT'></td>
</tr>

<tr>
 <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_FIRST_FRAME_PNG</abbr></th>
 <td>
  <select name="Flix2_FilterSetParam:FE2_PNGEX_EXPORT_FIRST_FRAME_PNG">
  <option value=""></option>
  <option value="on2false">FALSE</option>
  <option value="on2true">TRUE</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: FALSE">FE2_PNGEX_ENABLE_ALPHA</abbr></th>
 <td>
  <select name="Flix2_FilterSetParam:FE2_PNGEX_ENABLE_ALPHA">
  <option value=""></option>
  <option value="on2false">FALSE</option>
  <option value="on2true">TRUE</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="interval in ms; Default: disabled">FE2_PNGEX_EXPORT_INTERVAL</abbr></th>
 <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_EXPORT_INTERVAL'></td>
</tr>

<tr>
 <th><abbr title="comma delimited, e.g. t0,t1,t2,...tn">FE2_PNGEX_EXPORT_TIME_STRING</abbr></th>
 <td><input type='text' name='Flix2_FilterSetParamAsStr:FE2_PNGEX_EXPORT_TIME_STRING'></td>
</tr>

<tr>
 <th><abbr title="Default: FALSE">FE2_PNGEX_EXPORT_CUE_POINTS</abbr></th>
 <td>
  <select name="Flix2_FilterSetParam:FE2_PNGEX_EXPORT_CUE_POINTS">
  <option value=""></option>
  <option value="FE2_PNGEX_CP_ALL">All cue points (FE2_PNGEX_CP_ALL)</option>
  <option value="FE2_PNGEX_CP_NAV">Only navigation cue points (FE2_PNGEX_CP_NAV)</option>
  <option value="FE2_PNGEX_CP_EVENT">Only event cue points (FE2_PNGEX_CP_EVENT)</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="[-1,9] Default: -1 (Z_DEFAULT_COMPRESSION)">FE2_PNGEX_COMPRESSION_LEVEL</abbr></th>
 <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_COMPRESSION_LEVEL'></td>
</tr>

<tr><th><b>Automatic PNG Export Options:</b></th><td></td></tr>
```

```
   <tr>
    <th>FE2_PNGEX_AUTO_EXPORT_COUNT</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_COUNT'></td>
   </tr>

   <tr>
    <th><abbr title="start time in ms; Default: 0">FE2_PNGEX_AUTO_EXPORT_START_TIME</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_START_TIME'></td>
   </tr>

   <tr>
    <th><abbr title="stop time in ms; Default: <clip length>">FE2_PNGEX_AUTO_EXPORT_END_TIME</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_END_TIME'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD'></td>
   </tr>

  </table>

 </td>
</tr>
<!-- END - PNGEX filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_ROTATE' value='vfilter:'
          onchange="toggle_ftable('filter_rotate',this.checked)">
   FE2_FILTER_ROTATE
  </label>

  <table id='filter_rotate' class='disabled'>
   <tr>
    <th><abbr title="Default: 0. valid: {0,90,180,270}">FE2_ROTATE_ANGLE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_ROTATE_ANGLE'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - ROTATE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_SCALE' value='vfilter:'
          onchange="toggle_ftable('filter_scale',this.checked)">
   FE2_FILTER_SCALE
  </label>

  <table id='filter_scale' class='disabled'>
   <tr>
    <th><abbr title="Default: input image width">FE2_SCALE_WIDTH</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_SCALE_WIDTH'></td>
   </tr>

   <tr>
    <th><abbr title="Default: input image height">FE2_SCALE_HEIGHT</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_SCALE_HEIGHT'></td>
   </tr>
  </table>
```

```
 </td>
</tr>
<!-- END - SCALE filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_SHARPEN' value='vfilter:'
          onchange="toggle_ftable('filter_sharpen',this.checked)">
   FE2_FILTER_SHARPEN
  </label>

  <table id='filter_sharpen' class='disabled'>
   <tr>
    <th>(FE2_FILTER_SHARPEN defines no parameters)</th>
   </tr>
  </table>

 </td>
</tr>
<!-- END - SHARPEN filter parameters -->

<tr><th><b>Audio Filters</b></th></tr>

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_HIGHPASS' value='afilter:'
          onchange="toggle_ftable('filter_highpass',this.checked)">
   FE2_FILTER_HIGHPASS
  </label>

  <table id='filter_highpass' class='disabled'>
   <tr>
    <th><abbr title="Default: 0.707">FE2_HIGHPASS_Q</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_HIGHPASS_Q'></td>
   </tr>

   <tr>
    <th>FE2_HIGHPASS_CUTOFF</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_HIGHPASS_CUTOFF'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - HIGHPASS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_LOWPASS' value='afilter:'
          onchange="toggle_ftable('filter_lowpass',this.checked)">
   FE2_FILTER_LOWPASS
  </label>

  <table id='filter_lowpass' class='disabled'>
   <tr>
    <th><abbr title="Default: 0.707">FE2_LOWPASS_Q</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_LOWPASS_Q'></td>
   </tr>

   <tr>
```

```
    <th>FE2_LOWPASS_CUTOFF</th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_LOWPASS_CUTOFF'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - LOWPASS filter parameters -->

<tr>
 <td>
  <label>
   <input type='checkbox' class='filter'
          name='FE2_FILTER_RESAMPLE' value='afilter:'
          onchange="toggle_ftable('filter_resample',this.checked)">
   FE2_FILTER_RESAMPLE
  </label>

  <table id='filter_resample' class='disabled'>
   <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_RATE</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_RESAMPLE_RATE'></td>
   </tr>

   <tr>
    <th><abbr title="Default: 0">FE2_RESAMPLE_CHANNELS</abbr></th>
    <td><input type='text' name='Flix2_FilterSetParam:FE2_RESAMPLE_CHANNELS'></td>
   </tr>
  </table>

 </td>
</tr>
<!-- END - RESAMPLE filter parameters -->

</table>
</fieldset>

<!-- ##MUXERS########################################################################## -->
<hr>
<fieldset class='collapsed' id="muxers">
<legend onclick='toggle_expand(this)'>Muxers</legend>
<table>

<tr>
 <td>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3GP'
         onfocus="set_muxer_visible('tgptable')">
  FE2_MUXER_3GP 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_3G2'
         onfocus="set_muxer_visible('tg2table')">
  FE2_MUXER_3G2 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MOV'
         onfocus="set_muxer_visible('movtable')">
  FE2_MUXER_MOV 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_MP4'
         onfocus="set_muxer_visible('mp4table')">
  FE2_MUXER_MP4 
 </label>
 <br>
```

```
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FLV'
         onfocus="set_muxer_visible('flvtable',true)">
  FE2_MUXER_FLV 
 </label>
 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_SWF'
         onfocus="set_muxer_visible('swftable',true)">
  FE2_MUXER_SWF
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_FXM'
         onfocus="set_muxer_visible('fxmtable',true)">
  FE2_MUXER_FXM 
 </label>
 <br>

 <label>
  <input type='radio' name='muxer:' value='FE2_MUXER_WEBM'
         onfocus="set_muxer_visible('webmtable',true)">
  FE2_MUXER_WEBM 
 </label>
 </td>
</tr>

<tr>
 <td>
 <!-- 3GP muxer parameters -->
 <table id='tgptable' class='hidden'>
   <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
     <select name="Flix2_MuxerSetParam:FE2_3GP_FASTSTART">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
 </table>
 <!-- END - 3GP muxer parameters -->

 <!-- 3G2 muxer parameters -->
 <table id='tg2table' class='hidden'>
   <tr>
    <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
    <td>
     <select name="Flix2_MuxerSetParam:FE2_3G2_FASTSTART">
     <option value=""></option>
     <option value="on2false">FALSE</option>
     <option value="on2true">TRUE</option>
     </select>
    </td>
   </tr>
 </table>
 <!-- END - 3G2 muxer parameters -->

 <!-- FLV muxer parameters -->
 <table id='flvtable' class='hidden'>
  <tr>
   <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'">
       FE2_FLV_CUEPT_EVENT</abbr></th>
   <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_EVENT'></td>
  </tr>
```

```
<tr>
    <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'">
     FE2_FLV_CUEPT_NAV</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_NAV'></td>
</tr>

<tr>
 <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'">
     FE2_FLV_CUEPT_PARAM</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FLV_CUEPT_PARAM'></td>
</tr>

<tr>
  <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
            FE2_FLV_METADATA_ENABLE</abbr></th>
  <td>
    <table id='flv_metadata_enable' class=''>
     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Ena
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DURATION' name='Flix
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Ena
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_DATASIZE' name='Flix
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_SIZE' name='Fl
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_SIZE' name='Fl
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_DATARATE' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_DATARATE' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_AUDIO_CODECID' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_VIDEO_CODECID' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH'><abbr title="Default: Enable
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_WIDTH' name='Flix2_M
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT'><abbr title="Default: Enabl
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_HEIGHT' name='Flix2_
     </tr>
```

```
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_FRAMERATE' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_CANSEEKTOEND' name='
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTTIMESTAMP' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAM
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION'><abbr title="
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Di
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_ENABLE:MD_KEYFRAMES' name='Fli
      </tr>
     </table>
   </td>
 </tr>

 <tr>
   <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
             FE2_FLV_METADATA_DISABLE</abbr></th>
   <td>
     <table id='flv_metadata_disable' class=''>
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DURATION'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DURATION' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_DATASIZE' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_SIZE' name='F
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_SIZE' name='F
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_DATARATE' nam
      </tr>
```

```
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_AUDIO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_VIDEO_CODECID' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabl
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_WIDTH' name='Flix2_
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enab
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_HEIGHT' name='Flix2
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_FRAMERATE' name='Fl
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_CANSEEKTOEND' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Defaul
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTTIMESTAMP' name
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTA
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_LASTKEYFRAMELOCATIO
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: E
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FLV_METADATA_DISABLE:MD_KEYFRAMES' name='Fl
      </tr>
     </table>
   </td>
 </tr>
</table>
<!-- END - FLV muxer parameters -->

<!-- FXM muxer parameters -->
<table id='fxmtable' class='hidden'>
 <tr>
  <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'evtpt0=343.0'">
      FE2_FXM_CUEPT_EVENT</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_EVENT'></td>
 </tr>

 <tr>
     <th><abbr title="Format = cueptNAME '=' cueptTIME_SECONDS e.g. 'navpt0=343.0'">
      FE2_FXM_CUEPT_NAV</abbr></th>
```

```
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_NAV'></td>
</tr>

<tr>
 <th><abbr title="e.g. 'cuept_name&amp;n0=v0&amp;n1=v1...'">
     FE2_FXM_CUEPT_PARAM</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_FXM_CUEPT_PARAM'></td>
</tr>

<tr>
  <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
            FE2_FXM_METADATA_ENABLE</abbr></th>
  <td>
    <table id='fxm_metadata_enable' class=''>
     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DURATION'><abbr title="Default: Ena
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DURATION' name='Flix
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE'><abbr title="Default: Ena
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_DATASIZE' name='Flix
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_SIZE' name='Fl
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE'><abbr title="Default: E
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_SIZE' name='Fl
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_DATARATE' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE'><abbr title="Defaul
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_DATARATE' name
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_AUDIO_CODECID' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID'><abbr title="Default
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_VIDEO_CODECID' name=
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH'><abbr title="Default: Enable
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_WIDTH' name='Flix2_M
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT'><abbr title="Default: Enabl
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_HEIGHT' name='Flix2_
     </tr>

     <tr>
      <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE'><abbr title="Default: En
      <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_FRAMERATE' name='Fli
```

```
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_CANSEEKTOEND' name='
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP'><abbr title="Default
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTTIMESTAMP' name=
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title=
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMETIMESTAM
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION'><abbr title="
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_LASTKEYFRAMELOCATION
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES'><abbr title="Default: Di
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_ENABLE:MD_KEYFRAMES' name='Fli
      </tr>
     </table>
   </td>
 </tr>

 <tr>
   <th><abbr title="Select specific metadata entries to enable. Default for each item is provided.">
             FE2_FXM_METADATA_DISABLE</abbr></th>
   <td>
     <table id='fxm_metadata_disable' class=''>
      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DURATION'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DURATION' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE'><abbr title="Default: En
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_DATASIZE' name='Fli
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_SIZE' name='F
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE'><abbr title="Default:
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_SIZE' name='F
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE'><abbr title="Defau
       <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_DATARATE' nam
      </tr>

      <tr>
       <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID'><abbr title="Defaul
```

```
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_AUDIO_CODECID' name
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID'><abbr title="Defaul
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_VIDEO_CODECID' name
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH'><abbr title="Default: Enabl
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_WIDTH' name='Flix2_
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT'><abbr title="Default: Enab
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_HEIGHT' name='Flix2
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE'><abbr title="Default: E
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_FRAMERATE' name='Fl
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND'><abbr title="Default
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_CANSEEKTOEND' name=
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP'><abbr title="Defaul
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTTIMESTAMP' name
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTAMP'><abbr title
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMETIMESTA
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATION'><abbr title=
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_LASTKEYFRAMELOCATIO
       </tr>

       <tr>
        <th><label for='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES'><abbr title="Default: D
        <td><input type='checkbox' id='Flix2_MuxerSetParam:FE2_FXM_METADATA_DISABLE:MD_KEYFRAMES' name='Fl
       </tr>
      </table>
    </td>
  </tr>
</table>
<!-- END - FXM muxer parameters -->

<!-- MOV muxer parameters -->
<table id='movtable' class='hidden'>
  <tr>
   <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
   <td>
    <select name="Flix2_MuxerSetParam:FE2_MOV_FASTSTART">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
   </td>
  </tr>
</table>
<!-- END - MOV muxer parameters -->
```

```
<!-- MP4 muxer parameters -->
<table id='mp4table' class='hidden'>
  <tr>
   <th><abbr title="Default: FALSE">FE2_ISOMEDIA_FASTSTART</abbr></th>
   <td>
    <select name="Flix2_MuxerSetParam:FE2_MP4_FASTSTART">
    <option value=""></option>
    <option value="on2false">FALSE</option>
    <option value="on2true">TRUE</option>
    </select>
   </td>
  </tr>
</table>
<!-- END - MP4 muxer parameters -->

<!-- SWF muxer parameters -->
<table id='swftable' class='hidden'>
 <tr>
  <th><abbr title="Default: video width">FE2_SWF_WIDTH</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_WIDTH'></td>
 </tr>

 <tr>
  <th><abbr title="Default: video height">FE2_SWF_HEIGHT</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_HEIGHT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: video framerate">FE2_SWF_FRAMERATE</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_FRAMERATE'></td>
 </tr>

 <tr>
  <th>FE2_SWF_LOOP_COUNT</th>
  <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_LOOP_COUNT'></td>
 </tr>

 <tr>
  <th><abbr title="Default: none">FE2_SWF_EMBEDDED_URL</abbr></th>
  <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_EMBEDDED_URL'></td>
 </tr>

 <tr>
  <th><abbr title="Default: _self">FE2_SWF_EMBEDDED_URL_TARGET</abbr></th>
  <td>
   <select name='Flix2_MuxerSetParamAsStr:FE2_SWF_EMBEDDED_URL_TARGET'>
    <option value=""></option>
    <option value="_self">_self</option>
    <option value="_blank">_blank</option>
    <option value="_parent">_parent</option>
    <option value="_top">_top</option>
   </select>
   </td>
 </tr>

 <tr>
  <th><abbr title="Default: EmbeddedUrlIsLoadMovie">FE2_SWF_EMBEDDED_URL_TYPE</abbr></th>
  <td>
   <select name='Flix2_MuxerSetParam:FE2_SWF_EMBEDDED_URL_TYPE'>
    <option value=""></option>
    <option value="EmbeddedUrlIsGetUrl">EmbeddedUrlIsGetUrl</option>
    <option value="EmbeddedUrlIsLoadMovie">EmbeddedUrlIsLoadMovie</option>
   </select>
  </td>
 </tr>
```

```
<tr>
 <th><abbr title="e.g. n0=v0&amp;n1=v1...">FE2_SWF_ADD_VARIABLE</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_ADD_VARIABLE'></td>
</tr>

<tr><th><b>Preloader Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfPreloaderNone">FE2_SWF_PRELOAD_TYPE</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_PRELOAD_TYPE'>
   <option value=""></option>
   <option value="SwfPreloaderNone">SwfPreloaderNone</option>
   <option value="SwfFixedPreloader">SwfFixedPreloader</option>
   <option value="SwfAdaptivePreloader">SwfAdaptivePreloader</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 20">FE2_SWF_FIXED_PRELOAD_PCT</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_FIXED_PRELOAD_PCT'></td>
</tr>

<tr>
 <th><abbr title="Default: 1.1">FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR'></td>
</tr>

<tr><th><b>Start Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfOnMovieStartAutomatically">FE2_SWF_ON_START_OPTION</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_ON_START_OPTION'>
   <option value=""></option>
   <option value="SwfOnMovieStartAutomatically">SwfOnMovieStartAutomatically</option>
   <option value="SwfOnMovieStartOnClick">SwfOnMovieStartOnClick</option>
   <option value="SwfOnMovieStartWait">SwfOnMovieStartWait</option>
   <option value="SwfOnMovieStartEmbedSTOP">SwfOnMovieStartEmbedSTOP</option>
  </select>
 </td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_SWF_START_BLANK_FRAME</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_START_BLANK_FRAME'></td>
</tr>

<tr>
 <th><abbr title="Default: 0">FE2_SWF_START_WAIT_SEC</abbr></th>
 <td><input type='text' name='Flix2_MuxerSetParam:FE2_SWF_START_WAIT_SEC'></td>
</tr>

<tr><th><b>End Settings:</b></th><td></td></tr>

<tr>
 <th><abbr title="Default: SwfOnMovieEndNothing">FE2_SWF_ON_END_OPTION</abbr></th>
 <td>
  <select name='Flix2_MuxerSetParam:FE2_SWF_ON_END_OPTION'>
   <option value=""></option>
   <option value="SwfOnMovieEndNothing">SwfOnMovieEndNothing</option>
   <option value="SwfOnMovieEndSTOP">SwfOnMovieEndSTOP</option>
   <option value="SwfOnMovieEndLoop">SwfOnMovieEndLoop</option>
   <option value="SwfOnMovieEndUnload">SwfOnMovieEndUnload</option>
   <option value="SwfOnMovieEndLoadMovie">SwfOnMovieEndLoadMovie</option>
  </select>
```

```
   </td>
  </tr>

  <tr>
   <th>FE2_SWF_ON_END_URL</th>
   <td><input type='text' name='Flix2_MuxerSetParamAsStr:FE2_SWF_ON_END_URL'></td>
  </tr>
 </table>
 <!-- END - SWF muxer parameters -->

 <!-- WEBM muxer parameters -->
 <table id='webmtable' class='hidden'>
   <tr>
    <th>(FE2_MUXER_WEBM defines no parameters)</th>
   </tr>
 </table>
 <!-- END - WEBM muxer parameters -->

</table>
</fieldset>

<!-- ##END FORM########################################################## -->
<hr>
<p>
 <input type="submit" value="Start Encode">
 <input type='reset' value='Reset' onclick='reset_tables();'>
</p>
</form>
</div>
</body>
</html>
"""
```

16.2 process_sample.py

```
#!/usr/bin/python -u
##============================================================================
##
##  Copyright (c) On2 Technologies Inc. All Rights Reserved.
##
##----------------------------------------------------------------------------
##
##  File:        $Workfile$
##               $Revision$
##
##  Last Update: $DateUTC$
##
##----------------------------------------------------------------------------
##
import sys,os,cgi
import flixengine2
import time
import string
import re

def print_encoder_status():
    print "<br>Encoder Status<br>"

    res= flixengine2.Flix2_GetEncoderState(flix)
    print " Flix2_GetEncoderState: %d<br>" % res[1]

    res= flixengine2.Flix2_Errno(flix)
    print " Flix2_Errno: sc:%d flixerrno:%d syserrno:%d<br>" %\
        (res[0],res[1],res[2])
```

```
def process_sc(func, sc):
    print "<td align=\"center\">%d</td>" % sc

    if (sc != flixengine2.ON2_OK):
        if (sc == flixengine2.ON2_NET_ERROR): str = "rpcerr"
        else: str = "flixerrno"

        res = flixengine2.Flix2_Errno(flix)
        print "<td>Flix2_Errno: sc:%d %s:%d syserrno:%d</td>" %\
            (res[0],str,res[1],res[2])
        raise Exception(func)

    print "</tr>"

def SimpleGet(func):
    res = func(flix)
    if (res[0] != 0):
        raise Exception(func)
    return res[1]

def SimpleSet(func, arg):
    functry = func+'(flix, arg)'
    print "<tr><td>%s( %s )</td>"  % (func, arg)
    sc = eval(functry)
    process_sc(func,sc)

def SimpleSetNum(func, arg):
    print "<tr><td>%s( %s )</td>"  % (func, arg)
    if(re.search('[a-zA-Z_]', arg)):
        functry = func+'(flix, flixengine2.'+arg+')'
    else:
        functry = func+'(flix, int(arg))'
    sc = eval(functry)
    process_sc(func,sc)

def init_codec(name):
    global codec, codecptr

    ##if name is a codec name, e.g. FE2_CODEC_VP6, add an instance
    ##we'll assume all Flix2_CodecSetParam's relate to this codec until we
    ##hit the next codec name
    if(codecptr):
        flixengine2.delete_flix2plgnhandlep(codecptr)

    print "<tr><td>Flix2_AddCodec( %s )</td>" % name
    codecptr = flixengine2.new_flix2plgnhandlep()
    sc = flixengine2.Flix2_AddCodec(codecptr, flix, eval('flixengine2.'+name))
    process_sc('Flix2_AddCodec',sc)

    #retrieve the codec handle to be used in all Codec API function calls
    codec = flixengine2.flix2plgnhandlep_value(codecptr)

def codec_interface(funcname, name, value):
    print "<tr><td>%s( %s, %s )</td>" % (funcname,name,value)
    if 'AsStr' in funcname:
        functry = 'flixengine2.'+funcname+'(codec, flixengine2.'+name+' , \"'+value+'\")'
    elif(re.search('[a-zA-Z_]', value)):
        functry = 'flixengine2.'+funcname+'(codec, flixengine2.'+name+' , flixengine2.'+value+')'
    else:
        functry = 'flixengine2.'+funcname+'(codec, flixengine2.'+name+' , '+value+')'

    sc = eval(functry)
    process_sc(funcname,sc)

def init_filter(name):
    global filter, filterptr
```

```
    ##if name is a filter name, e.g. FE2_FILTER_CUT, add an instance
    ##we'll assume all Flix2_FilterSetParam's relate to this filter until we
    ##hit the next filter name
    if(filterptr):
        flixengine2.delete_flix2plgnhandlep(filterptr)

    print "<tr><td>Flix2_AddCodec( %s )</td>" % name
    filterptr = flixengine2.new_flix2plgnhandlep()
    sc = flixengine2.Flix2_AddFilter(filterptr, flix, eval('flixengine2.'+name))
    process_sc('Flix2_AddFilter',sc)

    #retrieve the filter handle to be used in all Filter API function calls
    filter = flixengine2.flix2plgnhandlep_value(filterptr)

def filter_interface(funcname, name, value):
    print "<tr><td>%s( %s, %s )</td>" % (funcname,name,value)
    if 'AsStr' in funcname:
        functry = 'flixengine2.'+funcname+'(filter, flixengine2.'+name+' , \"'+value+'\")'
    elif(re.search('[a-zA-Z_]', value)):
        functry = 'flixengine2.'+funcname+'(filter, flixengine2.'+name+' , flixengine2.'+value+')'
    else:
        functry = 'flixengine2.'+funcname+'(filter, flixengine2.'+name+' , '+value+')'

    sc = eval(functry)
    process_sc(funcname,sc)

def init_muxer(name):
    global muxer, muxerptr

    ##if name is a muxer name, e.g. FE2_MUXER_FLV, add an instance
    ##we'll assume all Flix2_MuxerSetParam's relate to this muxer until we
    ##hit the next muxer name
    if(muxerptr):
        flixengine2.delete_flix2plgnhandlep(muxerptr)

    print "<tr><td>Flix2_AddMuxer( %s )</td>" % name
    muxerptr = flixengine2.new_flix2plgnhandlep()
    sc = flixengine2.Flix2_AddMuxer(muxerptr, flix, eval('flixengine2.'+name))
    process_sc('Flix2_AddMuxer',sc)

    #retrieve the muxer handle to be used in all Muxer API function calls
    muxer = flixengine2.flix2plgnhandlep_value(muxerptr)

def muxer_interface(funcname, name, value):
    print "<tr><td>%s( %s, %s )</td>" % (funcname,name,value)
    if 'AsStr' in funcname:
        functry = 'flixengine2.'+funcname+'(muxer, flixengine2.'+name+' , \"'+value+'\")'
    elif(re.search('[a-zA-Z_]', value)):
        functry = 'flixengine2.'+funcname+'(muxer, flixengine2.'+name+' , flixengine2.'+value+')'
    else:
        functry = 'flixengine2.'+funcname+'(muxer, flixengine2.'+name+' , '+value+')'

    sc = eval(functry)
    process_sc(funcname,sc)

rpchost = 'localhost'
outdir  = '/var/www/cgi-bin/flixmedia/out/'

print """
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
<head>
<title>Flix CGI Process Sample - Python</title>
</head>
<body>
```

```
<hr>
"""

print "<p>process_sample.py version 1.8<br>\n"
print "Flix Engine client library v%s" % flixengine2.Flix2_Version()
print "<br>"
str = flixengine2.Flix2_Copyright()
str = string.replace(str,"\n","<br>")
print "%s</p>" % str

if ( os.environ['REQUEST_METHOD'] == 'GET' ) :
        if  os.environ['QUERY_STRING'] :
                # This will always yield a non-empty FieldStorage object.
                # That is, form.keys() will never be [].
                form = cgi.FieldStorage()

        else :
                # The query string is empty.
                # Calling cgi.FieldStorage() here would enter an infinite loop.
                form = None
elif ( os.environ['REQUEST_METHOD'] == 'POST' ) :
        # This will yield a non-empty FieldStorage object whose keys()
        # may or may not be [].
        form = cgi.FieldStorage()

        # If the form is blank, it's probably easiest to make this the equivalent
        # of there being no form at all.
        if not form.keys() :
                form = None

##verify outdir's presence and accessibility
if (rpchost == 'localhost' and\
    not (os.path.isdir(outdir) and os.access(outdir,os.W_OK))):
    (i,tt)=("<i>'"+outdir+"'</i>",'<tt>outdir</tt>')
    print "<p>****************************************<br>\n"\
          "<b>WARNING</b>: %s MUST exist and be writeable by <i>flixd</i>.\n"\
          "<br>Please make %s accessible or modify the %s"\
          " value defined in '%s'.<br>\n" % (i,i,tt,sys.argv[0])
    g=os.popen("grep -m 1 -H -n '^outdir' "+sys.argv[0]).read()
    if g:
        print "The definition of %s can be found here:<br>\n%s<br>\n" % (tt,g);
    print "****************************************</p>\n"

print "<table border='1' cellpadding='5'>"
print "<caption>Flix Function Calls</caption>\n"
print "<tr><th>Function Name</th><th>Return Value</th></tr>\n"
print "<tr><td>Flix2_CreateEx()</td>"

flixptr = flixengine2.new_flix2handlep()
sc = flixengine2.Flix2_CreateEx(flixptr, "localhost", 0)

#extract the handle value returned from _Create. flix will be used in
#every Flix API call that follows
flix = flixengine2.flix2handlep_value(flixptr)

process_sc('Flix2_CreateEx',sc)

codecptr = codec = 0
filterptr = filter = 0
muxerptr = muxer = 0

if (form) :
  for key in form:
    values = form[key]
    if not isinstance(values, list):
        value = [values.value]
        if key == 'Flix2_SetInputFile':
```

```
          funcname = 'flixengine2.'+key
          SimpleSet(funcname, value[0])
        elif key == 'Flix2_SetOutputFile':
          outfile = outdir+value[0]
          funcname = 'flixengine2.'+key
          SimpleSet(funcname, outfile)
        elif string.find(value[0],"FE2_CODEC") != -1:
          init_codec(value[0])
        elif string.find(key,"FE2_FILTER") != -1:
          init_filter(key)
        elif string.find(value[0],"FE2_MUXER") != -1:
          init_muxer(value[0])
        elif string.find(key, "CodecSetParam") != -1:
          temp = string.split(key,":")
          codec_interface(temp[0],temp[1],value[0])
        elif string.find(key, "FilterSetParam") != -1:
          temp = string.split(key,":")
          filter_interface(temp[0],temp[1],value[0])
        elif string.find(key, "MuxerSetParam") != -1:
          temp = string.split(key,":")
          muxer_interface(temp[0],temp[1],value[0])
        else:
          funcname = 'flixengine2.'+key
          SimpleSetNum(funcname, value[0])

print "<tr><td>Flix2_Encode()</td>"
sc = flixengine2.Flix2_Encode(flix)
print "<td align=\"center\">%d</td></tr>\n"  % (sc)
print "</table>\n"
print "<br>Encoding...(video frames encoded, percent complete). "
print "Total frames will reset when doing 2pass.<br>"

ier = 1
while(sc == 0 and ier != 0):
    ier = SimpleGet(flixengine2.Flix2_IsEncoderRunning)
    res = flixengine2.encoding_status_GetTotalFrames(flix)
    p = flixengine2.encoding_status_PercentComplete(flix)
    if (res[0] == 0):
        print "(%d, %d%%)<br>\n" % (res[1], p[1])
    time.sleep(1)

print "<br>Done!"
print_encoder_status()

sc = flixengine2.Flix2_Destroy(flix)

flixengine2.delete_flix2plgnhandlep(flixptr)
if codecptr: flixengine2.delete_flix2plgnhandlep(codecptr)
if filterptr: flixengine2.delete_flix2plgnhandlep(filterptr)
if muxerptr: flixengine2.delete_flix2plgnhandlep(muxerptr)

print """
</body>
</html>
"""
```

Chapter 17

Deprecated List

Global FE2_AudioSamplingrates With the addition of new Muxers the allowed sample rates will change. For this reason the desired value should be entered directly in Flix2_FilterSetParam().

Global FE2_FlvAudioFormat Use the Codec Interface. This enumeration will be removed in a future release. See also: Audio Codecs.

Global audio_options_GetBitrate Use the Codec Interface along with the FE2_CODECPARAM_-BITRATE parameter. This function will be removed in a future release. See also: Audio Codecs.

Global audio_options_GetFlvAudioFormat Use the Codec Interface. This function will be removed in a future release. See also: Audio Codecs.

Global audio_options_GetSamplingrate Use the Filter Interface along with FE2_FILTER_-RESAMPLE and the FE2_RESAMPLE_RATE parameter.

Global audio_options_GetStereo Use the Codec Interface along with the FE2_LAME_CHANNELS parameter. This function will be removed in a future release. See also: Audio Codecs.

Global audio_options_Reset Use the Codec and Filter interfaces.

Global audio_options_SetBitrate Use the Codec Interface along with the FE2_CODECPARAM_-BITRATE parameter. This function will be removed in a future release. See also: Audio Codecs.

Global audio_options_SetFlvAudioFormat Use the Codec Interface. This function will be removed in a future release. See also: Audio Codecs.

Global audio_options_SetSamplingrate Use the Filter Interface along with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

Global audio_options_SetStereo Use the Codec Interface along with the FE2_LAME_CHANNELS parameter. This function will be removed in a future release. See also: Audio Codecs.

Global audio_options_Validate Use the Codec and Filter interfaces.

Global FE2_LAME_CHANNELS Use the Filter Interface along with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

Global video_options_GetDeinterlace Use the Filter Interface along with FE2_FILTER_ADAPTIVE_-DEINTERLACE. This function will be removed in a future release.

Global video_options_SetDeinterlace Use the Filter Interface along with FE2_FILTER_ADAPTIVE_-DEINTERLACE. This function will be removed in a future release.

Global editor_options_GetBrightness Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

Global editor_options_SetBrightness Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

Global editor_options_GetUseBrightness Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

Global editor_options_SetUseBrightness Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

Global editor_options_GetContrast Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

Global editor_options_SetContrast Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

Global editor_options_GetUseContrast Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

Global editor_options_SetUseContrast Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

Global editor_options_GetHue Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_-BCHS_HUE parameter. This function will be removed in a future release.

Global editor_options_SetHue Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_-BCHS_HUE parameter. This function will be removed in a future release.

Global editor_options_GetUseHue Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_HUE parameter. This function will be removed in a future release.

Global editor_options_SetUseHue Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_HUE parameter. This function will be removed in a future release.

Global editor_options_GetSaturation Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

Global editor_options_SetSaturation Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

Global editor_options_GetUseSaturation Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

Global editor_options_SetUseSaturation Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

Global editor_options_GetCrop Use the Filter Interface along with FE2_FILTER_CROP. This function will be removed in a future release.

Global editor_options_SetCrop Use the Filter Interface along with FE2_FILTER_CROP. This function will be removed in a future release.

Global editor_options_GetCropBounds Use the Filter Interface along with FE2_FILTER_CROP and the FE2_CROP_TOP, FE2_CROP_LEFT, FE2_CROP_BOTTOM and FE2_CROP_RIGHT parameters. This function will be removed in a future release.

Global editor_options_SetCropBounds Use the Filter Interface along with FE2_FILTER_CROP and the FE2_CROP_TOP, FE2_CROP_LEFT, FE2_CROP_BOTTOM and FE2_CROP_RIGHT parameters. This function will be removed in a future release.

Global editor_options_GetUseCut Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

Global editor_options_SetUseCut Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT

- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

Global editor_options_GetCutStartTime Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

Global editor_options_SetCutStartTime Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

Global editor_options_GetCutStopTime Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

Global editor_options_SetCutStopTime Please use:

- Flix2_AddFilter()

- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

Global video_options_GetVideoFramerate Please use video_options_GetVideoFramerateAsDouble() to allow for non integer framerates.

Global video_options_SetVideoFramerate Please use video_options_SetVideoFramerateAsDouble() to allow for non integer framerates.

Global video_options_GetVideoFramerateAsDouble Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_FPS parameter. This function will be removed in a future release.

Global video_options_SetVideoFramerateAsDouble Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_FPS parameter. This function will be removed in a future release.

Global video_options_GetUseSourceFramerate Use the Filter Interface along with FE2_FILTER_FRAMERATE. Not adding FE2_FILTER_FRAMERATE with Flix2_AddFilter() implies use of the source frame rate. This function will be removed in a future release.

Global video_options_SetUseSourceFramerate Use the Filter Interface along with FE2_FILTER_FRAMERATE. Not adding FE2_FILTER_FRAMERATE with Flix2_AddFilter() implies use of the source frame rate. This function will be removed in a future release.

Global video_options_GetDecimateValue Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_DECIMATE parameter. This function will be removed in a future release.

Global video_options_SetDecimateValue Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_DECIMATE parameter. This function will be removed in a future release.

Global overlay_options_Reset Use the Filter Interface along with FE2_FILTER_OVERLAY. This function will be removed in a future release.

Global overlay_options_GetUseOverlay Use the Filter Interface along with FE2_FILTER_OVERLAY. This function will be removed in a future release.

Global overlay_options_SetUseOverlay Use the Filter Interface along with FE2_FILTER_OVERLAY. This function will be removed in a future release.

Global overlay_options_GetOverlayPath Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_FILE parameter. This function will be removed in a future release.

Global overlay_options_SetOverlayPath Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_FILE parameter. This function will be removed in a future release.

Global overlay_options_GetMaskPixelXY Use the Filter Interface along with FE2_FILTER_-OVERLAY and the FE2_OVERLAY_MASK_X and FE2_OVERLAY_MASK_Y parameters. This function will be removed in a future release.

Global overlay_options_SetMaskPixelXY Use the Filter Interface along with FE2_FILTER_-OVERLAY and the FE2_OVERLAY_MASK_X and FE2_OVERLAY_MASK_Y parameters. This function will be removed in a future release.

Global overlay_options_GetMaskPixelRGB Use the Filter Interface along with FE2_FILTER_-OVERLAY and the FE2_OVERLAY_MASK_R, FE2_OVERLAY_MASK_G and FE2_-OVERLAY_MASK_B parameters. This function will be removed in a future release.

Global overlay_options_SetMaskPixelRGB Use the Filter Interface along with FE2_FILTER_-OVERLAY and the FE2_OVERLAY_MASK_R, FE2_OVERLAY_MASK_G and FE2_-OVERLAY_MASK_B parameters. This function will be removed in a future release.

Global overlay_options_GetOverlayPosition Use the Filter Interface along with FE2_FILTER_-OVERLAY and the FE2_OVERLAY_POS, FE2_OVERLAY_POS_X, FE2_OVERLAY_POS_Y parameters. This function will be removed in a future release.

Global overlay_options_SetOverlayPosition Use the Filter Interface along with FE2_FILTER_-OVERLAY and the FE2_OVERLAY_POS, FE2_OVERLAY_POS_X, FE2_OVERLAY_POS_Y parameters. This function will be removed in a future release.

Global video_options_GetImageHeight Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

Global video_options_SetImageHeight Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

Global video_options_GetImageWidth Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

Global video_options_SetImageWidth Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

Global video_options_GetUseSourceDimensions The flix engine encodes video using the video source dimensions by default. If FE2_FILTER_SCALE has not been added to the filter chain, video will be encoded using source dimensions.

Global video_options_SetUseSourceDimensions The flix engine encodes video using the video source dimensions by default. If FE2_FILTER_SCALE has not been added to the filter chain, video will be encoded using source dimensions.

Global FE2_ExportedVideoType Use the Muxer Interface. This enumeration will be removed in a future release. See also: Muxers.

Global Flix2_GetExportVideoType Use the Muxer Interface. This function will be removed in a future release. See also: Muxers.

Global Flix2_SetExportVideoType Use the Muxer Interface. This function will be removed in a future release. See also: Muxers.

Global editor_options_Reset Use the Filter Interface

Global editor_options_Validate Use the Filter Interface

Global swf_options_Reset Use the Muxer Interface along with FE2_MUXER_SWF. This function will be removed in a future release.

Global swf_options_GetEmbeddedUrl Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL parameter. This function will be removed in a future release.

Global swf_options_GetEmbeddedUrlTarget Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_TARGET parameter. This function will be removed in a future release.

Global swf_options_SetEmbeddedUrl Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL parameter. This function will be removed in a future release.

Global swf_options_SetEmbeddedUrlTarget Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_TARGET parameter. This function will be removed in a future release.

Global swf_options_GetEmbeddedUrlType Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_TYPE parameter. This function will be removed in a future release.

Global swf_options_SetEmbeddedUrlType Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_TYPE parameter. This function will be removed in a future release.

Global swf_options_GetSwfFramerate This function has been deprecated in favor of swf_options_-GetSwfFramerateAsDouble() because this function can only handle integer framerates.

Global swf_options_SetSwfFramerate This function has been deprecated in favor of swf_options_-SetSwfFramerateAsDouble() because this function can only handle integer framerates.

Global swf_options_SetInsertBlankFrameOnStart Use the Muxer Interface along with FE2_-MUXER_SWF and the FE2_SWF_START_BLANK_FRAME parameter. This function will be removed in a future release.

Global swf_options_GetSwfFramerateAsDouble Use the Muxer Interface along with FE2_MUXER_-SWF and the FE2_SWF_FRAMERATE parameter. This function will be removed in a future release.

Global swf_options_SetSwfFramerateAsDouble Use the Muxer Interface along with FE2_MUXER_-SWF and the FE2_SWF_FRAMERATE parameter. This function will be removed in a future release.

Global swf_options_GetPercentToPreload Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_FIXED_PRELOAD_PCT parameter. This function will be removed in a future release.

Global swf_options_SetPercentToPreload Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_FIXED_PRELOAD_PCT parameter. This function will be removed in a future release.

Global swf_options_GetPreloaderType Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_PRELOAD_TYPE parameter. This function will be removed in a future release.

Global swf_options_SetPreloaderType Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_PRELOAD_TYPE parameter. This function will be removed in a future release.

Global swf_options_GetAdaptivePreloaderBufferFactor Use the Muxer Interface along with FE2_-MUXER_SWF and the FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR parameter. This function will be removed in a future release.

Global swf_options_SetAdaptivePreloaderBufferFactor Use the Muxer Interface along with FE2_-MUXER_SWF and the FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR parameter. This function will be removed in a future release.

Global swf_options_GetMovieOnEndOptions Use the Muxer Interface along with FE2_MUXER_-SWF and the FE2_SWF_ON_END_OPTION parameter. This function will be removed in a future release.

Global swf_options_SetMovieOnEndOptions Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ON_END_OPTION parameter. This function will be removed in a future release.

Global swf_options_GetLoopCount Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_LOOP_COUNT parameter. This function will be removed in a future release.

Global swf_options_SetLoopCount Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_LOOP_COUNT parameter. This function will be removed in a future release.

Global swf_options_GetLoadMovieOnEndUrl Use the Muxer Interface along with FE2_MUXER_-SWF and the FE2_SWF_ON_END_URL parameter. This function will be removed in a future release.

Global swf_options_SetLoadMovieOnEndUrl Use the Muxer Interface along with FE2_MUXER_-SWF and the FE2_SWF_ON_END_URL parameter. This function will be removed in a future release.

Global swf_options_GetMovieOnStartOptions Use the Muxer Interface along with FE2_MUXER_-SWF and the FE2_SWF_ON_START_OPTION parameter. This function will be removed in a future release.

Global swf_options_SetMovieOnStartOptions Use the Muxer Interface along with FE2_MUXER_-SWF and the FE2_SWF_ON_START_OPTION parameter. This function will be removed in a future release.

Global swf_options_GetWaitTimeToStart Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_START_WAIT_SEC parameter. This function will be removed in a future release.

Global swf_options_SetWaitTimeToStart Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_START_WAIT_SEC parameter. This function will be removed in a future release.

Global swf_options_AddVariable Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ADD_VARIABLE parameter. This function will be removed in a future release.

Global swf_options_DeleteVariable Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_DELETE_VARIABLE parameter. This function will be removed in a future release.

Global swf_options_GetVariableCount Use the Muxer Interface along with FE2_MUXER_SWF. This function will be removed in a future release.

Global swf_options_UpdateVariable Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ADD_VARIABLE parameter. This function will be removed in a future release.

Global video_options_GetSwfHeight Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_HEIGHT parameter. This function will be removed in a future release.

Global video_options_GetSwfWidth Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH parameter. This function will be removed in a future release.

Global video_options_GetUseCustomSwfDimensions Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH/FE2_SWF_HEIGHT parameters. This function will be removed in a future release.

Global video_options_SetSwfHeight Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_HEIGHT parameter. This function will be removed in a future release.

Global video_options_SetSwfWidth Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH parameter. This function will be removed in a future release.

Global video_options_SetUseCustomSwfDimensions Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH/FE2_SWF_HEIGHT parameters. This function will be removed in a future release.

Global FE2_CuePointType Use the Muxer Interface along with FE2_MUXER_FLV and the FE2_FLV_CUEPT_EVENT/FE2_FLV_CUEPT_NAV parameter. This enumeration will be removed in a future release.

Global FE2_VideoCodec Use the Codec Interface. This enumeration will be removed in a future release. See also: Video Codecs.

Global video_options_AddFLVCuePoint Use the Muxer Interface along with FE2_MUXER_FLV and the FE2_FLV_CUEPT_EVENT/FE2_FLV_CUEPT_NAV parameter. This function will be removed in a future release.

Global video_options_AddFLVCuePointParameter Use the Muxer Interface along with FE2_MUXER_FLV and the FE2_FLV_CUEPT_PARAM parameter. This function will be removed in a future release.

Global video_options_GetAlphaPercentage Use the Codec Interface along with FE2_CODEC_VP6ALPHA and the FE2_VP6A_ALPHA_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_GetCompressMode Use the Codec Interface along with FE2_CODEC_VP6 or FE2_CODEC_VP6ALPHA and the FE2_VP6_CXMODE or FE2_VP6A_CXMODE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_GetImageQuality The image quality functions have been deprecated in favor of the maximum bitrate functions. Use video_options_GetMaximumBitrate() instead.

Global video_options_GetKeyframeInterval Use the Codec Interface along with the FE2_-VCODECPARAM_KFFREQ parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_GetKeyframeIntervalType Use the Codec Interface along with the FE2_-VCODECPARAM_KFINTTYPE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_GetMaximumBitrate Use the Codec Interface along with the FE2_-CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_GetRateControlType Use the Codec Interface along with the FE2_-VCODECPARAM_RC_MODE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_GetSwfFramerate Please use swf_options_GetSwfFramerateAsDouble() to allow for non integer framerates.

Global video_options_GetUseMaximumBitrate Use the Codec Interface along with the FE2_-CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_GetVideoCodec Use the Codec Interface. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetAlphaPercentage Use the Codec Interface along with FE2_CODEC_-VP6ALPHA and the FE2_VP6A_ALPHA_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetCompressMode Use the Codec Interface along with FE2_CODEC_VP6 or FE2_CODEC_VP6ALPHA and the FE2_VP6_CXMODE or FE2_VP6A_CXMODE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetImageQuality The image quality functions have been deprecated in favor of the maximum bitrate functions. Use video_options_GetMaximumBitrate() instead.

Global video_options_SetKeyframeInterval Use the Codec Interface along with the FE2_-VCODECPARAM_KFFREQ parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetKeyframeIntervalType Use the Codec Interface along with the FE2_-VCODECPARAM_KFINTTYPE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetMaximumBitrate Use the Codec Interface along with the FE2_-CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetRateControlType Use the Codec Interface along with the FE2_-VCODECPARAM_RC_MODE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetSwfFramerate Please use swf_options_SetSwfFramerateAsDouble() to allow for non integer framerates.

Global video_options_SetUseMaximumBitrate Use the Codec Interface along with the FE2_-CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

Global video_options_SetVideoCodec Use the Codec Interface. This function will be removed in a future release. See also: Video Codecs.

Chapter 18

Module Index

18.1 Modules

Here is a list of all modules:

- Flix Engine API . . . 354
  - Audio Encoding Options . . . 213
    - Deprecated . . . 215
  - Codecs . . . 223
    - Common Codec Parameters . . . 224
    - Video Codecs . . . 226
      - H263 - FFmpeg . . . 234
      - H264 . . . 237
      - VP6 . . . 245
      - VP6 with Alpha . . . 255
      - VP8 . . . 264
    - Audio Codecs . . . 227
      - AAC . . . 228
      - AAC+ . . . 230
      - AMR_NB - FFmpeg . . . 232
      - MP3 - LAME . . . 241
      - libvorbis - FFmpeg . . . 244
  - Encoding Statistics . . . 273
  - Filters . . . 278
    - Video Filters . . . 279
      - Deinterlace . . . 281
      - Brightness/Contrast/Hue/Saturation . . . 284
      - Blur . . . 295
      - Crop . . . 298
      - Denoise . . . 309
      - Frame Rate . . . 310
      - Mirror . . . 320
      - Overlay (Watermark) . . . 322
      - PNG Image Export (Thumbnail) . . . 336
      - Rotate . . . 345
      - Scale . . . 346
      - Sharpen . . . 353
    - Audio Filters . . . 280

Highpass . . . 316
Lowpass . . . 318
Resample . . . 344
Cut . . . 303
Main Engine Interface . . . 355
Filter Manipulation . . . 375
Deprecated . . . 393
Codec Manipulation . . . 379
Muxer Manipulation . . . 383
Muxers . . . 394
FLV . . . 395
FXM . . . 400
3GPP - FFmpeg . . . 403
3GPP2 - FFmpeg . . . 405
MOV - FFmpeg . . . 407
MP4 - FFmpeg . . . 409
SWF . . . 411
WebM - FFmpeg . . . 444
Video Encoding Options . . . 450
Deprecated . . . 455
Flix Engine Core Library . . . 371
Frame Server API . . . 387
Base Types . . . 445

Chapter 19

File Index

19.1 File List

Here is a list of all files with brief descriptions:

flixengine2/audio_options.h (Reference module(s): Audio Encoding Options) 471
flixengine2/codec_constants.h (Reference module(s): Codecs) 473
flixengine2/encoding_status.h (Reference module(s): Encoding Statistics) 487
flixengine2/filter_constants.h (Reference module(s): Filters) 488
flixengine2/flixengine2.h (Reference module(s): Main Engine Interface, Flix Engine Core Library, Filter Manipulation, Codec Manipulation, Muxer Manipulation) 507
flixengine2/fs_options.h (Reference module(s): Frame Server API) 513
flixengine2/media_editor_options.h (Reference module(s): Filter Manipulation, Video Filters) . 514
flixengine2/muxer_constants.h (Reference module(s): Muxers) 516
flixengine2/on2types.h (Reference module(s): Base Types) 525
flixengine2/overlay_options.h (Reference module(s): Overlay (Watermark)) 527
flixengine2/swf_options.h (Reference module(s): SWF) 528
flixengine2/video_options.h (Reference module(s): Video Encoding Options, Frame Rate, Scale) 531
flixengine2/codecs/aac.h (Reference module(s): AAC, AAC+) 474
flixengine2/codecs/amr.h (Reference module(s): AMR_NB - FFmpeg) 475
flixengine2/codecs/codec_common.h 476
flixengine2/codecs/h263.h (Reference module(s): H263 - FFmpeg) 477
flixengine2/codecs/h264.h (Reference module(s): H264) 478
flixengine2/codecs/lame.h (Reference module(s): MP3 - LAME) 479
flixengine2/codecs/vorbis.h (Reference module(s): libvorbis - FFmpeg) 480
flixengine2/codecs/vp6.h (Reference module(s): VP6) 481
flixengine2/codecs/vp6_alpha.h (Reference module(s): VP6 with Alpha) 483
flixengine2/codecs/vp8.h (Reference module(s): VP8) 485
flixengine2/filters/adaptive_deinterlace.h (Reference module(s): Deinterlace) 489
flixengine2/filters/bchs.h (Reference module(s): Brightness/Contrast/Hue/Saturation) 490
flixengine2/filters/blur.h (Reference module(s): Blur) 491
flixengine2/filters/crop.h (Reference module(s): Crop) 492
flixengine2/filters/cut.h (Reference module(s): Cut) 493
flixengine2/filters/denoise.h (Reference module(s): Denoise) 494
flixengine2/filters/framerate.h (Reference module(s): Frame Rate) 495
flixengine2/filters/highpass.h (Reference module(s): Highpass) 496
flixengine2/filters/lowpass.h (Reference module(s): Lowpass) 497
flixengine2/filters/mirror.h (Reference module(s): Mirror) 498

flixengine2/filters/overlay.h (Reference module(s): Overlay (Watermark)) 499
flixengine2/filters/png_export.h (Reference module(s): PNG Image Export (Thumbnail)) 501
flixengine2/filters/resample.h (Reference module(s): Resample) 503
flixengine2/filters/rotate.h (Reference module(s): Rotate) 504
flixengine2/filters/scale.h (Reference module(s): Scale) 505
flixengine2/filters/sharpen.h (Reference module(s): Sharpen) 506
flixengine2/muxers/flv.h (Reference module(s): FLV Muxer) 517
flixengine2/muxers/fxm.h (Reference module(s): FXM Muxer) 519
flixengine2/muxers/isomedia.h (Reference module(s): 3GPP Muxer, 3GPP2 Muxer, MOV Muxer, MP4 Muxer) 520
flixengine2/muxers/swf.h (Reference module(s): SWF Muxer) 522
flixengine2/muxers/webm.h (Reference module(s): WebM Muxer) 524

Chapter 20

Module Documentation

20.1 Audio Encoding Options

Modules

- Deprecated

Enumerations

- enum FE2_AudioBitrates {
  Bitrate8kbps,
  Bitrate16kbps,
  Bitrate24kbps,
  Bitrate32kbps,
  Bitrate40kbps,
  Bitrate48kbps,
  Bitrate56kbps,
  Bitrate64kbps,
  Bitrate80kbps,
  Bitrate96kbps,
  Bitrate112kbps,
  Bitrate128kbps,
  Bitrate144kbps,
  Bitrate160kbps,
  Bitrate192kbps,
  Bitrate224kbps,
  Bitrate256kbps,
  Bitrate320kbps }

  Valid bitrates for FE2_CODEC_LAME. For use in calls to Flix2_CodecSetParam() and Flix2_CodecGetParam().

20.1.1 Enumeration Type Documentation

20.1.1.1 enum FE2_AudioBitrates

Valid bitrates for FE2_CODEC_LAME. For use in calls to Flix2_CodecSetParam() and Flix2_-CodecGetParam().

Enumerator:

Bitrate8kbps

Bitrate16kbps

Bitrate24kbps

Bitrate32kbps

Bitrate40kbps

Bitrate48kbps

Bitrate56kbps

Bitrate64kbps

Bitrate80kbps

Bitrate96kbps

Bitrate112kbps

Bitrate128kbps

Bitrate144kbps

Bitrate160kbps

Bitrate192kbps

Bitrate224kbps

Bitrate256kbps

Bitrate320kbps

Definition at line 32 of file audio_options.h.

20.2 Deprecated

Enumerations

- enum FE2_FlvAudioFormat {
  FlvAudioUncompressed,
  FlvAudioMp3 }

  Valid output audio formats for use in calls to audio_options_GetFlvAudioFormat() and audio_options_-SetFlvAudioFormat().

- enum FE2_AudioSamplingrates {
  Hertz11025,
  Hertz22050,
  Hertz44100 }

  Sample rates for use with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

Functions

- on2sc audio_options_Reset (FLIX2HANDLE flix)

  Reset the audio encoding options to their default values.

- on2sc audio_options_Validate (FLIX2HANDLE flix)

  Validate the current audio encoding options.

- on2sc audio_options_SetBitrate (FLIX2HANDLE flix, const FE2_AudioBitrates bitrate)

  Set the bitrate to use in mp3 encoding.

- on2sc audio_options_GetBitrate (const FLIX2HANDLE flix, FE2_AudioBitrates *pBitrate)

  Retrieve the current mp3 bitrate.

- on2sc audio_options_SetFlvAudioFormat (FLIX2HANDLE flix, const FE2_FlvAudioFormat format)

  Set the output audio format.

- on2sc audio_options_GetFlvAudioFormat (const FLIX2HANDLE flix, FE2_FlvAudioFormat *pFormat)

  Retrieve the current output audio format.

- on2sc audio_options_SetSamplingrate (FLIX2HANDLE flix, const FE2_AudioSamplingrates samplingrate)

  Set the output audio sample rate.

- on2sc audio_options_GetSamplingrate (const FLIX2HANDLE flix, FE2_AudioSamplingrates *pSamplingrate)

  Retrieve the current output sample rate.

- on2sc audio_options_SetStereo (FLIX2HANDLE flix, const on2bool stereo)

Set the number of output audio channels.

- on2sc audio_options_GetStereo (const FLIX2HANDLE flix, on2bool *pStereo)

  Retrieve the current number of output audio channels.

20.2.1 Enumeration Type Documentation

20.2.1.1 enum FE2_AudioSamplingrates

Sample rates for use with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

Deprecated

With the addition of new Muxers the allowed sample rates will change. For this reason the desired value should be entered directly in Flix2_FilterSetParam().

Enumerator:

Hertz11025

Hertz22050

Hertz44100

Definition at line 70 of file audio_options.h.

20.2.1.2 enum FE2_FlvAudioFormat

Valid output audio formats for use in calls to audio_options_GetFlvAudioFormat() and audio_options_-SetFlvAudioFormat().

Deprecated

Use the Codec Interface. This enumeration will be removed in a future release. See also: Audio Codecs.

Enumerator:

FlvAudioUncompressed

FlvAudioMp3

Definition at line 59 of file audio_options.h.

20.2.2 Function Documentation

20.2.2.1 on2sc audio_options_GetBitrate (const FLIX2HANDLE *flix*, FE2_AudioBitrates * *pBitrate*)

Retrieve the current mp3 bitrate.

Only has meaning when outputting FlvAudioMp3

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pBitrate*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pBitrate is not `NULL`

Note:

Default: Bitrate64kbps

Deprecated

Use the Codec Interface along with the FE2_CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Audio Codecs.

20.2.2.2 on2sc audio_options_GetFlvAudioFormat (const FLIX2HANDLE *flix*, FE2_FlvAudioFormat * *pFormat*)

Retrieve the current output audio format.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pFormat*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pFormat is not `NULL`

Deprecated

Use the Codec Interface. This function will be removed in a future release. See also: Audio Codecs.

20.2.2.3 on2sc audio_options_GetSamplingrate (const FLIX2HANDLE *flix*, FE2_AudioSamplingrates * *pSamplingrate*)

Retrieve the current output sample rate.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pSamplingrate*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pSamplingrate is not `NULL`

Note:

How the default output sample rate is determined is described by audio_options_SetSamplingrate()

Deprecated

Use the Filter Interface along with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

20.2.2.4 on2sc audio_options_GetStereo (const FLIX2HANDLE *flix*, on2bool * *pStereo*)

Retrieve the current number of output audio channels.

Value indicates stereo (`on2true`) or mono (`on2false`)

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pStereo*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pStereo is not `NULL`

Deprecated

Use the Codec Interface along with the FE2_LAME_CHANNELS parameter. This function will be removed in a future release. See also: Audio Codecs.

20.2.2.5 on2sc audio_options_Reset (FLIX2HANDLE *flix*)

Reset the audio encoding options to their default values.

Defaults are set as described by: audio_options_SetBitrate(), audio_options_SetFlvAudioFormat(), audio_options_SetSamplingrate() and audio_options_SetStereo()

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Deprecated

Use the Codec and Filter interfaces.

20.2.2.6 on2sc audio_options_SetBitrate (FLIX2HANDLE *flix*, const FE2_AudioBitrates *bitrate*)

Set the bitrate to use in mp3 encoding.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bitrate*** Bitrate to use encoding mp3 audio

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
bitrate is a valid member of FE2_AudioBitrates

Note:

Default: Bitrate64kbps

Deprecated

Use the Codec Interface along with the FE2_CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Audio Codecs.

20.2.2.7 on2sc audio_options_SetFlvAudioFormat (FLIX2HANDLE *flix*, const FE2_FlvAudioFormat *format*)

Set the output audio format.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***format*** Output audio format

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
format is a valid member of FE2_FlvAudioFormat

Note:

Default: FlvAudioMp3

Deprecated

Use the Codec Interface. This function will be removed in a future release. See also: Audio Codecs.

20.2.2.8 on2sc audio_options_SetSamplingrate (FLIX2HANDLE *flix*, const FE2_AudioSamplingrates *samplingrate*)

Set the output audio sample rate.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***samplingrate*** Output audio sample rate

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
samplingrate is a valid member of FE2_AudioSamplingrates

Remarks:

Flash video files are limited by what sample rate audio they can store. The default output sample rate is determined using the source file attributes in the following manner:

- source sample rate < 16000Hz, set default to Hertz11025
- source sample rate >= 16000Hz and < 32000Hz, set default to Hertz22050
- source sample rate >= 32000Hz, set default to Hertz44100

Deprecated

Use the Filter Interface along with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

20.2.2.9 on2sc audio_options_SetStereo (FLIX2HANDLE *flix*, const on2bool *stereo*)

Set the number of output audio channels.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***stereo*** Indicates whether to encode stereo (`on2true`) or mono (`on2false`)

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Remarks:

Flash video files are limited by the number of output audio channels they can store. The default output number of channels is determined using the source file attributes in the following manner:

- input number of channels == 2, set default to: stereo (`on2true`)
- otherwise, set default to: mono (`on2false`)

Deprecated

Use the Codec Interface along with the FE2_LAME_CHANNELS parameter. This function will be removed in a future release. See also: Audio Codecs.

20.2.2.10 on2sc audio_options_Validate (FLIX2HANDLE *flix*)

Validate the current audio encoding options.

Ensures the current audio options are sufficient to start encoding. The checks currently performed are:

- sample rate is a valid member of FE2_AudioSamplingrates
- output format is a valid member of FE2_FlvAudioFormat

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK on success
ON2_INVALID_PARAMS should the precondition fail
ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Deprecated

Use the Codec and Filter interfaces.

20.3 Codecs

20.3.1 Detailed Description

Codecs are configurable both by the codec interface and individual functions (e.g., (audio|video)_options_-SetXXX). The latter, however, are deprecated and will be removed in a future release. Therefore, any new code should use the codec interface.

The currently available codecs are:

- Video Codecs
  - H263 - FFmpeg
  - H264
  - VP6
  - VP6 with Alpha
- Audio Codecs
  - AAC
  - AAC+
  - AMR_NB - FFmpeg
  - MP3 - LAME

Modules

- Common Codec Parameters
- Video Codecs
- Audio Codecs

20.4 Common Codec Parameters

Defines

- #define FE2_CODECPARAM_BITRATE

  Codec parameter for stream bitrate.

- #define FE2_VCODECPARAM_RC_MODE

  Video codec parameter for the rate control mode.

- #define FE2_VCODECPARAM_KFINTTYPE

  Video codec parameter for the keyframe interval type.

- #define FE2_VCODECPARAM_KFFREQ

  Video codec parameter for keyframe frequency.

Enumerations

- enum FE2_VideoBitrateControls {
  CBR_1PASSControl,
  VBR_1PASSControl,
  CBR_2PASSControl,
  VBR_2PASSControl }

  Encoder rate control types, influences quality.

- enum FE2_VideoKeyframeTypes {
  MAX_KEYFRAMES,
  FIXED_KEYFRAMES }

  Key frame interval types, influences quality.

20.4.1 Define Documentation

20.4.1.1 #define FE2_CODECPARAM_BITRATE

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Definition at line 47 of file codec_common.h.

20.4.1.2 #define FE2_VCODECPARAM_KFFREQ

Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of FE2_VCODECPARAM_-KFINTTYPE

See also:

FE2_VideoKeyframeTypes

Definition at line 67 of file codec_common.h.

20.4.1.3 #define FE2_VCODECPARAM_KFINTTYPE

Video codec parameter for the keyframe interval type.

Valid values are defined by FE2_VideoKeyframeTypes.

Definition at line 59 of file codec_common.h.

20.4.1.4 #define FE2_VCODECPARAM_RC_MODE

Video codec parameter for the rate control mode.

Valid values are defined by FE2_VideoBitrateControls.

Definition at line 53 of file codec_common.h.

20.4.2 Enumeration Type Documentation

20.4.2.1 enum FE2_VideoBitrateControls

Encoder rate control types, influences quality.

For use in calls to Flix2_CodecSetParam() and Flix2_CodecGetParam() for the FE2_VCODECPARAM_-RC_MODE parameter.

Enumerator:

CBR_1PASSControl	single pass constant bitrate
VBR_1PASSControl	single pass variable bitrate
CBR_2PASSControl	two pass constant bitrate
VBR_2PASSControl	two pass variable bitrate

Definition at line 26 of file codec_common.h.

20.4.2.2 enum FE2_VideoKeyframeTypes

Key frame interval types, influences quality.

For use in calls to Flix2_CodecSetParam() and Flix2_CodecGetParam() for the FE2_VCODECPARAM_-KFINTTYPE parameter.

Enumerator:

MAX_KEYFRAMES	generate keyframes at MOST FE2_VCODECPARAM_KFFREQ apart
FIXED_KEYFRAMES	generate keyframes at a fixed interval

Definition at line 37 of file codec_common.h.

20.5 Video Codecs

Modules

- H263 - FFmpeg
- H264
- VP6
- VP6 with Alpha
- VP8

20.6 Audio Codecs

Modules

- AAC
- AAC+
- AMR_NB - FFmpeg
- MP3 - LAME
- libvorbis - FFmpeg

Defines

- #define FE2_CODEC_PCM

  'Codec' name for use with Flix2_AddCodec()

20.6.1 Define Documentation

20.6.1.1 #define FE2_CODEC_PCM

'Codec' name for use with Flix2_AddCodec()

Allows for PCM audio output.

Definition at line 65 of file codec_constants.h.

20.7 AAC

20.7.1 Detailed Description

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_AAC);
// Use 128kbps
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_AAC_BITRATE, 128);
```

Valid Configurations

Channels	Bitrates (kbps)	Sampling Rate (kHz)
1ch	8,10	8,11.025,12
1ch	12	8,11.025,12,16
1ch	16	8 to 24
1ch	20	11.025 to 24
1ch	24,28	11.025 to 32
1ch	32	11.025 to 48
1ch	40	16 to 48
1ch	48,56	22.05 to 48
1ch	64,80,96,112,128,160	32,44.1,48
2ch	16,20	8,11.025,12
2ch	24,28,32	11.025 to 24
2ch	40	16 to 32
2ch	48	22.05 to 32
2ch	56	22.05 to 48
2ch	64,80,96,112,128,160,192,224,256,288,320	32,44.1,48

Attention:

Should the requested bitrate under/overshoot the above bounds it will be adjusted accordingly.

See also:

Resample

Additional References:

- ISO - International Organization for Standardization
- ISO/IEC 14496-3:MPEG-4 Audio
- Coding Technologies aacPlus documentation for further details regarding AAC and aacPlus compatibility

Defines

- #define FE2_CODEC_AAC

  AAC. Codec name for use with Flix2_AddCodec().

- #define FE2_AAC_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

20.7.2 Define Documentation

20.7.2.1 #define FE2_AAC_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 64

Definition at line 75 of file aac.h.

20.7.2.2 #define FE2_CODEC_AAC

AAC. Codec name for use with Flix2_AddCodec().

Definition at line 69 of file aac.h.

20.8 AAC+

20.8.1 Detailed Description

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_AACPLUS);
// Use 128kbps
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_AACPLUS_BITRATE, 128);
```

Valid Configurations

```
Channels               | Bitrates (kbps)            | Sampling Rate (kHz)
==========================================================================
1ch                    | 8,10,12,16                 | 22.05,24,32,44.1,48
1ch                    | 20,24,28,32,40,48          | 32,44.1,48
1ch                    | 56,64                      | 32,44.1,48,64,88.2,96
1ch                    | 80,96,112,128              | 64,88.2,96
2ch                    | 16,20,24,28,32,40,48,56,64 | 32,44.1,48
2ch                    | 80,96,112,128              | 32,44.1,48,64,88.2,96
2ch                    | 160,192,224,256            | 64,88.2,96
2ch-Parametric-Stereo  | 16,20,24,28,32,40,48,56    | 32,44.1,48
```

Attention:

Should the requested bitrate under/overshoot the above bounds it will be adjusted accordingly.

See also:

Resample

Additional References:

- ISO - International Organization for Standardization
- ISO/IEC 14496-3:MPEG-4 Audio
- Coding Technologies aacPlus documentation for further details regarding AAC and aacPlus compatibility

Defines

- #define FE2_CODEC_AACPLUS

  AAC+ (HE-AAC). Codec name for use with Flix2_AddCodec().

- #define FE2_AACPLUS_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_AACPLUS_PARAMETRIC_STEREO

  Parameter for parametric stereo.

20.8.2 Define Documentation

20.8.2.1 #define FE2_AACPLUS_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 64

Definition at line 126 of file aac.h.

20.8.2.2 #define FE2_AACPLUS_PARAMETRIC_STEREO

Parameter for parametric stereo.

The Parametric Stereo technology was standardized by ISO 14496-3. It is a method to code stereo information and thus allows for further enhanced coding efficiency for lowest bit rate stereo coding. Combining Parametric Stereo with SBR and AAC yields the aacPlus v2 codec.

Definition at line 135 of file aac.h.

20.8.2.3 #define FE2_CODEC_AACPLUS

AAC+ (HE-AAC). Codec name for use with Flix2_AddCodec().

Definition at line 120 of file aac.h.

20.9 AMR_NB - FFmpeg

20.9.1 Detailed Description

AMR narrowband encoder.

Valid bitrates:

AMR narrowband only supports a fixed set of bitrates:

```
 codec mode | kbps
===================
  MR475     | 4.75
  MR515     | 5.15
  MR59      | 5.90
  MR67      | 6.70
  MR74      | 7.40
  MR795     | 7.95
  MR102     | 10.2
  MR122     | 12.2
```

FE2_AMR_BITRATE will be adjusted as follows, if necessary:

```
FE2_AMR_BITRATE = 0.0  <= FE2_AMR_BITRATE <  5.0,  MR475
                  5.0  <= FE2_AMR_BITRATE <  5.9,  MR515
                  5.9  <= FE2_AMR_BITRATE <  6.7,  MR59
                  6.7  <= FE2_AMR_BITRATE <= 7.0,  MR67
                  7.0  <  FE2_AMR_BITRATE <  7.95, MR74
                  7.95 <= FE2_AMR_BITRATE <  10.0, MR795
                  10.0 <= FE2_AMR_BITRATE <  12.0, MR102
                  12.0 <= FE2_AMR_BITRATE,         MR122
```

Attention:

In addition to the bitrate restrictions above, the sample rate MUST be 8000Hz and the number of channels MUST be 1.

See also:

Resample

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_AMR_NB);
// Use 12.2kbps
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_AMR_BITRATE, 12.2);
```

Additional References:

- 3GPP homepage
- AMR-NB Specification detail
- FFmpeg project homepage

Defines

- #define FE2_CODEC_AMR_NB

AMR Narrowband. Codec name for use with Flix2_AddCodec().

- #define FE2_AMR_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

20.9.2 Define Documentation

20.9.2.1 #define FE2_AMR_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 12.2kbps

Definition at line 80 of file amr.h.

20.9.2.2 #define FE2_CODEC_AMR_NB

AMR Narrowband. Codec name for use with Flix2_AddCodec().

Definition at line 73 of file amr.h.

20.10 H263 - FFmpeg

20.10.1 Detailed Description

The H263 codec ....

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_H263);
// Use 450kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_H263_BITRATE, 450.0);
```

Additional References:

- FFmpeg project homepage

Defines

- #define FE2_CODEC_H263
  Codec name for use with Flix2_AddCodec().
- #define FE2_CODEC_H263_BASELINE
  Codec name for use with Flix2_AddCodec().
- #define FE2_H263_BITRATE
  Alias for FE2_CODECPARAM_BITRATE.
- #define FE2_H263_KFINTTYPE
  Alias for FE2_VCODECPARAM_KFINTTYPE.
- #define FE2_H263_KFFREQ
  Alias for FE2_VCODECPARAM_KFFREQ.
- #define FE2_H263_RC_MODE
  Alias for FE2_VCODECPARAM_RC_MODE.
- #define FE2_H263_MIN_Q
  Codec parameter for minimum quantizer.
- #define FE2_H263_MAX_Q
  Codec parameter for maximum quantizer.

20.10.2 Define Documentation

20.10.2.1 #define FE2_CODEC_H263

Codec name for use with Flix2_AddCodec().

Specifically this codec refers to the H.263 profile used within FLV.

Definition at line 42 of file h263.h.

20.10.2.2 #define FE2_CODEC_H263_BASELINE

Codec name for use with Flix2_AddCodec().

Specifically this codec refers to the baseline H.263 profile (profile 0) supported within 3GPP files amongst others.

Attention:

The baseline profile only supports the following resolutions:

- sub-QCIF (128x96)
- QCIF (176x144)
- CIF (352x288)

See also:

Scale
ITU-T Recommendation H.263 (01/05): "Video coding for low bit rate communication".

Definition at line 56 of file h263.h.

20.10.2.3 #define FE2_H263_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 63 of file h263.h.

20.10.2.4 #define FE2_H263_KFFREQ

Alias for FE2_VCODECPARAM_KFFREQ.

Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of FE2_VCODECPARAM_-KFINTTYPE

See also:

FE2_VideoKeyframeTypes

Note:

Default: $12.0 \times fps$ or 360 frames if the framerate is unknown

Definition at line 78 of file h263.h.

20.10.2.5 #define FE2_H263_KFINTTYPE

Alias for FE2_VCODECPARAM_KFINTTYPE.

Video codec parameter for the keyframe interval type.

Valid values are defined by FE2_VideoKeyframeTypes.

Note:

Default: MAX_KEYFRAMES

Definition at line 70 of file h263.h.

20.10.2.6 #define FE2_H263_MAX_Q

Codec parameter for maximum quantizer.

Note:

Default: 31

Definition at line 94 of file h263.h.

20.10.2.7 #define FE2_H263_MIN_Q

Codec parameter for minimum quantizer.

Note:

Default: 2

Definition at line 89 of file h263.h.

20.10.2.8 #define FE2_H263_RC_MODE

Alias for FE2_VCODECPARAM_RC_MODE.

Note:

Default: VBR_2PASSControl

Definition at line 84 of file h263.h.

20.11 H264

20.11.1 Detailed Description

The H264 video codec.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_H264);
// Use 450kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_H264_BITRATE, 450.0);
```

Attention:

Both width and height MUST be multiples of 4 for codec setup to succeed

Additional References:

- ISO - International Organization for Standardization
- ISO/IEC 14496-10: Advanced Video Coding
- ITU - International Telecommunication Union
- ITU-T Rec. H.264: Recommendation H.264

20.11.2 Apple device support

As of version 8.0.10.1 preliminary support for the iPhone has been added.

The following settings are required to enable video playback:

- FE2_H264_PROFILE set to BASE_H264PROFILE
- Resolution <= 640x480. iPhone native is 480x320, 480x360 is recommended for 4:3 content

  See also:

  FE2_MUXER_MP4
  Apple Developer Connection video creation guide (registration required)

Defines

- #define FE2_CODEC_H264

  Codec name for use with Flix2_AddCodec().

- #define FE2_H264_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_H264_KFINTTYPE

  Alias for FE2_VCODECPARAM_KFINTTYPE.

- #define FE2_H264_KFFREQ

  Alias for FE2_VCODECPARAM_KFFREQ.

- #define FE2_H264_RC_MODE

  Alias for FE2_VCODECPARAM_RC_MODE.

- #define FE2_H264_PROFILE

  Codec parameter for encoding profile.

- #define FE2_H264_B_FRAME_RATE

  Codec parameter for B frame rate.

- #define FE2_H264_SPEED

  Controls frame analysis, affecting encoder speed and inversely output quality.

Typedefs

- typedef enum h264profile h264profile_t

Enumerations

- enum h264profile {
  BASE_H264PROFILE,
  MAIN_H264PROFILE,
  HIGH_H264PROFILE }

  Valid profiles for use with the FE2_H264_PROFILE parameter.

20.11.3 Define Documentation

20.11.3.1 #define FE2_CODEC_H264

Codec name for use with Flix2_AddCodec().

Attention:

Both width and height MUST be multiples of 4

Definition at line 57 of file h264.h.

20.11.3.2 #define FE2_H264_B_FRAME_RATE

Codec parameter for B frame rate.

Specifies number of B frames between I/P and next P frame.

Note:

Default: 0

Definition at line 102 of file h264.h.

20.11.3.3 #define FE2_H264_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 63 of file h264.h.

20.11.3.4 #define FE2_H264_KFFREQ

Alias for FE2_VCODECPARAM_KFFREQ.

Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of FE2_VCODECPARAM_-KFINTTYPE

See also:

FE2_VideoKeyframeTypes

Note:

Default: $12.0 \times fps$ or 360 frames if the framerate is unknown

Definition at line 77 of file h264.h.

20.11.3.5 #define FE2_H264_KFINTTYPE

Alias for FE2_VCODECPARAM_KFINTTYPE.

Video codec parameter for the keyframe interval type.

Valid values are defined by FE2_VideoKeyframeTypes.

Note:

Default: MAX_KEYFRAMES

Attention:

Only MAX_KEYFRAMES is supported at this time

Definition at line 70 of file h264.h.

20.11.3.6 #define FE2_H264_PROFILE

Codec parameter for encoding profile.

Note:

Valid values are defined by h264profile_t
Default: MAIN_H264PROFILE

Definition at line 89 of file h264.h.

20.11.3.7 #define FE2_H264_RC_MODE

Alias for FE2_VCODECPARAM_RC_MODE.

Note:

Default: VBR_1PASSControl

Attention:

Only VBR_1PASSControl or CBR_1PASSControl are supported at this time

Definition at line 84 of file h264.h.

20.11.3.8 #define FE2_H264_SPEED

Controls frame analysis, affecting encoder speed and inversely output quality.

Lower values will will produce faster overall encode times, with the potential for quality loss, depending on the input material. A value of 1 provides a good balance between speed and quality, 2 or above will begin to perceptibly lengthen the encode.

Note:

Default:

- BASE_H264PROFILE/MAIN_H264PROFILE: 1
- HIGH_H264PROFILE: 3

Valid Range: [0,5]

Definition at line 116 of file h264.h.

20.11.4 Typedef Documentation

20.11.4.1 typedef enum h264profile h264profile_t

20.11.5 Enumeration Type Documentation

20.11.5.1 enum h264profile

Valid profiles for use with the FE2_H264_PROFILE parameter.

Enumerator:

BASE_H264PROFILE

MAIN_H264PROFILE

HIGH_H264PROFILE

Definition at line 92 of file h264.h.

20.12 MP3 - LAME

20.12.1 Detailed Description

The LAME codec is used to produce MP3 audio streams.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_LAME);
// Use 128kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_LAME_BITRATE, Bitrate128kbps);
```

Additional References:

LAME project homepage

Defines

- #define FE2_CODEC_LAME

  Codec name for use with Flix2_AddCodec().

- #define FE2_LAME_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_LAME_QUALITY

  Codec parameter for controlling LAME library's algorithm selection.

- #define FE2_LAME_RC_MODE

  Codec parameter for controlling LAME library's rate control method.

- #define FE2_LAME_CHANNELS

  Codec parameter for number of output channels.

Typedefs

- typedef enum lame_rcmode lame_rcmode_t

Enumerations

- enum lame_rcmode {
  LAME_CBR,
  LAME_ABR,
  LAME_VBR_rh,
  LAME_VBR_mtrh }

  Rate control modes analogous to those found in `<lame/lame.h>`.

20.12.2 Define Documentation

20.12.2.1 #define FE2_CODEC_LAME

Codec name for use with Flix2_AddCodec().

Definition at line 41 of file lame.h.

20.12.2.2 #define FE2_LAME_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec. Valid bitrates are defined by FE2_AudioBitrates

Note:

Default: Bitrate64kbps

Attention:

Some combinations of bitrate and samplerate are invalid based on the MP3 specification. Setting an invalid pair will cause the codec setup to fail.
For further details please see: `http://www.mpgedit.org/mpgedit/mpeg_format/mpeghdr.htm`
Validity can be determined by inspecting the bitrate and samplerate indices.

Definition at line 55 of file lame.h.

20.12.2.3 #define FE2_LAME_CHANNELS

Codec parameter for number of output channels.

Note:

Default: input number of channels as described by audio_options_SetStereo()
This parameter is temporary and will be removed when a proper filter is added in a future release

Deprecated

Use the Filter Interface along with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

Definition at line 97 of file lame.h.

20.12.2.4 #define FE2_LAME_QUALITY

Codec parameter for controlling LAME library's algorithm selection.

From `<lame/lame.h>`:

```
internal algorithm selection.  True quality is determined by the bitrate
but this variable will effect quality by selecting expensive or cheap
algorithms.
```

```
quality=0..9.  0=best (very slow).  9=worst.
recommended:  2     near-best quality, not too slow
              5     good quality, fast
              7     ok quality, really fast
```

Note:

Valid range: [0,9]
Default: 5

Definition at line 72 of file lame.h.

20.12.2.5 #define FE2_LAME_RC_MODE

Codec parameter for controlling LAME library's rate control method.

Note:

Valid values are defined by lame_rcmode_t
Default: LAME_CBR

Definition at line 77 of file lame.h.

20.12.3 Typedef Documentation

20.12.3.1 typedef enum lame_rcmode lame_rcmode_t

20.12.4 Enumeration Type Documentation

20.12.4.1 enum lame_rcmode

Rate control modes analogous to those found in `<lame/lame.h>`.

Enumerator:

LAME_CBR constant bitrate

LAME_ABR average bitrate, produces predictable size output w/improved quality over CBR

LAME_VBR_rh variable bitrate, also referred to as vbr-old/standard

LAME_VBR_mtrh variable bitrate, also referred to as vbr-new/fast (over twice as fast as vbr-old)

Definition at line 80 of file lame.h.

20.13 libvorbis - FFmpeg

20.13.1 Detailed Description

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_VORBIS);
// Use 128kbps
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VORBIS_BITRATE, 128);
```

Defines

- #define FE2_CODEC_VORBIS

  Vorbis. Codec name for use with Flix2_AddCodec().

- #define FE2_VORBIS_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

20.13.2 Define Documentation

20.13.2.1 #define FE2_CODEC_VORBIS

Vorbis. Codec name for use with Flix2_AddCodec().

Definition at line 36 of file vorbis.h.

20.13.2.2 #define FE2_VORBIS_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 64kbps

Definition at line 43 of file vorbis.h.

20.14 VP6

20.14.1 Detailed Description

The VP6 (Flash 8) video codec.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_VP6);
// Use 450kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP6_BITRATE, 450.0);
```

20.14.2 Defaults dependent on bits per pixel

Defaults for some encoder parameters are determined based on the bits per pixel, which is determined as follows:

$$bitsperpixel = \frac{FE2\_VP6\_BITRATE * 1024}{imagew * imageh * fps}$$

The current defaults are:

	<.03	<.07	<.12	<.20	>=.20
FE2_VP6_MIN_Q	25	15	15	5	5
FE2_VP6_MAX_Q	62	56	50	40	32
FE2_VP6_SHARPNESS	1	1	5	7	7
FE2_VP6_NOISE_REDUCTION	3	0	0	0	0
FE2_VP6_TEMPORAL_RESAMPLING	1	1	0	0	0

20.14.3 Notes on datarate control

Using the above defaults some material may fail to achieve the requested bitrate. This is highly material dependent, but the result will be a clip with a reduced datarate. If you are intent on achieving a specific datarate and find that your clips are missing it the following settings will help:

FE2_VP6_MIN_Q	4
FE2_VP6_NOISE_REDUCTION	0
FE2_VP6_TEMPORAL_RESAMPLING	1
FE2_VP6_STREAM_PREBUFFER	5
FE2_VP6_STREAM_OPTIMAL_BUFFER	5

Reasoning:

- FE2_VP6_MIN_Q setting - The default minimum quantizer settings are an attempt to save bits for harder sections. Unfortunately this means that on easy but high framerate and/or resolution clips the desired quality cannot be achieved. The downside to this change is that more frames may be dropped on harder sections.
- FE2_VP6_NOISE_REDUCTION setting - The usage of noise sensitivity set to non-0 is an attempt to save bits by blurring the source. Sometimes the material is so easy that this attempt to save bits may not only be unnecessary, it may actually hurt the visual quality by blurring. Again this may cause more frames to be dropped in harder sections.
- FE2_VP6_TEMPORAL_RESAMPLING setting - This parameter should be enabled to guarantee there are no extreme datarate spikes throughout the clip.

General VP6 settings

- enum FE2_CompressMode {
  COMPRESSMODE_GOOD,
  COMPRESSMODE_BEST }

  Valid compress modes for VP6, influences encoder speed.

- enum vp6profile {
  VP6_E,
  VP6_S }

  Valid profiles for use with the FE2_VP6_PROFILE parameter.

- typedef enum vp6profile vp6profile_t
- #define FE2_VP6_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_VP6_KFINTTYPE

  Alias for FE2_VCODECPARAM_KFINTTYPE.

- #define FE2_VP6_KFFREQ

  Alias for FE2_VCODECPARAM_KFFREQ.

- #define FE2_VP6_RC_MODE

  Alias for FE2_VCODECPARAM_RC_MODE.

- #define FE2_VP6_CXMODE

  VP6 compress mode.

- #define FE2_VP6_SHARPNESS

  Codec parameter for sharpness.

- #define FE2_VP6_NOISE_REDUCTION

  Codec parameter for noise reduction.

- #define FE2_VP6_PROFILE

  Codec parameter for encoding profile.

Codec name

- #define FE2_CODEC_VP6

  Codec name for use with Flix2_AddCodec().

Advanced VP6 settings

- #define FE2_VP6_CONCURRENCY

  Codec parameter for concurrency level.

- #define FE2_VP6_UNDERSHOOT_PCT

  Codec parameter for undershoot percentage.

- #define FE2_VP6_MIN_Q

  Codec parameter for minimum quantizer.

- #define FE2_VP6_MAX_Q

  Codec parameter for maximum quantizer.

- #define FE2_VP6_TEMPORAL_RESAMPLING

  Codec parameter for temporal resampling.

- #define FE2_VP6_TEMPORAL_DOWN_WATERMARK

  Codec parameter for temporal down watermark percentage.

CBR-Specific

- #define FE2_VP6_STREAM_PEAK_BITRATE

  The maximum bitrate allowed in the stream.

- #define FE2_VP6_STREAM_PREBUFFER

  Seconds of preload that are necessary before starting playback.

- #define FE2_VP6_STREAM_OPTIMAL_BUFFER

  Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

- #define FE2_VP6_STREAM_MAX_BUFFER

  The maximum size of the buffer, in seconds.

VBR-Specific

- #define FE2_VP6_2PASS_MIN_SECTION

  VBR_2PASSControl minimum section datarate

- #define FE2_VP6_2PASS_MAX_SECTION

  VBR_2PASSControl maximum section datarate

20.14.4 Define Documentation

20.14.4.1 #define FE2_CODEC_VP6

Codec name for use with Flix2_AddCodec().

Definition at line 84 of file vp6.h.

20.14.4.2 #define FE2_VP6_2PASS_MAX_SECTION

VBR_2PASSControl maximum section datarate

This value is given as a percentage. The highest datarate, i.e., *FE2_VP6_2PASS_MAX_SECTION*% × *FE2_VP6_BITRATE*, that can be streamed, and also the highest datarate that the encoder will allow, no matter how difficult the section is.

Note:

Default: 400

Definition at line 336 of file vp6.h.

20.14.4.3 #define FE2_VP6_2PASS_MIN_SECTION

VBR_2PASSControl minimum section datarate

This value is given as a percentage. The lowest datarate, i.e., *FE2_VP6_2PASS_MIN_SECTION*% × *FE2_VP6_BITRATE*, that the encoder will allow for any section, no matter how easy the section is. This value is used to prevent difficult sections from stealing too many bits from easy sections.

Note:

Default: 40

Definition at line 326 of file vp6.h.

20.14.4.4 #define FE2_VP6_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 124 of file vp6.h.

20.14.4.5 #define FE2_VP6_CONCURRENCY

Codec parameter for concurrency level.

Setting this value to 1 will allow the encode process to take advantage of multiple cores/processors yielding a potentially significant gain in performance.

Usage Considerations:

The current implementation parallelizes the video encode on keyframe boundaries. To achieve this, each instance (1 per core/processor) must buffer both the raw input frames as well as those encoded. As a result, the memory requirement for each session increases dramatically.
The number of raw (YV12) input frames required per processor is FE2_VP6_KFFREQ or 60 (the current allowed maximum), whichever is smaller. The size of each raw frame can be calculated as follows:

$$framesiz\_bytes = imagew \times imageh \times \frac{3}{2}$$

Making the input buffer requirements for each processor:

$$raw\_bufreq = framesiz\_bytes \times \min\{FE2\_VP6\_KFFREQ, 60\}$$

The encoded buffer requirement is the same, as this would be the worst (though unlikely) case, making the per-processor memory increase:

$$memreq\_increase/processor = 2 \times raw\_bufreq$$

The current maximum number of cores/processors utilized is 8 with the load evenly distributed across those available.
Because of this some thought should be given to queueing encodes to avoid possible failures due to exhaustion of available memory.
Using the above, with FE2_VP6_KFFREQ=60, the per-processor memory increase for some common resolutions are:

```
Resolution      |  framesiz_bytes    | memreq_increase/processor |   memreq_increase/2    |   memreq_
=================================================================================================
320x240   (QVGA) | 115200   (.11MiB)  | 13824000   (13.18MiB)     | 27648000   (26.37MiB)  | 55296000
640x480   (VGA)  | 460800   (.44MiB)  | 55296000   (52.73MiB)     | 110592000 (105.47MiB)  | 221184000
720x480   (NTSC) | 518400   (.49MiB)  | 62208000   (59.33MiB)     | 124416000 (118.65MiB)  | 248832000
720x576   (PAL)  | 622080   (.59MiB)  | 74649600   (71.19MiB)     | 149299200 (142.38MiB)  | 298598400
1280x720  (720p) | 1382400  (1.32MiB) | 165888000  (158.2MiB)     | 331776000 (316.41MiB)  | 663552000
```

Note:

Default: 0 (disabled)

Attention:

Currently to enable this parameter:

- FE2_VP6_KFINTTYPE MUST be set to FIXED_KEYFRAMES.
- FE2_VP6_RC_MODE MUST be set to VBR_1PASSControl or CBR_1PASSControl.

Enabling this parameter on a processor that is solely Hyper-Threaded (e.g., early Pentium 4 models) MAY see a **decrease** in performance, especially under Linux.

See also:

FE2_VP6_CONCURRENCY Performance - Graphs comparing encode times with and without this parameter enabled.

Definition at line 228 of file vp6.h.

20.14.4.6 #define FE2_VP6_CXMODE

VP6 compress mode.

Valid values are defined by FE2_CompressMode

Note:

Default: COMPRESSMODE_GOOD

Definition at line 149 of file vp6.h.

20.14.4.7 #define FE2_VP6_KFFREQ

Alias for FE2_VCODECPARAM_KFFREQ.

Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of FE2_VCODECPARAM_-KFINTTYPE

See also:

FE2_VideoKeyframeTypes

Note:

Default: $12.0 \times output_{fps}$ or 360 frames if the framerate is unknown

Definition at line 137 of file vp6.h.

20.14.4.8 #define FE2_VP6_KFINTTYPE

Alias for FE2_VCODECPARAM_KFINTTYPE.

Video codec parameter for the keyframe interval type.

Valid values are defined by FE2_VideoKeyframeTypes.

Note:

Default: MAX_KEYFRAMES

Definition at line 130 of file vp6.h.

20.14.4.9 #define FE2_VP6_MAX_Q

Codec parameter for maximum quantizer.

Determines the quality of the output. A lower maximum number produces higher quality output.

Note:

Valid range: [0,63]
Default: Bits per pixel dependent

Attention:

Setting this value too low will likely cause the encoder to miss (i.e., overshoot) the target datarate specified by FE2_VP6_BITRATE and, should FE2_VP6_TEMPORAL_RESAMPLING be enabled, result in a large amount of dropped frames. The table mentioned above should be used as a guide in choosing a reasonable value based on the bitrate.

Definition at line 258 of file vp6.h.

20.14.4.10 #define FE2_VP6_MIN_Q

Codec parameter for minimum quantizer.

Determines the quality of the output. A lower minimum number produces higher quality output.

Note:

Valid range: [0,63]
Default: Bits per pixel dependent

Definition at line 244 of file vp6.h.

20.14.4.11 #define FE2_VP6_NOISE_REDUCTION

Codec parameter for noise reduction.

Determines the level of noise filtering to apply in the preprocessor. 0 is no preprocessing, 6 is extreme preprocessing.

Note:

Valid range: [0,6]
Setting this value to anything but 0 will result in slowing down the compression speed.
Default: Bits per pixel dependent

Definition at line 170 of file vp6.h.

20.14.4.12 #define FE2_VP6_PROFILE

Codec parameter for encoding profile.

Note:

Valid values are defined by vp6profile_t Default: VP6_E

Definition at line 175 of file vp6.h.

20.14.4.13 #define FE2_VP6_RC_MODE

Alias for FE2_VCODECPARAM_RC_MODE.

Video codec parameter for the rate control mode.

Valid values are defined by FE2_VideoBitrateControls.

Note:

Default: VBR_2PASSControl

Definition at line 143 of file vp6.h.

20.14.4.14 #define FE2_VP6_SHARPNESS

Codec parameter for sharpness.

Controls the sharpness of the image in the output. This setting does not impact any other setting and is largely a matter of personal preference. A low sharpness setting will result in fewer visible artifacts but may blur the image somewhat; a high sharpness will result in a sharper image but may result in more visible artifacts.

Note:

Valid range: [0,7]
Default: Bits per pixel dependent

Definition at line 160 of file vp6.h.

20.14.4.15 #define FE2_VP6_STREAM_MAX_BUFFER

The maximum size of the buffer, in seconds.

Attention:

Valid for CBR only

Definition at line 311 of file vp6.h.

20.14.4.16 #define FE2_VP6_STREAM_OPTIMAL_BUFFER

Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

Note:

Default: 10

Attention:

Valid for CBR only

Definition at line 307 of file vp6.h.

20.14.4.17 #define FE2_VP6_STREAM_PEAK_BITRATE

The maximum bitrate allowed in the stream.

This value is given as a percentage. The peak bitrate is calculated as follows: *FE2_VP6_STREAM_PEAK_BITRATE*% $\times$ *FE2_VP6_BITRATE*

Note:

Default: 100

Attention:

Valid for CBR only

Definition at line 293 of file vp6.h.

20.14.4.18 #define FE2_VP6_STREAM_PREBUFFER

Seconds of preload that are necessary before starting playback.

The buffer is used to maintain a consistent datarate and minimize playback interruption.

Note:

Default: 6

Attention:

Valid for CBR only

Definition at line 301 of file vp6.h.

20.14.4.19 #define FE2_VP6_TEMPORAL_DOWN_WATERMARK

Codec parameter for temporal down watermark percentage.

Specifies the percentage of the datarate buffer remaining below which the encoder is allowed to start dropping frames. Only used if FE2_VP6_TEMPORAL_RESAMPLING is enabled.

Note:

Default: 20
A larger percentage will make it more likely frames will be dropped to achieve the requested FE2_VP6_BITRATE. The converse is also true.

Definition at line 281 of file vp6.h.

20.14.4.20 #define FE2_VP6_TEMPORAL_RESAMPLING

Codec parameter for temporal resampling.

In particularly difficult regions, if enabled, the encoder will drop frames to achieve the target the data rate. The value of FE2_VP6_TEMPORAL_DOWN_WATERMARK determines the percentage of the datarate buffer below which the encoder is enabled to start dropping frames.

Note:

Default: Bits per pixel dependent

Attention:

Setting FE2_VP6_RC_MODE to CBR_1PASSControl or CBR_2PASSControl will unconditionally set this value to 1. In this case adjusting FE2_VP6_TEMPORAL_DOWN_WATERMARK may improve results.

Definition at line 271 of file vp6.h.

20.14.4.21 #define FE2_VP6_UNDERSHOOT_PCT

Codec parameter for undershoot percentage.

This value is given as a percentage. Creates output that targets a slightly lower datarate so there are bits available in the buffer to improve difficult sections.

Note:

Default: 90

Definition at line 236 of file vp6.h.

20.14.5 Typedef Documentation

20.14.5.1 typedef enum vp6profile vp6profile_t

20.14.6 Enumeration Type Documentation

20.14.6.1 enum FE2_CompressMode

Valid compress modes for VP6, influences encoder speed.

For use in calls to Flix2_CodecSetParam() and Flix2_CodecGetParam() for the FE2_VP6_CXMODE and FE2_VP6A_CXMODE. For general transcoding (i.e. decoding from one compressed format and encoding to VP6) `COMPRESSMODE_GOOD` will provide adequate results and the best possible encode time. The quality gain using `COMPRESSMODE_BEST` will only be visible when using clean raw source or compressing to extremely low bitrates.

Enumerator:

COMPRESSMODE_GOOD The encoder balances quality with the amount of time it takes to encode the output with the goal of making the encoding as fast as possible without losing too much quality. This is the default.

COMPRESSMODE_BEST The encoder places priority on the quality of the output over encoding speed. The output is compressed at the highest possible quality. This option takes the longest amount of time to encode.

Definition at line 100 of file vp6.h.

20.14.6.2 enum vp6profile

Valid profiles for use with the FE2_VP6_PROFILE parameter.

Enumerator:

VP6_E Default profile used in flash video encoding

VP6_S Profile optimized for use with high resolution/datarate video. Decreases decoding complexity allowing playback on resource constrained machines.

Definition at line 113 of file vp6.h.

20.15 VP6 with Alpha

20.15.1 Detailed Description

The VP6 codec with support for an alpha channel.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_VP6ALPHA);
//Use 380kbit for the video ..
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP6A_BITRATE, 380.0);
//..and 68kbit for the alpha channel
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP6A_ALPHA_BITRATE, 68.0);
```

As mentioned in the VP6 section Defaults dependent on bits per pixel some of the codec parameters are determined by the bits per pixel in the output. Currently the alpha channel parameters, e.g. FE2_VP6A_-ALPHA_MIN_Q, match their VP6 equivalents.

Codec name

- #define FE2_CODEC_VP6ALPHA

  Codec name for use with Flix2_AddCodec().

General VP6 settings

- #define FE2_VP6A_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_VP6A_ALPHA_BITRATE

  Compressed bitrate of the alpha channel in kbit/s.

- #define FE2_VP6A_KFINTTYPE

  Alias for FE2_VP6_KFINTTYPE.

- #define FE2_VP6A_KFFREQ

  Alias for FE2_VP6_KFFREQ.

- #define FE2_VP6A_RC_MODE

  Alias for FE2_VCODECPARAM_RC_MODE.

- #define FE2_VP6A_CXMODE

  Alias for FE2_VP6_CXMODE.

- #define FE2_VP6A_SHARPNESS

  Alias for FE2_VP6_SHARPNESS.

- #define FE2_VP6A_ALPHA_SHARPNESS

  Sharpness for the alpha channel.

- #define FE2_VP6A_NOISE_REDUCTION

  Alias for FE2_VP6_NOISE_REDUCTION.

- #define FE2_VP6A_ALPHA_NOISE_REDUCTION

  Noise reduction for the alpha channel.

Advanced VP6 settings

- #define FE2_VP6A_UNDERSHOOT_PCT

  Alias for FE2_VP6_UNDERSHOOT_PCT.

- #define FE2_VP6A_MIN_Q

  Alias for FE2_VP6_MIN_Q.

- #define FE2_VP6A_ALPHA_MIN_Q

  Minimum quantizer for the alpha channel.

- #define FE2_VP6A_MAX_Q

  Alias for FE2_VP6_MAX_Q.

- #define FE2_VP6A_ALPHA_MAX_Q

  Maximum quantizer for the alpha channel.

- #define FE2_VP6A_TEMPORAL_RESAMPLING

  Alias for FE2_VP6_TEMPORAL_RESAMPLING.

- #define FE2_VP6A_TEMPORAL_DOWN_WATERMARK

  Alias for FE2_VP6_TEMPORAL_DOWN_WATERMARK.

- #define FE2_VP6A_STREAM_PEAK_BITRATE

  Alias for FE2_VP6_STREAM_PEAK_BITRATE.

- #define FE2_VP6A_STREAM_PREBUFFER

  Alias for FE2_VP6_STREAM_PREBUFFER.

- #define FE2_VP6A_STREAM_OPTIMAL_BUFFER

  Alias for FE2_VP6_STREAM_OPTIMAL_BUFFER.

- #define FE2_VP6A_STREAM_MAX_BUFFER

  Alias for FE2_VP6_STREAM_MAX_BUFFER.

- #define FE2_VP6A_2PASS_MIN_SECTION

  Alias for FE2_VP6_2PASS_MIN_SECTION.

- #define FE2_VP6A_2PASS_MAX_SECTION

  Alias for FE2_VP6_2PASS_MAX_SECTION.

20.15.2 Define Documentation

20.15.2.1 #define FE2_CODEC_VP6ALPHA

Codec name for use with Flix2_AddCodec().

Definition at line 56 of file vp6_alpha.h.

20.15.2.2 #define FE2_VP6A_2PASS_MAX_SECTION

Alias for FE2_VP6_2PASS_MAX_SECTION.

VBR_2PASSControl maximum section datarate

This value is given as a percentage. The highest datarate, i.e., *FE2_VP6_2PASS_MAX_SECTION*% × *FE2_VP6_BITRATE*, that can be streamed, and also the highest datarate that the encoder will allow, no matter how difficult the section is.

Note:

Default: 400

Definition at line 202 of file vp6_alpha.h.

20.15.2.3 #define FE2_VP6A_2PASS_MIN_SECTION

Alias for FE2_VP6_2PASS_MIN_SECTION.

VBR_2PASSControl minimum section datarate

This value is given as a percentage. The lowest datarate, i.e., *FE2_VP6_2PASS_MIN_SECTION*% × *FE2_VP6_BITRATE*, that the encoder will allow for any section, no matter how easy the section is. This value is used to prevent difficult sections from stealing too many bits from easy sections.

Note:

Default: 40

Definition at line 196 of file vp6_alpha.h.

20.15.2.4 #define FE2_VP6A_ALPHA_BITRATE

Compressed bitrate of the alpha channel in kbit/s.

Note:

Default: 68 (15% of default 448kbps)

Definition at line 74 of file vp6_alpha.h.

20.15.2.5 #define FE2_VP6A_ALPHA_MAX_Q

Maximum quantizer for the alpha channel.

See also:

FE2_VP6A_MAX_Q

Definition at line 154 of file vp6_alpha.h.

20.15.2.6 #define FE2_VP6A_ALPHA_MIN_Q

Minimum quantizer for the alpha channel.

See also:

FE2_VP6A_MIN_Q

Definition at line 143 of file vp6_alpha.h.

20.15.2.7 #define FE2_VP6A_ALPHA_NOISE_REDUCTION

Noise reduction for the alpha channel.

See also:

FE2_VP6A_NOISE_REDUCTION

Definition at line 120 of file vp6_alpha.h.

20.15.2.8 #define FE2_VP6A_ALPHA_SHARPNESS

Sharpness for the alpha channel.

See also:

FE2_VP6A_SHARPNESS

Definition at line 109 of file vp6_alpha.h.

20.15.2.9 #define FE2_VP6A_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 380

Definition at line 69 of file vp6_alpha.h.

20.15.2.10 #define FE2_VP6A_CXMODE

Alias for FE2_VP6_CXMODE.

VP6 compress mode.

Valid values are defined by FE2_CompressMode

Note:

Default: COMPRESSMODE_GOOD

Definition at line 98 of file vp6_alpha.h.

20.15.2.11 #define FE2_VP6A_KFFREQ

Alias for FE2_VP6_KFFREQ.

Alias for FE2_VCODECPARAM_KFFREQ.

Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of FE2_VCODECPARAM_-KFINTTYPE

See also:

FE2_VideoKeyframeTypes

Note:

Default: $12.0 \times output_{fps}$ or 360 frames if the framerate is unknown

Definition at line 86 of file vp6_alpha.h.

20.15.2.12 #define FE2_VP6A_KFINTTYPE

Alias for FE2_VP6_KFINTTYPE.

Alias for FE2_VCODECPARAM_KFINTTYPE.

Video codec parameter for the keyframe interval type.

Valid values are defined by FE2_VideoKeyframeTypes.

Note:

Default: MAX_KEYFRAMES

Definition at line 80 of file vp6_alpha.h.

20.15.2.13 #define FE2_VP6A_MAX_Q

Alias for FE2_VP6_MAX_Q.

Codec parameter for maximum quantizer.

Determines the quality of the output. A lower maximum number produces higher quality output.

Note:

Valid range: [0,63]
Default: Bits per pixel dependent

Attention:

Setting this value too low will likely cause the encoder to miss (i.e., overshoot) the target datarate specified by FE2_VP6_BITRATE and, should FE2_VP6_TEMPORAL_RESAMPLING be enabled, result in a large amount of dropped frames. The table mentioned above should be used as a guide in choosing a reasonable value based on the bitrate.

Definition at line 149 of file vp6_alpha.h.

20.15.2.14 #define FE2_VP6A_MIN_Q

Alias for FE2_VP6_MIN_Q.

Codec parameter for minimum quantizer.

Determines the quality of the output. A lower minimum number produces higher quality output.

Note:

Valid range: [0,63]
Default: Bits per pixel dependent

Definition at line 138 of file vp6_alpha.h.

20.15.2.15 #define FE2_VP6A_NOISE_REDUCTION

Alias for FE2_VP6_NOISE_REDUCTION.

Codec parameter for noise reduction.

Determines the level of noise filtering to apply in the preprocessor. 0 is no preprocessing, 6 is extreme preprocessing.

Note:

Valid range: [0,6]
Setting this value to anything but 0 will result in slowing down the compression speed.
Default: Bits per pixel dependent

Definition at line 115 of file vp6_alpha.h.

20.15.2.16 #define FE2_VP6A_RC_MODE

Alias for FE2_VCODECPARAM_RC_MODE.

Note:

Default: VBR_2PASSControl

Definition at line 92 of file vp6_alpha.h.

20.15.2.17 #define FE2_VP6A_SHARPNESS

Alias for FE2_VP6_SHARPNESS.

Codec parameter for sharpness.

Controls the sharpness of the image in the output. This setting does not impact any other setting and is largely a matter of personal preference. A low sharpness setting will result in fewer visible artifacts but may blur the image somewhat; a high sharpness will result in a sharper image but may result in more visible artifacts.

Note:

Valid range: [0,7]
Default: Bits per pixel dependent

Definition at line 104 of file vp6_alpha.h.

20.15.2.18 #define FE2_VP6A_STREAM_MAX_BUFFER

Alias for FE2_VP6_STREAM_MAX_BUFFER.

The maximum size of the buffer, in seconds.

Attention:

Valid for CBR only

Definition at line 190 of file vp6_alpha.h.

20.15.2.19 #define FE2_VP6A_STREAM_OPTIMAL_BUFFER

Alias for FE2_VP6_STREAM_OPTIMAL_BUFFER.

Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

Note:

Default: 10

Attention:

Valid for CBR only

Definition at line 184 of file vp6_alpha.h.

20.15.2.20 #define FE2_VP6A_STREAM_PEAK_BITRATE

Alias for FE2_VP6_STREAM_PEAK_BITRATE.

The maximum bitrate allowed in the stream.

This value is given as a percentage. The peak bitrate is calculated as follows: $FE2\_VP6\_STREAM\_PEAK\_BITRATE\% \times FE2\_VP6\_BITRATE$

Note:

Default: 100

Attention:

Valid for CBR only

Definition at line 172 of file vp6_alpha.h.

20.15.2.21 #define FE2_VP6A_STREAM_PREBUFFER

Alias for FE2_VP6_STREAM_PREBUFFER.

Seconds of preload that are necessary before starting playback.

The buffer is used to maintain a consistent datarate and minimize playback interruption.

Note:

Default: 6

Attention:

Valid for CBR only

Definition at line 178 of file vp6_alpha.h.

20.15.2.22 #define FE2_VP6A_TEMPORAL_DOWN_WATERMARK

Alias for FE2_VP6_TEMPORAL_DOWN_WATERMARK.

Codec parameter for temporal down watermark percentage.

Specifies the percentage of the datarate buffer remaining below which the encoder is allowed to start dropping frames. Only used if FE2_VP6_TEMPORAL_RESAMPLING is enabled.

Note:

Default: 20
A larger percentage will make it more likely frames will be dropped to achieve the requested FE2_VP6_BITRATE. The converse is also true.

Definition at line 166 of file vp6_alpha.h.

20.15.2.23 #define FE2_VP6A_TEMPORAL_RESAMPLING

Alias for FE2_VP6_TEMPORAL_RESAMPLING.

Codec parameter for temporal resampling.

In particularly difficult regions, if enabled, the encoder will drop frames to achieve the target the data rate. The value of FE2_VP6_TEMPORAL_DOWN_WATERMARK determines the percentage of the datarate buffer below which the encoder is enabled to start dropping frames.

Note:

Default: Bits per pixel dependent

Attention:

Setting FE2_VP6_RC_MODE to CBR_1PASSControl or CBR_2PASSControl will unconditionally set this value to 1. In this case adjusting FE2_VP6_TEMPORAL_DOWN_WATERMARK may improve results.

Definition at line 160 of file vp6_alpha.h.

20.15.2.24 #define FE2_VP6A_UNDERSHOOT_PCT

Alias for FE2_VP6_UNDERSHOOT_PCT.

Codec parameter for undershoot percentage.

This value is given as a percentage. Creates output that targets a slightly lower datarate so there are bits available in the buffer to improve difficult sections.

Note:

Default: 90

Definition at line 132 of file vp6_alpha.h.

20.16 VP8

20.16.1 Detailed Description

The VP8 video codec.

Example Usage:

```
sc = Flix2_AddCodec(&codec, flix, FE2_CODEC_VP8);
// Use 450kbit
if(sc == ON2_OK)
    sc = Flix2_CodecSetParam(codec, FE2_VP8_BITRATE, 450.0);
```

Codec name

- #define FE2_CODEC_VP8

  Codec name for use with Flix2_AddCodec().

General VP8 settings

- #define FE2_VP8_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_VP8_KFINTTYPE

  Alias for FE2_VCODECPARAM_KFINTTYPE.

- #define FE2_VP8_KFFREQ

  Alias for FE2_VCODECPARAM_KFFREQ.

- #define FE2_VP8_RC_MODE

  Alias for FE2_VCODECPARAM_RC_MODE.

- #define FE2_VP8_CXMODE

  VP8 compress mode.

- #define FE2_VP8_SHARPNESS

  Codec parameter for sharpness.

- #define FE2_VP8_NOISE_REDUCTION

  Controls encoder noise reduction preprocessing.

Advanced VP8 settings

- #define FE2_VP8_UNDERSHOOT_PCT

  Codec parameter for undershoot percentage.

- #define FE2_VP8_OVERSHOOT_PCT

  Codec parameter for undershoot percentage.

- #define FE2_VP8_MIN_Q

  Codec parameter for minimum quantizer.

- #define FE2_VP8_MAX_Q

  Codec parameter for maximum quantizer.

- #define FE2_VP8_DROP_THRESH

  Threshold controlling encoder frame dropping.

CBR-Specific

- #define FE2_VP8_STREAM_INITIAL_BUFFER

  Seconds of preload that are necessary before starting playback.

- #define FE2_VP8_STREAM_OPTIMAL_BUFFER

  Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

- #define FE2_VP8_STREAM_MAX_BUFFER

  The maximum size of the buffer, in seconds.

VBR-Specific

- #define FE2_VP8_2PASS_MIN_SECTION

  VBR_2PASSControl minimum section datarate

- #define FE2_VP8_2PASS_MAX_SECTION

  VBR_2PASSControl maximum section datarate

Defines

- #define FE2_VP8_PROFILE

  Bitstream profile.

- #define FE2_VP8_ALTREF

  Enable the use of alternate reference frames.

- #define FE2_VP8_AR_MAX_FRAMES

  Max number of frames blurred creating alternate reference.

- #define FE2_VP8_AR_TYPE

  Filter type to use w/alternate reference.

- #define FE2_VP8_AR_STRENGTH

  Filter strength for the alternate reference.

- #define FE2_VP8_MB_STATIC_THRESHOLD

  Threshold for macroblocks treated static.

- #define FE2_VP8_TOKEN_PARTITIONS

  Number of token partitions.

- #define FE2_VP8_LAG

  Allow lagged encoding If set, this value allows the encoder to consume a number of input frames before producing output frames. This allows the encoder to base decisions for the current frame on future frames.

- #define FE2_VP8_THREADS

  Number of threads to use A reasonable selection would be the number of cores on the system.

20.16.2 Define Documentation

20.16.2.1 #define FE2_CODEC_VP8

Codec name for use with Flix2_AddCodec().

Definition at line 40 of file vp8.h.

20.16.2.2 #define FE2_VP8_2PASS_MAX_SECTION

VBR_2PASSControl maximum section datarate

This value is given as a percentage. The highest datarate, i.e., *FE2_VP8_2PASS_MAX_SECTION*% × *FE2_VP8_BITRATE*, that can be streamed, and also the highest datarate that the encoder will allow, no matter how difficult the section is.

Note:

Default: 400

Definition at line 198 of file vp8.h.

20.16.2.3 #define FE2_VP8_2PASS_MIN_SECTION

VBR_2PASSControl minimum section datarate

This value is given as a percentage. The lowest datarate, i.e., *FE2_VP8_2PASS_MIN_SECTION*% × *FE2_VP8_BITRATE*, that the encoder will allow for any section, no matter how easy the section is. This value is used to prevent difficult sections from stealing too many bits from easy sections.

Note:

Default: 40

Definition at line 188 of file vp8.h.

20.16.2.4 #define FE2_VP8_ALTREF

Enable the use of alternate reference frames.

Note:

Default: 0

Definition at line 211 of file vp8.h.

20.16.2.5 #define FE2_VP8_AR_MAX_FRAMES

Max number of frames blurred creating alternate reference.

Note:

Valid range: [0,25]
Default: 0

Definition at line 216 of file vp8.h.

20.16.2.6 #define FE2_VP8_AR_STRENGTH

Filter strength for the alternate reference.

Note:

Valid range: [0,6]
Default: 0

Definition at line 225 of file vp8.h.

20.16.2.7 #define FE2_VP8_AR_TYPE

Filter type to use w/alternate reference.

Note:

Default: 0

Definition at line 220 of file vp8.h.

20.16.2.8 #define FE2_VP8_BITRATE

Alias for FE2_CODECPARAM_BITRATE.

Codec parameter for stream bitrate.

Compressed stream bitrate in kbits/sec.

Note:

Default: 448

Definition at line 52 of file vp8.h.

20.16.2.9 #define FE2_VP8_CXMODE

VP8 compress mode.

Valid values are defined by FE2_CompressMode

Note:

Default: COMPRESSMODE_GOOD

Definition at line 77 of file vp8.h.

20.16.2.10 #define FE2_VP8_DROP_THRESH

Threshold controlling encoder frame dropping.

A value of 0 disables frame dropping. Larger values will increase the likelihood frames will be dropped to achieve data rate constraints. Recommended values are 0 for VBR mode and 70 for CBR mode.

Note:

Valid range: [0,100]
Default: 0

Definition at line 151 of file vp8.h.

20.16.2.11 #define FE2_VP8_KFFREQ

Alias for FE2_VCODECPARAM_KFFREQ.

Video codec parameter for keyframe frequency.

Value is in video frames. The interpretation depends on the setting of FE2_VCODECPARAM_-KFINTTYPE

See also:

FE2_VideoKeyframeTypes

Note:

Default: $12.0 \times output_fps$ or 360 frames if the framerate is unknown

Definition at line 65 of file vp8.h.

20.16.2.12 #define FE2_VP8_KFINTTYPE

Alias for FE2_VCODECPARAM_KFINTTYPE.

Video codec parameter for the keyframe interval type.

Valid values are defined by FE2_VideoKeyframeTypes.

Note:

Default: MAX_KEYFRAMES

Definition at line 58 of file vp8.h.

20.16.2.13 #define FE2_VP8_LAG

Allow lagged encoding If set, this value allows the encoder to consume a number of input frames before producing output frames. This allows the encoder to base decisions for the current frame on future frames.

Note:

Valid range: [0,25]
Default: 0

Definition at line 245 of file vp8.h.

20.16.2.14 #define FE2_VP8_MAX_Q

Codec parameter for maximum quantizer.

Determines the quality of the output. A lower maximum number produces higher quality output.

Note:

Valid range: [0,63]
Default: 63

Attention:

Setting this value too low will likely cause the encoder to miss (i.e., overshoot) the target datarate specified by FE2_VP8_BITRATE and, should FE2_VP8_DROP_THRESH be enabled, result in a large amount of dropped frames. The table mentioned above should be used as a guide in choosing a reasonable value based on the bitrate.

Definition at line 140 of file vp8.h.

20.16.2.15 #define FE2_VP8_MB_STATIC_THRESHOLD

Threshold for macroblocks treated static.

Note:

Default: 0

Definition at line 229 of file vp8.h.

20.16.2.16 #define FE2_VP8_MIN_Q

Codec parameter for minimum quantizer.

Determines the quality of the output. A lower minimum number produces higher quality output.

Note:

Valid range: [0,63]
Default: 4

Definition at line 126 of file vp8.h.

20.16.2.17 #define FE2_VP8_NOISE_REDUCTION

Controls encoder noise reduction preprocessing.

Determines the level of noise filtering to apply in the preprocessor. 0 is no preprocessing, 6 is extreme preprocessing.

Note:

Valid range: [0,6]
Setting this value to anything but 0 will result in slowing down the compression speed.
Default:

Definition at line 98 of file vp8.h.

20.16.2.18 #define FE2_VP8_OVERSHOOT_PCT

Codec parameter for undershoot percentage.

This value is given as a percentage.

Note:

Default: 200

Definition at line 118 of file vp8.h.

20.16.2.19 #define FE2_VP8_PROFILE

Bitstream profile.

Larger values may be more appropriate for HD material when targeting playback on lower end machines. Similar to VP6_S

Note:

Valid range: [0,3]
Default: 0

Definition at line 207 of file vp8.h.

20.16.2.20 #define FE2_VP8_RC_MODE

Alias for FE2_VCODECPARAM_RC_MODE.

Video codec parameter for the rate control mode.

Valid values are defined by FE2_VideoBitrateControls.

Note:

Default: VBR_2PASSControl

Definition at line 71 of file vp8.h.

20.16.2.21 #define FE2_VP8_SHARPNESS

Codec parameter for sharpness.

Controls the sharpness of the image in the output. This setting does not impact any other setting and is largely a matter of personal preference. A low sharpness setting will result in fewer visible artifacts but may blur the image somewhat; a high sharpness will result in a sharper image but may result in more visible artifacts.

Note:

Valid range: [0,7]
Default: 0

Definition at line 88 of file vp8.h.

20.16.2.22 #define FE2_VP8_STREAM_INITIAL_BUFFER

Seconds of preload that are necessary before starting playback.

The buffer is used to maintain a consistent datarate and minimize playback interruption.

Note:

Default: 6

Attention:

Valid for CBR only

Definition at line 163 of file vp8.h.

20.16.2.23 #define FE2_VP8_STREAM_MAX_BUFFER

The maximum size of the buffer, in seconds.

Attention:

Valid for CBR only

Definition at line 173 of file vp8.h.

20.16.2.24 #define FE2_VP8_STREAM_OPTIMAL_BUFFER

Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.

Note:

Default: 10

Attention:

Valid for CBR only

Definition at line 169 of file vp8.h.

20.16.2.25 #define FE2_VP8_THREADS

Number of threads to use A reasonable selection would be the number of cores on the system.

Note:

Default: 1

Definition at line 250 of file vp8.h.

20.16.2.26 #define FE2_VP8_TOKEN_PARTITIONS

Number of token partitions.

This defines VP8 partitioning mode for compressed data, i.e., the number of sub-streams in the bitstream. Used for parallelized decoding.

Note:

Valid range: {1,2,4,8}
Default: 1

Definition at line 237 of file vp8.h.

20.16.2.27 #define FE2_VP8_UNDERSHOOT_PCT

Codec parameter for undershoot percentage.

This value is given as a percentage. Creates output that targets a slightly lower datarate so there are bits available in the buffer to improve difficult sections.

Note:

Default: 95

Definition at line 112 of file vp8.h.

20.17 Encoding Statistics

Functions

- on2sc encoding_status_GetAverageBitrate (const FLIX2HANDLE flix, int32_t *pBitrate)

  Retrieve the encoder's average (video) bitrate.

- on2sc encoding_status_GetAverageFramesize (const FLIX2HANDLE flix, int32_t *pFramesize)

  Retrieve the encoder's average (video) frame size.

- on2sc encoding_status_GetMaximumFramesize (const FLIX2HANDLE flix, int32_t *pMaxFramesize)

  Retrieve the encoder's maximum (video) frame size.

- on2sc encoding_status_GetMinimumFramesize (const FLIX2HANDLE flix, int32_t *pMinFramesize)

  Retrieve the encoder's minimum (video) frame size.

- on2sc encoding_status_GetTotalFrames (const FLIX2HANDLE flix, int32_t *pTotalFrames)

  Retrieve the total number of (video) frames encoded.

- on2sc encoding_status_GetElapsedTime (const FLIX2HANDLE flix, int32_t *pElapsedTime)

  Retrieve the total elapsed encode time.

- on2sc encoding_status_GetEndTime (const FLIX2HANDLE flix, int32_t *pEndTime)

  Retrieve the encode completion time.

- on2sc encoding_status_GetStartTime (const FLIX2HANDLE flix, int32_t *pStartTime)

  Retrieve the encode start time.

- on2sc encoding_status_PercentComplete (const FLIX2HANDLE flix, int32_t *percent)

  Retrieve the percent of the encode complete.

20.17.1 Function Documentation

20.17.1.1 on2sc encoding_status_GetAverageBitrate (const FLIX2HANDLE *flix*, int32_t * *pBitrate*)

Retrieve the encoder's average (video) bitrate.

Value is in bits per second (bps).

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pBitrate*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pBitrate is not `NULL`

20.17.1.2 on2sc encoding_status_GetAverageFramesize (const FLIX2HANDLE *flix*, int32_t * *pFramesize*)

Retrieve the encoder's average (video) frame size.

Value is in bytes.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pFramesize*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pFramesize is not `NULL`

20.17.1.3 on2sc encoding_status_GetElapsedTime (const FLIX2HANDLE *flix*, int32_t * *pElapsedTime*)

Retrieve the total elapsed encode time.

Value is in milliseconds.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pElapsedTime*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pElapsedTime is not `NULL`

20.17.1.4 on2sc encoding_status_GetEndTime (const FLIX2HANDLE *flix*, int32_t * *pEndTime*)

Retrieve the encode completion time.

Value is in milliseconds.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pEndTime*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pEndTime is not `NULL`

20.17.1.5 on2sc encoding_status_GetMaximumFramesize (const FLIX2HANDLE *flix*, int32_t * *pMaxFramesize*)

Retrieve the encoder's maximum (video) frame size.

Value is in bytes.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pMaxFramesize*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pMaxFramesize is not `NULL`

20.17.1.6 on2sc encoding_status_GetMinimumFramesize (const FLIX2HANDLE *flix*, int32_t * *pMinFramesize*)

Retrieve the encoder's minimum (video) frame size.

Value is in bytes

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pMinFramesize*** Storage location to receive the result

Return values:

ON2_OK on success
ON2_INVALID_PARAMS should one or more of the preconditions fail
ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pMinFramesize is not `NULL`

20.17.1.7 on2sc encoding_status_GetStartTime (const FLIX2HANDLE *flix*, int32_t * *pStartTime*)

Retrieve the encode start time.

Value is in milliseconds.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()
→ ***pStartTime*** Storage location to receive the result

Return values:

ON2_OK on success
ON2_INVALID_PARAMS should one or more of the preconditions fail
ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pStartTime is not `NULL`

20.17.1.8 on2sc encoding_status_GetTotalFrames (const FLIX2HANDLE *flix*, int32_t * *pTotalFrames*)

Retrieve the total number of (video) frames encoded.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()
→ ***pTotalFrames*** Storage location to receive the result

Return values:

ON2_OK on success
ON2_INVALID_PARAMS should one or more of the preconditions fail
ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pTotalFrames is not `NULL`

20.17.1.9 on2sc encoding_status_PercentComplete (const FLIX2HANDLE *flix*, int32_t * *percent*)

Retrieve the percent of the encode complete.

Percent is returned as a whole number, e.g. 45.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***percent*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
percent is not `NULL`

Note:

If Flix2_GetSourceDuration() returns -1, percent complete will remain 0 until the first pass of the encode completes.
This value is only valid after Flix2_Encode() has been called.

20.18 Filters

20.18.1 Detailed Description

The Flix API supports a variety of filters. Filters are used to apply user controllable effects to the source material. For example, these include resizing, color adjustment, frame rate conversion, cuts, overlays, etc. All effect filters in Flix will be configured using this API. In a future release, all existing features will be available via this interface as well.

For those filters which are configurable both by this interface and individual functions (e.g., video_-options_SetXXX), this method is preferred.

Filters are generally processed in a fixed order, regardless of the order they are loaded in an encoding session. The order is:

- Video Filters
  - Preprocessing Filters
    1. Cut
  - Processing Filters
    1. Crop
    2. Deinterlace
    3. Framerate
    4. Denoise
    5. Blur/Sharpen
    6. Mirror
    7. Rotate
    8. Scale
    9. Brightness/Contrast/Hue/Saturation
  - Postprocessing Filters
    1. Overlay
    2. PNG Exporter
- Audio Filters
  - Preprocessing Filters
    1. Cut
  - Processing Filters
    1. Lowpass/Highpass
    2. Resample

Modules

- Video Filters
- Audio Filters
- Cut

20.19 Video Filters

Modules

- Deinterlace
- Brightness/Contrast/Hue/Saturation
- Blur
- Crop
- Denoise
- Frame Rate
- Mirror
- Overlay (Watermark)
- PNG Image Export (Thumbnail)
- Rotate
- Scale
- Sharpen

20.20 Audio Filters

Modules

- Highpass
- Lowpass
- Resample

20.21 Deinterlace

20.21.1 Detailed Description

The adaptive deinterlace filter implements a deinterlacer with three different modes - 1:2:1 blur, drop field, and adaptive

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |     Range
==============================================================================
FE2_ADAPTIVE_DEINTERLACE_MODE       | Numeric   | Required | deintmode_t
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_ADAPTIVE_DEINTERLACE);
// use adaptive mode
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_ADAPTIVE_DEINTERLACE_MODE, DEINTERLACE_ADAPTIVE);
```

Additional References:

Deinterlace entry in the Wikipedia

Deprecated functions

- on2sc video_options_GetDeinterlace (const FLIX2HANDLE flix, on2bool *lpDeinterlace)

  Determine if the deinterlace filter is enabled.

- on2sc video_options_SetDeinterlace (FLIX2HANDLE flix, const on2bool lDeinterlace)

  Enable/disable the deinterlace filter.

Defines

- #define FE2_FILTER_ADAPTIVE_DEINTERLACE

  Filter name for use with Flix2_AddFilter().

- #define FE2_ADAPTIVE_DEINTERLACE_MODE

  Specifies deinterlace mode to be applied to source image.

Typedefs

- typedef enum deinterlacemode deintmode_t

Enumerations

- enum deinterlacemode {

  DEINTERLACE_NONE,

DEINTERLACE_1_2_1_BLUR,
DEINTERLACE_DROP_FIELD,
DEINTERLACE_ADAPTIVE }

20.21.2 Define Documentation

20.21.2.1 #define FE2_ADAPTIVE_DEINTERLACE_MODE

Specifies deinterlace mode to be applied to source image.

The mode may be selected from deintmode_t.

Note:

Default: DEINTERLACE_NONE

Definition at line 65 of file adaptive_deinterlace.h.

20.21.2.2 #define FE2_FILTER_ADAPTIVE_DEINTERLACE

Filter name for use with Flix2_AddFilter().

Definition at line 58 of file adaptive_deinterlace.h.

20.21.3 Typedef Documentation

20.21.3.1 typedef enum deinterlacemode deintmode_t

20.21.4 Enumeration Type Documentation

20.21.4.1 enum deinterlacemode

Enumerator:

DEINTERLACE_NONE
DEINTERLACE_1_2_1_BLUR
DEINTERLACE_DROP_FIELD
DEINTERLACE_ADAPTIVE

Definition at line 50 of file adaptive_deinterlace.h.

20.21.5 Function Documentation

20.21.5.1 on2sc video_options_GetDeinterlace (const FLIX2HANDLE *flix*, on2bool * *lpDeinterlace*)

Determine if the deinterlace filter is enabled.

Parameters:

←***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpDeinterlace*** Variable to update with the current deinterlace enable status

Return values:

ON2_OK The deinterlace filter enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_ADAPTIVE_DEINTERLACE. This function will be removed in a future release.

20.21.5.2 on2sc video_options_SetDeinterlace (FLIX2HANDLE *flix*, const on2bool *lDeinterlace*)

Enable/disable the deinterlace filter.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lDeinterlace*** New deinterlace enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The deinterlace enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the deinterlace filter is disabled.
Using this function will force the use of 1:2:1 deinterlace.

Deprecated

Use the Filter Interface along with FE2_FILTER_ADAPTIVE_DEINTERLACE. This function will be removed in a future release.

20.22 Brightness/Contrast/Hue/Saturation

20.22.1 Detailed Description

The BCHS video filter is a filter used to modify the brightness, contrast, hue and/or saturation of the source image.

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
=====================================================================================
FE2_BCHS_BRIGHTNESS                 | Numeric   | Optional | [-255,255]
FE2_BCHS_CONTRAST                   | Numeric   | Optional | [-255,255]
FE2_BCHS_HUE                        | Numeric   | Optional | [-180,180]
FE2_BCHS_SATURATION                 | Numeric   | Optional | [-255,255]
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_BCHS);
//up the contrast by ~10%
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_BCHS_CONTRAST, 25.5);
```

Deprecated functions

- on2sc editor_options_GetBrightness (const FLIX2HANDLE flix, int32_t *pBrightness)

  Get the current brightness adjustment factor.

- on2sc editor_options_SetBrightness (FLIX2HANDLE flix, const int32_t brightness)

  Set the brightness adjustment factor.

- on2sc editor_options_GetUseBrightness (const FLIX2HANDLE flix, on2bool *pUseBrightness)

  Determine if the brightness filter is enabled.

- on2sc editor_options_SetUseBrightness (FLIX2HANDLE flix, const on2bool bUseBrightness)

  Enable/disable the brightness filter.

- on2sc editor_options_GetContrast (const FLIX2HANDLE flix, double *pContrast)

  Get the current contrast adjustment factor.

- on2sc editor_options_SetContrast (FLIX2HANDLE flix, const double contrast)

  Set the contrast adjustment factor.

- on2sc editor_options_GetUseContrast (const FLIX2HANDLE flix, on2bool *pUseContrast)

  Determine if the contrast filter is enabled.

- on2sc editor_options_SetUseContrast (FLIX2HANDLE flix, const on2bool bUseContrast)

  Enable/disable the contrast filter.

- on2sc editor_options_GetHue (const FLIX2HANDLE flix, int32_t *pHue)

  Get the current hue adjustment factor.

- on2sc editor_options_SetHue (FLIX2HANDLE flix, const int32_t hue)

  Set the hue adjustment factor.

- on2sc editor_options_GetUseHue (const FLIX2HANDLE flix, on2bool *pUseHue)

  Determine if the hue filter is enabled.

- on2sc editor_options_SetUseHue (FLIX2HANDLE flix, const on2bool bUseHue)

  Enable/disable the hue filter.

- on2sc editor_options_GetSaturation (const FLIX2HANDLE flix, double *pSaturation)

  Get the current saturation adjustment factor.

- on2sc editor_options_SetSaturation (FLIX2HANDLE flix, const double saturation)

  Set the saturation adjustment factor.

- on2sc editor_options_GetUseSaturation (const FLIX2HANDLE flix, on2bool *pUseSaturation)

  Determine if the saturation filter is enabled.

- on2sc editor_options_SetUseSaturation (FLIX2HANDLE flix, const on2bool bUseSaturation)

  Enable/disable the saturation filter.

Defines

- #define FE2_FILTER_BCHS

  Filter name for use with Flix2_AddFilter().

- #define FE2_BCHS_BRIGHTNESS

  Parameter for the brightness adjustment factor.

- #define FE2_BCHS_CONTRAST

  Parameter for the contrast adjustment factor.

- #define FE2_BCHS_HUE

  Parameter for the hue adjustment factor.

- #define FE2_BCHS_SATURATION

  Parameter for the saturation adjustment factor.

20.22.2 Define Documentation

20.22.2.1 #define FE2_BCHS_BRIGHTNESS

Parameter for the brightness adjustment factor.

Determines the level by which to adjust the brightness. Brightness is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative brightness of the video, use a positive number. To decrease the relative brightness of the video, use a negative number.

Note:

Valid range: [-255,255]
Default: 0 (no change)

Definition at line 58 of file bchs.h.

20.22.2.2 #define FE2_BCHS_CONTRAST

Parameter for the contrast adjustment factor.

Determines the level by which to adjust the contrast. Contrast is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative contrast of the video, use a positive number. To decrease the relative contrast of the video, use a negative number.

Note:

Valid range: [-255,255]
Default: 0 (no change)

Definition at line 69 of file bchs.h.

20.22.2.3 #define FE2_BCHS_HUE

Parameter for the hue adjustment factor.

Determines the level by which to adjust the hue. Hue is adjusted on a scale of -180 to 180, with 0 indicating no change. To adjust the hue of the video towards red, use a negative number. To adjust the hue of the video towards green, use a positive number.

Note:

Valid range: [-180,180]
Default: 0 (no change)

Definition at line 79 of file bchs.h.

20.22.2.4 #define FE2_BCHS_SATURATION

Parameter for the saturation adjustment factor.

Determines the level by which to adjust the saturation. Saturation is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative saturation of the video, use a positive number. To decrease the relative saturation of the video, use a negative number.

Note:

Valid range: [-255,255] Default: 0

Definition at line 90 of file bchs.h.

20.22.2.5 #define FE2_FILTER_BCHS

Filter name for use with Flix2_AddFilter().

Definition at line 47 of file bchs.h.

20.22.3 Function Documentation

20.22.3.1 on2sc editor_options_GetBrightness (const FLIX2HANDLE *flix*, int32_t * *pBrightness*)

Get the current brightness adjustment factor.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pBrightness*** Brightness factor (-255..255)

Return values:

ON2_OK The brightness value was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

20.22.3.2 on2sc editor_options_GetContrast (const FLIX2HANDLE *flix*, double * *pContrast*)

Get the current contrast adjustment factor.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pContrast*** Contrast factor (-255..255)

Return values:

ON2_OK The contrast value was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

20.22.3.3 on2sc editor_options_GetHue (const FLIX2HANDLE *flix*, int32_t * *pHue*)

Get the current hue adjustment factor.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pHue*** Hue factor (-180..180)

Return values:

ON2_OK The hue value was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_HUE parameter. This function will be removed in a future release.

20.22.3.4 on2sc editor_options_GetSaturation (const FLIX2HANDLE *flix*, double * *pSaturation*)

Get the current saturation adjustment factor.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pSaturation*** Saturation factor (-255..255)

Return values:

ON2_OK The saturation value was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

20.22.3.5 on2sc editor_options_GetUseBrightness (const FLIX2HANDLE *flix*, on2bool * *pUseBrightness*)

Determine if the brightness filter is enabled.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pUseBrightness*** Variable to update with the current brightness enable status

Return values:

ON2_OK The brightness filter enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

20.22.3.6 on2sc editor_options_GetUseContrast (const FLIX2HANDLE *flix*, on2bool * *pUseContrast*)

Determine if the contrast filter is enabled.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pUseContrast*** Variable to update with the current contrast enable status

Return values:

ON2_OK The contrast filter enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

20.22.3.7 on2sc editor_options_GetUseHue (const FLIX2HANDLE *flix*, on2bool * *pUseHue*)

Determine if the hue filter is enabled.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pUseHue*** Variable to update with the current hue enable status

Return values:

ON2_OK The hue filter enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_HUE parameter. This function will be removed in a future release.

20.22.3.8 on2sc editor_options_GetUseSaturation (const FLIX2HANDLE *flix*, on2bool * *pUseSaturation*)

Determine if the saturation filter is enabled.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pUseSaturation*** Variable to update with the current saturation enable status

Return values:

ON2_OK The saturation filter enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

20.22.3.9 on2sc editor_options_SetBrightness (FLIX2HANDLE *flix*, const int32_t *brightness*)

Set the brightness adjustment factor.

Determines the level by which to adjust the brightness. Brightness is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative brightness of the video, use a positive number. To decrease the relative brightness of the video, use a negative number.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***brightness*** Brightness Factor (-255 to 255)

Return values:

ON2_OK The brightness value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This call will have no effect on the encoded video unless the brightness filter is enabled with editor_options_SetUseBrightness(). The default value is 0 (no change).

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

20.22.3.10 on2sc editor_options_SetContrast (FLIX2HANDLE *flix*, const double *contrast*)

Set the contrast adjustment factor.

Determines the level by which to adjust the contrast. Contrast is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative contrast of the video, use a positive number. To decrease the relative contrast of the video, use a negative number.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***contrast*** Contrast Factor (-255 to 255)

Return values:

ON2_OK The contrast value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This call will have no effect on the encoded video unless the contrast filter is enabled with editor_options_SetUseContrast(). The default value is 1.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

20.22.3.11 on2sc editor_options_SetHue (FLIX2HANDLE *flix*, const int32_t *hue*)

Set the hue adjustment factor.

Determines the level by which to adjust the hue. Hue is adjusted on a scale of -180 to 180, with 0 indicating no change. To adjust the hue of the video towards red, use a negative number. To adjust the hue of the video towards green, use a positive number.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***hue*** Hue Factor (-180 to 180)

Return values:

ON2_OK The hue value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This call will have no effect on the encoded video unless the hue filter is enabled with editor_options_SetUseHue(). The default value is 0.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_HUE parameter. This function will be removed in a future release.

20.22.3.12 on2sc editor_options_SetSaturation (FLIX2HANDLE *flix*, const double *saturation*)

Set the saturation adjustment factor.

Determines the level by which to adjust the saturation. Saturation is adjusted on a scale of -255 to 255, with 0 indicating no change. To increase the relative saturation of the video, use a positive number. To decrease the relative saturation of the video, use a negative number.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***saturation*** Saturation Factor (-255 to 255)

Return values:

ON2_OK The saturation value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This call will have no effect on the encoded video unless the saturation filter is enabled with editor_-options_SetUseSaturation(). The default value is 1.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

20.22.3.13 on2sc editor_options_SetUseBrightness (FLIX2HANDLE *flix*, const on2bool *bUseBrightness*)

Enable/disable the brightness filter.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseBrightness*** New brightness enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The brightness enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the brightness filter is disabled.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_BRIGHTNESS parameter. This function will be removed in a future release.

20.22.3.14 on2sc editor_options_SetUseContrast (FLIX2HANDLE *flix*, const on2bool *bUseContrast*)

Enable/disable the contrast filter.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseContrast*** New contrast enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The contrast enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the contrast filter is disabled.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_CONTRAST parameter. This function will be removed in a future release.

20.22.3.15 on2sc editor_options_SetUseHue (FLIX2HANDLE *flix*, const on2bool *bUseHue*)

Enable/disable the hue filter.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseHue*** New hue enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The hue enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the hue filter is disabled.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_HUE parameter. This function will be removed in a future release.

20.22.3.16 on2sc editor_options_SetUseSaturation (FLIX2HANDLE *flix*, const on2bool *bUseSaturation*)

Enable/disable the saturation filter.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseSaturation*** New saturation enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The saturation enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the saturation filter is disabled.

Deprecated

Use the Filter Interface along with FE2_FILTER_BCHS and the FE2_BCHS_SATURATION parameter. This function will be removed in a future release.

20.23 Blur

20.23.1 Detailed Description

Blurs the source image

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |  Range
=============================================================================
FE2_BLUR_FILTER                     | Numeric   | Optional | blurfilter_t
FE2_BLUR_MASKSIZE                   | Numeric   | Optional | masksiz_t
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_BLUR);

//blur the image using a Gaussian kernel
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(&filter, FE2_BLUR_FILTER, BLUR_GAUSS);
```

Defines

- #define FE2_FILTER_BLUR

  Filter name for use with Flix2_AddFilter().

- #define FE2_BLUR_FILTER

  Specifies blur filter to be applied to source.

- #define FE2_BLUR_MASKSIZE

  Specifies the kernel/mask size to be used.

Typedefs

- typedef enum masksiz masksiz_t
- typedef enum blurfilter blurfilter_t

Enumerations

- enum masksiz {

  MASK_3x3,

  MASK_5x5 }

  Filter mask/kernel size.

- enum blurfilter {

  BLUR_LOWPASS,

  BLUR_GAUSS }

20.23.2 Define Documentation

20.23.2.1 #define FE2_BLUR_FILTER

Specifies blur filter to be applied to source.

The blur filter may be selected from blurfilter_t.

Note:

Default: BLUR_LOWPASS

Definition at line 88 of file blur.h.

20.23.2.2 #define FE2_BLUR_MASKSIZE

Specifies the kernel/mask size to be used.

The mask size may be selected from masksiz_t.

Note:

Default: MASK_3x3

See also:

blurfilter_t for kernel details

Definition at line 95 of file blur.h.

20.23.2.3 #define FE2_FILTER_BLUR

Filter name for use with Flix2_AddFilter().

Definition at line 82 of file blur.h.

20.23.3 Typedef Documentation

20.23.3.1 typedef enum blurfilter blurfilter_t

20.23.3.2 typedef enum masksiz masksiz_t

20.23.4 Enumeration Type Documentation

20.23.4.1 enum blurfilter

Enumerator:

BLUR_LOWPASS Filter image using a lowpass kernel.

```
            1/9 1/9 1/9
MASK_3x3:   1/9 1/9 1/9
            1/9 1/9 1/9

            1/25 1/25 1/25 1/25 1/25
            1/25 1/25 1/25 1/25 1/25
```

```
MASK_5x5: 1/25 1/25 1/25 1/25 1/25
          1/25 1/25 1/25 1/25 1/25
          1/25 1/25 1/25 1/25 1/25
```

BLUR_GAUSS Filter image using a Gaussian kernel.

```
           1/16 2/16 1/16
MASK_3x3:  2/16 4/16 2/16
           1/16 2/16 1/16

           2/571  7/571  12/571  7/571  2/571
           7/571 31/571  52/571 31/571  7/571
MASK_5x5: 12/571 52/571 127/571 52/571 12/571
           7/571 31/571  52/571 31/571  7/571
           2/571  7/571  12/571  7/571  2/571
```

Definition at line 52 of file blur.h.

20.23.4.2 enum masksiz

Filter mask/kernel size.

Enumerator:

MASK_3x3 kernel/mask will be a matrix of size 3x3

MASK_5x5 kernel/mask will be a matrix of size 5x5

Definition at line 47 of file blur.h.

20.24 Crop

20.24.1 Detailed Description

The crop video filter is a filter used to isolate a subsection of an input image.

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |     Range
==============================================================================
FE2_CROP_TOP                        | Numeric   | Optional | [0,video height]
FE2_CROP_BOTTOM                     | Numeric   | Optional | [0,video height]
FE2_CROP_LEFT                       | Numeric   | Optional | [0,video width]
FE2_CROP_RIGHT                      | Numeric   | Optional | [0,video width]
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_CROP);
//set crop bounding box to (0,0,240,320) (t,l,b,r)
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_CROP_BOTTOM, 240);
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_CROP_RIGHT, 320);
```

Deprecated functions

- on2sc editor_options_GetCrop (const FLIX2HANDLE flix, on2bool *pCrop)

  Determine if the crop filter is enabled.

- on2sc editor_options_SetCrop (FLIX2HANDLE flix, const on2bool crop)

  Enable/disable the crop filter.

- on2sc editor_options_GetCropBounds (const FLIX2HANDLE flix, int32_t *pTop, int32_t *pLeft, int32_t *pBottom, int32_t *pRight)

  Get the current bounding box used for cropping.

- on2sc editor_options_SetCropBounds (FLIX2HANDLE flix, const int32_t top, const int32_t left, const int32_t bottom, const int32_t right)

  Set the current bounding box used for cropping.

Defines

- #define FE2_FILTER_CROP

  Filter name for use with Flix2_AddFilter().

- #define FE2_CROP_TOP

  Filter parameter for setting bounding box's top coordinate.

- #define FE2_CROP_BOTTOM

  Filter parameter for setting bounding box's bottom coordinate.

- #define FE2_CROP_LEFT

  Filter parameter for setting bounding box's left coordinate.

- #define FE2_CROP_RIGHT

  Filter parameter for setting bounding box's right coordinate.

20.24.2 Define Documentation

20.24.2.1 #define FE2_CROP_BOTTOM

Filter parameter for setting bounding box's bottom coordinate.

Note:

Default: input image height
Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 61 of file crop.h.

20.24.2.2 #define FE2_CROP_LEFT

Filter parameter for setting bounding box's left coordinate.

Note:

Default: 0
Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 67 of file crop.h.

20.24.2.3 #define FE2_CROP_RIGHT

Filter parameter for setting bounding box's right coordinate.

Note:

Default: input image width
Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 73 of file crop.h.

20.24.2.4 #define FE2_CROP_TOP

Filter parameter for setting bounding box's top coordinate.

Note:

Default: 0
Value must be a multiple of 2. Odd values will be silently adjusted down.

Definition at line 55 of file crop.h.

20.24.2.5 #define FE2_FILTER_CROP

Filter name for use with Flix2_AddFilter().

Definition at line 49 of file crop.h.

20.24.3 Function Documentation

20.24.3.1 on2sc editor_options_GetCrop (const FLIX2HANDLE *flix*, on2bool ∗ *pCrop*)

Determine if the crop filter is enabled.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pCrop*** Variable to update with the current crop enable status

Return values:

ON2_OK The crop enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_CROP. This function will be removed in a future release.

20.24.3.2 on2sc editor_options_GetCropBounds (const FLIX2HANDLE *flix*, int32_t ∗ *pTop*, int32_t ∗ *pLeft*, int32_t ∗ *pBottom*, int32_t ∗ *pRight*)

Get the current bounding box used for cropping.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pTop*** Topmost row of the cropped area

→ ***pLeft*** Leftmost column of the cropped area

→ ***pBottom*** Bottommost row of the cropped area

→ ***pRight*** Rightmost row of the cropped area

Return values:

ON2_OK The crop bounding box was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_CROP and the FE2_CROP_TOP, FE2_CROP_-LEFT, FE2_CROP_BOTTOM and FE2_CROP_RIGHT parameters. This function will be removed in a future release.

20.24.3.3 on2sc editor_options_SetCrop (FLIX2HANDLE *flix*, const on2bool *crop*)

Enable/disable the crop filter.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***crop*** New crop enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The crop enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the crop filter is disabled.

Deprecated

Use the Filter Interface along with FE2_FILTER_CROP. This function will be removed in a future release.

20.24.3.4 on2sc editor_options_SetCropBounds (FLIX2HANDLE *flix*, const int32_t *top*, const int32_t *left*, const int32_t *bottom*, const int32_t *right*)

Set the current bounding box used for cropping.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***top*** Topmost row of the cropped area

← ***left*** Leftmost column of the cropped area

← ***bottom*** Bottommost row of the cropped area

← ***right*** Rightmost row of the cropped area

Return values:

ON2_OK The crop bounding box was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This call will have no effect on the encoded video unless the crop filter is enabled with editor_options_-SetCrop()

Deprecated

Use the Filter Interface along with FE2_FILTER_CROP and the FE2_CROP_TOP, FE2_CROP_-LEFT, FE2_CROP_BOTTOM and FE2_CROP_RIGHT parameters. This function will be removed in a future release.

20.25 Cut

20.25.1 Detailed Description

The cut filter allows for the encoding of a user specified range of time from the input media file.

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
=================================================================================
FE2_CUT_START_SEC                   | Numeric   | Optional | [0, media duration]
FE2_CUT_STOP_SEC                    | Numeric   | Optional | [-1, media duration]
FE2_CUT_USE_SEEK                    | Numeric   | Optional | [0, 1]
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_CUT);
// start the cut at 5 seconds
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_CUT_START_SEC, 5.0);
// end the cut at 10 seconds
if (sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_CUT_STOP_SEC, 10.0);
```

Deprecated functions

- on2sc editor_options_GetUseCut (const FLIX2HANDLE flix, on2bool *pUseCut)

  Determine if the cut filter is enabled.

- on2sc editor_options_SetUseCut (FLIX2HANDLE flix, const on2bool bUseCut)

  Enable/disable the cut filter.

- on2sc editor_options_GetCutStartTime (const FLIX2HANDLE flix, double *pStartTime)

  Get the current cut start time.

- on2sc editor_options_SetCutStartTime (FLIX2HANDLE flix, const double start_time)

  Set the cut start time.

- on2sc editor_options_GetCutStopTime (const FLIX2HANDLE flix, double *pEndTime)

  Get the current cut stop time.

- on2sc editor_options_SetCutStopTime (FLIX2HANDLE flix, const double end_time)

  Set the cut stop time.

Defines

- #define FE2_FILTER_CUT

  Filter name for use with Flix2_AddFilter().

- #define FE2_CUT_START_SEC

  Filter parameter for cut start time.

- #define FE2_CUT_STOP_SEC

  Filter parameter for setting the cut stop time.

- #define FE2_CUT_USE_SEEK

  Set FE2_CUT_START_SEC application method.

20.25.2 Define Documentation

20.25.2.1 #define FE2_CUT_START_SEC

Filter parameter for cut start time.

Sets the time (in seconds relative to the beginning of the source) to begin the media cut. A value of `0` indicates that the cut should start at the beginning of the input media.

Note:

Default: 0

Definition at line 64 of file cut.h.

20.25.2.2 #define FE2_CUT_STOP_SEC

Filter parameter for setting the cut stop time.

Sets the time (in seconds relative to the beginning of the input media) to end the media cut. A value of `-1` indicates the cut should stop at the end of the input media.

Note:

Default: -1

Definition at line 75 of file cut.h.

20.25.2.3 #define FE2_CUT_USE_SEEK

Set FE2_CUT_START_SEC application method.

When FE2_CUT_START_SEC is non-zero this parameter determines how the cut is applied. A value of `0` indicates that the entire source file should be read and discarded until FE2_CUT_START_SEC is reached. This method will increase the time necessary to process the source media as it must be decoded before it be discarded. This was the default behavior prior to 8.0.7.0. A value of `1` indicates that the engine should attempt to perform a seek on the source media to arrive at FE2_CUT_START_SEC. Should this method fail the engine will fall back to the legacy method.

Note:

Default: 1
If output is produced with the value set to 1, but the results are undesirable the only workaround is to re-encode with the value set to 0.

Definition at line 95 of file cut.h.

20.25.2.4 #define FE2_FILTER_CUT

Filter name for use with Flix2_AddFilter().

Definition at line 52 of file cut.h.

20.25.3 Function Documentation

20.25.3.1 on2sc editor_options_GetCutStartTime (const FLIX2HANDLE *flix*, double * *pStartTime*)

Get the current cut start time.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pStartTime*** Clip start time (in seconds)

Return values:

ON2_OK The cut start time was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

20.25.3.2 on2sc editor_options_GetCutStopTime (const FLIX2HANDLE *flix*, double * *pEndTime*)

Get the current cut stop time.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pEndTime*** Clip stop time (in seconds, -1 for end)

Return values:

ON2_OK The cut stop time was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

20.25.3.3 on2sc editor_options_GetUseCut (const FLIX2HANDLE *flix*, on2bool * *pUseCut*)

Determine if the cut filter is enabled.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pUseCut*** Variable to update with the current saturation enable status

Return values:

ON2_OK The cut filter enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

20.25.3.4 on2sc editor_options_SetCutStartTime (FLIX2HANDLE *flix*, const double *start_time*)

Set the cut start time.

Sets the time (in seconds from the beginning of the source) to begin the media cut. 0 indicates start immediately.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***start_time*** Start time (in seconds, relative to source)

Return values:

ON2_OK The start time value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This call will have no effect on the encoded video unless the cut filter is enabled with editor_options_-SetUseCut(). The default value is 0.0.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

20.25.3.5 on2sc editor_options_SetCutStopTime (FLIX2HANDLE *flix*, const double *end_time*)

Set the cut stop time.

Sets the time (in seconds from the beginning of the source) to end the media cut. -1 indicates end of source.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***end_time*** End time (in seconds, relative to source)

Return values:

ON2_OK The end time value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This call will have no effect on the encoded video unless the cut filter is enabled with editor_options_-SetUseCut(). The default value is -1.0 (end of source).

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

20.25.3.6 on2sc editor_options_SetUseCut (FLIX2HANDLE *flix*, const on2bool *bUseCut*)

Enable/disable the cut filter.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseCut*** New cut enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The cut enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the cut filter is disabled.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_CUT
- FE2_CUT_START_SEC
- FE2_CUT_STOP_SEC

for access to the flix engine cut filter.

20.26 Denoise

20.26.1 Detailed Description

Removes noise from source image

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |  Range
=====================================================================
FE2_DENOISE_NOISE_LEVEL             | Numeric   | Optional | [0.0,1.0)
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_DENOISE);
if(sc == ON2_OK)
    ; //adaptive noise removal will be applied to the source image
```

Defines

- #define FE2_FILTER_DENOISE

  Filter name for use with Flix2_AddFilter().

- #define FE2_DENOISE_NOISE_LEVEL

  Specifies noise level of the source.

20.26.2 Define Documentation

20.26.2.1 #define FE2_DENOISE_NOISE_LEVEL

Specifies noise level of the source.

Any value greater than 0 implies a constant and known level of noise within the source. A value of 0 will cause the filter to estimate the noise level, allowing it to vary from frame to frame. Larger noise levels indicate a noisier source resulting in increased removal by the filter, at the cost of sharpness.

Note:

Default: 0 (adaptive)

Definition at line 57 of file denoise.h.

20.26.2.2 #define FE2_FILTER_DENOISE

Filter name for use with Flix2_AddFilter().

Definition at line 45 of file denoise.h.

20.27 Frame Rate

20.27.1 Detailed Description

The frame rate filter modifies the source frame rate either by applying a specific frame rate (e.g. 12.0) or a decimation interval (e.g. 2 = 1/2 source frame rate).

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
==============================================================================
FE2_FRAMERATE_FPS                   | Numeric   | Optional | [0.0,)
FE2_FRAMERATE_DECIMATE              | Numeric   | Optional | [1,)
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_FRAMERATE);
// decimate by 2. ex. 29.97fps becomes 14.985
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_FRAMERATE_DECIMATE, 2);
```

Deprecated functions

- on2sc video_options_GetVideoFramerate (const FLIX2HANDLE flix, int32_t *lpVideoFramerate)

  Get the current video framerate.

- on2sc video_options_SetVideoFramerate (FLIX2HANDLE flix, const int32_t lVideoFramerate)

  Set the video framerate.

- on2sc video_options_GetVideoFramerateAsDouble (const FLIX2HANDLE flix, double *p_fps)

  Get the current video framerate.

- on2sc video_options_SetVideoFramerateAsDouble (FLIX2HANDLE flix, const double fps)

  Set the video framerate.

- on2sc video_options_GetUseSourceFramerate (FLIX2HANDLE flix, on2bool *bpUseSourceFramerate)

  Determine if the source's framerate will be used instead of the modified framerate.

- on2sc video_options_SetUseSourceFramerate (FLIX2HANDLE flix, const on2bool bUseSourceFramerate)

  Switch between source and scaled framerate.

- on2sc video_options_GetDecimateValue (const FLIX2HANDLE flix, uint32_t *pValue)

  Get the current decimation of the video framerate.

- on2sc video_options_SetDecimateValue (FLIX2HANDLE flix, const uint32_t value)

  Set the decimation of the video framerate.

Defines

- #define FE2_FILTER_FRAMERATE

  Filter name for use with Flix2_AddFilter().

- #define FE2_FRAMERATE_FPS

  Filter parameter name for frames per second value.

- #define FE2_FRAMERATE_DECIMATE

  Filter parameter for the decimation interval.

20.27.2 Define Documentation

20.27.2.1 #define FE2_FILTER_FRAMERATE

Filter name for use with Flix2_AddFilter().

Definition at line 49 of file framerate.h.

20.27.2.2 #define FE2_FRAMERATE_DECIMATE

Filter parameter for the decimation interval.

Decimation removes frames from the source video at regular intervals. This is useful for dropping the rate without introducing jerkiness due to uneven frame rate.

Note:

Must be a positive integer
Default: disabled

Definition at line 67 of file framerate.h.

20.27.2.3 #define FE2_FRAMERATE_FPS

Filter parameter name for frames per second value.

Allows the frame rate to be adjusted to a specific rate.

Note:

Default: disabled

Definition at line 56 of file framerate.h.

20.27.3 Function Documentation

20.27.3.1 on2sc video_options_GetDecimateValue (const FLIX2HANDLE *flix*, uint32_t * *pValue*)

Get the current decimation of the video framerate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pValue*** Decimate value

Return values:

ON2_OK Success.

ON2_INVALID_PARAMS Should one or more of the preconditions fail.

ON2_NET_ERROR The underlying communication layer failed.

Precondition:

flix is not `NULL`
pValue is not `NULL`

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_-DECIMATE parameter. This function will be removed in a future release.

20.27.3.2 on2sc video_options_GetUseSourceFramerate (FLIX2HANDLE *flix*, on2bool * *bpUseSourceFramerate*)

Determine if the source's framerate will be used instead of the modified framerate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***bpUseSourceFramerate*** Variable to update with the current framerate selection

Return values:

ON2_OK The framerate status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_FRAMERATE. Not adding FE2_FILTER_-FRAMERATE with Flix2_AddFilter() implies use of the source frame rate. This function will be removed in a future release.

20.27.3.3 on2sc video_options_GetVideoFramerate (const FLIX2HANDLE *flix*, int32_t * *lpVideoFramerate*)

Get the current video framerate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpVideoFramerate*** Frames per second

Return values:

ON2_OK The framerate was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Deprecated

Please use video_options_GetVideoFramerateAsDouble() to allow for non integer framerates.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

20.27.3.4 on2sc video_options_GetVideoFramerateAsDouble (const FLIX2HANDLE *flix*, double * *p_fps*)

Get the current video framerate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***p_fps*** Frames per second

Return values:

ON2_OK The framerate was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_FPS parameter. This function will be removed in a future release.

20.27.3.5 on2sc video_options_SetDecimateValue (FLIX2HANDLE *flix*, const uint32_t *value*)

Set the decimation of the video framerate.

This value will be used to drop frames according to the video framerate. For example if the framerate is 30fps and the decimate value is 3 then the encoded framerate will be 10fps.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***value*** Decimate value

Return values:

ON2_OK The target framerate was successfully set in the engine.

ON2_INVALID_PARAMS Should one or more of the preconditions fail.

ON2_NET_ERROR The underlying communication layer failed.

Precondition:

flix is not `NULL`

Note:

Default value is 0.
A value of <= 1 means to not drop any frames.

Deprecated

Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_-DECIMATE parameter. This function will be removed in a future release.

20.27.3.6 on2sc video_options_SetUseSourceFramerate (FLIX2HANDLE *flix*, const on2bool *bUseSourceFramerate*)

Switch between source and scaled framerate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseSourceFramerate*** New framerate status. on2true to use source framerate, on2false to use scaled framerate.

Return values:

ON2_OK The framerate status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the source framerate will be used, unless a call to video_options_SetVideoFramerate(), video_options_SetVideoFramerateAsDouble() or video_options_SetDecimateValue() has been made.

Deprecated

Use the Filter Interface along with FE2_FILTER_FRAMERATE. Not adding FE2_FILTER_-FRAMERATE with Flix2_AddFilter() implies use of the source frame rate. This function will be removed in a future release.

20.27.3.7 on2sc video_options_SetVideoFramerate (FLIX2HANDLE *flix*, const int32_t *lVideoFramerate*)

Set the video framerate.

Sets the framerate to be used when encoding the video. Frames will be duplicated or dropped as necessary to achieve the desired framerate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lVideoFramerate*** Framerate target (in frames per second)

Return values:

ON2_OK The target framerate was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Deprecated

Please use video_options_SetVideoFramerateAsDouble() to allow for non integer framerates.

Note:

The default is to use the source framerate.

20.27.3.8 on2sc video_options_SetVideoFramerateAsDouble (FLIX2HANDLE *flix*, const double *fps*)

Set the video framerate.

Sets the framerate to be used when encoding the video. Frames will be duplicated or dropped as necessary to achieve the desired framerate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***fps*** Framerate target (in frames per second)

Return values:

ON2_OK The target framerate was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

The default is to use the source framerate.

Deprecated

Use the Filter Interface along with FE2_FILTER_FRAMERATE and the FE2_FRAMERATE_FPS parameter. This function will be removed in a future release.

20.28 Highpass

20.28.1 Detailed Description

The highpass audio filter is a filter to attenuate sounds in the audio track that are lower than the cutoff frequency. In other words, high frequencies are passed by the filter, and low frequencies are stopped.

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_HIGHPASS);
// Cut off frequencies below 2khz
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_HIGHPASS_CUTOFF, 2000.0);
```

Additional References:

Cookbook formulae for audio EQ biquad filter coefficients by Robert Bristow-Johnson

Defines

- #define FE2_FILTER_HIGHPASS

  Filter name for use with Flix2_AddFilter().

- #define FE2_HIGHPASS_Q

  Filter parameter for shape constant ("Q" coefficient).

- #define FE2_HIGHPASS_CUTOFF

  Filter parameter for cutoff frequency.

20.28.2 Define Documentation

20.28.2.1 #define FE2_FILTER_HIGHPASS

Filter name for use with Flix2_AddFilter().

Definition at line 46 of file highpass.h.

20.28.2.2 #define FE2_HIGHPASS_CUTOFF

Filter parameter for cutoff frequency.

This is the filter's "corner" frequency. Components of the sound track with frequencies lower than this frequency will be attenuated. If very precise control of the cutoff frequency is required, note that the response of the filter at this frequency is -3db. Note that the maximum frequency is always half of the sampling rate of the produced file (5512.5/11025/22050 for sample rates Hertz11025/Hertz22050/Hertz44100 respectively).

Definition at line 69 of file highpass.h.

20.28.2.3 #define FE2_HIGHPASS_Q

Filter parameter for shape constant ("Q" coefficient).

The Q coefficient controls the "shape" of the filter. In general, higher numbers mean a sharper response curve. The default value is 0.707, and should be appropriate in most cases. If you feel the default value is inadequate, iterative experimentation is the best way to select a new value. Alternatively, a mathematical description of this filter is linked in the references above.

Definition at line 57 of file highpass.h.

20.29 Lowpass

20.29.1 Detailed Description

The lowpass audio filter is a filter to attenuate sounds in the audio track that are higher than the cutoff frequency. In other words, low frequencies are passed by the filter, and high frequencies are stopped.

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_LOWPASS);
// Cut off frequencies above 10khz
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_LOWPASS_CUTOFF, 10000.0);
```

Additional References:

Cookbook formulae for audio EQ biquad filter coefficients by Robert Bristow-Johnson

Defines

- #define FE2_FILTER_LOWPASS

  Filter name for use with Flix2_AddFilter().

- #define FE2_LOWPASS_Q

  Filter parameter for shape constant ("Q" coefficient).

- #define FE2_LOWPASS_CUTOFF

  Filter parameter for cutoff frequency.

20.29.2 Define Documentation

20.29.2.1 #define FE2_FILTER_LOWPASS

Filter name for use with Flix2_AddFilter().

Definition at line 46 of file lowpass.h.

20.29.2.2 #define FE2_LOWPASS_CUTOFF

Filter parameter for cutoff frequency.

This is the filter's "corner" frequency. Components of the sound track with frequencies higher than this frequency will be attenuated. If very precise control of the cutoff frequency is required, note that the response of the filter at this frequency is -3db. Note that the maximum frequency is always half of the sampling rate of the produced file (5512.5/11025/22050 for sample rates Hertz11025/Hertz22050/Hertz44100 respectively).

Definition at line 69 of file lowpass.h.

20.29.2.3 #define FE2_LOWPASS_Q

Filter parameter for shape constant ("Q" coefficient).

The Q coefficient controls the "shape" of the filter. In general, higher numbers mean a sharper response curve. The default value is 0.707, and should be appropriate in most cases. If you feel the default value is inadequate, iterative experimentation is the best way to select a new value. Alternatively, a mathematical description of this filter is linked in the references above.

Definition at line 57 of file lowpass.h.

20.30 Mirror

20.30.1 Detailed Description

The mirror filter mirrors (flips) the source along the horizontal axis, vertical axis or both.

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
==============================================================================
FE2_MIRROR_HORIZONTAL               | Numeric   | Optional | {on2false,on2true}
FE2_MIRROR_VERTICAL                 | Numeric   | Optional | {on2false,on2true}
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_MIRROR);
// flip the image horizontally
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_MIRROR_HORIZONTAL, on2true);
```

Note:

Setting both FE2_MIRROR_HORIZONTAL and FE2_MIRROR_VERTICAL is equivalent to setting FE2_ROTATE_ANGLE to 180

See also:

FE2_FILTER_ROTATE

Defines

- #define FE2_FILTER_MIRROR

  Filter name for use with Flix2_AddFilter().

- #define FE2_MIRROR_HORIZONTAL

  Specifies horizontal disposition.

- #define FE2_MIRROR_VERTICAL

  Specifies vertical disposition.

20.30.2 Define Documentation

20.30.2.1 #define FE2_FILTER_MIRROR

Filter name for use with Flix2_AddFilter().

Definition at line 49 of file mirror.h.

20.30.2.2 #define FE2_MIRROR_HORIZONTAL

Specifies horizontal disposition.

Mirror (flip) the image along the horizontal axis

Note:

Default: 0 (disabled)
non-zero: enabled

Definition at line 56 of file mirror.h.

20.30.2.3 #define FE2_MIRROR_VERTICAL

Specifies vertical disposition.

Mirror (flip) the image along the vertical axis

Note:

Default: 0 (disabled)
non-zero: enabled

Definition at line 63 of file mirror.h.

20.31 Overlay (Watermark)

20.31.1 Detailed Description

The overlay video filter is a filter used to apply a PNG image to the output video. In addition this filter supports positioning the overlay and treating sections of the image as transparent either by specifying a pixel of the image to use as a reference, specifying a RGB value to use as a reference or using the alpha channel of the image.

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
=====================================================================================
FE2_OVERLAY_FILE                    | String    | Required | N/A
FE2_OVERLAY_MASK_XY                 | Boolean   | Optional | [on2false,on2true]
FE2_OVERLAY_MASK_X                  | Numeric   | Optional | [0,video width)
FE2_OVERLAY_MASK_Y                  | Numeric   | Optional | [0,video height)
FE2_OVERLAY_MASK_RGB                | Boolean   | Optional | [on2false,on2true]
FE2_OVERLAY_MASK_R                  | Numeric   | Optional | [0,255]
FE2_OVERLAY_MASK_G                  | Numeric   | Optional | [0,255]
FE2_OVERLAY_MASK_B                  | Numeric   | Optional | [0,255]
FE2_OVERLAY_POS                     | Numeric   | Optional | [FE2_OverlayPositionMode]
FE2_OVERLAY_POS_X                   | Numeric   | Optional | [0,video width)
FE2_OVERLAY_POS_Y                   | Numeric   | Optional | [0,video height)
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_OVERLAY);
//apply image.png to the top left of the output video
if(sc == ON2_OK)
    sc = Flix2_FilterSetParamAsStr(filter, FE2_OVERLAY_FILE, "/tmp/image.png");
```

Deprecated functions

- on2sc overlay_options_Reset (const FLIX2HANDLE flix)

  Resets the overlay options.

- on2sc overlay_options_GetUseOverlay (const FLIX2HANDLE flix, on2bool *pUseOverlay)

  Determines if an overlay is to be used.

- on2sc overlay_options_SetUseOverlay (FLIX2HANDLE flix, const on2bool bUseOverlay)

  Enables or disables overlay usage.

- on2sc overlay_options_GetOverlayPath (const FLIX2HANDLE flix, on2tc *pOverlayFilePath, int32_t *pLen)

  Returns the path to the overlay image file.

- on2sc overlay_options_SetOverlayPath (FLIX2HANDLE flix, const on2tc *pOverlayFilePath)

  Set the path to the overlay image file.

- on2sc overlay_options_GetMaskPixelXY (const FLIX2HANDLE flix, int32_t *pMaskPixelX, int32_t *pMaskPixelY)

  Return the X and Y coordinates of the mask pixel.

- on2sc overlay_options_SetMaskPixelXY (FLIX2HANDLE flix, int32_t maskPixelX, int32_t maskPixelY)

  Set the X and Y coordinates of the mask pixel.

- on2sc overlay_options_GetMaskPixelRGB (FLIX2HANDLE flix, uint8_t *pMaskPixelR, uint8_t *pMaskPixelG, uint8_t *pMaskPixelB)

  Return the Red, Green, and Blue component values of the mask pixel.

- on2sc overlay_options_SetMaskPixelRGB (FLIX2HANDLE flix, uint8_t maskPixelR, uint8_t maskPixelG, uint8_t maskPixelB)

  Set the Red, Green, and Blue component values of the mask pixel.

- on2sc overlay_options_GetOverlayPosition (FLIX2HANDLE flix, FE2_OverlayPositionMode *pMode, uint32_t *pX, uint32_t *pY)

  Return the overlay position.

- on2sc overlay_options_SetOverlayPosition (FLIX2HANDLE flix, FE2_OverlayPositionMode mode, uint32_t x, uint32_t y)

  Set the overlay position.

Defines

- #define FE2_FILTER_OVERLAY

  Filter name for use with Flix2_AddFilter().

- #define FE2_OVERLAY_FILE

  Set the path to the overlay image file.

- #define FE2_OVERLAY_MASK_XY

  Use the pixel at coordinates (FE2_OVERLAY_MASK_X,FE2_OVERLAY_MASK_Y) to determine the transparent color.

- #define FE2_OVERLAY_MASK_X

  X coordinate of pixel to use for transparency.

- #define FE2_OVERLAY_MASK_Y

  Y coordinate of pixel to use for transparency.

- #define FE2_OVERLAY_MASK_RGB

  Use the RGB value (FE2_OVERLAY_MASK_R, FE2_OVERLAY_MASK_G, FE2_OVERLAY_MASK_B) as the transparency color.

- #define FE2_OVERLAY_MASK_R

  Red component of the transparency color.

- #define FE2_OVERLAY_MASK_G

  Green component of the transparency color.

- #define FE2_OVERLAY_MASK_B

Blue component of the transparency color.

- #define FE2_OVERLAY_POS

  Set the overlay position. Valid modes are defined by FE2_OverlayPositionMode.

- #define FE2_OVERLAY_POS_X

  X coordinate of overlay top left on video.

- #define FE2_OVERLAY_POS_Y

  Y coordinate of overlay top left on video.

Enumerations

- enum FE2_OverlayPositionMode {
  FE2_OVERLAY_POS_MODE_TOPLEFT,
  FE2_OVERLAY_POS_MODE_BOTLEFT,
  FE2_OVERLAY_POS_MODE_CENTER,
  FE2_OVERLAY_POS_MODE_TOPRIGHT,
  FE2_OVERLAY_POS_MODE_BOTRIGHT,
  FE2_OVERLAY_POS_MODE_XY }

  Position modes for use with FE2_OVERLAY_POS.

20.31.2 Define Documentation

20.31.2.1 #define FE2_FILTER_OVERLAY

Filter name for use with Flix2_AddFilter().

Definition at line 76 of file overlay.h.

20.31.2.2 #define FE2_OVERLAY_FILE

Set the path to the overlay image file.

Remarks:

Only PNG images, 24 bit bitmaps, and 32 bit bitmaps are supported. Use of PNG images is highly recommended.
Images with an uneven width or uneven height will be **cropped**.
Overlay images larger than the video image will be **cropped**.
When both the video and overlay images contain an alpha channel, the overlay and video alpha will be averaged. For example, if the video at the overlay position was fully transparent, and the overlay was fully opaque, the region of the exported video where the overlay was placed would become 50% opaque.

Note:

Linux Only: flixd, the encoding server, gets its input from, and writes its output to, the host file system. `FE2_OVERLAY_FILE` must be accessible in this context. An absolute path should be given to help ensure this is the case.

See also:

Flix2_SetInputFile()

Definition at line 96 of file overlay.h.

20.31.2.3 #define FE2_OVERLAY_MASK_B

Blue component of the transparency color.

Remarks:

Default value: 0

Note:

Only used if FE2_OVERLAY_MASK_RGB is set to on2true

Definition at line 156 of file overlay.h.

20.31.2.4 #define FE2_OVERLAY_MASK_G

Green component of the transparency color.

Remarks:

Default value: 0

Note:

Only used if FE2_OVERLAY_MASK_RGB is set to on2true

Definition at line 150 of file overlay.h.

20.31.2.5 #define FE2_OVERLAY_MASK_R

Red component of the transparency color.

Remarks:

Default value: 0

Note:

Only used if FE2_OVERLAY_MASK_RGB is set to on2true

Definition at line 144 of file overlay.h.

20.31.2.6 #define FE2_OVERLAY_MASK_RGB

Use the RGB value (FE2_OVERLAY_MASK_R, FE2_OVERLAY_MASK_G, FE2_OVERLAY_-MASK_B) as the transparency color.

Remarks:

The color specified will become transparent when the overlay image is drawn on the video image.
Setting this parameter will disable FE2_OVERLAY_MASK_XY if it has been set.
Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.
If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Definition at line 138 of file overlay.h.

20.31.2.7 #define FE2_OVERLAY_MASK_X

X coordinate of pixel to use for transparency.

Remarks:

Default value: 0

Definition at line 117 of file overlay.h.

20.31.2.8 #define FE2_OVERLAY_MASK_XY

Use the pixel at coordinates (FE2_OVERLAY_MASK_X,FE2_OVERLAY_MASK_Y) to determine the transparent color.

Remarks:

The color at the specified coordinate will become transparent when the overlay image is drawn on the video image.
Setting this parameter will disable FE2_OVERLAY_MASK_RGB if it has been set.
Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.
If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Definition at line 112 of file overlay.h.

20.31.2.9 #define FE2_OVERLAY_MASK_Y

Y coordinate of pixel to use for transparency.

Remarks:

Default value: 0

Definition at line 122 of file overlay.h.

20.31.2.10 #define FE2_OVERLAY_POS

Set the overlay position. Valid modes are defined by FE2_OverlayPositionMode.

Remarks:

If the mode is equal to FE2_OVERLAY_POS_MODE_XY, and either coordinate value is invalid (x > video width or y > video height), the invalid coordinate will be set to 0.
If the value of the x and/or y coordinates cause any portion of the overlay to be outside of the video image bounds, the portion of the overlay image that is out of bounds will be cropped. Default: FE2_-OVERLAY_POS_MODE_TOPLEFT

Definition at line 169 of file overlay.h.

20.31.2.11 #define FE2_OVERLAY_POS_X

X coordinate of overlay top left on video.

Note:

value is only used when FE2_OVERLAY_POS has been set to FE2_OVERLAY_POS_MODE_XY

Definition at line 175 of file overlay.h.

20.31.2.12 #define FE2_OVERLAY_POS_Y

Y coordinate of overlay top left on video.

Note:

value is only used when FE2_OVERLAY_POS has been set to FE2_OVERLAY_POS_MODE_XY

Definition at line 181 of file overlay.h.

20.31.3 Enumeration Type Documentation

20.31.3.1 enum FE2_OverlayPositionMode

Position modes for use with FE2_OVERLAY_POS.

Enumerator:

FE2_OVERLAY_POS_MODE_TOPLEFT draw overlay image on the top left corner of the video

FE2_OVERLAY_POS_MODE_BOTLEFT draw overlay image on the bottom left corner of the video

FE2_OVERLAY_POS_MODE_CENTER draw overlay image on the center of the video

FE2_OVERLAY_POS_MODE_TOPRIGHT draw overlay image on the top right corner of the video

FE2_OVERLAY_POS_MODE_BOTRIGHT draw overlay image on the bottom right corner of the video

FE2_OVERLAY_POS_MODE_XY draw overlay image on the video at the x and y coordinates specified

Definition at line 57 of file overlay.h.

20.31.4 Function Documentation

20.31.4.1 on2sc overlay_options_GetMaskPixelRGB (FLIX2HANDLE *flix*, uint8_t * *pMaskPixelR*, uint8_t * *pMaskPixelG*, uint8_t * *pMaskPixelB*)

Return the Red, Green, and Blue component values of the mask pixel.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pMaskPixelR*** pointer to variable that will receive the R value

→ ***pMaskPixelG*** pointer to variable that will receive the G value

→ ***pMaskPixelB*** pointer to variable that will receive the B value

Return values:

ON2_OK the color component values were retrieved successfully

ON2_INVALID_PARAMS a precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pMaskPixelR is not `NULL`
pMaskPixelG is not `NULL`
pMaskPixelB is not `NULL`

Remarks:

This method returns the values set in previous call to overlay_options_SetMaskPixelRGB(), if no coordinates were previously set, the Red, Green, and Blue component values will be set to `0`.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_MASK_R, FE2_OVERLAY_MASK_G and FE2_OVERLAY_MASK_B parameters. This function will be removed in a future release.

20.31.4.2 on2sc overlay_options_GetMaskPixelXY (const FLIX2HANDLE *flix*, int32_t * *pMaskPixelX*, int32_t * *pMaskPixelY*)

Return the X and Y coordinates of the mask pixel.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pMaskPixelX*** pointer to variable that will receive the X value

→ ***pMaskPixelY*** pointer to variable that will receive the Y value

Return values:

ON2_OK the coordinate values were retrieved successfully

ON2_INVALID_PARAMS a precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pMaskPixelX is not `NULL`
pMaskPixelY is not `NULL`

Remarks:

This method returns the values set in previous call to overlay_options_SetMaskPixelXY(), if no coordinates were previously set, the X and Y coordinate values will be set to `0`.
The color at the specified coordinate will become transparent when the overlay image is drawn on the video image.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_MASK_X and FE2_OVERLAY_MASK_Y parameters. This function will be removed in a future release.

20.31.4.3 on2sc overlay_options_GetOverlayPath (const FLIX2HANDLE *flix*, on2tc * *pOverlayFilePath*, int32_t * *pLen*)

Returns the path to the overlay image file.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pOverlayFilePath*** string that will receive the file path

↔ ***pLen*** length of `pOverlayFilePath`

Return values:

ON2_OK success

ON2_INVALID_PARAMS a precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pLen is not `NULL`

Remarks:

if `pOverlayFilePath` is not `NULL`, the value pointed to by pLen must be set to the maximum number of characters that can be written to `pOverlayFilePath`.
If `pOverlayFilePath` is `NULL`, `pLen` will be set to the length of the path to the overlay image

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_FILE parameter. This function will be removed in a future release.

20.31.4.4 on2sc overlay_options_GetOverlayPosition (FLIX2HANDLE *flix*, FE2_OverlayPositionMode * *pMode*, uint32_t * *pX*, uint32_t * *pY*)

Return the overlay position.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pMode*** pointer to variable that will receive the mode value

→ ***pX*** pointer to variable that will receive the X value

→ ***pY*** pointer to variable that will receive the Y value

Return values:

ON2_OK the values were retrieved successfully

ON2_INVALID_PARAMS a precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pMode is not `NULL`
pX is not `NULL`
pY is not `NULL`

Remarks:

This method returns the values set in previous call to overlay_options_SetOverlayPosition(), if no call was made previously, mode will default to FE2_OVERLAY_POS_MODE_TOPLEFT. The X/Y coordinates will default to `0`.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_POS, FE2_-OVERLAY_POS_X, FE2_OVERLAY_POS_Y parameters. This function will be removed in a future release.

20.31.4.5 on2sc overlay_options_GetUseOverlay (const FLIX2HANDLE *flix*, on2bool * *pUseOverlay*)

Determines if an overlay is to be used.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pUseOverlay*** pointer to the on2bool variable that will receive the usage state

Return values:

ON2_OK `pUseOverlay` was set successfully

ON2_INVALID_PARAMS a precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pUseOverlay is not `NULL`

Remarks:

Call this function to determine if an overlay is to be drawn on the encoded video output.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY. This function will be removed in a future release.

20.31.4.6 on2sc overlay_options_Reset (const FLIX2HANDLE *flix*)

Resets the overlay options.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK reset of overlay options was successful

ON2_INVALID_PARAMS the precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Remarks:

Call this function if you wish to reset all overlay options to their default values.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY. This function will be removed in a future release.

20.31.4.7 on2sc overlay_options_SetMaskPixelRGB (FLIX2HANDLE *flix*, uint8_t *maskPixelR*, uint8_t *maskPixelG*, uint8_t *maskPixelB*)

Set the Red, Green, and Blue component values of the mask pixel.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***maskPixelR*** the R component value

← ***maskPixelG*** the G component value

← ***maskPixelB*** the B component value

Return values:

ON2_OK the component values were set successfully

ON2_INVALID_PARAMS the precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Remarks:

The color specified will become transparent when the overlay image is drawn on the video image.
Once this function has been called successfully, overlay_options_Reset() must be called to turn transparency off.
Calling this method will disable masking based on XY coordinate.
Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.
If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_MASK_R, FE2_OVERLAY_MASK_G and FE2_OVERLAY_MASK_B parameters. This function will be removed in a future release.

20.31.4.8 on2sc overlay_options_SetMaskPixelXY (FLIX2HANDLE *flix*, int32_t *maskPixelX*, int32_t *maskPixelY*)

Set the X and Y coordinates of the mask pixel.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***maskPixelX*** the X coordinate value

← ***maskPixelY*** the Y coordinate value

Return values:

ON2_OK the coordinate values were set successfully

ON2_INVALID_PARAMS the precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Remarks:

The color at the specified coordinate will become transparent when the overlay image is drawn on the video image.

Once this function has been called successfully, overlay_options_Reset() must be called to turn transparency off.
Calling this method will disable masking based on RGB components.
Using a PNG or bitmap with full alpha channel will produce better results than using a transparency color.
If using this method to add an overlay to video with an existing alpha channel, the transparent portions of the overlay will be transparent in the output video. Opaque portions of the overlay will be averaged with the existing alpha data.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_MASK_X and FE2_OVERLAY_MASK_Y parameters. This function will be removed in a future release.

20.31.4.9 on2sc overlay_options_SetOverlayPath (FLIX2HANDLE *flix*, const on2tc * *pOverlayFilePath*)

Set the path to the overlay image file.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***pOverlayFilePath*** the path to the overlay image

Return values:

ON2_OK the path to the overlay image was set successfully

ON2_INVALID_PARAMS a precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pOverlayFilePath is not `NULL`

Remarks:

This method does **not** check for the existence of the file at the specified path.
Only PNG images, 24 bit bitmaps, and 32 bit bitmaps are supported. Use of PNG images is highly recommended.
Images with an uneven width or uneven height will be **cropped**.
Overlay images larger than the video image will be **cropped**.
When both the video and overlay images contain an alpha channel, the overlay and video alpha will be averaged. For example, if the video at the overlay position was fully transparent, and the overlay was fully opaque, the region of the exported video where the overlay was placed would become 50% opaque.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_FILE parameter. This function will be removed in a future release.

20.31.4.10 on2sc overlay_options_SetOverlayPosition (FLIX2HANDLE *flix*, FE2_OverlayPositionMode *mode*, uint32_t *x*, uint32_t *y*)

Set the overlay position.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***mode*** the position mode

→ ***x*** the x coordinate value

→ ***y*** the y coordinate value

Return values:

ON2_OK success

ON2_INVALID_PARAMS the precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Remarks:

The x and y values are ignored unless `mode` is equal to FE2_OVERLAY_POS_MODE_XY.
If `mode` is equal to FE2_OVERLAY_POS_MODE_XY, and either coordinate value is invalid (`x` > video width or `y` > video height), the invalid coordinate will be set to `0`.
If the the value of the `x` and/or `y` coordinates cause any portion of the overlay to be outside of the video image bounds, the portion of the overlay image that is out of bounds will be cropped.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY and the FE2_OVERLAY_POS, FE2_-OVERLAY_POS_X, FE2_OVERLAY_POS_Y parameters. This function will be removed in a future release.

20.31.4.11 on2sc overlay_options_SetUseOverlay (FLIX2HANDLE *flix*, const on2bool *bUseOverlay*)

Enables or disables overlay usage.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseOverlay*** on2bool value set to the desired overlay state

Return values:

ON2_OK the overlay status was changed successfully

ON2_INVALID_PARAMS the precondition failed

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is `NULL`

Remarks:

Set `bUseOverlay` to on2true to enable overlay usage. Set `bUseOverlay` to on2false to disable usage.
Only PNG images, 24 bit bitmaps, and 32 bit bitmaps are supported. Use of PNG images is highly recommended.
Images with an uneven width or uneven height will be **cropped**.
Overlay images larger than the video image will be **cropped**.
When both the video and overlay images contain an alpha channel, the overlay and video alpha will be averaged. For example, if the video at the overlay position was fully transparent, and the overlay was fully opaque, the region of the exported video where the overlay was placed would become 50% opaque.

Deprecated

Use the Filter Interface along with FE2_FILTER_OVERLAY. This function will be removed in a future release.

20.32 PNG Image Export (Thumbnail)

20.32.1 Detailed Description

The PNG image export filter creates PNG images from the source video data of the input file passed to Flix2_SetInputFile().

Filter Parameters:

Name	Type	Opt/Reqd	Range
FE2_PNGEX_OUTPUT_DIRECTORY	String	Optional	N/A
FE2_PNGEX_FILENAME_PREFIX	String	Optional	N/A
FE2_PNGEX_FILENAME_SUFFIX	String	Optional	N/A
FE2_PNGEX_ENABLE_ALPHA	Boolean	Optional	[0,1]
FE2_PNGEX_EXPORT_INTERVAL	Numeric	Optional	[1,video stream duration]
FE2_PNGEX_EXPORT_TIME_STRING	String	Optional	N/A
FE2_PNGEX_EXPORT_CUE_POINTS	Numeric	Optional	[0,3]
FE2_PNGEX_AUTO_EXPORT_COUNT	Numeric	Optional	[0,10000]
FE2_PNGEX_AUTO_EXPORT_START_TIME	Numeric	Optional	[0,video stream duration]
FE2_PNGEX_AUTO_EXPORT_END_TIME	Numeric	Optional	[0,video stream duration]
FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD	Numeric	Optional	[1,)
FE2_PNGEX_EXPORT_FIRST_FRAME_PNG	Boolean	Optional	[0,1]
FE2_PNGEX_COMPRESSION_LEVEL	Numeric	Optional	[-1,9]
FE2_PNGEX_WIDTH	Numeric	Optional	FE2_SCALE_WIDTH
FE2_PNGEX_HEIGHT	Numeric	Optional	FE2_SCALE_HEIGHT

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_PNGEX);
// Automatically export 10 PNG images spaced evenly through the clip
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_PNGEX_AUTO_EXPORT_COUNT, 10);
```

Notes:

- All PNG exporter parameters are optional, but at least one of the following parameters must be used in order to export images:
  - FE2_PNGEX_AUTO_EXPORT_COUNT
  - FE2_PNGEX_EXPORT_CUE_POINTS
  - FE2_PNGEX_EXPORT_FIRST_FRAME_PNG
  - FE2_PNGEX_EXPORT_INTERVAL
  - FE2_PNGEX_EXPORT_TIME_STRING
- Image file naming:
  - The general form of an output image name is:
    FE2_PNGEX_OUTPUT_DIRECTORY '/' *FE2_PNGEX_FILENAME_PREFIX* *nnnn* *FE2_PNGEX_FILENAME_SUFFIX* '*.png*'
    using values supplied or defaults as described by each parameter.
  - Image file numbering starts at 0, the numeric portion of the file name will be expanded to 4 columns (e.g., 0 will be 0000). This limits the export total to 10000 images. If 10000 images per encoded file is inadequate, please contact support.
- Duplicate PNG images:

  - If any of FE2_PNGEX_AUTO_EXPORT_COUNT, FE2_PNGEX_EXPORT_CUE_POINTS or FE2_PNGEX_EXPORT_TIME_STRING are enabled and a time is duplicated, the duplicate time will be discarded.
  - If FE2_PNGEX_EXPORT_INTERVAL is enabled and at least one of FE2_PNGEX_AUTO_-EXPORT_COUNT, FE2_PNGEX_EXPORT_CUE_POINTS or FE2_PNGEX_EXPORT_-TIME_STRING is enabled, the time interval based PNG image will be generated using the frame subsequent to the frame that would otherwise have been duplicated if times overlap.

- Incorrect duration:
  - If the duration is unavailable (i.e., Flix2_GetSourceDuration() returns -1) the first FE2_-PNGEX_AUTO_EXPORT_COUNT frames will be output.
  - If the duration is misreported (i.e., >actual) and not all requested PNGs have been output, one additional PNG will be generated on the last frame.

Additional References:

For information about the PNG file format, please visit: `www.libpng.org`

Defines

- #define FE2_FILTER_PNGEX

  Filter name for use with Flix2_AddFilter().

- #define FE2_PNGEX_OUTPUT_DIRECTORY

  Filter parameter for setting PNG image output directory.

- #define FE2_PNGEX_FILENAME_PREFIX

  Filter parameter for setting PNG image file name prefix.

- #define FE2_PNGEX_FILENAME_SUFFIX

  Filter parameter for setting PNG image file name suffix.

- #define FE2_PNGEX_ENABLE_ALPHA

  Filter parameter for preserving source video alpha channel data in exported PNG images.

- #define FE2_PNGEX_EXPORT_INTERVAL

  Filter parameter for enabling PNG image export at a millisecond interval.

- #define FE2_PNGEX_EXPORT_TIME_STRING

  Filter parameter for setting a string of PNG export times in milliseconds.

- #define FE2_PNGEX_EXPORT_CUE_POINTS

  Filter parameter for enabling export of PNG images at cue points.

- #define FE2_PNGEX_AUTO_EXPORT_COUNT

  Filter parameter for enabling automatic PNG image generation.

- #define FE2_PNGEX_AUTO_EXPORT_START_TIME

  Filter parameter for setting PNG auto generation start time.

- #define FE2_PNGEX_AUTO_EXPORT_END_TIME

  Filter parameter for setting PNG auto generation end time.

- #define FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD

  Filter parameter for randomizing auto generated times.

- #define FE2_PNGEX_EXPORT_FIRST_FRAME_PNG

  Filter parameter for enabling PNG image creation using the first video frame.

- #define FE2_PNGEX_COMPRESSION_LEVEL

  Filter parameter for setting compression level used by libpng.

- #define FE2_PNGEX_WIDTH

  Filter parameter for setting PNG image width.

- #define FE2_PNGEX_HEIGHT

  Filter parameter for setting PNG image height.

Enumerations

- enum FE2_PNGExCuePtMode {
  FE2_PNGEX_CP_EVENT,
  FE2_PNGEX_CP_NAV,
  FE2_PNGEX_CP_ALL }

  Constants for use with FE2_PNGEX_EXPORT_CUE_POINTS.

20.32.2 Define Documentation

20.32.2.1 #define FE2_FILTER_PNGEX

Filter name for use with Flix2_AddFilter().

Definition at line 99 of file png_export.h.

20.32.2.2 #define FE2_PNGEX_AUTO_EXPORT_COUNT

Filter parameter for enabling automatic PNG image generation.

If enabled, the PNG exporter will export the specified number of PNG images at times spaced evenly throughout the video stream. See the notes section for important information about this feature and single pass encoding. Time slice duration is calculated using the following:

```
time_slice_duration = video_stream_duration / FE2_PNGEX_AUTO_EXPORT_COUNT
```

The first auto generated PNG image time is calculated using the following equation:

```
first_png_time = FE2_PNGEX_AUTO_EXPORT_START_TIME > 0 ?
    FE2_PNGEX_AUTO_EXPORT_START_TIME : time_slice_duration / 2
```

Additional PNG time(s) is/are calculated using the following equation:

```
n: 0 based PNG image index
png_time = first_png_time + (time_slice_duration * n)
```

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // auto export 10 PNG images
    sc = Flix2_FilterSetParam(png_exporter_plgn,
             FE2_PNGEX_AUTO_EXPORT_COUNT, 10);
```

Definition at line 263 of file png_export.h.

20.32.2.3 #define FE2_PNGEX_AUTO_EXPORT_END_TIME

Filter parameter for setting PNG auto generation end time.

The PNG exporter will use this value as the video stream duration in auto export time calculations. The value specified MUST be in milliseconds.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // set the auto generation end time to 10 seconds
    sc = Flix2_FilterSetParam(png_exporter_plgn,
             FE2_PNGEX_AUTO_EXPORT_END_TIME, 10000);
```

Definition at line 305 of file png_export.h.

20.32.2.4 #define FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD

Filter parameter for randomizing auto generated times.

When set to a value greater than zero, PNG image generation will occur at random offsets from the times at which images would normally be generated when FE2_PNGEX_AUTO_EXPORT_COUNT is enabled. The value of this parameter controls the time period (offset range) within which the image generation will occur. The duration of each time slice is calculated using the following equation:

```
time_slice_duration =
    video_stream_duration / FE2_PNGEX_AUTO_EXPORT_COUNT
```

FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD is used to calculate the range of time within which to randomize the PNG generation time:

```
random_offset_range = time_slice_duration /
    FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD
```

The time of each automatically generated PNG is then calculated:

```
f: first png time
n: 0 based PNG image index
r: random offset in the range 0-random_range
    png_time = f + (n * time_slice_duration) + r;
```

Definition at line 336 of file png_export.h.

20.32.2.5 #define FE2_PNGEX_AUTO_EXPORT_START_TIME

Filter parameter for setting PNG auto generation start time.

The value specified MUST be in milliseconds. This parameter affects the following changes on time calculations explained in the FE2_PNGEX_AUTO_EXPORT_COUNT section:

```
time_slice_duration =
    (video_stream_duration - FE2_PNGEX_AUTO_EXPORT_START_TIME) /
    FE2_PNGEX_AUTO_EXPORT_COUNT
first_png_time = FE2_PNGEX_AUTO_EXPORT_START_TIME
```

Subsequent PNG times will be calculated the final equation outlined in the FE2_PNGEX_AUTO_-EXPORT_COUNT section.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // set the first auto export time to 1 second
    sc = Flix2_FilterSetParam(png_exporter_plgn,
             FE2_PNGEX_AUTO_EXPORT_START_TIME, 1000);
```

Definition at line 289 of file png_export.h.

20.32.2.6 #define FE2_PNGEX_COMPRESSION_LEVEL

Filter parameter for setting compression level used by libpng.

Notes from png.h:

Set the library compression level. Currently, valid values range from 0 - 9, corresponding directly to the zlib compression levels 0 - 9 (0 - no compression, 9 - "maximal" compression). Note that tests have shown that zlib compression levels 3-6 usually perform as well as level 9 for PNG images, and do considerably fewer caclulations. In the future, these values may not correspond directly to the zlib compression levels.

In addition to the above range, -1 corresponds to zlib's default value.

Note:

Default: -1 (Z_DEFAULT_COMPRESSION)

Definition at line 370 of file png_export.h.

20.32.2.7 #define FE2_PNGEX_ENABLE_ALPHA

Filter parameter for preserving source video alpha channel data in exported PNG images.

If enabled, the PNG export filter preserves the source video alpha when exporting PNG images. This setting will only produce PNG images with alpha channels if encoding Flash 8 Video with Alpha.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // enable alpha preservation
    sc = Flix2_FilterSetParam(png_exporter_plgn,
             FE2_PNGEX_ENABLE_ALPHA, on2true);
```

Definition at line 172 of file png_export.h.

20.32.2.8 #define FE2_PNGEX_EXPORT_CUE_POINTS

Filter parameter for enabling export of PNG images at cue points.

If enabled, PNG images will be exported at event, navigation, or all cue points. See video_options_-AddFLVCuePoint(), for information with regard to adding cue points to your output FLV file.

Use FE2_PNGEX_CP_EVENT, FE2_PNGEX_CP_NAV, or FE2_PNGEX_CP_ALL to specify the types of cue points for which the PNG exporter should export PNG images.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // generate PNGs for all cue points
    sc = Flix2_FilterSetParam(png_exporter_plgn,
             FE2_PNGEX_EXPORT_CUE_POINTS, FE2_PNGEX_CP_ALL);
```

Definition at line 229 of file png_export.h.

20.32.2.9 #define FE2_PNGEX_EXPORT_FIRST_FRAME_PNG

Filter parameter for enabling PNG image creation using the first video frame.

This parameter will force the PNG exporter to generate a PNG image from the first video frame in the input file.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // create a PNG image of the first video frame
    sc = Flix2_FilterSetParam(png_exporter_plgn,
             FE2_PNGEX_EXPORT_FIRST_FRAME_PNG, on2true);
```

Definition at line 353 of file png_export.h.

20.32.2.10 #define FE2_PNGEX_EXPORT_INTERVAL

Filter parameter for enabling PNG image export at a millisecond interval.

The PNG image exporter will export a PNG once every FE2_PNGEX_EXPORT_INTERVAL milliseconds if this parameter is specified in a call to Flix2_FilterSetParam()

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // generate a PNG every 10 seconds
    sc = Flix2_FilterSetParam(png_exporter_plgn,
             FE2_PNGEX_EXPORT_INTERVAL, 10000);
```

Definition at line 190 of file png_export.h.

20.32.2.11 #define FE2_PNGEX_EXPORT_TIME_STRING

Filter parameter for setting a string of PNG export times in milliseconds.

The PNG exporter will create PNG images from the source video stream at the times specified in the string. The string MUST be in the format: `t0`,t1,t2,tn

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // produce PNG images from the source video
    // at times of 5 seconds, 10 seconds, and 15 seconds
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
             FE2_PNGEX_EXPORT_TIME_STRING, "5000,10000,15000");
```

Definition at line 209 of file png_export.h.

20.32.2.12 #define FE2_PNGEX_FILENAME_PREFIX

Filter parameter for setting PNG image file name prefix.

If a prefix is specified, PNG image files will be named in the format *FE2_PNGEX_FILENAME_-PREFIX*png_number.png. If a prefix is not specified, the file name passed to Flix2_SetOutputFile() will be used. If no prefix is desired, the prefix **MUST** be set to an empty string.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // use prefix for PNG output file names
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
             FE2_PNGEX_FILENAME_PREFIX, "pngprefix");
```

Definition at line 136 of file png_export.h.

20.32.2.13 #define FE2_PNGEX_FILENAME_SUFFIX

Filter parameter for setting PNG image file name suffix.

If a suffix is specified, PNG image files will be named in the format png_number*FE2_PNGEX_-FILENAME_SUFFIX*.png.

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // use suffix for PNG output file names
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
             FE2_PNGEX_FILENAME_SUFFIX, "pngsuffix");
```

Note:

Default: empty

Definition at line 154 of file png_export.h.

20.32.2.14 #define FE2_PNGEX_HEIGHT

Filter parameter for setting PNG image height.

Note:

Default: input height
From **8.0.16.0**, use of the reserved values defined for FE2_SCALE_HEIGHT is valid

Definition at line 388 of file png_export.h.

20.32.2.15 #define FE2_PNGEX_OUTPUT_DIRECTORY

Filter parameter for setting PNG image output directory.

If a directory is not specified, the PNG exporter will write PNG images to the directory specified in the path passed to Flix2_SetOutputFile().

For example, the following code will set the PNG export directory to /png/output/directory:

```
sc = Flix2_AddFilter(&png_exporter_plgn, flix, FE2_FILTER_PNGEX);
if (sc == ON2_OK)
    // output PNGs to directory
    sc = Flix2_FilterSetParamAsStr(png_exporter_plgn,
             FE2_PNGEX_OUTPUT_DIRECTORY, "/png/output/directory");
```

Definition at line 118 of file png_export.h.

20.32.2.16 #define FE2_PNGEX_WIDTH

Filter parameter for setting PNG image width.

Note:

Default: input width
From **8.0.16.0**, use of the reserved values defined for FE2_SCALE_WIDTH is valid

Definition at line 379 of file png_export.h.

20.32.3 Enumeration Type Documentation

20.32.3.1 enum FE2_PNGExCuePtMode

Constants for use with FE2_PNGEX_EXPORT_CUE_POINTS.

Enumerator:

FE2_PNGEX_CP_EVENT PNG image exporter will export PNGs for only event cue points

FE2_PNGEX_CP_NAV PNG image exporter will export PNGs for only navigation cue points

FE2_PNGEX_CP_ALL PNG image exporter will export PNGs for all cue points

Definition at line 391 of file png_export.h.

20.33 Resample

20.33.1 Detailed Description

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |     Range
==============================================================================
FE2_RESAMPLE_RATE                   | Numeric   | Optional | [8000,192000]Hz
FE2_RESAMPLE_CHANNELS               | Numeric   | Optional | [1,2]
```

Usage notes:

- FLV/SWF files are limited to sample rates of Hertz11025, Hertz22050 and Hertz44100.

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_RESAMPLE);
// Resample to 22050Hz
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_RESAMPLE_RATE, Hertz22050);
```

Defines

- #define FE2_FILTER_RESAMPLE

  Filter name for use with Flix2_AddFilter().

- #define FE2_RESAMPLE_RATE

  Desired sample rate in Hertz (Hz).

- #define FE2_RESAMPLE_CHANNELS

  Filter parameter for channels.

20.33.2 Define Documentation

20.33.2.1 #define FE2_FILTER_RESAMPLE

Filter name for use with Flix2_AddFilter().

Definition at line 50 of file resample.h.

20.33.2.2 #define FE2_RESAMPLE_CHANNELS

Filter parameter for channels.

Definition at line 58 of file resample.h.

20.33.2.3 #define FE2_RESAMPLE_RATE

Desired sample rate in Hertz (Hz).

Definition at line 54 of file resample.h.

20.34 Rotate

20.34.1 Detailed Description

The rotate filter rotates the source clockwise by the angle specified by FE2_ROTATE_ANGLE.

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |     Range
==============================================================================
FE2_ROTATE_ANGLE                    | Numeric   | Optional | {0,90,180,270}
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_ROTATE);
// rotate source 90 degrees clockwise
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_ROTATE_ANGLE, 90.0);
```

Defines

- #define FE2_FILTER_ROTATE

  Filter name for use with Flix2_AddFilter().

- #define FE2_ROTATE_ANGLE

  Specifies the image rotation angle in degrees.

20.34.2 Define Documentation

20.34.2.1 #define FE2_FILTER_ROTATE

Filter name for use with Flix2_AddFilter().

Definition at line 45 of file rotate.h.

20.34.2.2 #define FE2_ROTATE_ANGLE

Specifies the image rotation angle in degrees.

The source is rotated clockwise.

Note:

Default: 0.0 (no rotation)
Valid range: {0,90,180,270}

Definition at line 52 of file rotate.h.

20.35 Scale

20.35.1 Detailed Description

The video scale (resize) filter implements a high quality bicubic scaler to change the dimensions of the video.

Filter Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
=====================================================================================
FE2_SCALE_WIDTH                     | Numeric   | Optional | [-19,)
FE2_SCALE_HEIGHT                    | Numeric   | Optional | [-19,)
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_SCALE);
// scale to 320xXXX maintaining the display aspect ratio
// e.g., 640x480 -> 320x240 (4:3)
//       640x360 -> 320x180 (16:9)
if(sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_SCALE_WIDTH, 320);
if (sc == ON2_OK)
    sc = Flix2_FilterSetParam(filter, FE2_SCALE_HEIGHT, -2);
```

20.35.2 Reserved values

The following values have special meaning and are applicable to both FE2_SCALE_WIDTH and FE2_-SCALE_HEIGHT:

- `0`: display dimension
- `-1`: original dimension (default)
- `-2`: calculate dimension based on its complement and the display aspect ratio
- `-3`: calculate dimension based on its complement and the original aspect ratio
- `n-4*x`: where `n` is one of the values above with the resulting dimension being an even multiple of `2^x`

Display aspect ratio is calculated based on the desired resolution, as set by the container, e.g., the input is encoded at 4:3, but displays at 16:9. Needless to say, this is clip dependent and requires the correct values be set within the container and forwarded from the input source. If unavailable this value will be set to the encoded resolution.

Attention:

Only one dimension may be set to `{-2-4*[0,4],-3-4*[0,4]}`. The other MUST be `{0-4*[0,4],-1-4*[0,4]}` or > `0`.

Deprecated functions

- on2sc video_options_GetImageHeight (const FLIX2HANDLE flix, int32_t *lpImageHeight)

  Get the current scaled image height.

- on2sc video_options_SetImageHeight (FLIX2HANDLE flix, const int32_t lImageHeight)

  Set the scaled image height.

- on2sc video_options_GetImageWidth (const FLIX2HANDLE flix, int32_t *lpImageWidth)

  Get the current scaled image width.

- on2sc video_options_SetImageWidth (FLIX2HANDLE flix, const int32_t lImageWidth)

  Set the scaled image width.

- on2sc video_options_GetUseSourceDimensions (const FLIX2HANDLE flix, on2bool *bpUseSourceDimensions)

  Determine if the source's dimensions will be used instead of the scaled dimensions.

- on2sc video_options_SetUseSourceDimensions (FLIX2HANDLE flix, const on2bool bUseSourceDimensions)

  Switch between source and scaled dimensions.

Defines

- #define FE2_FILTER_SCALE

  Filter name for use with Flix2_AddFilter().

- #define FE2_SCALE_WIDTH

  Filter parameter name for scaled width.

- #define FE2_SCALE_HEIGHT

  Filter parameter name for scaled height.

20.35.3 Define Documentation

20.35.3.1 #define FE2_FILTER_SCALE

Filter name for use with Flix2_AddFilter().

Definition at line 73 of file scale.h.

20.35.3.2 #define FE2_SCALE_HEIGHT

Filter parameter name for scaled height.

Use FE2_SCALE_HEIGHT with Flix2_FilterSetParam() and Flix2_FilterGetParam() to set and/or get the scaled height of the video.

Note:

The value returned by a call to Flix2_FilterGetParam() when this name is specified should only be considered valid if it follows a call to Flix2_FilterSetParam().

Attention:

This function can NOT be used to obtain information about the unscaled input video, use video_options_GetSourceHeight().

See also:

Reserved values

Definition at line 103 of file scale.h.

20.35.3.3 #define FE2_SCALE_WIDTH

Filter parameter name for scaled width.

Use FE2_SCALE_WIDTH with Flix2_FilterSetParam() and Flix2_FilterGetParam() to set and/or get the scaled width of the video.

Note:

The value returned by a call to Flix2_FilterGetParam() when this name is specified should only be considered valid if it follows a call to Flix2_FilterSetParam().

Attention:

This function can NOT be used to obtain information about the unscaled input video, use video_options_GetSourceWidth().

See also:

Reserved values

Definition at line 88 of file scale.h.

20.35.4 Function Documentation

20.35.4.1 on2sc video_options_GetImageHeight (const FLIX2HANDLE *flix*, int32_t * *lpImageHeight*)

Get the current scaled image height.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpImageHeight*** Image height (in pixels)

Return values:

ON2_OK The height value was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This value should only be considered valid if it follows a call to video_options_SetImageHeight(). It can not be used to get information about the unscaled input video.
If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

20.35.4.2 on2sc video_options_GetImageWidth (const FLIX2HANDLE *flix*, int32_t * *lpImageWidth*)

Get the current scaled image width.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpImageWidth*** Image width (in pixels)

Return values:

ON2_OK The width value was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This value should only be considered valid if it follows a call to video_options_SetImageWidth(). It can not be used to get information about the unscaled input video.
If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

20.35.4.3 on2sc video_options_GetUseSourceDimensions (const FLIX2HANDLE *flix*, on2bool * *bpUseSourceDimensions*)

Determine if the source's dimensions will be used instead of the scaled dimensions.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()
→ ***bpUseSourceDimensions*** Variable to update with the current dimension status

Return values:

ON2_OK The dimension status was successfully retrieved from the engine.
ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

The flix engine encodes video using the video source dimensions by default. If FE2_FILTER_SCALE has not been added to the filter chain, video will be encoded using source dimensions.

20.35.4.4 on2sc video_options_SetImageHeight (FLIX2HANDLE *flix*, const int32_t *lImageHeight*)

Set the scaled image height.

Determines the scaled height of the output video.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()
← ***lImageHeight*** Image height (in pixels)

Return values:

ON2_OK The image height value was successfully set in the engine.
ON2_INVALID_PARAMS The value is out of range.
ON2_NET_ERROR The underlying communication layer failed.

Note:

The default is to use the source's height.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

20.35.4.5 on2sc video_options_SetImageWidth (FLIX2HANDLE *flix*, const int32_t *lImageWidth*)

Set the scaled image width.

Determines the scaled width of the output video.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lImageWidth*** Image width (in pixels)

Return values:

ON2_OK The image width value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

The default is to use the source's width.

Deprecated

Please use:

- Flix2_AddFilter()
- Flix2_FilterGetParam()
- Flix2_FilterSetParam()

With the constants:

- FE2_FILTER_SCALE
- FE2_SCALE_WIDTH
- FE2_SCALE_HEIGHT

for access to the flix engine video scaling filter.

20.35.4.6 on2sc video_options_SetUseSourceDimensions (FLIX2HANDLE *flix*, const on2bool *bUseSourceDimensions*)

Switch between source and scaled dimensions.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseSourceDimensions*** New dimension status. on2true to use source dimensions, on2false to use scaled dimensions.

Return values:

ON2_OK The dimension status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the source dimensions will be used, unless a call to video_options_SetImageHeight() or video_options_SetImageWidth() has been made.

Deprecated

The flix engine encodes video using the video source dimensions by default. If FE2_FILTER_SCALE has not been added to the filter chain, video will be encoded using source dimensions.

Attention:

Use of this function will cause FE2_FILTER_SCALE settings to be ignored. If FE2_FILTER_SCALE has been added to the filter chain via the Filter Interface this function is unnecessary. Scaling can be disabled by calling Flix2_RemoveFilter() with the handle returned from Flix2_AddFilter().

20.36 Sharpen

20.36.1 Detailed Description

Sharpens the source image using a 3x3 kernel with the values:

```
-1/8 -1/8 -1/8
-1/8 16/8 -1/8
-1/8 -1/8 -1/8
```

Example Usage:

```
sc = Flix2_AddFilter(&filter, flix, FE2_FILTER_SHARPEN);
if(sc == ON2_OK)
    ; //a sharpening filter will be applied to the source image
```

Defines

- #define FE2_FILTER_SHARPEN

  Filter name for use with Flix2_AddFilter().

20.36.2 Define Documentation

20.36.2.1 #define FE2_FILTER_SHARPEN

Filter name for use with Flix2_AddFilter().

Definition at line 41 of file sharpen.h.

20.37 Flix Engine API

20.37.1 Detailed Description

The Flix Engine for Linux consists of two main parts: the encoding server (*flixd*) and the client library (*libflixengine2.so*). The client library is what the user application interfaces with, using the Flix Engine API. Further details can be found in the system overview and install guides included in this release.

See also:

Support Options & FAQ

Version:

8.0.17.2

Modules

- Audio Encoding Options
- Codecs
- Encoding Statistics
- Filters
- Main Engine Interface
- Filter Manipulation
- Codec Manipulation
- Muxer Manipulation
- Muxers
- Video Encoding Options

20.38 Main Engine Interface

20.38.1 Detailed Description

LINUX

Enumerations

- enum FE2_ExportedVideoType {
  ExportSWF3To6Video,
  ExportSWF7PlusVideo,
  ExportSWFVectorVideo,
  ExportFLVVideo }

  Output file types for use in calls to Flix2_SetExportVideoType() and Flix2_GetExportVideoType().

- enum FE2_EncState {
  EncStateIdle,
  EncStateRunning,
  EncStateQueued }

  Encoder state returned by Flix2_GetEncoderState().

- enum FE2_errno {
  ErrNone,
  ErrSys,
  ErrFileIO,
  ErrFileOpen,
  ErrFileDecode,
  ErrFileDecodeA,
  ErrFileDecodeV,
  ErrEncodeA,
  ErrEncodeV }

  Flix engine error state returned by Flix2_Errno().

Functions

- on2sc Flix2_Create (FLIX2HANDLE *pFlix)

  Create a handle to the flix engine.

- on2sc Flix2_CreateEx (FLIX2HANDLE *pFlix, const char *rpchost, int32_t timeout)

  Create a handle to the flix engine.

- on2sc Flix2_Destroy (FLIX2HANDLE flix)

  Frees resources associated with a `FLIX2HANDLE`.

- const char * Flix2_Version ()

  Returns the library version as a string.

- const on2tc * Flix2_Copyright ()

  Returns copyright information for this library as a string.

- on2sc Flix2_SetOutputFile (FLIX2HANDLE flix, const on2tc *outputFile)

  Set the destination file for the encode session.

- on2sc Flix2_GetOutputFile (const FLIX2HANDLE flix, on2tc *pOutputFile, int32_t *len)

  Get the destination file for the encode session.

- on2sc Flix2_SetOverwriteExistingFiles (FLIX2HANDLE flix, const on2bool bOverwriteExistingFiles)

  Enable/disable overwriting of existing output files.

- on2sc Flix2_GetOverwriteExistingFiles (const FLIX2HANDLE flix, on2bool *bpOverwriteExistingFiles)

  Retrieve the engine's current behavior regarding existing output files.

- on2sc Flix2_SetExportAudio (FLIX2HANDLE flix, const on2bool bExportAudio)

  Enable/disable audio output in the current encoding session.

- on2sc Flix2_GetExportAudio (const FLIX2HANDLE flix, on2bool *bpExportAudio)

  Retrieve the engine's current behavior regarding audio output.

- on2sc Flix2_SetExportVideo (FLIX2HANDLE flix, const on2bool bExportVideo)

  Enable/disable video output in the current encoding session.

- on2sc Flix2_GetExportVideo (const FLIX2HANDLE flix, on2bool *bpExportVideo)

  Retrieve the engine's current behavior regarding video output.

- on2sc Flix2_SetExportVideoType (FLIX2HANDLE flix, const FE2_ExportedVideoType exportVideoType)

  Set the output file type.

- on2sc Flix2_GetExportVideoType (const FLIX2HANDLE flix, FE2_ExportedVideoType *pExportVideoType)

  Retrieve the current output file type.

- on2sc Flix2_SetInputFile (FLIX2HANDLE flix, const on2tc *inputFile)

  Set the source file for the encode session.

- on2sc Flix2_GetInputFile (const FLIX2HANDLE flix, on2tc *pInputFile, int32_t *len)

  Get the source file for the encode session.

- on2sc Flix2_GetSourceDuration (const FLIX2HANDLE flix, int32_t *duration)

  Get the duration, in milliseconds, of the source file.

- on2sc Flix2_Encode (FLIX2HANDLE flix)

Start encoding an output file.

- on2sc Flix2_StopEncoding (FLIX2HANDLE flix)
  Cancel a running encoding session.
- on2sc Flix2_Reset (FLIX2HANDLE flix)
  Reset the engine to its defaults.
- on2sc Flix2_Validate (const FLIX2HANDLE flix)
  Validate the current encoder settings.
- on2sc Flix2_IsEncoderRunning (const FLIX2HANDLE flix, on2bool *bpIsEncoderRunning)
  Check the status of an encode.
- on2sc Flix2_GetEncoderState (const FLIX2HANDLE flix, FE2_EncState *pEncState)
  Retrieve the current state of the encoder.
- on2sc Flix2_Errno (const FLIX2HANDLE flix, FE2_errno *flixerrno, int32_t *syserrno)
  Retrieve the current error state of the engine.

20.38.2 Enumeration Type Documentation

20.38.2.1 enum FE2_EncState

Encoder state returned by Flix2_GetEncoderState().

Enumerator:

EncStateIdle the encoder is not running
EncStateRunning the encoder is running
EncStateQueued the session has been queued until space is available

Definition at line 101 of file flixengine2.h.

20.38.2.2 enum FE2_errno

Flix engine error state returned by Flix2_Errno().

Enumerator:

ErrNone internal error was not set by the library
ErrSys generic system error
ErrFileIO generic file io error
ErrFileOpen error opening file
ErrFileDecode
ErrFileDecodeA
ErrFileDecodeV
ErrEncodeA
ErrEncodeV

Definition at line 108 of file flixengine2.h.

20.38.2.3 enum FE2_ExportedVideoType

Output file types for use in calls to Flix2_SetExportVideoType() and Flix2_GetExportVideoType().

Deprecated

Use the Muxer Interface. This enumeration will be removed in a future release. See also: Muxers.

Enumerator:

ExportSWF3To6Video

ExportSWF7PlusVideo

ExportSWFVectorVideo

ExportFLVVideo

Definition at line 93 of file flixengine2.h.

20.38.3 Function Documentation

20.38.3.1 const on2tc* Flix2_Copyright ()

Returns copyright information for this library as a string.

20.38.3.2 on2sc Flix2_Create (FLIX2HANDLE * *pFlix*)

Create a handle to the flix engine.

Initializes a `FLIX2HANDLE` for use in subsequent `FLIXENGINE_API` calls.

Parameters:

→ ***pFlix*** storage location to receive the value of the created `FLIX2HANDLE`

Return values:

ON2_OK the engine successfully created and initialized a `FLIX2HANDLE` for use in `FLIXENGINE_API` functions.

ON2_INVALID_PARAMS should the precondition fail

ON2_NO_MEM memory could not be allocated for the `FLIX2HANDLE`

ON2_NET_ERROR the underlying communication layer failed. Call Flix2_Errno() for further information.

Common `clnt_stat` values in this case are:

- RPC_SYSTEMERROR (12) errno=ECONNREFUSED (111) - connection to portmap on *rpchost* failed
- RPC_UNKNOWNHOST (13) - *rpchost* name lookup failed
- RPC_PROGNOTREGISTERED (15) - *flixd* is not registered with *portmap* on *rpchost*

For a complete listing see: `rpc/clnt.h`

Precondition:

pFlix is not `NULL`

Remarks:

Linux Only: Attempts first to connect *portmap* on localhost. Upon successful query of *flixd's* port, initiates a connection to the rpc server, *flixd*, which registers the encoding session. Failing either of these ON2_NET_ERROR is returned.

20.38.3.3 on2sc Flix2_CreateEx (FLIX2HANDLE * *pFlix*, const char * *rpchost*, int32_t *timeout*)

Create a handle to the flix engine.

Initializes a `FLIX2HANDLE` for use in subsequent `FLIXENGINE_API` calls.

Parameters:

→ ***pFlix*** storage location to receive the value of the created `FLIX2HANDLE`

← ***rpchost*** hostname/IPv4 address where *flixd* is running `NULL` may be used in which case the connection will be attempted on localhost.

← ***timeout*** timeout in seconds for the underlying communication layer. Passing 0 will use the default, typically 25s.

Return values:

ON2_OK the engine successfully created and initialized a `FLIX2HANDLE` for use in `FLIXENGINE_API` functions.

ON2_INVALID_PARAMS should the precondition fail

ON2_NO_MEM memory could not be allocated for the `FLIX2HANDLE`

ON2_NET_ERROR the underlying communication layer failed. Call Flix2_Errno() for further information.

Common `clnt_stat` values in this case are:

- RPC_SYSTEMERROR (12) errno=ECONNREFUSED (111) - connection to portmap on *rpchost* failed
- RPC_UNKNOWNHOST (13) - *rpchost* name lookup failed
- RPC_PROGNOTREGISTERED (15) - *flixd* is not registered with *portmap* on *rpchost*

For a complete listing see: `rpc/clnt.h`

Precondition:

pFlix is not `NULL`

Note:

Available on Linux only.

Remarks:

Attempts first to connect to portmap on *rpchost*. Upon successful query of *flixd's* port, initiates a connection to the rpc server, flixd, which registers the encoding session. Failing either of these ON2_NET_ERROR is returned.
rpchost may be given in the form:

```
'server:port'
```

to connect to *flixd* directly (if it was run with the `-port` option) to bypass the connection to *portmap*.

20.38.3.4 on2sc Flix2_Destroy (FLIX2HANDLE *flix*)

Frees resources associated with a `FLIX2HANDLE`.

Parameters:

←***flix*** `FLIXHANDLE` to be destroyed

Return values:

ON2_OK the engine successfully destroyed the handle

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Remarks:

If an encoding session started with Flix2_Encode() is still running when this function is called. It will be stopped with Flix2_StopEncoding() before destroying the handle.

20.38.3.5 on2sc Flix2_Encode (FLIX2HANDLE *flix*)

Start encoding an output file.

Encode the input file set in Flix2_SetInputFile() to the output file set in Flix2_SetOutputFile(), using the options set in the accessor functions or the engine defaults where applicable. Before starting the encode session, calls Flix2_Validate() to ensure the engine is ready to encode the input.

Parameters:

←***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK the engine successfully started the encode

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

<***ON2_OK*** Flix2_Validate() failed / there was an error starting the encode session. Call Flix2_Errno() for more detail.

Precondition:

flix is not `NULL`

Note:

Linux Only: upon success *flixd* starts an encode thread on behalf of the `FLIX2HANDLE`.

Detail about the state of the encoder can be obtained by calling Flix2_IsEncoderRunning(), Flix2_-GetEncoderState() or any of the Encoding Statistics functions

20.38.3.6 on2sc Flix2_Errno (const FLIX2HANDLE *flix*, FE2_errno * *flixerrno*, int32_t * *syserrno*)

Retrieve the current error state of the engine.

If an API call has returned an error, this function can be used to retrieve a more specific reason.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***flixerrno*** Storage location to receive the engine error

→ ***syserrno*** Storage location to receive the system wide error

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Remarks:

syserrno is the value of `errno`
If the last library status code was ON2_NET_ERROR *flixerrno* will be the RPC `clnt_stat` returned from same. enum `clnt_stat` values are defined in */usr/include/rpc/clnt.h*.

20.38.3.7 on2sc Flix2_GetEncoderState (const FLIX2HANDLE *flix*, FE2_EncState * *pEncState*)

Retrieve the current state of the encoder.

If using the function to check for encode completion: when EncStateIdle is returned, the completion status can be obtained by calling Flix2_Errno()

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pEncState*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pEncState is not `NULL`

Remarks:

Linux Only: if *flixd* was run with the `-max-sessions` option, EncStateQueued may be returned if the number of concurrent encoding sessions was exceeded. This session will run in the order it was received when a slot becomes available

20.38.3.8 on2sc Flix2_GetExportAudio (const FLIX2HANDLE *flix*, on2bool * *bpExportAudio*)

Retrieve the engine's current behavior regarding audio output.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***bpExportAudio*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
bpExportAudio is not `NULL`

Note:

Default: on2true

20.38.3.9 on2sc Flix2_GetExportVideo (const FLIX2HANDLE *flix*, on2bool * *bpExportVideo*)

Retrieve the engine's current behavior regarding video output.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***bpExportVideo*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
bpExportVideo is not `NULL`

Note:

Default: on2true

20.38.3.10 on2sc Flix2_GetExportVideoType (const FLIX2HANDLE *flix*, FE2_ExportedVideoType * *pExportVideoType*)

Retrieve the current output file type.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()
→ ***pExportVideoType*** Storage location to receive the result

Return values:

ON2_OK on success
ON2_INVALID_PARAMS should one or more of the preconditions fail
ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pExportVideoType is not `NULL`

Note:

Default: ExportFLVVideo

Deprecated

Use the Muxer Interface. This function will be removed in a future release. See also: Muxers.

20.38.3.11 on2sc Flix2_GetInputFile (const FLIX2HANDLE *flix*, on2tc * *pInputFile*, int32_t * *len*)

Get the source file for the encode session.

Returns the input file name.

If *pInputFile* is `NULL` this function will return the size in bytes required to store the current input file in *len*, not including the null terminator.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()
→ ***pInputFile*** Storage location for the input file path
↔ ***len*** on input, the max number of bytes available in *pInputFile*. On return, the number of bytes used.

Return values:

ON2_OK on success
ON2_INVALID_PARAMS should one or more of the preconditions fail
ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
len is not `NULL`

Attention:

Currently only supported in C/C++

20.38.3.12 on2sc Flix2_GetOutputFile (const FLIX2HANDLE *flix*, on2tc * *pOutputFile*, int32_t * *len*)

Get the destination file for the encode session.

Returns the (possibly modified see Flix2_SetOutputFile()) output file name.

If *pOutputFile* is `NULL` this function will return the size in bytes required to store the current output file in *len*, not including the null terminator.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pOutputFile*** Storage location for the output file path

↔ ***len*** on input, the max number of bytes available in *pOutputFile*. On return, the number of bytes used.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
len is not `NULL`

Attention:

Currently only supported in C/C++

20.38.3.13 on2sc Flix2_GetOverwriteExistingFiles (const FLIX2HANDLE *flix*, on2bool * *bpOverwriteExistingFiles*)

Retrieve the engine's current behavior regarding existing output files.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***bpOverwriteExistingFiles*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
bpOverwriteExistingFiles is not `NULL`

Note:

Default: on2false

20.38.3.14 on2sc Flix2_GetSourceDuration (const FLIX2HANDLE *flix*, int32_t * *duration*)

Get the duration, in milliseconds, of the source file.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***duration*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
duration is not `NULL`

Note:

This value is only valid after a call to Flix2_SetInputFile(). If the duration for the file could not be determined *duration* will be set to -1.

20.38.3.15 on2sc Flix2_IsEncoderRunning (const FLIX2HANDLE *flix*, on2bool * *bpIsEncoderRunning*)

Check the status of an encode.

Returns whether an encode session started with Flix2_Encode() is still running. If the function returns on2false the completion status can be obtained by calling Flix2_Errno()

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***bpIsEncoderRunning*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

20.38.3.16 on2sc Flix2_Reset (FLIX2HANDLE *flix*)

Reset the engine to its defaults.

Resets all encoding related options to their default values. Input and output files are retained.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

20.38.3.17 on2sc Flix2_SetExportAudio (FLIX2HANDLE *flix*, const on2bool *bExportAudio*)

Enable/disable audio output in the current encoding session.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bExportAudio*** Enable/Disable

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

Default: on2true

20.38.3.18 on2sc Flix2_SetExportVideo (FLIX2HANDLE *flix*, const on2bool *bExportVideo*)

Enable/disable video output in the current encoding session.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bExportVideo*** Enable/Disable

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not NULL

Note:

Default: on2true

20.38.3.19 on2sc Flix2_SetExportVideoType (FLIX2HANDLE *flix*, const FE2_ExportedVideoType *exportVideoType*)

Set the output file type.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***exportVideoType*** File type to output

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NOT_SUPP if the video type is unsupported by the engine

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not NULL
exportVideoType is a valid member of FE2_ExportedVideoType

Attention:

ExportSWFVectorVideo is unsupported by this version of the engine

Note:

Default: ExportFLVVideo
When using ExportSWF7PlusVideo the video codec should be set to CODEC_H263 with video_options_SetVideoCodec()

Deprecated

Use the Muxer Interface. This function will be removed in a future release. See also: Muxers.

20.38.3.20 on2sc Flix2_SetInputFile (FLIX2HANDLE *flix*, const on2tc * *inputFile*)

Set the source file for the encode session.

Attempts to open the input file, extracting the available source information, e.g. video width/height, duration. Addtionally configures the encoder to match the source attributes by calling video_options_Reset() and audio_options_Reset(). For this reason this function should be called before setting any encoder options.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***inputFile*** Path to the source file.

Return values:

ON2_OK the engine successfully opened the input file for decoding

ON2_INVALID_PARAMS a parameter is invalid.

ON2_NET_ERROR the underlying communication layer failed

<***ON2_OK*** An error occurred opening the file. Call Flix2_Errno() for more detail.

Precondition:

flix is not `NULL`
inputFile is not `NULL`

Remarks:

Linux Only: flixd, the encoding server, gets its input from, and writes its output to, the host file system. *inputFile* must be accessible in this context. An absolute path should be given to help ensure this is the case.

20.38.3.21 on2sc Flix2_SetOutputFile (FLIX2HANDLE *flix*, const on2tc * *outputFile*)

Set the destination file for the encode session.

When Flix2_Encode() is called an attempt is made to open *outputFile*.

If the file exists the behavior is determined by the value of Flix2_GetOverwriteExistingFiles():

- on2true the file will be overwritten
- on2false a new file will be created of the form: *outputFilennn.ext*, e.g. out000.flv

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***outputFile*** Path to the output file.

Return values:

ON2_OK the engine successfully set the output file

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for *outputFile*

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
outputFile is not `NULL`

Remarks:

Linux Only: flixd, the encoding server, gets its input from, and writes its output to, the host file system. *inputFile* must be accessible in this context. An absolute path should be given to help ensure this is the case.

20.38.3.22 on2sc Flix2_SetOverwriteExistingFiles (FLIX2HANDLE *flix*, const on2bool *bOverwriteExistingFiles*)

Enable/disable overwriting of existing output files.

The engine's behavior based on this setting is described by Flix2_SetOutputFile()

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bOverwriteExistingFiles*** Enable/Disable

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

Default: on2false

20.38.3.23 on2sc Flix2_StopEncoding (FLIX2HANDLE *flix*)

Cancel a running encoding session.

Stops an encoding session started by Flix2_Encode(). The output file will be removed.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK the engine successfully stopped the encode

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

20.38.3.24 on2sc Flix2_Validate (const FLIX2HANDLE *flix*)

Validate the current encoder settings.

Ensures that the current options set in the engine will allow for an encoding session to be started.

The validation steps performed are:

- input file has been set
- at least one of Flix2_GetExportVideo() and Flix2_GetExportAudio() is set to on2true
- the export video type is ExportFLVVideo or ExportSWF7PlusVideo
- if Flix2_GetExportVideo() is on2true, calls video_options_Validate()
- if Flix2_GetExportAudio() is on2true, calls audio_options_Validate()

Parameters:

←***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK on success

ON2_ERROR on failure

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

20.38.3.25 const char* Flix2_Version ()

Returns the library version as a string.

20.39 Flix Engine Core Library

20.39.1 Detailed Description

This library, *libflixengine2_core.so*, encompasses the Flix Engine API and adds the Frame Server API.

The Flix Engine Core Library consists of a single library, *libflixengine2_core.so*, that exports the entire Flix Engine API. In addition to this it adds the Frame Server API which allows the user to feed the engine raw audio/video frames as opposed to entire files. This is only available through C/C++. There are no language bindings for the core library.

Modules

- Frame Server API

  Flix Engine Frame Server API.

Functions

- on2sc Flix2_Init ()

  Perform library initialization.

- on2sc Flix2_Deinit ()

  Cleanup resources allocated by Flix2_Init().

- on2sc Flix2_SetLogLevel (FLIX2HANDLE flix, int32_t level)

  Set the library-wide debug log level.

- on2sc Flix2_GetLogLevel (const FLIX2HANDLE flix, int32_t *level)

  Retrieve current library-wide debug log level.

- on2sc Flix2_SetLogPath (FLIX2HANDLE flix, on2tc *logpath)

  Set the library's log file path.

- on2sc Flix2_GetLogPath (const FLIX2HANDLE flix, on2tc *logpath, int32_t *len)

  Retrieve the library's current log file path.

20.39.2 Function Documentation

20.39.2.1 on2sc Flix2_Deinit ()

Cleanup resources allocated by Flix2_Init().

Call once per program instance

Attention:

Available only in the Flix Engine Core Library

20.39.2.2 on2sc Flix2_GetLogLevel (const FLIX2HANDLE *flix*, int32_t * *level*)

Retrieve current library-wide debug log level.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create()

→ ***level*** Storage location to receive the result

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not `NULL`
level is not `NULL`

20.39.2.3 on2sc Flix2_GetLogPath (const FLIX2HANDLE *flix*, on2tc * *logpath*, int32_t * *len*)

Retrieve the library's current log file path.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create()

→ ***logpath*** Storage location for the log file path

↔ ***len*** on input, the max number of bytes available in *logpath*. On return, the number of bytes used.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not `NULL`
len is not `NULL`

20.39.2.4 on2sc Flix2_Init ()

Perform library initialization.

Call once per program instance

Attention:

Available only in the Flix Engine Core Library

20.39.2.5 on2sc Flix2_SetLogLevel (FLIX2HANDLE *flix*, int32_t *level*)

Set the library-wide debug log level.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create()

← ***level*** The log level to set the library to

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not NULL
level is within the range 0-4

Remarks:

Valid values for *level* are:

- 0= none (Default)
- 1= informational
- 2= errors+asserts
- 3= debug
- 4= heavy/program trace logging

20.39.2.6 on2sc Flix2_SetLogPath (FLIX2HANDLE *flix*, on2tc ∗ *logpath*)

Set the library's log file path.

The log file is opened for appending whenever a log message needs to be written and closed immediately thereafter.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create()

← ***logpath*** Path to the log file

Return values:

ON2_OK the engine successfully set the log file

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for *logpath*

Precondition:

flix is not `NULL`
logpath is not `NULL`

Note:

No attempt is made to ensure the file can be created by this function, nor will it create any non-existent directories in *logpath*.

Remarks:

logpath can be the name of a device, e.g. `/dev/stderr`

20.40 Filter Manipulation

Modules

- Deprecated

Functions

- on2sc Flix2_AddFilter (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)

  Add a filter to the encoder's filter chain.

- on2sc Flix2_RemoveFilter (FLIX2PLGNHANDLE plgn)

  Remove a filter to the encoder's filter chain.

- on2sc Flix2_FilterSetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)

  Set the value of a parameter in a filter instance using a string representation.

- on2sc Flix2_FilterGetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)

  Retrieve the value of a parameter in a filter instance represented as a string.

- on2sc Flix2_FilterSetParam (FLIX2PLGNHANDLE plgn, const char *name, double inDblVal)

  Set the value of a parameter in a filter instance using a double representation.

- on2sc Flix2_FilterGetParam (FLIX2PLGNHANDLE plgn, const char *name, double *outDblVal)

  Retrieve the value of a parameter in a filter instance represented as a double.

20.40.1 Function Documentation

20.40.1.1 on2sc Flix2_AddFilter (FLIX2PLGNHANDLE * *pPlgn*, const FLIX2HANDLE *flix*, const char * *plgn_name*)

Add a filter to the encoder's filter chain.

Initializes a `FLIX2PLGNHANDLE` mapped to `plgn_name` for use in subsequent filter calls.

Parameters:

→ ***pPlgn*** Storage location to receive the value of the created `FLIX2PLGNHANDLE`

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***plgn_name*** Name of the filter to add to the chain

Return values:

ON2_OK the engine successfully added the filter and initialized a `FLIX2PLGNHANDLE` for use in filter related functions.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM memory could not be allocated for the `FLIX2PLGNHANDLE`

ON2_NET_ERROR the underlying communication layer failed

Precondition:

pPlgn is not `NULL`
flix is not `NULL`
plgn_name is not `NULL`

See also:

Filters

20.40.1.2 on2sc Flix2_FilterGetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double * *outDblVal*)

Retrieve the value of a parameter in a filter instance represented as a double.

Parameters:

← ***plgn*** Handle to the filter returned from Flix2_AddFilter()

← ***name*** Name of the parameter to retrieve

→ ***outDblVal*** Storage location to receive the value

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the filter does not support this parameter/representation

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`
outDblVal is not `NULL`

See also:

Filters

20.40.1.3 on2sc Flix2_FilterGetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, on2tc * *value*, int32_t * *len*)

Retrieve the value of a parameter in a filter instance represented as a string.

Parameters:

← ***plgn*** Handle to the filter returned from Flix2_AddFilter()

← ***name*** Name of the parameter to retrieve

→ ***value*** Storage location to receive the value

↔ ***len*** Pointer to buffer length variable. Should contain the maximum size of the buffer. This call updates this variable with the size of the returned string.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NOT_SUPP the filter does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`
value is not `NULL`
len is not `NULL`

Attention:

This function is currently unsupported and will return ON2_NOT_SUPP.

See also:

Filters

20.40.1.4 on2sc Flix2_FilterSetParam (FLIX2PLGNHANDLE *plgn*, const char ∗ *name*, double *inDblVal*)

Set the value of a parameter in a filter instance using a double representation.

Parameters:

← ***plgn*** Handle to the filter returned from Flix2_AddFilter()

← ***name*** Name of the parameter to set

← ***inDblVal*** New value for the parameter

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the filter does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

See also:

Filters

20.40.1.5 on2sc Flix2_FilterSetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char ∗ *name*, const on2tc ∗ *value*)

Set the value of a parameter in a filter instance using a string representation.

Parameters:

← ***plgn*** Handle to the filter returned from Flix2_AddFilter()

← ***name*** Name of the parameter to set

← ***value*** New value for the parameter

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the filter does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

See also:

Filters

20.40.1.6 on2sc Flix2_RemoveFilter (FLIX2PLGNHANDLE *plgn*)

Remove a filter to the encoder's filter chain.

Parameters:

← ***plgn*** Handle to the filter returned from Flix2_AddFilter()

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

20.41 Codec Manipulation

Functions

- on2sc Flix2_AddCodec (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)

  Add a codec to the encoder's codec chain.

- on2sc Flix2_RemoveCodec (FLIX2PLGNHANDLE plgn)

  Remove a codec from the encoder's codec chain.

- on2sc Flix2_CodecSetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)

  Set the value of a parameter in a codec instance using a string representation.

- on2sc Flix2_CodecGetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)

  Retrieve the value of a parameter in a codec instance represented as a string.

- on2sc Flix2_CodecSetParam (FLIX2PLGNHANDLE plgn, const char *name, double inDblVal)

  Set the value of a parameter in a codec instance using a double representation.

- on2sc Flix2_CodecGetParam (FLIX2PLGNHANDLE plgn, const char *name, double *outDblVal)

  Retrieve the value of a parameter in a codec instance represented as a double.

20.41.1 Function Documentation

20.41.1.1 on2sc Flix2_AddCodec (FLIX2PLGNHANDLE * *pPlgn*, const FLIX2HANDLE *flix*, const char * *plgn_name*)

Add a codec to the encoder's codec chain.

Initializes a `FLIX2PLGNHANDLE` mapped to `plgn_name` for use in subsequent codec calls.

Parameters:

→ ***pPlgn*** Storage location to receive the value of the created `FLIX2PLGNHANDLE`

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***plgn_name*** Name of the codec to add to the chain

Return values:

ON2_OK the engine successfully added the codec and initialized a `FLIX2PLGNHANDLE` for use in codec related functions.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM memory could not be allocated for the `FLIX2PLGNHANDLE`

ON2_NET_ERROR the underlying communication layer failed

Precondition:

pPlgn is not `NULL`
flix is not `NULL`
plgn_name is not `NULL`

See also:

Codecs

20.41.1.2 on2sc Flix2_CodecGetParam (FLIX2PLGNHANDLE *plgn*, const char ∗ *name*, double ∗ *outDblVal*)

Retrieve the value of a parameter in a codec instance represented as a double.

Parameters:

← ***plgn*** Handle to the codec returned from Flix2_AddCodec()

← ***name*** Name of the parameter to retrieve

→ ***outDblVal*** Storage location to receive the value

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the codec does not support this parameter/representation

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`
outDblVal is not `NULL`

See also:

Codecs

20.41.1.3 on2sc Flix2_CodecGetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char ∗ *name*, on2tc ∗ *value*, int32_t ∗ *len*)

Retrieve the value of a parameter in a codec instance represented as a string.

Parameters:

← ***plgn*** Handle to the codec returned from Flix2_AddCodec()

← ***name*** Name of the parameter to retrieve

→ ***value*** Storage location to receive the value

↔ ***len*** Pointer to buffer length variable. Should contain the maximum size of the buffer. This call updates this variable with the size of the returned string.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NOT_SUPP the codec does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`
value is not `NULL`
len is not `NULL`

Attention:

This function is currently unsupported and will return ON2_NOT_SUPP.

See also:

Codecs

20.41.1.4 on2sc Flix2_CodecSetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double *inDblVal*)

Set the value of a parameter in a codec instance using a double representation.

Parameters:

← ***plgn*** Handle to the codec returned from Flix2_AddCodec()

← ***name*** Name of the parameter to set

← ***inDblVal*** New value for the parameter

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the codec does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

See also:

Codecs

20.41.1.5 on2sc Flix2_CodecSetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, const on2tc * *value*)

Set the value of a parameter in a codec instance using a string representation.

Parameters:

← ***plgn*** Handle to the codec returned from Flix2_AddCodec()

← ***name*** Name of the parameter to set

← ***value*** New value for the parameter

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the codec does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

See also:

Codecs

20.41.1.6 on2sc Flix2_RemoveCodec (FLIX2PLGNHANDLE *plgn*)

Remove a codec from the encoder's codec chain.

Parameters:

← ***plgn*** Handle to the codec returned from Flix2_AddCodec()

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

20.42 Muxer Manipulation

Functions

- on2sc Flix2_AddMuxer (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)

  Set the desired muxer.

- on2sc Flix2_RemoveMuxer (FLIX2PLGNHANDLE plgn)

  Remove the specified muxer.

- on2sc Flix2_MuxerSetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)

  Set the value of a parameter in a muxer instance using a string representation.

- on2sc Flix2_MuxerGetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)

  Retrieve the value of a parameter in a muxer instance represented as a string.

- on2sc Flix2_MuxerSetParam (FLIX2PLGNHANDLE plgn, const char *name, double inDblVal)

  Set the value of a parameter in a muxer instance using a double representation.

- on2sc Flix2_MuxerGetParam (FLIX2PLGNHANDLE plgn, const char *name, double *outDblVal)

  Retrieve the value of a parameter in a muxer instance represented as a double.

20.42.1 Function Documentation

20.42.1.1 on2sc Flix2_AddMuxer (FLIX2PLGNHANDLE * *pPlgn*, const FLIX2HANDLE *flix*, const char * *plgn_name*)

Set the desired muxer.

Initializes a `FLIX2PLGNHANDLE` mapped to `plgn_name` for use in subsequent muxer calls.

Parameters:

→ ***pPlgn*** Storage location to receive the value of the created `FLIX2PLGNHANDLE`

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***plgn_name*** Name of the desired muxer

Return values:

ON2_OK the engine successfully added the muxer and initialized a `FLIX2PLGNHANDLE` for use in muxer related functions.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM memory could not be allocated for the `FLIX2PLGNHANDLE`

ON2_NET_ERROR the underlying communication layer failed

Precondition:

pPlgn is not NULL
flix is not NULL
plgn_name is not NULL

See also:

Muxers

20.42.1.2 on2sc Flix2_MuxerGetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double * *outDblVal*)

Retrieve the value of a parameter in a muxer instance represented as a double.

Parameters:

← ***plgn*** Handle to the muxer returned from Flix2_AddMuxer()

← ***name*** Name of the parameter to retrieve

→ ***outDblVal*** Storage location to receive the value

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the muxer does not support this parameter/representation

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not NULL
outDblVal is not NULL

See also:

Codecs

20.42.1.3 on2sc Flix2_MuxerGetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, on2tc * *value*, int32_t * *len*)

Retrieve the value of a parameter in a muxer instance represented as a string.

Parameters:

← ***plgn*** Handle to the muxer returned from Flix2_AddMuxer()

← ***name*** Name of the parameter to retrieve

→ ***value*** Storage location to receive the value

↔ ***len*** Pointer to buffer length variable. Should contain the maximum size of the buffer. This call updates this variable with the size of the returned string.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NOT_SUPP the muxer does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`
value is not `NULL`
len is not `NULL`

Attention:

This function is currently unsupported and will return ON2_NOT_SUPP.

See also:

muxer

20.42.1.4 on2sc Flix2_MuxerSetParam (FLIX2PLGNHANDLE *plgn*, const char * *name*, double *inDblVal*)

Set the value of a parameter in a muxer instance using a double representation.

Parameters:

← ***plgn*** Handle to the muxer returned from Flix2_AddMuxer()

← ***name*** Name of the parameter to set

← ***inDblVal*** New value for the parameter

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the muxer does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

See also:

Muxers

20.42.1.5 on2sc Flix2_MuxerSetParamAsStr (FLIX2PLGNHANDLE *plgn*, const char * *name*, const on2tc * *value*)

Set the value of a parameter in a muxer instance using a string representation.

Parameters:

← ***plgn*** Handle to the muxer returned from Flix2_AddMuxer()

← ***name*** Name of the parameter to set

← ***value*** New value for the parameter

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NOT_SUPP the muxer does not support this parameter

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

See also:

Muxers

20.42.1.6 on2sc Flix2_RemoveMuxer (FLIX2PLGNHANDLE *plgn*)

Remove the specified muxer.

Parameters:

← ***plgn*** Handle to the muxer returned from Flix2_AddMuxer()

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

plgn is not `NULL`

20.43 Frame Server API

20.43.1 Detailed Description

Flix Engine Frame Server API.

The Frame Server API allows an application to provide the engine with raw audio and video frames rather than entire files. This service is only available via the core Flix Engine library (*libflixengine2_-core.so*). It is not available through the daemon (*flixd*) / rpc client library (*libflixengine2.so*) pair or the language extensions that are built to interface with *libflixengine2.so*.

20.43.2 Example Usage

```
//initialize the library, create a handle to the engine and set our outfile
Flix2_Init();
Flix2_Create(&flix);
Flix2_SetOutputFile(flix,"frameserver-out.flv");

//enable the frame server
Flix2_SetFrameServer(flix,on2true);

//setup our video input, 320x240@29.97fps, bgra
Flix2_SetVideoProperties(flix,VideoFormatBGRA,320,240,320*4,2997,1000);
//setup our audio input, 2ch/16bitsps/44.1kHz
Flix2_SetAudioProperties(flix,2,16,44100);

//encode audio/video using the library's default encoding parameters
Flix2_EncodeFrame(flix, FrameTypeVideo, on2false,
    vidframe, 320*240*4, 0);
Flix2_EncodeFrame(flix, FrameTypeAudio, on2false,
    audframe, audframesiz, 0);
...
Flix2_EncodeFrame(flix, FrameTypeVideo, on2false,
    vidframe, 320*240*4, 300300);
Flix2_EncodeFrame(flix, FrameTypeAudio, on2false,
    audframe, audframesiz, 0);
...
//set eos on both inputs
Flix2_EncodeFrame(flix, FrameTypeVideo, on2true,
    NULL, 0, 303303);
Flix2_EncodeFrame(flix, FrameTypeAudio, on2true,
    NULL, 0, 0);
//finish encoding any queued frames
do {
    Flix2_IsEncoderRunning(flix,&b);
} while(b);

//cleanup
Flix2_Destroy(flix);
Flix2_Deinit();
```

Attention:

The Frame Server API does not currently support resampling of the input audio.

Note:

As the Frame Server API deals with raw video and audio, which use a large amount of storage, no sample is distributed with the Flix Engine. If you are interested in using the Frame Server API and require a sample please contact support.

Enumerations

- enum FE2_FrameType {
  FrameTypeAudio,
  FrameTypeVideo }
  Frame type being passed to Flix2_EncodeFrame().

- enum FE2_VideoFormat {
  VideoFormatYV12,
  VideoFormatBGRA }
  Format of input video used in calls to Flix2_SetVideoProperties().

Functions

- on2sc Flix2_SetFrameServer (FLIX2HANDLE flix, on2bool enable)
  Enable/disable the frame server for this session.

- on2sc Flix2_GetFrameServer (FLIX2HANDLE flix, on2bool *enabled)
  Retrieve the current frame server status.

- on2sc Flix2_SetAudioProperties (FLIX2HANDLE flix, uint8_t channels, int16_t bitspersample, int32_t samplerate)
  Set the input audio properties for the frame server.

- on2sc Flix2_SetVideoProperties (FLIX2HANDLE flix, FE2_VideoFormat format, int32_t width, int32_t height, int32_t pitch, int32_t rate, int32_t scale)
  Set the input video properties for the frame server.

- on2sc Flix2_EncodeFrame (FLIX2HANDLE flix, FE2_FrameType type, on2bool eos, uint8_t *frame, uint32_t len, int64_t ts90k)
  Queue an input frame for encoding.

20.43.3 Enumeration Type Documentation

20.43.3.1 enum FE2_FrameType

Frame type being passed to Flix2_EncodeFrame().

Attention:

Available only in the Flix Engine Core Library

Enumerator:

FrameTypeAudio

FrameTypeVideo

Definition at line 88 of file fs_options.h.

20.43.3.2 enum FE2_VideoFormat

Format of input video used in calls to Flix2_SetVideoProperties().

Attention:

Available only in the Flix Engine Core Library

Enumerator:

VideoFormatYV12

VideoFormatBGRA

Definition at line 95 of file fs_options.h.

20.43.4 Function Documentation

20.43.4.1 on2sc Flix2_EncodeFrame (FLIX2HANDLE *flix*, FE2_FrameType *type*, on2bool *eos*, uint8_t * *frame*, uint32_t *len*, int64_t *ts90k*)

Queue an input frame for encoding.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***type*** Input type

← ***eos*** End of stream indicator. Must be set for all streams to complete an encode (a null frame may be passed after all valid frames have been sent)

← ***frame*** Pointer to raw data buffer. (1 video frame or a chunk of audio samples based on the type parameter).

← ***len*** Size in bytes of the raw data pointed to by *frame*

← ***ts90k*** Timestamp for this frame in units of 90kHz. Currently only used for FrameTypeVideo

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

<***ON2_OK*** if there was an error completing encoder setup. Call Flix2_Errno() for further detail

Precondition:

flix is not `NULL`
frame is not `NULL` or *eos* is set
input of type *type* was setup using the corresponding Flix2_SetXXXProperties call

Attention:

Available only in the Flix Engine Core Library

Note:

The engine handles interleaving of the output file so input need not be perfectly interleaved. Making a reasonable attempt at interleaving the input will reduce memory usage, however.
Flix2_IsEncoderRunning() should be called after setting eos to ensure all frames have been encoded.

See also:

Example Usage

20.43.4.2 on2sc Flix2_GetFrameServer (FLIX2HANDLE *flix*, on2bool * *enabled*)

Retrieve the current frame server status.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***enabled*** Enabled (`on2true`) or disabled (`on2false`)

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not `NULL`

Attention:

Available only in the Flix Engine Core Library

Note:

Default: `on2false` (disabled)
If enabled the user should call at least one of Flix2_SetAudioProperties() or Flix2_-SetVideoProperties() before calling Flix2_EncodeFrame()

20.43.4.3 on2sc Flix2_SetAudioProperties (FLIX2HANDLE *flix*, uint8_t *channels*, int16_t *bitspersample*, int32_t *samplerate*)

Set the input audio properties for the frame server.

When FrameTypeAudio is passed to Flix2_EncodeFrame() the frame server will assume the input is of this format. The sample rate given will be used to calculate a timestamp.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***channels*** Number of audio channels (1/2)

← ***bitspersample*** The bits per input sample (8/16)

← ***samplerate*** The input sample rate in Hz (11025/22050/44100)

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not `NULL`
channels is >0 and <= 2
bitspersample is 8 or 16
samplerate is 11025, 22050 or 44100

Attention:

Available only in the Flix Engine Core Library

Remarks:

Taken together with Flix2_SetVideoProperties(), can be viewed as the analogue of Flix2_SetInputFile()

Note:

MUST be called before Flix2_EncodeFrame() if audio output is desired
Calling this function after calling Flix2_EncodeFrame(FrameTypeAudio) is NOT supported

20.43.4.4 on2sc Flix2_SetFrameServer (FLIX2HANDLE *flix*, on2bool *enable*)

Enable/disable the frame server for this session.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***enable*** Enable (`on2true`) or disable (`on2false`) frame server mode

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should the precondition fail

Precondition:

flix is not `NULL`

Attention:

Available only in the Flix Engine Core Library

Note:

Default: `on2false` (disabled)
If enabled the user should call at least one of Flix2_SetAudioProperties() or Flix2_SetVideoProperties() before calling Flix2_EncodeFrame()

20.43.4.5 on2sc Flix2_SetVideoProperties (FLIX2HANDLE *flix*, FE2_VideoFormat *format*, int32_t *width*, int32_t *height*, int32_t *pitch*, int32_t *rate*, int32_t *scale*)

Set the input video properties for the frame server.

When FrameTypeVideo is passed to Flix2_EncodeFrame() the frame server will assume the input is of this format.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***format*** Raw video format

← ***width*** Video width

← ***height*** Video height

← ***pitch*** Video pitch (stride) in bytes. For YV12 stride is typically equal to width, for BGRA it's width $*$ 4

← ***rate*** Video rate (rate / scale = framerate)

← ***scale*** Video scale (rate / scale = framerate)

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

Precondition:

flix is not `NULL`
format is a valid member of FE2_VideoFormat
width and *height* are > 0
pitch is non-zero

Attention:

Available only in the Flix Engine Core Library

Remarks:

Taken together with Flix2_SetAudioProperties(), can be viewed as the analogue of Flix2_SetInputFile()

Note:

MUST be called before Flix2_EncodeFrame() if video output is desired
Calling this function after calling Flix2_EncodeFrame(FrameTypeVideo) is NOT supported
If the input video is of variable framerate rate and scale may be omitted (i.e. set to 0)

20.44 Deprecated

Functions

- on2sc editor_options_Reset (FLIX2HANDLE flix)
  Reset the media editor options to their defaults.
- on2sc editor_options_Validate (const FLIX2HANDLE flix)
  Ensure the current media editor settings are valid.

20.44.1 Function Documentation

20.44.1.1 on2sc editor_options_Reset (FLIX2HANDLE *flix*)

Reset the media editor options to their defaults.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK The editor options were reset successfully.

ON2_NET_ERROR The underlying communication layer failed.

Deprecated

Use the Filter Interface

20.44.1.2 on2sc editor_options_Validate (const FLIX2HANDLE *flix*)

Ensure the current media editor settings are valid.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK The editor options are valid and encoding may proceed.

ON2_INVALID_PARAMS One or more options are invalid. Proceeding with encoding may cause encoding to fail or a file that does not match the specified options will be created.

ON2_NET_ERROR The underlying communication layer failed.

Deprecated

Use the Filter Interface

20.45 Muxers

20.45.1 Detailed Description

Muxers are configurable both by the muxer interface and individual functions (e.g. (swf|video)_options_-SetXXX). The latter, however, are deprecated and will be removed in a future release. Therefore, any new code should use the muxer interface.

Modules

- FLV
- FXM
- 3GPP - FFmpeg
- 3GPP2 - FFmpeg
- MOV - FFmpeg
- MP4 - FFmpeg
- SWF
- WebM - FFmpeg

20.46 FLV

20.46.1 Detailed Description

Muxer Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
==============================================================================
FE2_FLV_CUEPT_EVENT                 |  String   | Optional | [0.0,)
FE2_FLV_CUEPT_NAV                   |  String   | Optional | [0.0,)
FE2_FLV_CUEPT_PARAM                 |  String   | Optional | N/A
FE2_FLV_METADATA_ENABLE             |  Numeric  | Optional | flvmetadata_t
FE2_FLV_METADATA_DISABLE            |  Numeric  | Optional | flvmetadata_t
```

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_FLV);
// set a navigation cue point at 10s
if(sc == ON2_OK)
    sc = Flix2_MuxerSetParamAsStr(muxer, FE2_FLV_CUEPT_NAV, "nav0=10.0");
```

20.46.2 Supported Codecs

- Video:
  - FE2_CODEC_H263
  - FE2_CODEC_VP6
  - FE2_CODEC_VP6ALPHA
- Audio:
  - FE2_CODEC_AAC
  - FE2_CODEC_AACPLUS
  - FE2_CODEC_LAME
  - FE2_CODEC_PCM

20.46.3 Format Restrictions

- Audio:
  - FE2_CODEC_AAC/FE2_CODEC_AACPLUS: 44.1kHz & 2ch
  - Others: 44.1/22.05/11.025kHz & 1/2ch

  If the input does not meet the above conditions FE2_FILTER_RESAMPLE will be automatically added, overriding user settings if necessary.

Defines

- #define FE2_MUXER_FLV

  FLV muxer. For use with Flix2_AddMuxer().

- #define FE2_FLV_CUEPT_EVENT

  Set an event cue point.

- #define FE2_FLV_CUEPT_NAV

  Set a navigation cue point.

- #define FE2_FLV_CUEPT_PARAM

  Add a name/value pair to an existing cue point.

- #define FE2_FLV_METADATA_ENABLE

  Enable output of meta data element.

- #define FE2_FLV_METADATA_DISABLE

  Disable output of meta data element.

Typedefs

- typedef enum flv_metadata flvmetadata_t

Enumerations

- enum flv_metadata {
  MD_DURATION,
  MD_DATASIZE,
  MD_AUDIO_SIZE,
  MD_VIDEO_SIZE,
  MD_AUDIO_DATARATE,
  MD_VIDEO_DATARATE,
  MD_AUDIO_CODECID,
  MD_VIDEO_CODECID,
  MD_WIDTH,
  MD_HEIGHT,
  MD_FRAMERATE,
  MD_CANSEEKTOEND,
  MD_LASTTIMESTAMP,
  MD_LASTKEYFRAMETIMESTAMP,
  MD_LASTKEYFRAMELOCATION,
  MD_KEYFRAMES }

  Supported FLV onMetaData elements.

20.46.4 Define Documentation

20.46.4.1 #define FE2_FLV_CUEPT_EVENT

Set an event cue point.

Parameter format:

`Format = cueptNAME '=' cueptTIME_SECONDS`
e.g., "evtpt0=343.0".

Precondition:

time must be >= 0.0

Note:

Multiple cue points may be added.

See also:

http://livedocs.macromedia.com/flash/8/main/wwhelp/wwhimpl/common/html/wwhelp.htm?Parts&file=00001574.html

Definition at line 133 of file flv.h.

20.46.4.2 #define FE2_FLV_CUEPT_NAV

Set a navigation cue point.

This parameter allows seeking to the specified time by generating an I-Frame (keyframe) in the video stream and adding the entry to the metadata.

Parameter format:

`Format = cueptNAME '=' cueptTIME_SECONDS`
e.g., "navpt0=343.0".

Precondition:

time must be >= 0.0

Note:

Multiple cue points may be added.

See also:

http://livedocs.macromedia.com/flash/8/main/wwhelp/wwhimpl/common/html/wwhelp.htm?Parts&file=00001574.html

Definition at line 145 of file flv.h.

20.46.4.3 #define FE2_FLV_CUEPT_PARAM

Add a name/value pair to an existing cue point.

Parameter format:

```
Format = cueptNAME *[ '&' Pair ]
Pair = Name '=' Value
```
e.g., "cuept0&name0=value0&name1=value1"

Precondition:

The cue point has already been added

Definition at line 154 of file flv.h.

20.46.4.4 #define FE2_FLV_METADATA_DISABLE

Disable output of meta data element.

Note:

Valid elements are defined by flvmetadata_t

Definition at line 164 of file flv.h.

20.46.4.5 #define FE2_FLV_METADATA_ENABLE

Enable output of meta data element.

Note:

Valid elements are defined by flvmetadata_t

Definition at line 159 of file flv.h.

20.46.4.6 #define FE2_MUXER_FLV

FLV muxer. For use with Flix2_AddMuxer().

Definition at line 124 of file flv.h.

20.46.5 Typedef Documentation

20.46.5.1 typedef enum flv_metadata flvmetadata_t

20.46.6 Enumeration Type Documentation

20.46.6.1 enum flv_metadata

Supported FLV onMetaData elements.

Descriptions indicate the default in bold, followed by the element name in italics, its type and a description of the units if applicable.

Note:

Audio/Video specific entries are disabled should there be no stream of the type

Enumerator:

MD_DURATION **Enabled**. *duration* <Number> seconds

MD_DATASIZE **Enabled**. *datasize* <Number> bytes

MD_AUDIO_SIZE **Enabled**. *audiosize* <Number> bytes

MD_VIDEO_SIZE **Enabled**. *videosize* <Number> bytes

MD_AUDIO_DATARATE **Enabled**. *audiodatarate* <Number> kbps

MD_VIDEO_DATARATE **Enabled**. *videodatarate* <Number> kbps

MD_AUDIO_CODECID **Enabled**. *audiocodecid* <Number>

MD_VIDEO_CODECID **Enabled**. *videocodecid* <Number>

MD_WIDTH **Enabled**. *width* <Number>

MD_HEIGHT **Enabled**. *height* <Number>

MD_FRAMERATE **Enabled**. *framerate* <Number> frames/sec

MD_CANSEEKTOEND **Enabled**. *canSeekToEnd* <Boolean>

Indicates last video tag is a key frame.

MD_LASTTIMESTAMP **Enabled**. *lasttimestamp* <Number> seconds

MD_LASTKEYFRAMETIMESTAMP **Disabled**. *lastkeyframetimestamp* <Number> seconds

MD_LASTKEYFRAMELOCATION **Disabled**. *lastkeyframelocation* <Number> byte offset

MD_KEYFRAMES **Disabled**. *keyframes* <Object>

Contains 2 arrays:

- *filepositions:* <Number> byte offset
- *times:* <Number> timestamp in seconds

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the 'keyframes' object is added, with the result then replacing the original output file.

Definition at line 71 of file flv.h.

20.47 FXM

20.47.1 Detailed Description

The FXM muxer produces output compatible with the Sun Java VM.

Muxer Parameters:

```
Name                                |   Type    | Opt/Reqd |    Range
==============================================================================
FE2_FXM_CUEPT_EVENT                 |  String   | Optional | [0.0,)
FE2_FXM_CUEPT_NAV                   |  String   | Optional | [0.0,)
FE2_FXM_CUEPT_PARAM                 |  String   | Optional | N/A
FE2_FXM_METADATA_ENABLE             |  Numeric  | Optional | fxmmetadata_t
FE2_FXM_METADATA_DISABLE            |  Numeric  | Optional | fxmmetadata_t
```

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_FXM);
if(sc == ON2_OK)
    ; //now using the FXM muxer for output
```

20.47.2 Supported Codecs

- Video:
  - FE2_CODEC_VP6
- Audio:
  - FE2_CODEC_LAME

20.47.3 Format Restrictions

- Audio:
  - 44.1/22.05/11.025kHz & 1/2ch

  If the input does not meet the above conditions FE2_FILTER_RESAMPLE will be automatically added, overriding user settings if necessary.

Defines

- #define FE2_MUXER_FXM

  FXM muxer. For use with Flix2_AddMuxer().

- #define FE2_FXM_CUEPT_EVENT

  Set an event cue point.

- #define FE2_FXM_CUEPT_NAV

  Set a navigation cue point.

- #define FE2_FXM_CUEPT_PARAM

Add a name/value pair to an existing cue point.

- #define FE2_FXM_METADATA_ENABLE

  Enable output of meta data element.

- #define FE2_FXM_METADATA_DISABLE

  Disable output of meta data element.

Typedefs

- typedef enum flv_metadata fxmmetadata_t

20.47.4 Define Documentation

20.47.4.1 #define FE2_FXM_CUEPT_EVENT

Set an event cue point.

Parameter format:

`Format = cueptNAME '=' cueptTIME_SECONDS`
e.g., "evtpt0=343.0".

Precondition:

time must be >= 0.0

Note:

Multiple cue points may be added.

Definition at line 70 of file fxm.h.

20.47.4.2 #define FE2_FXM_CUEPT_NAV

Set a navigation cue point.

This parameter allows seeking to the specified time by generating an I-Frame (keyframe) in the video stream and adding the entry to the metadata.

Parameter format:

`Format = cueptNAME '=' cueptTIME_SECONDS`
e.g., "navpt0=343.0".

Precondition:

time must be >= 0.0

Note:

Multiple cue points may be added.

Definition at line 81 of file fxm.h.

20.47.4.3 #define FE2_FXM_CUEPT_PARAM

Add a name/value pair to an existing cue point.

Parameter format:

```
Format = cueptNAME *[ '&' Pair ]
Pair = Name '=' Value
```
e.g., "cuept0&name0=value0&name1=value1"

Precondition:

The cue point has already been added

Definition at line 90 of file fxm.h.

20.47.4.4 #define FE2_FXM_METADATA_DISABLE

Disable output of meta data element.

Note:

Valid elements are defined by fxmmetadata_t

Definition at line 100 of file fxm.h.

20.47.4.5 #define FE2_FXM_METADATA_ENABLE

Enable output of meta data element.

Note:

Valid elements are defined by fxmmetadata_t

Definition at line 95 of file fxm.h.

20.47.4.6 #define FE2_MUXER_FXM

FXM muxer. For use with Flix2_AddMuxer().

Definition at line 62 of file fxm.h.

20.47.5 Typedef Documentation

20.47.5.1 typedef enum flv_metadata fxmmetadata_t

Definition at line 59 of file fxm.h.

20.48 3GPP - FFmpeg

20.48.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_3GP);
if(sc == ON2_OK)
    ; //Now using the 3GPP muxer for output
```

20.48.2 Supported Codecs

- Video:
  - FE2_CODEC_H263_BASELINE
  - FE2_CODEC_H264
- Audio:
  - FE2_CODEC_AAC
  - FE2_CODEC_AACPLUS
  - FE2_CODEC_AMR_NB

Additional References:

- 3GPP homepage

Defines

- #define FE2_MUXER_3GP

  3GPP muxer. For use with Flix2_AddMuxer()

- #define FE2_3GP_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

20.48.3 Define Documentation

20.48.3.1 #define FE2_3GP_FASTSTART

FE2_ISOMEDIA_FASTSTART alias

Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 67 of file isomedia.h.

20.48.3.2 #define FE2_MUXER_3GP

3GPP muxer. For use with Flix2_AddMuxer()

Definition at line 62 of file isomedia.h.

20.49 3GPP2 - FFmpeg

20.49.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_3G2);
if(sc == ON2_OK)
    ; //Now using the 3GPP2 muxer for output
```

20.49.2 Supported Codecs

- Video:
  - FE2_CODEC_H263_BASELINE
  - FE2_CODEC_H264
- Audio:
  - FE2_CODEC_AAC
  - FE2_CODEC_AACPLUS
  - FE2_CODEC_AMR_NB

Additional References:

- 3GPP2 homepage

Defines

- #define FE2_MUXER_3G2

  3GPP2 muxer. For use with Flix2_AddMuxer()

- #define FE2_3G2_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

20.49.3 Define Documentation

20.49.3.1 #define FE2_3G2_FASTSTART

FE2_ISOMEDIA_FASTSTART alias

Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 102 of file isomedia.h.

20.49.3.2 #define FE2_MUXER_3G2

3GPP2 muxer. For use with Flix2_AddMuxer()

Definition at line 97 of file isomedia.h.

20.50 MOV - FFmpeg

20.50.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_MOV);
if(sc == ON2_OK)
    ; //Now using the MOV muxer for output
```

20.50.2 Supported Codecs

- Video:
  - FE2_CODEC_H263_BASELINE
  - FE2_CODEC_H264
- Audio:
  - FE2_CODEC_AAC
  - FE2_CODEC_AACPLUS
  - FE2_CODEC_AMR_NB

Additional References:

- ISO - International Organization for Standardization
- ISO/IEC 14496-12:ISO base media file format
- QuickTime File Format

Attention:

The current implementation does not write a 'ctts' table (cf., ISO/IEC 14496-12) to the output. This will cause playback under QuickTime to stutter should FE2_H264_B_FRAME_RATE be used. Note playback under the Flash Player is unaffected. This will be addressed in a future release.

Defines

- #define FE2_MUXER_MOV

  MOV muxer. For use with Flix2_AddMuxer().

- #define FE2_MOV_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

20.50.3 Define Documentation

20.50.3.1 #define FE2_MOV_FASTSTART

FE2_ISOMEDIA_FASTSTART alias

Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 146 of file isomedia.h.

20.50.3.2 #define FE2_MUXER_MOV

MOV muxer. For use with Flix2_AddMuxer().

Definition at line 141 of file isomedia.h.

20.51 MP4 - FFmpeg

20.51.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_MP4);
if(sc == ON2_OK)
    ; //Now using the MP4 muxer for output
```

20.51.2 Supported Codecs

- Video:
  - FE2_CODEC_H264
- Audio:
  - FE2_CODEC_AAC
  - FE2_CODEC_AACPLUS

Additional References:

- ISO - International Organization for Standardization
- ISO/IEC 14496-12:ISO base media file format

Attention:

The current implementation does not write a 'ctts' table (cf., ISO/IEC 14496-12) to the output. This will cause playback under QuickTime to stutter should FE2_H264_B_FRAME_RATE be used. Note playback under the Flash Player is unaffected. This will be addressed in a future release.

See also:

H.264 notes for Apple device support

Defines

- #define FE2_MUXER_MP4

  MP4 muxer. For use with Flix2_AddMuxer().

- #define FE2_MP4_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

20.51.3 Define Documentation

20.51.3.1 #define FE2_MP4_FASTSTART

FE2_ISOMEDIA_FASTSTART alias

Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 188 of file isomedia.h.

20.51.3.2 #define FE2_MUXER_MP4

MP4 muxer. For use with Flix2_AddMuxer().

Definition at line 183 of file isomedia.h.

20.52 SWF

20.52.1 Detailed Description

Muxer Parameters:

```
Name                                    |   Type    | Opt/Reqd |        Range
================================================================================
FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR |  Number   | Optional | [0.0,2.0] step:0.1
FE2_SWF_ADD_VARIABLE                    |  String   | Optional | N/A
FE2_SWF_DELETE_VARIABLE                 |  String   | Optional | N/A
FE2_SWF_EMBEDDED_URL                    |  String   | Optional | N/A
FE2_SWF_EMBEDDED_URL_TARGET             |  String   | Optional | N/A
FE2_SWF_EMBEDDED_URL_TYPE               |  String   | Optional | FE2_EmbeddedUrlType
FE2_SWF_FIXED_PRELOAD_PCT               |  Number   | Optional | [0,100]
FE2_SWF_FRAMERATE                       |  Number   | Optional | N/A
FE2_SWF_HEIGHT                          |  Number   | Optional | N/A
FE2_SWF_LOOP_COUNT                      |  Number   | Optional | [0,)
FE2_SWF_ON_END_OPTION                   |  String   | Optional | FE2_SwfOnEndOptions
FE2_SWF_ON_END_URL                      |  String   | Optional | N/A
FE2_SWF_ON_START_OPTION                 |  String   | Optional | FE2_SwfOnStartOptions
FE2_SWF_PRELOAD_TYPE                    |  Number   | Optional | FE2_SwfPreloaderOptions
FE2_SWF_START_BLANK_FRAME               |  Number   | Optional | [0,1]
FE2_SWF_START_WAIT_SEC                  |  Number   | Optional | [0.0,)
FE2_SWF_WIDTH                           |  Number   | Optional | N/A
```

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_SWF);
// enable the adaptive preloader
if(sc == ON2_OK)
    sc = Flix2_MuxerSetParam(muxer, FE2_SWF_PRELOAD_TYPE, SwfAdaptivePreloader);
```

20.52.2 Supported Codecs

- Video:
  - FE2_CODEC_H263
  - FE2_CODEC_VP6
  - FE2_CODEC_VP6ALPHA
- Audio:
  - FE2_CODEC_LAME
  - FE2_CODEC_PCM

Deprecated functions

- on2sc swf_options_Reset (FLIX2HANDLE flix)

  Resets the swf options.

- on2sc swf_options_GetEmbeddedUrl (FLIX2HANDLE flix, char *embeddedUrl, int32_t *len)

  Gets the embedded URL.

- on2sc swf_options_GetEmbeddedUrlTarget (FLIX2HANDLE flix, char *embeddedUrlTarget, int32_t *len)

  Gets the target of embedded URL.

- on2sc swf_options_SetEmbeddedUrl (FLIX2HANDLE flix, const char *embeddedUrl)

  Sets the embedded URL of the SWF or HTML that will be loaded when the user clicks on the video.

- on2sc swf_options_SetEmbeddedUrlTarget (FLIX2HANDLE flix, const char *embeddedUrlTarget)

  Sets the target of the embedded URL.

- on2sc swf_options_GetEmbeddedUrlType (FLIX2HANDLE flix, FE2_EmbeddedUrlType *embeddedUrlType)

  Gets the type of embedded URL.

- on2sc swf_options_SetEmbeddedUrlType (FLIX2HANDLE flix, const FE2_EmbeddedUrlType embeddedUrlType)

  Sets the type of embedded URL.

- on2sc swf_options_GetSwfFramerate (FLIX2HANDLE flix, int32_t *pSwfFramerate)

  Gets the SWF framerate.

- on2sc swf_options_SetSwfFramerate (FLIX2HANDLE flix, const int32_t swfFramerate)

  Sets the SWF framerate.

- on2sc swf_options_GetInsertBlankFrameOnStart (FLIX2HANDLE flix, on2bool *pInsertBlankFrameOnStart)

  Gets if the engine is to insert a blank frame as the first frame of the SWF or not.

- on2sc swf_options_SetInsertBlankFrameOnStart (FLIX2HANDLE flix, const on2bool insertBlankFrameOnStart)

  Tells the engine to insert a blank frame as the first frame of the SWF or not.

- on2sc swf_options_GetSwfFramerateAsDouble (FLIX2HANDLE flix, double *pSwfFramerate)

  Gets the SWF framerate as a double.

- on2sc swf_options_SetSwfFramerateAsDouble (FLIX2HANDLE flix, const double swfFramerate)

  Sets the SWF framerate as a double.

- on2sc swf_options_GetEnablePreloader (FLIX2HANDLE flix, on2bool *pEnablePreloader)

  Gets if a video preloader is enabled or disabled.

- on2sc swf_options_SetEnablePreloader (FLIX2HANDLE flix, const on2bool enablePreloader)

  Enables or disables the video preloader.

- on2sc swf_options_GetPercentToPreload (FLIX2HANDLE flix, int32_t *pPercentToPreload)

  Gets the percent of the SWF movie to preload before playback begins.

- on2sc swf_options_SetPercentToPreload (FLIX2HANDLE flix, const int32_t percentToPreload)

Sets the percent of the SWF movie to preload before playback begins.

- on2sc swf_options_GetPreloaderType (FLIX2HANDLE flix, FE2_SwfPreloaderOptions *pPreloaderType)

  Gets the type of preloader.

- on2sc swf_options_SetPreloaderType (FLIX2HANDLE flix, const FE2_SwfPreloaderOptions preloaderType)

  Sets the type of preloader.

- on2sc swf_options_GetAdaptivePreloaderBufferFactor (FLIX2HANDLE flix, double *pPreloaderBufferFactor)

  Gets the adaptive preload buffer factor.

- on2sc swf_options_SetAdaptivePreloaderBufferFactor (FLIX2HANDLE flix, const double preloaderBufferFactor)

  Sets the adaptive preload buffer factor.

- on2sc swf_options_GetMovieOnEndOptions (FLIX2HANDLE flix, FE2_SwfOnEndOptions *pOnEndOptions)

  Gets the options for the end of the SWF.

- on2sc swf_options_SetMovieOnEndOptions (FLIX2HANDLE flix, const FE2_SwfOnEndOptions onEndOptions)

  Sets the options for the end of the SWF.

- on2sc swf_options_GetLoopCount (FLIX2HANDLE flix, int32_t *pLoopCount)

  Gets the number of times the SWF should loop.

- on2sc swf_options_SetLoopCount (FLIX2HANDLE flix, const int32_t loopCount)

  Sets the number of times the SWF should loop.

- on2sc swf_options_GetLoadMovieOnEndUrl (FLIX2HANDLE flix, char *pLoadMovieOnEndUrl, int32_t *len)

  Gets the URL of a SWF movie to load after the current movie ends.

- on2sc swf_options_SetLoadMovieOnEndUrl (FLIX2HANDLE flix, const char *loadMovieOnEndUrl)

  Sets the URL of a SWF movie to load after the current movie ends.

- on2sc swf_options_GetMovieOnStartOptions (FLIX2HANDLE flix, FE2_SwfOnStartOptions *pOnStartOptions)

  Gets the options for the start of the SWF.

- on2sc swf_options_SetMovieOnStartOptions (FLIX2HANDLE flix, const FE2_SwfOnStartOptions onStartOptions)

  Sets the options for the start of the SWF.

- on2sc swf_options_GetWaitTimeToStart (FLIX2HANDLE flix, int32_t *pWaitTimeToStart)

  Gets the number of seconds to wait before playback begins.

- on2sc swf_options_SetWaitTimeToStart (FLIX2HANDLE flix, const int32_t waitTimeToStart)

  Sets the number of seconds to wait before playback begins.

- on2sc swf_options_AddVariable (FLIX2HANDLE flix, const char *name, const char *value)

  Adds a custom SWF variable as a name/value pair.

- on2sc swf_options_DeleteVariable (FLIX2HANDLE flix, const int32_t index)

  Deletes a custom SWF variable.

- on2sc swf_options_GetVariableCount (FLIX2HANDLE flix, int32_t *pVariableCount)

  Gets the number of custom SWF variables already added.

- on2sc swf_options_UpdateVariable (FLIX2HANDLE flix, const int32_t index, const char *name, const char *value)

  Updates an already existing SWF variable.

Defines

- #define FE2_MUXER_SWF

  SWF muxer. For use with Flix2_AddMuxer().

- #define FE2_SWF_HEIGHT

  Set the SWF height.

- #define FE2_SWF_WIDTH

  Set the SWF width.

- #define FE2_SWF_FRAMERATE

  Set the SWF framerate.

- #define FE2_SWF_EMBEDDED_URL

  Set the SWF's target URL.

- #define FE2_SWF_EMBEDDED_URL_TARGET

  Set the target of FE2_SWF_EMBEDDED_URL.

- #define FE2_SWF_EMBEDDED_URL_TYPE

  Set how FE2_SWF_EMBEDDED_URL is interpreted.

- #define FE2_SWF_LOOP_COUNT

  Sets the number of times the SWF should loop.

- #define FE2_SWF_PRELOAD_TYPE

  Sets the type of preloader.

- #define FE2_SWF_FIXED_PRELOAD_PCT

  Sets the percent of the SWF movie to preload before playback begins.

- #define FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR

Sets the adaptive preload buffer factor.

- #define FE2_SWF_ON_END_OPTION

  Sets the options for the end of the SWF.

- #define FE2_SWF_ON_END_URL

  Sets the URL a SWF movie will load after the current movie ends.

- #define FE2_SWF_ON_START_OPTION

  Sets the options for the start of the SWF.

- #define FE2_SWF_START_BLANK_FRAME

  Control the insertion of a blank first frame in the SWF.

- #define FE2_SWF_START_WAIT_SEC

  Sets the number of seconds to wait before playback begins.

- #define FE2_SWF_ADD_VARIABLE

  Add or update a variable in the SWF.

- #define FE2_SWF_DELETE_VARIABLE

  Delete an existing variable in the SWF.

Enumerations

- enum FE2_EmbeddedUrlType {
  EmbeddedUrlIsGetUrl,
  EmbeddedUrlIsLoadMovie }

  Differentiates between the type of file (HTML or SWF) set through FE2_SWF_EMBEDDED_URL.

- enum FE2_SwfOnEndOptions {
  SwfOnMovieEndNothing,
  SwfOnMovieEndSTOP,
  SwfOnMovieEndLoop,
  SwfOnMovieEndUnload,
  SwfOnMovieEndLoadMovie }

  Actions that can be added to the last frame of a SWF file.

- enum FE2_SwfOnStartOptions {
  SwfOnMovieStartAutomatically,
  SwfOnMovieStartOnClick,
  SwfOnMovieStartWait,
  SwfOnMovieStartEmbedSTOP }

  Actions that can be added to the start frame of the SWF file.

- enum FE2_SwfPreloaderOptions {
  SwfPreloaderNone,
  SwfFixedPreloader,
  SwfAdaptivePreloader }
  Determines the type of preloader added to the SWF file.

20.52.3 Define Documentation

20.52.3.1 #define FE2_MUXER_SWF

SWF muxer. For use with Flix2_AddMuxer().

Definition at line 105 of file swf.h.

20.52.3.2 #define FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR

Sets the adaptive preload buffer factor.

An adaptive preloader is an "intelligent" preloader that will preload a variable amount of video based on the size of the video and the viewer's internet connection speed so that the viewer can watch the video without stops and starts - no matter what speed they are connected to the internet at. The adaptive preloader determines the user's connection speed and based on that figure and the size of the video file preloading, it preloads precisely the amount of the video required to enable the video to begin playing as soon as it possibly can and still not stop at any point until it is done. In this way, the preload process is optimized so the amount of time taken for a video to preload is minimized as much as possible. If the adaptive preloader is selected, a buffer factor must also be selected. The buffer factor will cause the preloader to preload more (or less) of the video than the initial calculations determine is the minimum amount of time required. This is useful because a viewer's connection speed often varies, even during the course of watching a single video. The time required to load the file up to the end is multiplied by this factor. If a buffer is not needed, set the factor to 1.0 (a factor of 1.0 means no buffer - loadingTime $*$ 1.0 = loadingTime). To be on the safe side, the recommendation is to use a factor larger than 1.0. If the viewer's connection should get faster during the visit (e.g., perhaps a download ends) then the buffer factor can be changed to a number smaller than 1.0. If the viewer's connection will likely get slower during the visit (e.g., because the viewer starts a new download) the buffer factor should be changed to a number greater than 1.0. As a general rule, it is a good idea to make the buffer factor larger than 1.0 to be on the safe side.

Note:

Default: 1.1
Valid range: [0.0,2.0] step: 0.1
This implicitly sets FE2_SWF_PRELOAD_TYPE to SwfAdaptivePreloader
Works with any Flash player version 5 and higher.

Definition at line 222 of file swf.h.

20.52.3.3 #define FE2_SWF_ADD_VARIABLE

Add or update a variable in the SWF.

Multiple variables may be specified.

Parameter format:

```
Format = Variable *[ '&' Variable ]
Variable = varName '=' varValue
```
e.g. "n0=v0&n1=v1"

Definition at line 262 of file swf.h.

20.52.3.4 #define FE2_SWF_DELETE_VARIABLE

Delete an existing variable in the SWF.

Multiple variables may be specified.

Parameter format:

```
Format = varName *[ '&' varName ]
```
e.g. "var0&var1"

Definition at line 270 of file swf.h.

20.52.3.5 #define FE2_SWF_EMBEDDED_URL

Set the SWF's target URL.

When a user clicks on the video of the encoded SWF file, the SWF file will immediately try and load the embedded URL.

See also:

FE2_SWF_EMBEDDED_URL_TARGET, FE2_SWF_EMBEDDED_URL_TYPE

Definition at line 138 of file swf.h.

20.52.3.6 #define FE2_SWF_EMBEDDED_URL_TARGET

Set the target of FE2_SWF_EMBEDDED_URL.

Valid values:

- `"_self"`
- `"_blank"`
- `"_parent"`
- `"_top"`

Note:

Default: "_self"

See also:

http://www.w3.org/TR/REC-WebCGM/REC-03-CGM-IC.html#webcgm_3_1_2_2

Definition at line 151 of file swf.h.

20.52.3.7 #define FE2_SWF_EMBEDDED_URL_TYPE

Set how FE2_SWF_EMBEDDED_URL is interpreted.

Valid values are defined by FE2_EmbeddedUrlType

Note:

Default: EmbeddedUrlIsLoadMovie

Definition at line 158 of file swf.h.

20.52.3.8 #define FE2_SWF_FIXED_PRELOAD_PCT

Sets the percent of the SWF movie to preload before playback begins.

Note:

Default: 20%
Valid range: [0,100]
This implicitly sets FE2_SWF_PRELOAD_TYPE to SwfFixedPreloader
Works with any Flash player version 4 and higher.

Definition at line 184 of file swf.h.

20.52.3.9 #define FE2_SWF_FRAMERATE

Set the SWF framerate.

A few things should be considered before setting the SWF framerate. First, SWF framerate should always be equal to or a multiple of the video framerate to prevent problems with encoding such as loss of audio sync and audio distortion. Second, the fractional part of the output SWF framerate has to be rounded to the nearest 1/256th, i.e. 29.97 becomes 29.96875.

Note:

Changing the SWF framerate will NOT change the duration or the audio/video synchronization except as mentioned above.
Default: video framerate

See also:

Frame Rate

Definition at line 130 of file swf.h.

20.52.3.10 #define FE2_SWF_HEIGHT

Set the SWF height.

Note:

Default: video height

Definition at line 110 of file swf.h.

20.52.3.11 #define FE2_SWF_LOOP_COUNT

Sets the number of times the SWF should loop.

Note:

Default: 0
Setting this to a value >0 will see the SWF ignore the loop command of the Flash player.
This will implicitly set FE2_SWF_ON_END_OPTION to SwfOnMovieEndLoop

Definition at line 168 of file swf.h.

20.52.3.12 #define FE2_SWF_ON_END_OPTION

Sets the options for the end of the SWF.

Valid values are defined by FE2_SwfOnEndOptions

Note:

Default: SwfOnMovieEndNothing

Definition at line 228 of file swf.h.

20.52.3.13 #define FE2_SWF_ON_END_URL

Sets the URL a SWF movie will load after the current movie ends.

Note:

This will implicitly set FE2_SWF_ON_END_OPTION to SwfOnMovieEndLoadMovie

Definition at line 234 of file swf.h.

20.52.3.14 #define FE2_SWF_ON_START_OPTION

Sets the options for the start of the SWF.

Valid values are defined by FE2_SwfOnStartOptions.

Note:

Default: SwfOnMovieStartAutomatically

Definition at line 241 of file swf.h.

20.52.3.15 #define FE2_SWF_PRELOAD_TYPE

Sets the type of preloader.

Valid types are defined by FE2_SwfPreloaderOptions.

Note:

Default: SwfPreloaderNone

Definition at line 175 of file swf.h.

20.52.3.16 #define FE2_SWF_START_BLANK_FRAME

Control the insertion of a blank first frame in the SWF.

Note:

Default: 0

Definition at line 247 of file swf.h.

20.52.3.17 #define FE2_SWF_START_WAIT_SEC

Sets the number of seconds to wait before playback begins.

Note:

Default: 0

Definition at line 252 of file swf.h.

20.52.3.18 #define FE2_SWF_WIDTH

Set the SWF width.

Note:

Default: video width

Definition at line 115 of file swf.h.

20.52.4 Enumeration Type Documentation

20.52.4.1 enum FE2_EmbeddedUrlType

Differentiates between the type of file (HTML or SWF) set through FE2_SWF_EMBEDDED_URL.

Enumerator:

EmbeddedUrlIsGetUrl

EmbeddedUrlIsLoadMovie

Definition at line 68 of file swf.h.

20.52.4.2 enum FE2_SwfOnEndOptions

Actions that can be added to the last frame of a SWF file.

For use with FE2_SWF_ON_END_OPTION

Enumerator:

SwfOnMovieEndNothing

SwfOnMovieEndSTOP
SwfOnMovieEndLoop
SwfOnMovieEndUnload
SwfOnMovieEndLoadMovie

Definition at line 76 of file swf.h.

20.52.4.3 enum FE2_SwfOnStartOptions

Actions that can be added to the start frame of the SWF file.

For use with FE2_SWF_ON_START_OPTION

Enumerator:

SwfOnMovieStartAutomatically
SwfOnMovieStartOnClick
SwfOnMovieStartWait
SwfOnMovieStartEmbedSTOP

Definition at line 87 of file swf.h.

20.52.4.4 enum FE2_SwfPreloaderOptions

Determines the type of preloader added to the SWF file.

For use with FE2_SWF_PRELOAD_TYPE

Enumerator:

SwfPreloaderNone
SwfFixedPreloader
SwfAdaptivePreloader

Definition at line 97 of file swf.h.

20.52.5 Function Documentation

20.52.5.1 on2sc swf_options_AddVariable (FLIX2HANDLE *flix*, const char * *name*, const char * *value*)

Adds a custom SWF variable as a name/value pair.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()
← ***name*** The name of the variable.
← ***value*** The value of the variable.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for *name* or *value*

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
name is not `NULL`
value is not `NULL`

See also:

swf_options_DeleteVariable(), swf_options_GetVariableCount(), swf_options_UpdateVariable()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ADD_VARIABLE parameter. This function will be removed in a future release.

20.52.5.2 on2sc swf_options_DeleteVariable (FLIX2HANDLE *flix*, const int32_t *index*)

Deletes a custom SWF variable.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***index*** The zero based index of the variable to delete.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred reallocating memory for the variable list.

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
index is inside the range of current variables.

See also:

swf_options_AddVariable(), swf_options_GetVariableCount(), swf_options_UpdateVariable()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_DELETE_VARIABLE parameter. This function will be removed in a future release.

20.52.5.3 on2sc swf_options_GetAdaptivePreloaderBufferFactor (FLIX2HANDLE *flix*, double * *pPreloaderBufferFactor*)

Gets the adaptive preload buffer factor.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pPreloaderBufferFactor*** The adaptive preloader buffer factor.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pPreloaderBufferFactor is not `NULL`

See also:

swf_options_SetAdaptivePreloaderBufferFactor()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ADAPTIVE_-PRELOAD_BUFFER_FACTOR parameter. This function will be removed in a future release.

20.52.5.4 on2sc swf_options_GetEmbeddedUrl (FLIX2HANDLE *flix*, char * *embeddedUrl*, int32_t * *len*)

Gets the embedded URL.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***embeddedUrl*** The embedded URL.

↔ ***len***
- IN: The length of *embeddedUrl*.
- OUT: The length of the URL (minus the null terminator).

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM *embeddedUrl* does not have enough allocated space to return the URL. The size in bytes needed (minus null terminator) will be returned in *len*.

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
len is not `NULL`

Note:

If *embeddedUrl* is `NULL` this function will return the size in bytes required to store the current embedded URL in *len*, not including the null terminator.

See also:

swf_options_SetEmbeddedUrl(), swf_options_GetEmbeddedUrlType(), swf_options_SetEmbeddedUrlType()

Attention:

Currently only supported in C/C++

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL parameter. This function will be removed in a future release.

20.52.5.5 on2sc swf_options_GetEmbeddedUrlTarget (FLIX2HANDLE *flix*, char * *embeddedUrlTarget*, int32_t * *len*)

Gets the target of embedded URL.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***embeddedUrlTarget*** The target of embedded URL.

↔ ***len***
- IN: The length of *embeddedUrlTarget*.
- OUT: The length of the URL (minus the null terminator).

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM *embeddedUrlTarget* does not have enough allocated space to return the target. The size in bytes needed (minus null terminator) will be returned in *len*.

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
len is not `NULL`

Note:

If *embeddedUrlTarget* is `NULL` this function will return the size in bytes required to store the current target in *len*, not including the null terminator.

See also:

swf_options_GetEmbeddedUrl(), swf_options_SetEmbeddedUrl(), swf_options_SetEmbeddedUrlType()

Attention:

Currently only supported in C/C++

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_-TARGET parameter. This function will be removed in a future release.

20.52.5.6 on2sc swf_options_GetEmbeddedUrlType (FLIX2HANDLE *flix*, FE2_EmbeddedUrlType * *embeddedUrlType*)

Gets the type of embedded URL.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***embeddedUrlType*** The type of embedded URL.

Return values:

ON2_OK success.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
embeddedUrlType is not `NULL`

See also:

swf_options_SetEmbeddedUrl(), swf_options_SetEmbeddedUrlType()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_-TYPE parameter. This function will be removed in a future release.

20.52.5.7 on2sc swf_options_GetEnablePreloader (FLIX2HANDLE *flix*, on2bool * *pEnablePreloader*)

Gets if a video preloader is enabled or disabled.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pEnablePreloader***

Return values:

ON2_OK Success.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pEnablePreloader is not `NULL`

See also:

swf_options_SetEnablePreloader()

20.52.5.8 on2sc swf_options_GetInsertBlankFrameOnStart (FLIX2HANDLE *flix*, on2bool * *pInsertBlankFrameOnStart*)

Gets if the engine is to insert a blank frame as the first frame of the SWF or not.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pInsertBlankFrameOnStart***

Return values:

ON2_OK Successfully returned if the engine will insert a blank frame or not.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pInsertBlankFrameOnStart is not `NULL`

See also:

swf_options_SetInsertBlankFrameOnStart()

20.52.5.9 on2sc swf_options_GetLoadMovieOnEndUrl (FLIX2HANDLE *flix*, char * *pLoadMovieOnEndUrl*, int32_t * *len*)

Gets the URL of a SWF movie to load after the current movie ends.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pLoadMovieOnEndUrl*** The URL of a SWF movie.

↔ ***len***
- IN: The length of *pLoadMovieOnEndUrl*.
- OUT: The length of the URL (minus the null terminator).

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM *pLoadMovieOnEndUrl* does not have enough allocated space to return the URL. The size in bytes needed (minus null terminator) will be returned in *len*.

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
len is not `NULL`

Note:

If *pLoadMovieOnEndUrl* is `NULL` this function will return the size in bytes required to store the current URL in *len*, not including the null terminator.

See also:

swf_options_SetLoadMovieOnEndUrl()

Attention:

Currently only supported in C/C++

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ON_END_URL parameter. This function will be removed in a future release.

20.52.5.10 on2sc swf_options_GetLoopCount (FLIX2HANDLE *flix*, int32_t * *pLoopCount*)

Gets the number of times the SWF should loop.

Parameters:

←***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→***pLoopCount*** The number of times the SWF should loop.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pLoopCount is not `NULL`

See also:

swf_options_SetLoopCount()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_LOOP_COUNT parameter. This function will be removed in a future release.

20.52.5.11 on2sc swf_options_GetMovieOnEndOptions (FLIX2HANDLE *flix*, FE2_SwfOnEndOptions * *pOnEndOptions*)

Gets the options for the end of the SWF.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pOnEndOptions*** The options for the end of the SWF.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pOnEndOptions is not `NULL`

See also:

swf_options_SetMovieOnEndOptions()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ON_END_OPTION parameter. This function will be removed in a future release.

20.52.5.12 on2sc swf_options_GetMovieOnStartOptions (FLIX2HANDLE *flix*, FE2_SwfOnStartOptions * *pOnStartOptions*)

Gets the options for the start of the SWF.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pOnStartOptions*** The options for the start of the SWF.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pOnStartOptions is not `NULL`

See also:

swf_options_SetMovieOnStartOptions()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ON_START_OPTION parameter. This function will be removed in a future release.

20.52.5.13 on2sc swf_options_GetPercentToPreload (FLIX2HANDLE *flix*, int32_t * *pPercentToPreload*)

Gets the percent of the SWF movie to preload before playback begins.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pPercentToPreload*** Percent of the SWF movie to preload before playback begins.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pPercentToPreload is not `NULL`

See also:

swf_options_SetPercentToPreload()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_FIXED_PRELOAD_-PCT parameter. This function will be removed in a future release.

20.52.5.14 on2sc swf_options_GetPreloaderType (FLIX2HANDLE *flix*, FE2_SwfPreloaderOptions * *pPreloaderType*)

Gets the type of preloader.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pPreloaderType*** The preloader type.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pPreloaderType is not `NULL`

See also:

swf_options_SetPreloaderType()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_PRELOAD_TYPE parameter. This function will be removed in a future release.

20.52.5.15 on2sc swf_options_GetSwfFramerate (FLIX2HANDLE *flix*, int32_t * *pSwfFramerate*)

Gets the SWF framerate.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***pSwfFramerate*** The SWF framerate.

Return values:

ON2_OK The engine successfully returned SWF framerate.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pSwfFramerate is not `NULL`

Deprecated

This function has been deprecated in favor of swf_options_GetSwfFramerateAsDouble() because this function can only handle integer framerates.

See also:

swf_options_GetSwfFramerateAsDouble()

20.52.5.16 on2sc swf_options_GetSwfFramerateAsDouble (FLIX2HANDLE *flix*, double * *pSwfFramerate*)

Gets the SWF framerate as a double.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pSwfFramerate*** The SWF framerate.

Return values:

ON2_OK The engine successfully returned SWF framerate.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pSwfFramerate is not `NULL`

See also:

swf_options_SetSwfFramerateAsDouble()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_FRAMERATE parameter. This function will be removed in a future release.

20.52.5.17 on2sc swf_options_GetVariableCount (FLIX2HANDLE *flix*, int32_t * *pVariableCount*)

Gets the number of custom SWF variables already added.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pVariableCount*** The number of custom SWF variables already added

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pVariableCount is not `NULL`

See also:

swf_options_AddVariable(), swf_options_DeleteVariable(), swf_options_UpdateVariable()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF. This function will be removed in a future release.

20.52.5.18 on2sc swf_options_GetWaitTimeToStart (FLIX2HANDLE *flix*, int32_t * *pWaitTimeToStart*)

Gets the number of seconds to wait before playback begins.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pWaitTimeToStart*** Number of seconds to wait.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pWaitTimeToStart is not `NULL`

See also:

swf_options_SetWaitTimeToStart()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_START_WAIT_SEC parameter. This function will be removed in a future release.

20.52.5.19 on2sc swf_options_Reset (FLIX2HANDLE *flix*)

Resets the swf options.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK reset of swf options was successful

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

Call this function if you wish to reset all swf options to their default values. This will free any memory that was allocated to the swf object.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF. This function will be removed in a future release.

20.52.5.20 on2sc swf_options_SetAdaptivePreloaderBufferFactor (FLIX2HANDLE *flix*, const double *preloaderBufferFactor*)

Sets the adaptive preload buffer factor.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***preloaderBufferFactor*** The adaptive preloader buffer factor.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

The preloader has to be enabled.
The default value is 1.1.
The range is 0-2 in increments of 0.1.
The preloader type must be SwfAdaptivePreloader.
Works with any Flash player version 5 and higher.

See also:

swf_options_GetAdaptivePreloaderBufferFactor()

Remarks:

An adaptive preloader is an "intelligent" preloader that will preload a variable amount of video based on the size of the video and the viewer's internet connection speed so that the viewer can watch the video without stops and starts - no matter what speed they are connected to the internet at. The adaptive preloader determines the user's connection speed and based on that figure and the size of the video file preloading, it preloads precisely the amount of the video required to enable the video to begin playing as soon as it possibly can and still not stop at any point until it is done. In this way, the preload process is optimized so the amount of time taken for a video to preload is minimized as much as possible. If the adaptive preloader is selected, a buffer factor must also be selected. The buffer factor will cause the preloader to preload more (or less) of the video than the initial calculations determine is the minimum amount of time required. This is useful because a viewer's connection speed often varies, even during the course of watching a single video. The time required to load the file up to the end is multiplied by this factor. If a buffer is not needed, set the factor to 1.0 (a factor of 1.0 means no buffer - loadingTime $*$ 1.0 = loadingTime). To be on the safe side, the recommendation is to use a factor larger than 1.0. If the viewer's connection should get faster during the visit (e.g., perhaps a download ends) then the buffer factor can be changed to a number smaller than 1.0. If the viewer's connection will likely get slower during the visit (e.g., because the viewer starts a new download) the buffer factor should be changed to a number greater than 1.0. As a general rule, it is a good idea to make the buffer factor larger than 1.0 to be on the safe side.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ADAPTIVE_-PRELOAD_BUFFER_FACTOR parameter. This function will be removed in a future release.

20.52.5.21 on2sc swf_options_SetEmbeddedUrl (FLIX2HANDLE *flix*, const char * *embeddedUrl*)

Sets the embedded URL of the SWF or HTML that will be loaded when the user clicks on the video.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***embeddedUrl*** The embedded URL.

Return values:

ON2_OK success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for *embeddedUrl*

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

When a user clicks on the video of the encoded SWF file, the SWF file will imediately try and load the embedded URL.
If *embeddedUrl* is `NULL` then the engine will delete the embedded URL.

See also:

swf_options_GetEmbeddedUrl(), swf_options_GetEmbeddedUrlType(), swf_options_-SetEmbeddedUrlType()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL parameter. This function will be removed in a future release.

20.52.5.22 on2sc swf_options_SetEmbeddedUrlTarget (FLIX2HANDLE *flix*, const char * *embeddedUrlTarget*)

Sets the target of the embedded URL.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***embeddedUrlTarget*** The target of the embedded URL.

Return values:

ON2_OK success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for *embeddedUrlTarget*

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
If *embeddedUrlTarget* is not `NULL` then it must be one of the valid values.

Note:

The *embeddedUrlTarget* will be ignored if the embedded URL is not set.
Valid values for *embeddedUrlTarget* are:

- `"_self"`
- `"_blank"`
- `"_parent"`
- `"_top"`
- see http://www.w3.org/TR/REC-WebCGM/REC-03-CGM-IC.html#webcgm_3_-1_2_2 for more information.

The default value is `"_self"`.
If *embeddedUrlTarget* is `NULL` then the engine will delete the target of the embedded URL.

See also:

swf_options_GetEmbeddedUrl(), swf_options_SetEmbeddedUrl(), swf_options_-GetEmbeddedUrlType()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_-TARGET parameter. This function will be removed in a future release.

20.52.5.23 on2sc swf_options_SetEmbeddedUrlType (FLIX2HANDLE *flix*, const FE2_EmbeddedUrlType *embeddedUrlType*)

Sets the type of embedded URL.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***embeddedUrlType*** The type of embedded URL.

Return values:

ON2_OK success.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
embeddedUrlType is a valid member of FE2_EmbeddedUrlType

Note:

The *embeddedUrlType* will be ignored if the embedded URL is not set.

Remarks:

The default value is EmbeddedUrlIsLoadMovie.

See also:

swf_options_SetEmbeddedUrl(), swf_options_GetEmbeddedUrlType()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_EMBEDDED_URL_-TYPE parameter. This function will be removed in a future release.

20.52.5.24 on2sc swf_options_SetEnablePreloader (FLIX2HANDLE *flix*, const on2bool *enablePreloader*)

Enables or disables the video preloader.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***enablePreloader***

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

on2true will enable a preloader.
on2false will disable a preloader.

See also:

swf_options_GetEnablePreloader()

20.52.5.25 on2sc swf_options_SetInsertBlankFrameOnStart (FLIX2HANDLE *flix*, const on2bool *insertBlankFrameOnStart*)

Tells the engine to insert a blank frame as the first frame of the SWF or not.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***insertBlankFrameOnStart*** on2true will insert a blank frame. on2false will not.

Return values:

ON2_OK The engine set the variable of whether or not to insert a blank frame.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

on2true will insert a blank frame.
on2false will not.

See also:

swf_options_GetInsertBlankFrameOnStart()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_START_BLANK_-FRAME parameter. This function will be removed in a future release.

20.52.5.26 on2sc swf_options_SetLoadMovieOnEndUrl (FLIX2HANDLE *flix*, const char * *loadMovieOnEndUrl*)

Sets the URL of a SWF movie to load after the current movie ends.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***loadMovieOnEndUrl*** The URL of a SWF movie.

Return values:

ON2_OK success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for *loadMovieOnEndUrl*

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

If *loadMovieOnEndUrl* is `NULL` then the engine will delete the URL.

See also:

swf_options_GetLoadMovieOnEndUrl()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ON_END_URL parameter. This function will be removed in a future release.

20.52.5.27 on2sc swf_options_SetLoopCount (FLIX2HANDLE *flix*, const int32_t *loopCount*)

Sets the number of times the SWF should loop.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***loopCount*** The number of times the SWF should loop.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

The default value is 0.
Setting this to a value >0 will have the SWF ignore the loop command of the Flash player.

See also:

swf_options_GetLoopCount()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_LOOP_COUNT parameter. This function will be removed in a future release.

20.52.5.28 on2sc swf_options_SetMovieOnEndOptions (FLIX2HANDLE *flix*, const FE2_SwfOnEndOptions *onEndOptions*)

Sets the options for the end of the SWF.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***onEndOptions*** The options for the end of the SWF.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
onEndOptions is a valid member of FE2_SwfOnEndOptions

Note:

The default value is SwfOnMovieEndNothing.

See also:

swf_options_GetMovieOnEndOptions()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ON_END_OPTION parameter. This function will be removed in a future release.

20.52.5.29 on2sc swf_options_SetMovieOnStartOptions (FLIX2HANDLE *flix*, const FE2_SwfOnStartOptions *onStartOptions*)

Sets the options for the start of the SWF.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***onStartOptions*** The options for the start of the SWF.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
onStartOptions is a valid member of FE2_SwfOnStartOptions

Note:

The default value is SwfOnMovieStartAutomatically.

See also:

swf_options_GetMovieOnStartOptions()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ON_START_OPTION parameter. This function will be removed in a future release.

20.52.5.30 on2sc swf_options_SetPercentToPreload (FLIX2HANDLE *flix*, const int32_t *percentToPreload*)

Sets the percent of the SWF movie to preload before playback begins.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***percentToPreload*** Percent of the SWF movie to preload before playback begins.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Note:

The preloader has to be enabled.
The default value is 20%.
The range is 1-100.
The preloader type must be SwfFixedPreloader.
Works with any Flash player version 4 and higher.

See also:

swf_options_GetPercentToPreload()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_FIXED_PRELOAD_-PCT parameter. This function will be removed in a future release.

20.52.5.31 on2sc swf_options_SetPreloaderType (FLIX2HANDLE *flix*, const FE2_SwfPreloaderOptions *preloaderType*)

Sets the type of preloader.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***preloaderType*** The preloader type.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
preloaderType is a valid member of FE2_SwfPreloaderOptions

Note:

The preloader has to be enabled.
The default value is SwfFixedPreloader.

See also:

swf_options_GetPreloaderType()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_PRELOAD_TYPE parameter. This function will be removed in a future release.

20.52.5.32 on2sc swf_options_SetSwfFramerate (FLIX2HANDLE *flix*, const int32_t *swfFramerate*)

Sets the SWF framerate.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***swfFramerate*** The SWF framerate.

Return values:

ON2_OK The engine successfully set the SWF framerate.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Deprecated

This function has been deprecated in favor of swf_options_SetSwfFramerateAsDouble() because this function can only handle integer framerates.

See also:

swf_options_SetSwfFramerateAsDouble()

20.52.5.33 on2sc swf_options_SetSwfFramerateAsDouble (FLIX2HANDLE *flix*, const double *swfFramerate*)

Sets the SWF framerate as a double.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***swfFramerate*** The SWF framerate.

Return values:

ON2_OK The engine successfully set SWF framerate.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not NULL

Note:

The default value will be the video framerate.
The SWF framerate may be changed when the encoding process starts because of two conditions. Changing the SWF framerate will NOT change the duration or the audio/video synchronization. The first condition is the SWF framerate should always be equal to or a multiple of the video framerate to prevent problems with encoding such as loss of audio sync and audio distortion. The second condition is the fractional part of the output SWF framerate has to be rounded to the nearest 1/256th. I.E. If the SWF framerate is set to 29.97 then the output SWF framerate will be converted to 29.96875.

See also:

swf_options_GetSwfFramerateAsDouble(), video_options_SetVideoFramerateAsDouble()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_FRAMERATE parameter. This function will be removed in a future release.

20.52.5.34 on2sc swf_options_SetWaitTimeToStart (FLIX2HANDLE *flix*, const int32_t *waitTimeToStart*)

Sets the number of seconds to wait before playback begins.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***waitTimeToStart*** Number of seconds to wait.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not NULL

Note:

The default value is 0.

See also:

swf_options_GetWaitTimeToStart()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_START_WAIT_SEC parameter. This function will be removed in a future release.

20.52.5.35 on2sc swf_options_UpdateVariable (FLIX2HANDLE *flix*, const int32_t *index*, const char * *name*, const char * *value*)

Updates an already existing SWF variable.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***index*** The zero based index of the variable to update.

← ***name*** The name of the variable.

← ***value*** The value of the variable.

Return values:

ON2_OK Success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred reallocating memory for the variable list.

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
name is not `NULL`
value is not `NULL`
index is inside the range of current variables

See also:

swf_options_AddVariable(), swf_options_DeleteVariable(), swf_options_GetVariableCount()

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_ADD_VARIABLE parameter. This function will be removed in a future release.

20.53 WebM - FFmpeg

20.53.1 Detailed Description

Example Usage:

```
sc = Flix2_AddMuxer(&muxer, flix, FE2_MUXER_WEBM);
if(sc == ON2_OK)
    ; //Now using the WebM muxer for output
```

20.53.2 Supported Codecs

- Video:
  - FE2_CODEC_VP8
- Audio:
  - FE2_CODEC_VORBIS

Additional References:

- WebM Project

Defines

- #define FE2_MUXER_WEBM

  WebM muxer. For use with Flix2_AddMuxer().

20.53.3 Define Documentation

20.53.3.1 #define FE2_MUXER_WEBM

WebM muxer. For use with Flix2_AddMuxer().

Definition at line 45 of file webm.h.

20.54 Base Types

Defines

- #define OTC(str)

  a macro suitable for declaring a constant on2tc

- #define ON2TC

  printf format string suitable for printing an on2tc

- #define ON264

  printf format string suitable for printing an on2s64

- #define PRId64

Typedefs

- typedef char int8_t
- typedef short int16_t
- typedef int int32_t
- typedef unsigned char uint8_t
- typedef unsigned short uint16_t
- typedef unsigned int uint32_t
- typedef int8_t on2s8
- typedef uint8_t on2u8
- typedef int16_t on2s16
- typedef uint16_t on2u16
- typedef int32_t on2s32
- typedef uint32_t on2u32
- typedef int32_t on2bool
- typedef char on2tc
- typedef long long on2s64
- typedef unsigned long long on2u64
- typedef on2s64 int64_t
- typedef on2u64 uint64_t

Enumerations

- enum _on2bool {
  on2false,
  on2true }
- enum on2sc {
  ON2_NOT_FOUND,
  ON2_BUFFER_EMPTY,
  ON2_BUFFER_FULL,
  ON2_CONNREFUSED,
  ON2_TIMEDOUT,

ON2_WOULDBLOCK,
ON2_NET_ERROR,
ON2_INVALID_VERSION,
ON2_INPROGRESS,
ON2_NOT_SUPP,
ON2_NO_MEM,
ON2_INVALID_PARAMS,
ON2_ERROR,
ON2_OK,
ON2_DONE }

Common return type.

20.54.1 Define Documentation

20.54.1.1 #define ON264

printf format string suitable for printing an on2s64

Definition at line 96 of file on2types.h.

20.54.1.2 #define ON2TC

printf format string suitable for printing an on2tc

Definition at line 81 of file on2types.h.

20.54.1.3 #define OTC(str)

a macro suitable for declaring a constant on2tc

Definition at line 80 of file on2types.h.

20.54.1.4 #define PRId64

Definition at line 95 of file on2types.h.

20.54.2 Typedef Documentation

20.54.2.1 typedef short int16_t

Definition at line 43 of file on2types.h.

20.54.2.2 typedef int int32_t

Definition at line 44 of file on2types.h.

20.54.2.3 typedef on2s64 int64_t

Definition at line 130 of file on2types.h.

20.54.2.4 typedef char int8_t

Definition at line 42 of file on2types.h.

20.54.2.5 typedef int32_t on2bool

Definition at line 57 of file on2types.h.

20.54.2.6 typedef int16_t on2s16

Definition at line 53 of file on2types.h.

20.54.2.7 typedef int32_t on2s32

Definition at line 55 of file on2types.h.

20.54.2.8 typedef long long on2s64

Definition at line 97 of file on2types.h.

20.54.2.9 typedef int8_t on2s8

Definition at line 51 of file on2types.h.

20.54.2.10 typedef char on2tc

Definition at line 79 of file on2types.h.

20.54.2.11 typedef uint16_t on2u16

Definition at line 54 of file on2types.h.

20.54.2.12 typedef uint32_t on2u32

Definition at line 56 of file on2types.h.

20.54.2.13 typedef unsigned long long on2u64

Definition at line 98 of file on2types.h.

20.54.2.14 typedef uint8_t on2u8

Definition at line 52 of file on2types.h.

20.54.2.15 typedef unsigned short uint16_t

Definition at line 47 of file on2types.h.

20.54.2.16 typedef unsigned int uint32_t

Definition at line 48 of file on2types.h.

20.54.2.17 typedef on2u64 uint64_t

Definition at line 131 of file on2types.h.

20.54.2.18 typedef unsigned char uint8_t

Definition at line 46 of file on2types.h.

20.54.3 Enumeration Type Documentation

20.54.3.1 enum _on2bool

Enumerator:

on2false
on2true

Definition at line 59 of file on2types.h.

20.54.3.2 enum on2sc

Common return type.

Enumerator:

ON2_NOT_FOUND
ON2_BUFFER_EMPTY
ON2_BUFFER_FULL
ON2_CONNREFUSED
ON2_TIMEDOUT
ON2_WOULDBLOCK
ON2_NET_ERROR
ON2_INVALID_VERSION
ON2_INPROGRESS
ON2_NOT_SUPP

ON2_NO_MEM
ON2_INVALID_PARAMS
ON2_ERROR
ON2_OK
ON2_DONE

Definition at line 137 of file on2types.h.

20.55 Video Encoding Options

Modules

- Deprecated

Functions

- on2sc video_options_Reset (FLIX2HANDLE flix)

  Reset the video options to their defaults.

- on2sc video_options_Validate (const FLIX2HANDLE flix)

  Ensure the current video settings are valid.

- on2sc video_options_GetSwfHeight (const FLIX2HANDLE flix, int32_t *lpSwfHeight)

  Gets the SWF height.

- on2sc video_options_SetSwfHeight (FLIX2HANDLE flix, const int32_t lSwfHeight)

  Sets the SWF height.

- on2sc video_options_GetSwfWidth (const FLIX2HANDLE flix, int32_t *lpSwfWidth)

  Gets the SWF width.

- on2sc video_options_SetSwfWidth (FLIX2HANDLE flix, const int32_t lSwfWidth)

  Sets the SWF width.

- on2sc video_options_GetUseCustomSwfDimensions (const FLIX2HANDLE flix, on2bool *bpUseCustomSwfDimensions)

  Determine if the engine is using the SWF width and height for the SWF.

- on2sc video_options_SetUseCustomSwfDimensions (FLIX2HANDLE flix, const on2bool bUseCustomSwfDimensions)

  Tells the engine to use the SWF width and SWF height for the SWF.

- on2sc video_options_GetSourceHeight (const FLIX2HANDLE flix, int32_t *height)

  Get the height of the source video.

- on2sc video_options_GetSourceWidth (const FLIX2HANDLE flix, int32_t *width)

  Get the width of the source video.

20.55.1 Function Documentation

20.55.1.1 on2sc video_options_GetSourceHeight (const FLIX2HANDLE *flix*, int32_t * *height*)

Get the height of the source video.

Parameters:

←*flix* Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***height*** Source video height, in pixels

Return values:

ON2_OK The source video size was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

20.55.1.2 on2sc video_options_GetSourceWidth (const FLIX2HANDLE *flix*, int32_t * *width*)

Get the width of the source video.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***width*** Source video width, in pixels

Return values:

ON2_OK The source video size was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

20.55.1.3 on2sc video_options_GetSwfHeight (const FLIX2HANDLE *flix*, int32_t * *lpSwfHeight*)

Gets the SWF height.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpSwfHeight*** SWF height.

Return values:

ON2_OK Success.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_HEIGHT parameter. This function will be removed in a future release.

20.55.1.4 on2sc video_options_GetSwfWidth (const FLIX2HANDLE *flix*, int32_t * *lpSwfWidth*)

Gets the SWF width.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpSwfWidth*** SWF width.

Return values:

ON2_OK Success.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH parameter. This function will be removed in a future release.

20.55.1.5 on2sc video_options_GetUseCustomSwfDimensions (const FLIX2HANDLE *flix*, on2bool * *bpUseCustomSwfDimensions*)

Determine if the engine is using the SWF width and height for the SWF.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***bpUseCustomSwfDimensions*** Variable to update with the current custom SWF dimensions enable status

Return values:

ON2_OK Success.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH/FE2_SWF_-HEIGHT parameters. This function will be removed in a future release.

20.55.1.6 on2sc video_options_Reset (FLIX2HANDLE *flix*)

Reset the video options to their defaults.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK The video options were reset successfully.

ON2_NET_ERROR The underlying communication layer failed.

20.55.1.7 on2sc video_options_SetSwfHeight (FLIX2HANDLE *flix*, const int32_t *lSwfHeight*)

Sets the SWF height.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lSwfHeight*** SWF height.

Return values:

ON2_OK Success.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the SWF height will be the video height.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_HEIGHT parameter. This function will be removed in a future release.

20.55.1.8 on2sc video_options_SetSwfWidth (FLIX2HANDLE *flix*, const int32_t *lSwfWidth*)

Sets the SWF width.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lSwfWidth*** SWF width.

Return values:

ON2_OK Success.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the SWF width will be the video width.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH parameter. This function will be removed in a future release.

20.55.1.9 on2sc video_options_SetUseCustomSwfDimensions (FLIX2HANDLE *flix*, const on2bool *bUseCustomSwfDimensions*)

Tells the engine to use the SWF width and SWF height for the SWF.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseCustomSwfDimensions*** on2true to enable, on2false to disable.

Return values:

ON2_OK Success.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, this is disabled.
This will automatically be set to enabled if video_options_SetSwfWidth() or video_options_-SetSwfHeight() is called.

Deprecated

Use the Muxer Interface along with FE2_MUXER_SWF and the FE2_SWF_WIDTH/FE2_SWF_-HEIGHT parameters. This function will be removed in a future release.

20.55.1.10 on2sc video_options_Validate (const FLIX2HANDLE *flix*)

Ensure the current video settings are valid.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

Return values:

ON2_OK The video options are valid and encoding may proceed.

ON2_INVALID_PARAMS One or more options are invalid. Proceeding with encoding may cause encoding to fail or a file that does not match the specified options will be created.

ON2_NET_ERROR The underlying communication layer failed.

20.56 Deprecated

Enumerations

- enum FE2_VideoCodec {
  CODEC_NULL,
  CODEC_H263,
  CODEC_SCREENVIDEO,
  CODEC_VP6,
  CODEC_VP6ALPHA }

  Output video codec types, influences quality/compatibility.

- enum FE2_CuePointType {
  CUE_EVENT,
  CUE_NAVIGATION }

  Cue point type for use with video_options_AddFLVCuePoint().

Functions

- on2sc video_options_GetImageQuality (const FLIX2HANDLE flix, int32_t *lpImageQuality)

  Get the current image quality factor.

- on2sc video_options_SetImageQuality (FLIX2HANDLE flix, const int32_t lImageQuality)

  Set the image quality factor.

- on2sc video_options_GetKeyframeInterval (const FLIX2HANDLE flix, int32_t *lpKeyframeInterval)

  Get the current interval between keyfames.

- on2sc video_options_SetKeyframeInterval (FLIX2HANDLE flix, const int32_t lKeyframeInterval)

  Set the interval between keyfames.

- on2sc video_options_GetKeyframeIntervalType (const FLIX2HANDLE flix, FE2_VideoKeyframeTypes *pKeyframeIntervalType)

  Get the current keyframe interval type.

- on2sc video_options_SetKeyframeIntervalType (FLIX2HANDLE flix, const FE2_VideoKeyframeTypes keyframeIntervalType)

  Set the keyframe interval type.

- on2sc video_options_GetMaximumBitrate (const FLIX2HANDLE flix, int32_t *lpMaximumBitrate)

  Get the current maximum bitrate target.

- on2sc video_options_SetMaximumBitrate (FLIX2HANDLE flix, const int32_t lMaximumBitrate)

  Set the maximum bitrate target.

- on2sc video_options_GetRateControlType (const FLIX2HANDLE flix, FE2_VideoBitrateControls *pRateControlType)

  Get the current rate control type.

- on2sc video_options_SetRateControlType (FLIX2HANDLE flix, const FE2_VideoBitrateControls rateControlType)

  Set the rate control type.

- on2sc video_options_GetUseMaximumBitrate (const FLIX2HANDLE flix, on2bool *bpUseMaximumBitrate)

  Determine if the maximum bitrate target will be used.

- on2sc video_options_SetUseMaximumBitrate (FLIX2HANDLE flix, const on2bool bUseMaximumBitrate)

  Enable/disable the maximum bitrate target value.

- on2sc video_options_GetSwfFramerate (const FLIX2HANDLE flix, int32_t *pSwfFramerate)

  Gets the SWF framerate.

- on2sc video_options_SetSwfFramerate (FLIX2HANDLE flix, const int32_t swfFramerate)

  Sets the SWF framerate.

- on2sc video_options_GetVideoCodec (const FLIX2HANDLE flix, FE2_VideoCodec *pVideoCodec)

  Determine which video codec will be used.

- on2sc video_options_SetVideoCodec (FLIX2HANDLE flix, const FE2_VideoCodec videoCodec)

  Select the video codec to be used.

- on2sc video_options_GetAlphaPercentage (const FLIX2HANDLE flix, int32_t *percentage)

  Gets the percentage of the video bitrate to be used to encode the alpha channel.

- on2sc video_options_SetAlphaPercentage (FLIX2HANDLE flix, const int32_t percentage)

  Sets the percentage of the video bitrate to be used to encode the alpha channel.

- on2sc video_options_AddFLVCuePoint (FLIX2HANDLE flix, const char *pName, const double time, const FE2_CuePointType type)

  Adds a cue point to the FLV.

- on2sc video_options_AddFLVCuePointParameter (FLIX2HANDLE flix, const char *pCuePointName, const char *pName, const char *pValue)

  Adds a name/value parameter to an already added cue point.

- on2sc video_options_GetCompressMode (const FLIX2HANDLE flix, FE2_CompressMode *mode)

  Get the current VP6 compression mode.

- on2sc video_options_SetCompressMode (FLIX2HANDLE flix, FE2_CompressMode mode)

  Set the VP6 compression mode.

20.56.1 Enumeration Type Documentation

20.56.1.1 enum FE2_CuePointType

Cue point type for use with video_options_AddFLVCuePoint().

Deprecated

Use the Muxer Interface along with FE2_MUXER_FLV and the FE2_FLV_CUEPT_EVENT/FE2_-FLV_CUEPT_NAV parameter. This enumeration will be removed in a future release.

Enumerator:

CUE_EVENT Trigger an event at a specified point

CUE_NAVIGATION Allows the user to seek to a specified point

Definition at line 52 of file video_options.h.

20.56.1.2 enum FE2_VideoCodec

Output video codec types, influences quality/compatibility.

For use in calls to video_options_SetVideoCodec() and video_options_GetVideoCodec()

Deprecated

Use the Codec Interface. This enumeration will be removed in a future release. See also: Video Codecs.

Enumerator:

CODEC_NULL place holder, not for external use

CODEC_H263 H263 codec

CODEC_SCREENVIDEO Screen Video codec, NOT supported

CODEC_VP6 VP6 (Flash8) codec

CODEC_VP6ALPHA VP6 + encoded alpha channel

Definition at line 38 of file video_options.h.

20.56.2 Function Documentation

20.56.2.1 on2sc video_options_AddFLVCuePoint (FLIX2HANDLE *flix*, const char * *pName*, const double *time*, const FE2_CuePointType *type*)

Adds a cue point to the FLV.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***pName*** Name of the cue point.

← ***time*** Time of the cue point in seconds.

← ***type*** Cue point type.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for the new cue point.

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pName is not `NULL`
time >= 0.0
type is a valid member of FE2_CuePointType

Note:

Cue points can only be added to FLV files.
May add multiple cue points.
For more information on cue points please see `http://livedocs.macromedia.com/flash/8/main/wwhelp/Parts&file=00001574.html`

Deprecated

Use the Muxer Interface along with FE2_MUXER_FLV and the FE2_FLV_CUEPT_EVENT/FE2_-FLV_CUEPT_NAV parameter. This function will be removed in a future release.

20.56.2.2 on2sc video_options_AddFLVCuePointParameter (FLIX2HANDLE *flix*, const char * *pCuePointName*, const char * *pName*, const char * *pValue*)

Adds a name/value parameter to an already added cue point.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***pCuePointName*** Name of the cue point to add the parameter to.

← ***pName*** Name of the parameter.

← ***pValue*** Value of the parameter.

Return values:

ON2_OK on success

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NO_MEM an error occurred allocating memory for the new cue point.

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pCuePointName is not `NULL`
pName is not `NULL`
pValue is not `NULL`
pCuePointName has already been added to the cue point list

Note:

Cue points can only be added to FLV files.
May add multiple parameters to a cue point.

Deprecated

Use the Muxer Interface along with FE2_MUXER_FLV and the FE2_FLV_CUEPT_PARAM parameter. This function will be removed in a future release.

20.56.2.3 on2sc video_options_GetAlphaPercentage (const FLIX2HANDLE *flix*, int32_t * *percentage*)

Gets the percentage of the video bitrate to be used to encode the alpha channel.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***percentage*** Alpha percentage

Return values:

ON2_OK Success.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface along with FE2_CODEC_VP6ALPHA and the FE2_VP6A_ALPHA_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.4 on2sc video_options_GetCompressMode (const FLIX2HANDLE *flix*, FE2_CompressMode * *mode*)

Get the current VP6 compression mode.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***mode*** Compression mode (one of FE2_CompressMode)

Return values:

ON2_OK The rate control type was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface along with FE2_CODEC_VP6 or FE2_CODEC_VP6ALPHA and the FE2_-VP6_CXMODE or FE2_VP6A_CXMODE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.5 on2sc video_options_GetImageQuality (const FLIX2HANDLE *flix*, int32_t * *lpImageQuality*)

Get the current image quality factor.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpImageQuality*** Image quality (0..100)

Return values:

ON2_OK The quality value was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

The image quality functions have been deprecated in favor of the maximum bitrate functions. Use video_options_GetMaximumBitrate() instead.

20.56.2.6 on2sc video_options_GetKeyframeInterval (const FLIX2HANDLE *flix*, int32_t * *lpKeyframeInterval*)

Get the current interval between keyfames.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpKeyframeInterval*** Keyframe interval (in frames)

Return values:

ON2_OK The keyframe interval was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface along with the FE2_VCODECPARAM_KFFREQ parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.7 on2sc video_options_GetKeyframeIntervalType (const FLIX2HANDLE *flix*, FE2_VideoKeyframeTypes * *pKeyframeIntervalType*)

Get the current keyframe interval type.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pKeyframeIntervalType*** Keyframe interval type (one of FE2_VideoKeyframeTypes)

Return values:

ON2_OK The keyframe interval type was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface along with the FE2_VCODECPARAM_KFINTTYPE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.8 on2sc video_options_GetMaximumBitrate (const FLIX2HANDLE *flix*, int32_t * *lpMaximumBitrate*)

Get the current maximum bitrate target.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***lpMaximumBitrate*** Target bitrate (in Kbps)

Return values:

ON2_OK The keyframe interval type was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface along with the FE2_CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.9 on2sc video_options_GetRateControlType (const FLIX2HANDLE *flix*, FE2_VideoBitrateControls * *pRateControlType*)

Get the current rate control type.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***pRateControlType*** Rate control type (one of FE2_VideoBitrateControls)

Return values:

ON2_OK The rate control type was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface along with the FE2_VCODECPARAM_RC_MODE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.10 on2sc video_options_GetSwfFramerate (const FLIX2HANDLE *flix*, int32_t * *pSwfFramerate*)

Gets the SWF framerate.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***pSwfFramerate*** The SWF framerate.

Return values:

ON2_OK The engine successfully returned SWF framerate.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`
pSwfFramerate is not `NULL`

Deprecated

Please use swf_options_GetSwfFramerateAsDouble() to allow for non integer framerates.

20.56.2.11 on2sc video_options_GetUseMaximumBitrate (const FLIX2HANDLE *flix*, on2bool * *bpUseMaximumBitrate*)

Determine if the maximum bitrate target will be used.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ ***bpUseMaximumBitrate*** Variable to update with the current bitrate enable status

Return values:

ON2_OK The bitrate enable status was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface along with the FE2_CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.12 on2sc video_options_GetVideoCodec (const FLIX2HANDLE *flix*, FE2_VideoCodec * *pVideoCodec*)

Determine which video codec will be used.

Parameters:

← *flix* Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

→ *pVideoCodec* Current video codec (one of FE2_VideoCodec)

Return values:

ON2_OK The selected video codec was successfully retrieved from the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

If a value other than ON2_OK is returned, the output variables should not be considered valid.

Deprecated

Use the Codec Interface. This function will be removed in a future release. See also: Video Codecs.

20.56.2.13 on2sc video_options_SetAlphaPercentage (FLIX2HANDLE *flix*, const int32_t *percentage*)

Sets the percentage of the video bitrate to be used to encode the alpha channel.

Parameters:

← *flix* Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← *percentage* Alpha percentage

Return values:

ON2_OK Success.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the percentage is 15%.
The range is 0-100%.
The codec type must be CODEC_VP6ALPHA.
For example if the maximum video bitrate was set to 500 through video_options_SetMaximumBitrate(), and the alpha percentage was set to 15% then the bitrate for the VP6 video would be 425 and the alpha bitrate would be 75.

Deprecated

Use the Codec Interface along with FE2_CODEC_VP6ALPHA and the FE2_VP6A_ALPHA_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.14 on2sc video_options_SetCompressMode (FLIX2HANDLE *flix*, FE2_CompressMode *mode*)

Set the VP6 compression mode.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***mode*** Compression mode (one of FE2_CompressMode)

Return values:

ON2_OK The compression mode was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This setting affects VP6 only. COMPRESSMODE_BEST provides a slightly better quality image, but takes approximately twice as long to encode as VP6 set to COMPRESSMODE_GOOD.
The default value is COMPRESSMODE_GOOD.

Deprecated

Use the Codec Interface along with FE2_CODEC_VP6 or FE2_CODEC_VP6ALPHA and the FE2_VP6_CXMODE or FE2_VP6A_CXMODE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.15 on2sc video_options_SetImageQuality (FLIX2HANDLE *flix*, const int32_t *lImageQuality*)

Set the image quality factor.

Determines the image "quality" level. Higher numbers will generally result in higher quality video, at the cost of higher bitrates and file sizes.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lImageQuality*** Quality Factor (0 to 100)

Return values:

ON2_OK The quality value was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

The default value is 75.

Deprecated

The image quality functions have been deprecated in favor of the maximum bitrate functions. Use video_options_GetMaximumBitrate() instead.

20.56.2.16 on2sc video_options_SetKeyframeInterval (FLIX2HANDLE *flix*, const int32_t *lKeyframeInterval*)

Set the interval between keyfames.

Determines the interval (in frames) between keyframes. Keyframes "refresh" the player with the best possible quality image, and subsequent images are derived from that image. In addition, the keyframe interval determines the granularity at which seeking can happen (i.e., the player can only seek to a keyframe). Reducing the keyframe interval increases the number of seekable points in the video. However, keyframes take significantly more bits to encode than non-keyframes, which can result in unnecessarily large files if the value is set too low. For most purposes, the default value is preferred.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lKeyframeInterval*** Keyframe interval (in frames)

Return values:

ON2_OK The keyframe interval was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

The default value is $12 * framerate$ (12 seconds worth)

Deprecated

Use the Codec Interface along with the FE2_VCODECPARAM_KFFREQ parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.17 on2sc video_options_SetKeyframeIntervalType (FLIX2HANDLE *flix*, const FE2_VideoKeyframeTypes *keyframeIntervalType*)

Set the keyframe interval type.

Two keyframe modes are supported, MAX_KEYFRAMES (the default) and FIXED_KEYFRAMES. These specify that the keyframe interval set using video_options_SetKeyframeInterval() is either the maximum interval between keyframes or a fixed interval, respectively. In general, the compression codec does the best job of deciding when keyframes should be used, so setting the maximum interval is usually more appropriate.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***keyframeIntervalType*** Keyframe interval type (one of FE2_VideoKeyframeTypes)

Return values:

ON2_OK The keyframe interval was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

The default value is MAX_KEYFRAMES

Deprecated

Use the Codec Interface along with the FE2_VCODECPARAM_KFINTTYPE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.18 on2sc video_options_SetMaximumBitrate (FLIX2HANDLE *flix*, const int32_t *lMaximumBitrate*)

Set the maximum bitrate target.

Sets the bitrate the compressor will target when encoding the video.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***lMaximumBitrate*** Bitrate target (in Kbps)

Return values:

ON2_OK The target bitrate was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

This bitrate should be a reasonable number with respect to the other video parameters. To determine if a particular value is reasonable or not, first calculate the number of bits per pixel it represents, given the bitrate in Kbps, framerate in frames per second, and width and height in pixels, using this equation:

$$bpp = \frac{bitrate * 1024}{width * height * framerate}$$

The H.263 codec is usable down to 0.05 bpp. For lower bitrates, VP6 must be used.
The default value is 448 Kbps.
Setting this value will have no effect unless enabled via video_options_SetUseMaximumBitrate()

Deprecated

Use the Codec Interface along with the FE2_CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

See also:

Video Codecs

20.56.2.19 on2sc video_options_SetRateControlType (FLIX2HANDLE *flix*, const FE2_VideoBitrateControls *rateControlType*)

Set the rate control type.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***rateControlType*** Rate control type (one of FE2_VideoBitrateControls)

Return values:

ON2_OK The rate control type was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

The default value is VBR_2PASSControl.

Deprecated

Use the Codec Interface along with the FE2_VCODECPARAM_RC_MODE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.20 on2sc video_options_SetSwfFramerate (FLIX2HANDLE *flix*, const int32_t *swfFramerate*)

Sets the SWF framerate.

Parameters:

← ***flix*** Handle to the flix engine returned from Flix2_Create() or Flix2_CreateEx()

← ***swfFramerate*** The SWF framerate.

Return values:

ON2_OK The engine successfully set the SWF framerate.

ON2_INVALID_PARAMS should one or more of the preconditions fail

ON2_NET_ERROR the underlying communication layer failed

Precondition:

flix is not `NULL`

Deprecated

Please use swf_options_SetSwfFramerateAsDouble() to allow for non integer framerates.

20.56.2.21 on2sc video_options_SetUseMaximumBitrate (FLIX2HANDLE *flix*, const on2bool *bUseMaximumBitrate*)

Enable/disable the maximum bitrate target value.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***bUseMaximumBitrate*** New bitrate enable status. on2true to enable, on2false to disable.

Return values:

ON2_OK The bitrate enable status was successfully set in the engine.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the deinterlace filter is disabled.

Deprecated

Use the Codec Interface along with the FE2_CODECPARAM_BITRATE parameter. This function will be removed in a future release. See also: Video Codecs.

20.56.2.22 on2sc video_options_SetVideoCodec (FLIX2HANDLE *flix*, const FE2_VideoCodec *videoCodec*)

Select the video codec to be used.

Parameters:

← ***flix*** Handle to the Flix Engine returned from Flix2_Create() or Flix2_CreateEx()

← ***videoCodec*** New video codec (one of FE2_VideoCodec)

Return values:

ON2_OK The video codec was successfully set in the engine.

ON2_INVALID_PARAMS The value is out of range.

ON2_NET_ERROR The underlying communication layer failed.

Note:

By default, the CODEC_VP6 will be used.

Deprecated

Use the Codec Interface. This function will be removed in a future release. See also: Video Codecs.

Chapter 21

File Documentation

21.1 about.dox File Reference

21.2 binding_deviations.dox File Reference

21.3 flixengine2/audio_options.h File Reference

21.3.1 Detailed Description

Reference module(s): Audio Encoding Options.

Definition in file audio_options.h.

Enumerations

- enum FE2_AudioBitrates {
  Bitrate8kbps,
  Bitrate16kbps,
  Bitrate24kbps,
  Bitrate32kbps,
  Bitrate40kbps,
  Bitrate48kbps,
  Bitrate56kbps,
  Bitrate64kbps,
  Bitrate80kbps,
  Bitrate96kbps,
  Bitrate112kbps,
  Bitrate128kbps,
  Bitrate144kbps,
  Bitrate160kbps,
  Bitrate192kbps,
  Bitrate224kbps,
  Bitrate256kbps,
  Bitrate320kbps }

  Valid bitrates for FE2_CODEC_LAME. For use in calls to Flix2_CodecSetParam() and Flix2_CodecGetParam().

- enum FE2_FlvAudioFormat {
  FlvAudioUncompressed,
  FlvAudioMp3 }

  Valid output audio formats for use in calls to audio_options_GetFlvAudioFormat() and audio_options_SetFlvAudioFormat().

- enum FE2_AudioSamplingrates {
  Hertz11025,
  Hertz22050,
  Hertz44100 }

  Sample rates for use with FE2_FILTER_RESAMPLE and the FE2_RESAMPLE_RATE parameter.

Functions

- on2sc audio_options_Reset (FLIX2HANDLE flix)

  Reset the audio encoding options to their default values.

- on2sc audio_options_Validate (FLIX2HANDLE flix)

  Validate the current audio encoding options.

- on2sc audio_options_SetBitrate (FLIX2HANDLE flix, const FE2_AudioBitrates bitrate)

  Set the bitrate to use in mp3 encoding.

- on2sc audio_options_GetBitrate (const FLIX2HANDLE flix, FE2_AudioBitrates *pBitrate)

  Retrieve the current mp3 bitrate.

- on2sc audio_options_SetFlvAudioFormat (FLIX2HANDLE flix, const FE2_FlvAudioFormat format)

  Set the output audio format.

- on2sc audio_options_GetFlvAudioFormat (const FLIX2HANDLE flix, FE2_FlvAudioFormat *pFormat)

  Retrieve the current output audio format.

- on2sc audio_options_SetSamplingrate (FLIX2HANDLE flix, const FE2_AudioSamplingrates samplingrate)

  Set the output audio sample rate.

- on2sc audio_options_GetSamplingrate (const FLIX2HANDLE flix, FE2_AudioSamplingrates *pSamplingrate)

  Retrieve the current output sample rate.

- on2sc audio_options_SetStereo (FLIX2HANDLE flix, const on2bool stereo)

  Set the number of output audio channels.

- on2sc audio_options_GetStereo (const FLIX2HANDLE flix, on2bool *pStereo)

  Retrieve the current number of output audio channels.

21.4 flixengine2/codec_constants.h File Reference

21.4.1 Detailed Description

Reference module(s): Codecs.

Definition in file codec_constants.h.

```
#include "codecs/codec_common.h"
#include "codecs/h263.h"
#include "codecs/h264.h"
#include "codecs/vp6.h"
#include "codecs/vp6_alpha.h"
#include "codecs/vp8.h"
#include "codecs/aac.h"
#include "codecs/amr.h"
#include "codecs/lame.h"
#include "codecs/vorbis.h"
```

Defines

- #define FE2_CODEC_PCM

  'Codec' name for use with Flix2_AddCodec()

21.5 flixengine2/codecs/aac.h File Reference

21.5.1 Detailed Description

Reference module(s): AAC, AAC+.

Definition in file aac.h.

Defines

- #define FE2_CODEC_AAC

  AAC. Codec name for use with Flix2_AddCodec().

- #define FE2_AAC_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_CODEC_AACPLUS

  AAC+ (HE-AAC). Codec name for use with Flix2_AddCodec().

- #define FE2_AACPLUS_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_AACPLUS_PARAMETRIC_STEREO

  Parameter for parametric stereo.

21.6 flixengine2/codecs/amr.h File Reference

21.6.1 Detailed Description

Reference module(s): AMR_NB - FFmpeg.

Definition in file amr.h.

Defines

- #define FE2_CODEC_AMR_NB

  AMR Narrowband. Codec name for use with Flix2_AddCodec().

- #define FE2_AMR_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

21.7 flixengine2/codecs/codec_common.h File Reference

Defines

- #define FE2_CODECPARAM_BITRATE

  Codec parameter for stream bitrate.

- #define FE2_VCODECPARAM_RC_MODE

  Video codec parameter for the rate control mode.

- #define FE2_VCODECPARAM_KFINTTYPE

  Video codec parameter for the keyframe interval type.

- #define FE2_VCODECPARAM_KFFREQ

  Video codec parameter for keyframe frequency.

Enumerations

- enum FE2_VideoBitrateControls {
  CBR_1PASSControl,
  VBR_1PASSControl,
  CBR_2PASSControl,
  VBR_2PASSControl }

  Encoder rate control types, influences quality.

- enum FE2_VideoKeyframeTypes {
  MAX_KEYFRAMES,
  FIXED_KEYFRAMES }

  Key frame interval types, influences quality.

21.8 flixengine2/codecs/h263.h File Reference

21.8.1 Detailed Description

Reference module(s): H263 - FFmpeg.

Definition in file h263.h.

Defines

- #define FE2_CODEC_H263

  Codec name for use with Flix2_AddCodec().

- #define FE2_CODEC_H263_BASELINE

  Codec name for use with Flix2_AddCodec().

- #define FE2_H263_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_H263_KFINTTYPE

  Alias for FE2_VCODECPARAM_KFINTTYPE.

- #define FE2_H263_KFFREQ

  Alias for FE2_VCODECPARAM_KFFREQ.

- #define FE2_H263_RC_MODE

  Alias for FE2_VCODECPARAM_RC_MODE.

- #define FE2_H263_MIN_Q

  Codec parameter for minimum quantizer.

- #define FE2_H263_MAX_Q

  Codec parameter for maximum quantizer.

21.9 flixengine2/codecs/h264.h File Reference

21.9.1 Detailed Description

Reference module(s): H264.

Definition in file h264.h.

Defines

- #define FE2_CODEC_H264

  Codec name for use with Flix2_AddCodec().

- #define FE2_H264_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_H264_KFINTTYPE

  Alias for FE2_VCODECPARAM_KFINTTYPE.

- #define FE2_H264_KFFREQ

  Alias for FE2_VCODECPARAM_KFFREQ.

- #define FE2_H264_RC_MODE

  Alias for FE2_VCODECPARAM_RC_MODE.

- #define FE2_H264_PROFILE

  Codec parameter for encoding profile.

- #define FE2_H264_B_FRAME_RATE

  Codec parameter for B frame rate.

- #define FE2_H264_SPEED

  Controls frame analysis, affecting encoder speed and inversely output quality.

Typedefs

- typedef enum h264profile h264profile_t

Enumerations

- enum h264profile {
  BASE_H264PROFILE,
  MAIN_H264PROFILE,
  HIGH_H264PROFILE }

  Valid profiles for use with the FE2_H264_PROFILE parameter.

21.10 flixengine2/codecs/lame.h File Reference

21.10.1 Detailed Description

Reference module(s): MP3 - LAME.

Definition in file lame.h.

Defines

- #define FE2_CODEC_LAME

  Codec name for use with Flix2_AddCodec().

- #define FE2_LAME_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_LAME_QUALITY

  Codec parameter for controlling LAME library's algorithm selection.

- #define FE2_LAME_RC_MODE

  Codec parameter for controlling LAME library's rate control method.

- #define FE2_LAME_CHANNELS

  Codec parameter for number of output channels.

Typedefs

- typedef enum lame_rcmode lame_rcmode_t

Enumerations

- enum lame_rcmode {
  LAME_CBR,
  LAME_ABR,
  LAME_VBR_rh,
  LAME_VBR_mtrh }

  Rate control modes analogous to those found in `<lame/lame.h>`.

21.11 flixengine2/codecs/vorbis.h File Reference

21.11.1 Detailed Description

Reference module(s): libvorbis - FFmpeg.

Definition in file vorbis.h.

Defines

- #define FE2_CODEC_VORBIS

  Vorbis. Codec name for use with Flix2_AddCodec().

- #define FE2_VORBIS_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

21.12 flixengine2/codecs/vp6.h File Reference

21.12.1 Detailed Description

Reference module(s): VP6.

Definition in file vp6.h.

General VP6 settings

- #define FE2_VP6_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_VP6_KFINTTYPE

  Alias for FE2_VCODECPARAM_KFINTTYPE.

- #define FE2_VP6_KFFREQ

  Alias for FE2_VCODECPARAM_KFFREQ.

- #define FE2_VP6_RC_MODE

  Alias for FE2_VCODECPARAM_RC_MODE.

- #define FE2_VP6_CXMODE

  VP6 compress mode.

- #define FE2_VP6_SHARPNESS

  Codec parameter for sharpness.

- #define FE2_VP6_NOISE_REDUCTION

  Codec parameter for noise reduction.

- #define FE2_VP6_PROFILE

  Codec parameter for encoding profile.

- enum FE2_CompressMode {
  COMPRESSMODE_GOOD,
  COMPRESSMODE_BEST }

  Valid compress modes for VP6, influences encoder speed.

- enum vp6profile {
  VP6_E,
  VP6_S }

  Valid profiles for use with the FE2_VP6_PROFILE parameter.

- typedef enum vp6profile vp6profile_t

Defines

Codec name

- #define FE2_CODEC_VP6
  Codec name for use with Flix2_AddCodec().

Advanced VP6 settings

- #define FE2_VP6_CONCURRENCY
  Codec parameter for concurrency level.
- #define FE2_VP6_UNDERSHOOT_PCT
  Codec parameter for undershoot percentage.
- #define FE2_VP6_MIN_Q
  Codec parameter for minimum quantizer.
- #define FE2_VP6_MAX_Q
  Codec parameter for maximum quantizer.
- #define FE2_VP6_TEMPORAL_RESAMPLING
  Codec parameter for temporal resampling.
- #define FE2_VP6_TEMPORAL_DOWN_WATERMARK
  Codec parameter for temporal down watermark percentage.

CBR-Specific

- #define FE2_VP6_STREAM_PEAK_BITRATE
  The maximum bitrate allowed in the stream.
- #define FE2_VP6_STREAM_PREBUFFER
  Seconds of preload that are necessary before starting playback.
- #define FE2_VP6_STREAM_OPTIMAL_BUFFER
  Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.
- #define FE2_VP6_STREAM_MAX_BUFFER
  The maximum size of the buffer, in seconds.

VBR-Specific

- #define FE2_VP6_2PASS_MIN_SECTION
  VBR_2PASSControl minimum section datarate
- #define FE2_VP6_2PASS_MAX_SECTION
  VBR_2PASSControl maximum section datarate

21.13 flixengine2/codecs/vp6_alpha.h File Reference

21.13.1 Detailed Description

Reference module(s): VP6 with Alpha.

Definition in file vp6_alpha.h.

Defines

Codec name

- #define FE2_CODEC_VP6ALPHA
  Codec name for use with Flix2_AddCodec().

General VP6 settings

- #define FE2_VP6A_BITRATE
  Alias for FE2_CODECPARAM_BITRATE.
- #define FE2_VP6A_ALPHA_BITRATE
  Compressed bitrate of the alpha channel in kbit/s.
- #define FE2_VP6A_KFINTTYPE
  Alias for FE2_VP6_KFINTTYPE.
- #define FE2_VP6A_KFFREQ
  Alias for FE2_VP6_KFFREQ.
- #define FE2_VP6A_RC_MODE
  Alias for FE2_VCODECPARAM_RC_MODE.
- #define FE2_VP6A_CXMODE
  Alias for FE2_VP6_CXMODE.
- #define FE2_VP6A_SHARPNESS
  Alias for FE2_VP6_SHARPNESS.
- #define FE2_VP6A_ALPHA_SHARPNESS
  Sharpness for the alpha channel.
- #define FE2_VP6A_NOISE_REDUCTION
  Alias for FE2_VP6_NOISE_REDUCTION.
- #define FE2_VP6A_ALPHA_NOISE_REDUCTION
  Noise reduction for the alpha channel.

Advanced VP6 settings

- #define FE2_VP6A_UNDERSHOOT_PCT
  Alias for FE2_VP6_UNDERSHOOT_PCT.

- #define FE2_VP6A_MIN_Q
  Alias for FE2_VP6_MIN_Q.
- #define FE2_VP6A_ALPHA_MIN_Q
  Minimum quantizer for the alpha channel.
- #define FE2_VP6A_MAX_Q
  Alias for FE2_VP6_MAX_Q.
- #define FE2_VP6A_ALPHA_MAX_Q
  Maximum quantizer for the alpha channel.
- #define FE2_VP6A_TEMPORAL_RESAMPLING
  Alias for FE2_VP6_TEMPORAL_RESAMPLING.
- #define FE2_VP6A_TEMPORAL_DOWN_WATERMARK
  Alias for FE2_VP6_TEMPORAL_DOWN_WATERMARK.
- #define FE2_VP6A_STREAM_PEAK_BITRATE
  Alias for FE2_VP6_STREAM_PEAK_BITRATE.
- #define FE2_VP6A_STREAM_PREBUFFER
  Alias for FE2_VP6_STREAM_PREBUFFER.
- #define FE2_VP6A_STREAM_OPTIMAL_BUFFER
  Alias for FE2_VP6_STREAM_OPTIMAL_BUFFER.
- #define FE2_VP6A_STREAM_MAX_BUFFER
  Alias for FE2_VP6_STREAM_MAX_BUFFER.
- #define FE2_VP6A_2PASS_MIN_SECTION
  Alias for FE2_VP6_2PASS_MIN_SECTION.
- #define FE2_VP6A_2PASS_MAX_SECTION
  Alias for FE2_VP6_2PASS_MAX_SECTION.

21.14 flixengine2/codecs/vp8.h File Reference

21.14.1 Detailed Description

Reference module(s): VP8.

Definition in file vp8.h.

Defines

- #define FE2_VP8_PROFILE

  Bitstream profile.

- #define FE2_VP8_ALTREF

  Enable the use of alternate reference frames.

- #define FE2_VP8_AR_MAX_FRAMES

  Max number of frames blurred creating alternate reference.

- #define FE2_VP8_AR_TYPE

  Filter type to use w/alternate reference.

- #define FE2_VP8_AR_STRENGTH

  Filter strength for the alternate reference.

- #define FE2_VP8_MB_STATIC_THRESHOLD

  Threshold for macroblocks treated static.

- #define FE2_VP8_TOKEN_PARTITIONS

  Number of token partitions.

- #define FE2_VP8_LAG

  Allow lagged encoding If set, this value allows the encoder to consume a number of input frames before producing output frames. This allows the encoder to base decisions for the current frame on future frames.

- #define FE2_VP8_THREADS

  Number of threads to use A reasonable selection would be the number of cores on the system.

Codec name

- #define FE2_CODEC_VP8

  Codec name for use with Flix2_AddCodec().

General VP8 settings

- #define FE2_VP8_BITRATE

  Alias for FE2_CODECPARAM_BITRATE.

- #define FE2_VP8_KFINTTYPE

Alias for FE2_VCODECPARAM_KFINTTYPE.

- #define FE2_VP8_KFFREQ
  Alias for FE2_VCODECPARAM_KFFREQ.
- #define FE2_VP8_RC_MODE
  Alias for FE2_VCODECPARAM_RC_MODE.
- #define FE2_VP8_CXMODE
  VP8 compress mode.
- #define FE2_VP8_SHARPNESS
  Codec parameter for sharpness.
- #define FE2_VP8_NOISE_REDUCTION
  Controls encoder noise reduction preprocessing.

Advanced VP8 settings

- #define FE2_VP8_UNDERSHOOT_PCT
  Codec parameter for undershoot percentage.
- #define FE2_VP8_OVERSHOOT_PCT
  Codec parameter for undershoot percentage.
- #define FE2_VP8_MIN_Q
  Codec parameter for minimum quantizer.
- #define FE2_VP8_MAX_Q
  Codec parameter for maximum quantizer.
- #define FE2_VP8_DROP_THRESH
  Threshold controlling encoder frame dropping.

CBR-Specific

- #define FE2_VP8_STREAM_INITIAL_BUFFER
  Seconds of preload that are necessary before starting playback.
- #define FE2_VP8_STREAM_OPTIMAL_BUFFER
  Buffer size that the encoder strives to reach or maintain in case of specific frame overshoots.
- #define FE2_VP8_STREAM_MAX_BUFFER
  The maximum size of the buffer, in seconds.

VBR-Specific

- #define FE2_VP8_2PASS_MIN_SECTION
  VBR_2PASSControl minimum section datarate
- #define FE2_VP8_2PASS_MAX_SECTION
  VBR_2PASSControl maximum section datarate

21.15 flixengine2/encoding_status.h File Reference

21.15.1 Detailed Description

Reference module(s): Encoding Statistics.

Definition in file encoding_status.h.

Functions

- on2sc encoding_status_GetAverageBitrate (const FLIX2HANDLE flix, int32_t *pBitrate)

  Retrieve the encoder's average (video) bitrate.

- on2sc encoding_status_GetAverageFramesize (const FLIX2HANDLE flix, int32_t *pFramesize)

  Retrieve the encoder's average (video) frame size.

- on2sc encoding_status_GetMaximumFramesize (const FLIX2HANDLE flix, int32_t *pMaxFramesize)

  Retrieve the encoder's maximum (video) frame size.

- on2sc encoding_status_GetMinimumFramesize (const FLIX2HANDLE flix, int32_t *pMinFramesize)

  Retrieve the encoder's minimum (video) frame size.

- on2sc encoding_status_GetTotalFrames (const FLIX2HANDLE flix, int32_t *pTotalFrames)

  Retrieve the total number of (video) frames encoded.

- on2sc encoding_status_GetElapsedTime (const FLIX2HANDLE flix, int32_t *pElapsedTime)

  Retrieve the total elapsed encode time.

- on2sc encoding_status_GetEndTime (const FLIX2HANDLE flix, int32_t *pEndTime)

  Retrieve the encode completion time.

- on2sc encoding_status_GetStartTime (const FLIX2HANDLE flix, int32_t *pStartTime)

  Retrieve the encode start time.

- on2sc encoding_status_PercentComplete (const FLIX2HANDLE flix, int32_t *percent)

  Retrieve the percent of the encode complete.

21.16 flixengine2/filter_constants.h File Reference

21.16.1 Detailed Description

Reference module(s): Filters.

Definition in file filter_constants.h.

```
#include "filters/cut.h"
#include "filters/adaptive_deinterlace.h"
#include "filters/bchs.h"
#include "filters/blur.h"
#include "filters/crop.h"
#include "filters/denoise.h"
#include "filters/framerate.h"
#include "filters/mirror.h"
#include "filters/overlay.h"
#include "filters/png_export.h"
#include "filters/rotate.h"
#include "filters/scale.h"
#include "filters/sharpen.h"
#include "filters/highpass.h"
#include "filters/lowpass.h"
#include "filters/resample.h"
```

21.17 flixengine2/filters/adaptive_deinterlace.h File Reference

21.17.1 Detailed Description

Reference module(s): Deinterlace.

Definition in file adaptive_deinterlace.h.

Defines

- #define FE2_FILTER_ADAPTIVE_DEINTERLACE

  Filter name for use with Flix2_AddFilter().

- #define FE2_ADAPTIVE_DEINTERLACE_MODE

  Specifies deinterlace mode to be applied to source image.

Typedefs

- typedef enum deinterlacemode deintmode_t

Enumerations

- enum deinterlacemode {
  DEINTERLACE_NONE,
  DEINTERLACE_1_2_1_BLUR,
  DEINTERLACE_DROP_FIELD,
  DEINTERLACE_ADAPTIVE }

21.18 flixengine2/filters/bchs.h File Reference

21.18.1 Detailed Description

Reference module(s): Brightness/Contrast/Hue/Saturation.

Definition in file bchs.h.

Defines

- #define FE2_FILTER_BCHS

  Filter name for use with Flix2_AddFilter().

- #define FE2_BCHS_BRIGHTNESS

  Parameter for the brightness adjustment factor.

- #define FE2_BCHS_CONTRAST

  Parameter for the contrast adjustment factor.

- #define FE2_BCHS_HUE

  Parameter for the hue adjustment factor.

- #define FE2_BCHS_SATURATION

  Parameter for the saturation adjustment factor.

21.19 flixengine2/filters/blur.h File Reference

21.19.1 Detailed Description

Reference module(s): Blur.

Definition in file blur.h.

Defines

- #define FE2_FILTER_BLUR

  Filter name for use with Flix2_AddFilter().

- #define FE2_BLUR_FILTER

  Specifies blur filter to be applied to source.

- #define FE2_BLUR_MASKSIZE

  Specifies the kernel/mask size to be used.

Typedefs

- typedef enum masksiz masksiz_t
- typedef enum blurfilter blurfilter_t

Enumerations

- enum masksiz {
  MASK_3x3,
  MASK_5x5 }

  Filter mask/kernel size.

- enum blurfilter {
  BLUR_LOWPASS,
  BLUR_GAUSS }

21.20 flixengine2/filters/crop.h File Reference

21.20.1 Detailed Description

Reference module(s): Crop.

Definition in file crop.h.

Defines

- #define FE2_FILTER_CROP

  Filter name for use with Flix2_AddFilter().

- #define FE2_CROP_TOP

  Filter parameter for setting bounding box's top coordinate.

- #define FE2_CROP_BOTTOM

  Filter parameter for setting bounding box's bottom coordinate.

- #define FE2_CROP_LEFT

  Filter parameter for setting bounding box's left coordinate.

- #define FE2_CROP_RIGHT

  Filter parameter for setting bounding box's right coordinate.

21.21 flixengine2/filters/cut.h File Reference

21.21.1 Detailed Description

Reference module(s): Cut.

Definition in file cut.h.

Defines

- #define FE2_FILTER_CUT

  Filter name for use with Flix2_AddFilter().

- #define FE2_CUT_START_SEC

  Filter parameter for cut start time.

- #define FE2_CUT_STOP_SEC

  Filter parameter for setting the cut stop time.

- #define FE2_CUT_USE_SEEK

  Set FE2_CUT_START_SEC application method.

21.22 flixengine2/filters/denoise.h File Reference

21.22.1 Detailed Description

Reference module(s): Denoise.

Definition in file denoise.h.

Defines

- #define FE2_FILTER_DENOISE

  Filter name for use with Flix2_AddFilter().

- #define FE2_DENOISE_NOISE_LEVEL

  Specifies noise level of the source.

21.23 flixengine2/filters/framerate.h File Reference

21.23.1 Detailed Description

Reference module(s): Frame Rate.

Definition in file framerate.h.

Defines

- #define FE2_FILTER_FRAMERATE

  Filter name for use with Flix2_AddFilter().

- #define FE2_FRAMERATE_FPS

  Filter parameter name for frames per second value.

- #define FE2_FRAMERATE_DECIMATE

  Filter parameter for the decimation interval.

21.24 flixengine2/filters/highpass.h File Reference

21.24.1 Detailed Description

Reference module(s): Highpass.

Definition in file highpass.h.

Defines

- #define FE2_FILTER_HIGHPASS

  Filter name for use with Flix2_AddFilter().

- #define FE2_HIGHPASS_Q

  Filter parameter for shape constant ("Q" coefficient).

- #define FE2_HIGHPASS_CUTOFF

  Filter parameter for cutoff frequency.

21.25 flixengine2/filters/lowpass.h File Reference

21.25.1 Detailed Description

Reference module(s): Lowpass.

Definition in file lowpass.h.

Defines

- #define FE2_FILTER_LOWPASS

  Filter name for use with Flix2_AddFilter().

- #define FE2_LOWPASS_Q

  Filter parameter for shape constant ("Q" coefficient).

- #define FE2_LOWPASS_CUTOFF

  Filter parameter for cutoff frequency.

21.26 flixengine2/filters/mirror.h File Reference

21.26.1 Detailed Description

Reference module(s): Mirror.

Definition in file mirror.h.

Defines

- #define FE2_FILTER_MIRROR

  Filter name for use with Flix2_AddFilter().

- #define FE2_MIRROR_HORIZONTAL

  Specifies horizontal disposition.

- #define FE2_MIRROR_VERTICAL

  Specifies vertical disposition.

21.27 flixengine2/filters/overlay.h File Reference

21.27.1 Detailed Description

Reference module(s): Overlay (Watermark).

Definition in file overlay.h.

Defines

- #define FE2_FILTER_OVERLAY

  Filter name for use with Flix2_AddFilter().

- #define FE2_OVERLAY_FILE

  Set the path to the overlay image file.

- #define FE2_OVERLAY_MASK_XY

  Use the pixel at coordinates (FE2_OVERLAY_MASK_X,FE2_OVERLAY_MASK_Y) to determine the transparent color.

- #define FE2_OVERLAY_MASK_X

  X coordinate of pixel to use for transparency.

- #define FE2_OVERLAY_MASK_Y

  Y coordinate of pixel to use for transparency.

- #define FE2_OVERLAY_MASK_RGB

  Use the RGB value (FE2_OVERLAY_MASK_R, FE2_OVERLAY_MASK_G, FE2_OVERLAY_MASK_B) as the transparency color.

- #define FE2_OVERLAY_MASK_R

  Red component of the transparency color.

- #define FE2_OVERLAY_MASK_G

  Green component of the transparency color.

- #define FE2_OVERLAY_MASK_B

  Blue component of the transparency color.

- #define FE2_OVERLAY_POS

  Set the overlay position. Valid modes are defined by FE2_OverlayPositionMode.

- #define FE2_OVERLAY_POS_X

  X coordinate of overlay top left on video.

- #define FE2_OVERLAY_POS_Y

  Y coordinate of overlay top left on video.

Enumerations

- enum FE2_OverlayPositionMode {
  FE2_OVERLAY_POS_MODE_TOPLEFT,
  FE2_OVERLAY_POS_MODE_BOTLEFT,
  FE2_OVERLAY_POS_MODE_CENTER,
  FE2_OVERLAY_POS_MODE_TOPRIGHT,
  FE2_OVERLAY_POS_MODE_BOTRIGHT,
  FE2_OVERLAY_POS_MODE_XY }

  Position modes for use with FE2_OVERLAY_POS.

21.28 flixengine2/filters/png_export.h File Reference

21.28.1 Detailed Description

Reference module(s): PNG Image Export (Thumbnail).

Definition in file png_export.h.

Defines

- #define FE2_FILTER_PNGEX

  Filter name for use with Flix2_AddFilter().

- #define FE2_PNGEX_OUTPUT_DIRECTORY

  Filter parameter for setting PNG image output directory.

- #define FE2_PNGEX_FILENAME_PREFIX

  Filter parameter for setting PNG image file name prefix.

- #define FE2_PNGEX_FILENAME_SUFFIX

  Filter parameter for setting PNG image file name suffix.

- #define FE2_PNGEX_ENABLE_ALPHA

  Filter parameter for preserving source video alpha channel data in exported PNG images.

- #define FE2_PNGEX_EXPORT_INTERVAL

  Filter parameter for enabling PNG image export at a millisecond interval.

- #define FE2_PNGEX_EXPORT_TIME_STRING

  Filter parameter for setting a string of PNG export times in milliseconds.

- #define FE2_PNGEX_EXPORT_CUE_POINTS

  Filter parameter for enabling export of PNG images at cue points.

- #define FE2_PNGEX_AUTO_EXPORT_COUNT

  Filter parameter for enabling automatic PNG image generation.

- #define FE2_PNGEX_AUTO_EXPORT_START_TIME

  Filter parameter for setting PNG auto generation start time.

- #define FE2_PNGEX_AUTO_EXPORT_END_TIME

  Filter parameter for setting PNG auto generation end time.

- #define FE2_PNGEX_AUTO_EXPORT_RANDOM_PERIOD

  Filter parameter for randomizing auto generated times.

- #define FE2_PNGEX_EXPORT_FIRST_FRAME_PNG

  Filter parameter for enabling PNG image creation using the first video frame.

- #define FE2_PNGEX_COMPRESSION_LEVEL

Filter parameter for setting compression level used by libpng.

- #define FE2_PNGEX_WIDTH

  Filter parameter for setting PNG image width.

- #define FE2_PNGEX_HEIGHT

  Filter parameter for setting PNG image height.

Enumerations

- enum FE2_PNGExCuePtMode {
  FE2_PNGEX_CP_EVENT,
  FE2_PNGEX_CP_NAV,
  FE2_PNGEX_CP_ALL }

  Constants for use with FE2_PNGEX_EXPORT_CUE_POINTS.

21.29 flixengine2/filters/resample.h File Reference

21.29.1 Detailed Description

Reference module(s): Resample.

Definition in file resample.h.

Defines

- #define FE2_FILTER_RESAMPLE

  Filter name for use with Flix2_AddFilter().

- #define FE2_RESAMPLE_RATE

  Desired sample rate in Hertz (Hz).

- #define FE2_RESAMPLE_CHANNELS

  Filter parameter for channels.

21.30 flixengine2/filters/rotate.h File Reference

21.30.1 Detailed Description

Reference module(s): Rotate.

Definition in file rotate.h.

Defines

- #define FE2_FILTER_ROTATE

  Filter name for use with Flix2_AddFilter().

- #define FE2_ROTATE_ANGLE

  Specifies the image rotation angle in degrees.

21.31 flixengine2/filters/scale.h File Reference

21.31.1 Detailed Description

Reference module(s): Scale.

Definition in file scale.h.

Defines

- #define FE2_FILTER_SCALE

  Filter name for use with Flix2_AddFilter().

- #define FE2_SCALE_WIDTH

  Filter parameter name for scaled width.

- #define FE2_SCALE_HEIGHT

  Filter parameter name for scaled height.

21.32 flixengine2/filters/sharpen.h File Reference

21.32.1 Detailed Description

Reference module(s): Sharpen.

Definition in file sharpen.h.

Defines

- #define FE2_FILTER_SHARPEN

  Filter name for use with Flix2_AddFilter().

21.33 flixengine2/flixengine2.h File Reference

21.33.1 Detailed Description

Reference module(s): Main Engine Interface, Flix Engine Core Library, Filter Manipulation, Codec Manipulation, Muxer Manipulation.

Definition in file flixengine2.h.

```
#include "on2types.h"
#include "filter_constants.h"
#include "codec_constants.h"
#include "muxer_constants.h"
#include "video_options.h"
#include "audio_options.h"
#include "swf_options.h"
#include "encoding_status.h"
#include "media_editor_options.h"
#include "overlay_options.h"
```

Defines

- #define FLIXENGINE_API
- #define FLIXENGINE_VERSION_CHIEF
- #define FLIXENGINE_VERSION_MAJOR
- #define FLIXENGINE_VERSION_MINOR
- #define FLIXENGINE_VERSION_PATCH
- #define FLIXENGINE_VERSION_EXTRA
- #define FLIXENGINE_VERSION_STR

Typedefs

- typedef void * FLIX2HANDLE
- typedef void * FLIX2PLGNHANDLE

Enumerations

- enum FE2_ExportedVideoType {
  ExportSWF3To6Video,
  ExportSWF7PlusVideo,
  ExportSWFVectorVideo,
  ExportFLVVideo }

  Output file types for use in calls to Flix2_SetExportVideoType() and Flix2_GetExportVideoType().

- enum FE2_EncState {
  EncStateIdle,
  EncStateRunning,
  EncStateQueued }
  Encoder state returned by Flix2_GetEncoderState().

- enum FE2_errno {
  ErrNone,
  ErrSys,
  ErrFileIO,
  ErrFileOpen,
  ErrFileDecode,
  ErrFileDecodeA,
  ErrFileDecodeV,
  ErrEncodeA,
  ErrEncodeV }
  Flix engine error state returned by Flix2_Errno().

Functions

- on2sc Flix2_Init ()
  Perform library initialization.

- on2sc Flix2_Deinit ()
  Cleanup resources allocated by Flix2_Init().

- on2sc Flix2_Create (FLIX2HANDLE *pFlix)
  Create a handle to the flix engine.

- on2sc Flix2_CreateEx (FLIX2HANDLE *pFlix, const char *rpchost, int32_t timeout)
  Create a handle to the flix engine.

- on2sc Flix2_Destroy (FLIX2HANDLE flix)
  Frees resources associated with a `FLIX2HANDLE`.

- const char * Flix2_Version ()
  Returns the library version as a string.

- const on2tc * Flix2_Copyright ()
  Returns copyright information for this library as a string.

- on2sc Flix2_SetOutputFile (FLIX2HANDLE flix, const on2tc *outputFile)
  Set the destination file for the encode session.

- on2sc Flix2_GetOutputFile (const FLIX2HANDLE flix, on2tc *pOutputFile, int32_t *len)
  Get the destination file for the encode session.

- on2sc Flix2_SetOverwriteExistingFiles (FLIX2HANDLE flix, const on2bool bOverwriteExistingFiles)

  Enable/disable overwriting of existing output files.

- on2sc Flix2_GetOverwriteExistingFiles (const FLIX2HANDLE flix, on2bool *bpOverwriteExistingFiles)

  Retrieve the engine's current behavior regarding existing output files.

- on2sc Flix2_SetExportAudio (FLIX2HANDLE flix, const on2bool bExportAudio)

  Enable/disable audio output in the current encoding session.

- on2sc Flix2_GetExportAudio (const FLIX2HANDLE flix, on2bool *bpExportAudio)

  Retrieve the engine's current behavior regarding audio output.

- on2sc Flix2_SetExportVideo (FLIX2HANDLE flix, const on2bool bExportVideo)

  Enable/disable video output in the current encoding session.

- on2sc Flix2_GetExportVideo (const FLIX2HANDLE flix, on2bool *bpExportVideo)

  Retrieve the engine's current behavior regarding video output.

- on2sc Flix2_SetExportVideoType (FLIX2HANDLE flix, const FE2_ExportedVideoType exportVideoType)

  Set the output file type.

- on2sc Flix2_GetExportVideoType (const FLIX2HANDLE flix, FE2_ExportedVideoType *pExportVideoType)

  Retrieve the current output file type.

- on2sc Flix2_SetInputFile (FLIX2HANDLE flix, const on2tc *inputFile)

  Set the source file for the encode session.

- on2sc Flix2_GetInputFile (const FLIX2HANDLE flix, on2tc *pInputFile, int32_t *len)

  Get the source file for the encode session.

- on2sc Flix2_GetSourceDuration (const FLIX2HANDLE flix, int32_t *duration)

  Get the duration, in milliseconds, of the source file.

- on2sc Flix2_Encode (FLIX2HANDLE flix)

  Start encoding an output file.

- on2sc Flix2_StopEncoding (FLIX2HANDLE flix)

  Cancel a running encoding session.

- on2sc Flix2_Reset (FLIX2HANDLE flix)

  Reset the engine to its defaults.

- on2sc Flix2_Validate (const FLIX2HANDLE flix)

  Validate the current encoder settings.

- on2sc Flix2_IsEncoderRunning (const FLIX2HANDLE flix, on2bool *bpIsEncoderRunning)

  Check the status of an encode.

- on2sc Flix2_GetEncoderState (const FLIX2HANDLE flix, FE2_EncState *pEncState)

  Retrieve the current state of the encoder.

- on2sc Flix2_SetLogLevel (FLIX2HANDLE flix, int32_t level)

  Set the library-wide debug log level.

- on2sc Flix2_GetLogLevel (const FLIX2HANDLE flix, int32_t *level)

  Retrieve current library-wide debug log level.

- on2sc Flix2_SetLogPath (FLIX2HANDLE flix, on2tc *logpath)

  Set the library's log file path.

- on2sc Flix2_GetLogPath (const FLIX2HANDLE flix, on2tc *logpath, int32_t *len)

  Retrieve the library's current log file path.

- on2sc Flix2_AddFilter (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)

  Add a filter to the encoder's filter chain.

- on2sc Flix2_RemoveFilter (FLIX2PLGNHANDLE plgn)

  Remove a filter to the encoder's filter chain.

- on2sc Flix2_FilterSetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)

  Set the value of a parameter in a filter instance using a string representation.

- on2sc Flix2_FilterGetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)

  Retrieve the value of a parameter in a filter instance represented as a string.

- on2sc Flix2_FilterSetParam (FLIX2PLGNHANDLE plgn, const char *name, double inDblVal)

  Set the value of a parameter in a filter instance using a double representation.

- on2sc Flix2_FilterGetParam (FLIX2PLGNHANDLE plgn, const char *name, double *outDblVal)

  Retrieve the value of a parameter in a filter instance represented as a double.

- on2sc Flix2_AddCodec (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)

  Add a codec to the encoder's codec chain.

- on2sc Flix2_RemoveCodec (FLIX2PLGNHANDLE plgn)

  Remove a codec from the encoder's codec chain.

- on2sc Flix2_CodecSetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)

  Set the value of a parameter in a codec instance using a string representation.

- on2sc Flix2_CodecGetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)

  Retrieve the value of a parameter in a codec instance represented as a string.

- on2sc Flix2_CodecSetParam (FLIX2PLGNHANDLE plgn, const char *name, double inDblVal)

  Set the value of a parameter in a codec instance using a double representation.

- on2sc Flix2_CodecGetParam (FLIX2PLGNHANDLE plgn, const char *name, double *outDblVal)

  Retrieve the value of a parameter in a codec instance represented as a double.

- on2sc Flix2_AddMuxer (FLIX2PLGNHANDLE *pPlgn, const FLIX2HANDLE flix, const char *plgn_name)

  Set the desired muxer.

- on2sc Flix2_RemoveMuxer (FLIX2PLGNHANDLE plgn)

  Remove the specified muxer.

- on2sc Flix2_MuxerSetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, const on2tc *value)

  Set the value of a parameter in a muxer instance using a string representation.

- on2sc Flix2_MuxerGetParamAsStr (FLIX2PLGNHANDLE plgn, const char *name, on2tc *value, int32_t *len)

  Retrieve the value of a parameter in a muxer instance represented as a string.

- on2sc Flix2_MuxerSetParam (FLIX2PLGNHANDLE plgn, const char *name, double inDblVal)

  Set the value of a parameter in a muxer instance using a double representation.

- on2sc Flix2_MuxerGetParam (FLIX2PLGNHANDLE plgn, const char *name, double *outDblVal)

  Retrieve the value of a parameter in a muxer instance represented as a double.

- on2sc Flix2_Errno (const FLIX2HANDLE flix, FE2_errno *flixerrno, int32_t *syserrno)

  Retrieve the current error state of the engine.

21.33.2 Define Documentation

21.33.2.1 #define FLIXENGINE_API

Definition at line 24 of file flixengine2.h.

21.33.2.2 #define FLIXENGINE_VERSION_CHIEF

Definition at line 33 of file flixengine2.h.

21.33.2.3 #define FLIXENGINE_VERSION_EXTRA

Definition at line 37 of file flixengine2.h.

21.33.2.4 #define FLIXENGINE_VERSION_MAJOR

Definition at line 34 of file flixengine2.h.

21.33.2.5 #define FLIXENGINE_VERSION_MINOR

Definition at line 35 of file flixengine2.h.

21.33.2.6 #define FLIXENGINE_VERSION_PATCH

Definition at line 36 of file flixengine2.h.

21.33.2.7 #define FLIXENGINE_VERSION_STR

Definition at line 43 of file flixengine2.h.

21.33.3 Typedef Documentation

21.33.3.1 typedef void∗ FLIX2HANDLE

handle type passed to all Flix Engine API functions

Definition at line 50 of file flixengine2.h.

21.33.3.2 typedef void ∗ FLIX2PLGNHANDLE

filter plgn handle passed to Filter Manipulation functions

Definition at line 50 of file flixengine2.h.

21.34 flixengine2/fs_options.h File Reference

21.34.1 Detailed Description

Reference module(s): Frame Server API.

Definition in file fs_options.h.

Enumerations

- enum FE2_FrameType {
  FrameTypeAudio,
  FrameTypeVideo }

  Frame type being passed to Flix2_EncodeFrame().

- enum FE2_VideoFormat {
  VideoFormatYV12,
  VideoFormatBGRA }

  Format of input video used in calls to Flix2_SetVideoProperties().

Functions

- on2sc Flix2_SetFrameServer (FLIX2HANDLE flix, on2bool enable)

  Enable/disable the frame server for this session.

- on2sc Flix2_GetFrameServer (FLIX2HANDLE flix, on2bool *enabled)

  Retrieve the current frame server status.

- on2sc Flix2_SetAudioProperties (FLIX2HANDLE flix, uint8_t channels, int16_t bitspersample, int32_t samplerate)

  Set the input audio properties for the frame server.

- on2sc Flix2_SetVideoProperties (FLIX2HANDLE flix, FE2_VideoFormat format, int32_t width, int32_t height, int32_t pitch, int32_t rate, int32_t scale)

  Set the input video properties for the frame server.

- on2sc Flix2_EncodeFrame (FLIX2HANDLE flix, FE2_FrameType type, on2bool eos, uint8_t *frame, uint32_t len, int64_t ts90k)

  Queue an input frame for encoding.

21.35 flixengine2/media_editor_options.h File Reference

21.35.1 Detailed Description

Reference module(s): Filter Manipulation, Video Filters.

Definition in file media_editor_options.h.

Functions

- on2sc editor_options_Reset (FLIX2HANDLE flix)
  Reset the media editor options to their defaults.
- on2sc editor_options_Validate (const FLIX2HANDLE flix)
  Ensure the current media editor settings are valid.

Deprecated functions

- on2sc editor_options_GetCrop (const FLIX2HANDLE flix, on2bool *pCrop)
  Determine if the crop filter is enabled.
- on2sc editor_options_SetCrop (FLIX2HANDLE flix, const on2bool crop)
  Enable/disable the crop filter.
- on2sc editor_options_GetCropBounds (const FLIX2HANDLE flix, int32_t *pTop, int32_t *pLeft, int32_t *pBottom, int32_t *pRight)
  Get the current bounding box used for cropping.
- on2sc editor_options_SetCropBounds (FLIX2HANDLE flix, const int32_t top, const int32_t left, const int32_t bottom, const int32_t right)
  Set the current bounding box used for cropping.

Deprecated functions

- on2sc editor_options_GetBrightness (const FLIX2HANDLE flix, int32_t *pBrightness)
  Get the current brightness adjustment factor.
- on2sc editor_options_SetBrightness (FLIX2HANDLE flix, const int32_t brightness)
  Set the brightness adjustment factor.
- on2sc editor_options_GetUseBrightness (const FLIX2HANDLE flix, on2bool *pUseBrightness)
  Determine if the brightness filter is enabled.
- on2sc editor_options_SetUseBrightness (FLIX2HANDLE flix, const on2bool bUseBrightness)
  Enable/disable the brightness filter.
- on2sc editor_options_GetContrast (const FLIX2HANDLE flix, double *pContrast)
  Get the current contrast adjustment factor.
- on2sc editor_options_SetContrast (FLIX2HANDLE flix, const double contrast)

Set the contrast adjustment factor.

- on2sc editor_options_GetUseContrast (const FLIX2HANDLE flix, on2bool *pUseContrast)
  Determine if the contrast filter is enabled.
- on2sc editor_options_SetUseContrast (FLIX2HANDLE flix, const on2bool bUseContrast)
  Enable/disable the contrast filter.
- on2sc editor_options_GetHue (const FLIX2HANDLE flix, int32_t *pHue)
  Get the current hue adjustment factor.
- on2sc editor_options_SetHue (FLIX2HANDLE flix, const int32_t hue)
  Set the hue adjustment factor.
- on2sc editor_options_GetUseHue (const FLIX2HANDLE flix, on2bool *pUseHue)
  Determine if the hue filter is enabled.
- on2sc editor_options_SetUseHue (FLIX2HANDLE flix, const on2bool bUseHue)
  Enable/disable the hue filter.
- on2sc editor_options_GetSaturation (const FLIX2HANDLE flix, double *pSaturation)
  Get the current saturation adjustment factor.
- on2sc editor_options_SetSaturation (FLIX2HANDLE flix, const double saturation)
  Set the saturation adjustment factor.
- on2sc editor_options_GetUseSaturation (const FLIX2HANDLE flix, on2bool *pUseSaturation)
  Determine if the saturation filter is enabled.
- on2sc editor_options_SetUseSaturation (FLIX2HANDLE flix, const on2bool bUseSaturation)
  Enable/disable the saturation filter.

Deprecated functions

- on2sc editor_options_GetUseCut (const FLIX2HANDLE flix, on2bool *pUseCut)
  Determine if the cut filter is enabled.
- on2sc editor_options_SetUseCut (FLIX2HANDLE flix, const on2bool bUseCut)
  Enable/disable the cut filter.
- on2sc editor_options_GetCutStartTime (const FLIX2HANDLE flix, double *pStartTime)
  Get the current cut start time.
- on2sc editor_options_SetCutStartTime (FLIX2HANDLE flix, const double start_time)
  Set the cut start time.
- on2sc editor_options_GetCutStopTime (const FLIX2HANDLE flix, double *pEndTime)
  Get the current cut stop time.
- on2sc editor_options_SetCutStopTime (FLIX2HANDLE flix, const double end_time)
  Set the cut stop time.

21.36 flixengine2/muxer_constants.h File Reference

21.36.1 Detailed Description

Reference module(s): Muxers.

Definition in file muxer_constants.h.

```
#include "muxers/flv.h"
#include "muxers/fxm.h"
#include "muxers/isomedia.h"
#include "muxers/swf.h"
#include "muxers/webm.h"
```

21.37 flixengine2/muxers/flv.h File Reference

21.37.1 Detailed Description

Reference module(s): FLV Muxer.

Definition in file flv.h.

Defines

- #define FE2_MUXER_FLV

  FLV muxer. For use with Flix2_AddMuxer().

- #define FE2_FLV_CUEPT_EVENT

  Set an event cue point.

- #define FE2_FLV_CUEPT_NAV

  Set a navigation cue point.

- #define FE2_FLV_CUEPT_PARAM

  Add a name/value pair to an existing cue point.

- #define FE2_FLV_METADATA_ENABLE

  Enable output of meta data element.

- #define FE2_FLV_METADATA_DISABLE

  Disable output of meta data element.

Typedefs

- typedef enum flv_metadata flvmetadata_t

Enumerations

- enum flv_metadata {
  MD_DURATION,
  MD_DATASIZE,
  MD_AUDIO_SIZE,
  MD_VIDEO_SIZE,
  MD_AUDIO_DATARATE,
  MD_VIDEO_DATARATE,
  MD_AUDIO_CODECID,
  MD_VIDEO_CODECID,
  MD_WIDTH,
  MD_HEIGHT,
  MD_FRAMERATE,

MD_CANSEEKTOEND,
MD_LASTTIMESTAMP,
MD_LASTKEYFRAMETIMESTAMP,
MD_LASTKEYFRAMELOCATION,
MD_KEYFRAMES }

Supported FLV onMetaData elements.

21.38 flixengine2/muxers/fxm.h File Reference

21.38.1 Detailed Description

Reference module(s): FXM Muxer.

Definition in file fxm.h.

Defines

- #define FE2_MUXER_FXM

  FXM muxer. For use with Flix2_AddMuxer().

- #define FE2_FXM_CUEPT_EVENT

  Set an event cue point.

- #define FE2_FXM_CUEPT_NAV

  Set a navigation cue point.

- #define FE2_FXM_CUEPT_PARAM

  Add a name/value pair to an existing cue point.

- #define FE2_FXM_METADATA_ENABLE

  Enable output of meta data element.

- #define FE2_FXM_METADATA_DISABLE

  Disable output of meta data element.

Typedefs

- typedef enum flv_metadata fxmmetadata_t

21.39 flixengine2/muxers/isomedia.h File Reference

21.39.1 Detailed Description

Reference module(s): 3GPP Muxer, 3GPP2 Muxer, MOV Muxer, MP4 Muxer.

Definition in file isomedia.h.

Defines

- #define FE2_ISOMEDIA_FASTSTART

  Influence 'moov' box placement.

- #define FE2_MUXER_3GP

  3GPP muxer. For use with Flix2_AddMuxer()

- #define FE2_3GP_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

- #define FE2_MUXER_3G2

  3GPP2 muxer. For use with Flix2_AddMuxer()

- #define FE2_3G2_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

- #define FE2_MUXER_MOV

  MOV muxer. For use with Flix2_AddMuxer().

- #define FE2_MOV_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

- #define FE2_MUXER_MP4

  MP4 muxer. For use with Flix2_AddMuxer().

- #define FE2_MP4_FASTSTART

  FE2_ISOMEDIA_FASTSTART alias

21.39.2 Define Documentation

21.39.2.1 #define FE2_ISOMEDIA_FASTSTART

Influence 'moov' box placement.

When enabled places the 'moov' box near the beginning of the file allowing for progressive download.

Note:

Default: 0 (disabled)

Attention:

The current implementation is a post-process. A temporary file is created within the output directory as the necessary boxes are rewritten, with the result then replacing the original output file.

Definition at line 34 of file isomedia.h.

21.40 flixengine2/muxers/swf.h File Reference

21.40.1 Detailed Description

Reference module(s): SWF Muxer.

Definition in file swf.h.

Defines

- #define FE2_MUXER_SWF

  SWF muxer. For use with Flix2_AddMuxer().

- #define FE2_SWF_HEIGHT

  Set the SWF height.

- #define FE2_SWF_WIDTH

  Set the SWF width.

- #define FE2_SWF_FRAMERATE

  Set the SWF framerate.

- #define FE2_SWF_EMBEDDED_URL

  Set the SWF's target URL.

- #define FE2_SWF_EMBEDDED_URL_TARGET

  Set the target of FE2_SWF_EMBEDDED_URL.

- #define FE2_SWF_EMBEDDED_URL_TYPE

  Set how FE2_SWF_EMBEDDED_URL is interpreted.

- #define FE2_SWF_LOOP_COUNT

  Sets the number of times the SWF should loop.

- #define FE2_SWF_PRELOAD_TYPE

  Sets the type of preloader.

- #define FE2_SWF_FIXED_PRELOAD_PCT

  Sets the percent of the SWF movie to preload before playback begins.

- #define FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_FACTOR

  Sets the adaptive preload buffer factor.

- #define FE2_SWF_ON_END_OPTION

  Sets the options for the end of the SWF.

- #define FE2_SWF_ON_END_URL

  Sets the URL a SWF movie will load after the current movie ends.

- #define FE2_SWF_ON_START_OPTION

Sets the options for the start of the SWF.

- #define FE2_SWF_START_BLANK_FRAME

  Control the insertion of a blank first frame in the SWF.

- #define FE2_SWF_START_WAIT_SEC

  Sets the number of seconds to wait before playback begins.

- #define FE2_SWF_ADD_VARIABLE

  Add or update a variable in the SWF.

- #define FE2_SWF_DELETE_VARIABLE

  Delete an existing variable in the SWF.

Enumerations

- enum FE2_EmbeddedUrlType {
  EmbeddedUrlIsGetUrl,
  EmbeddedUrlIsLoadMovie }

  Differentiates between the type of file (HTML or SWF) set through FE2_SWF_EMBEDDED_URL.

- enum FE2_SwfOnEndOptions {
  SwfOnMovieEndNothing,
  SwfOnMovieEndSTOP,
  SwfOnMovieEndLoop,
  SwfOnMovieEndUnload,
  SwfOnMovieEndLoadMovie }

  Actions that can be added to the last frame of a SWF file.

- enum FE2_SwfOnStartOptions {
  SwfOnMovieStartAutomatically,
  SwfOnMovieStartOnClick,
  SwfOnMovieStartWait,
  SwfOnMovieStartEmbedSTOP }

  Actions that can be added to the start frame of the SWF file.

- enum FE2_SwfPreloaderOptions {
  SwfPreloaderNone,
  SwfFixedPreloader,
  SwfAdaptivePreloader }

  Determines the type of preloader added to the SWF file.

21.41 flixengine2/muxers/webm.h File Reference

21.41.1 Detailed Description

Reference module(s): WebM Muxer.

Definition in file webm.h.

Defines

- #define FE2_MUXER_WEBM

  WebM muxer. For use with Flix2_AddMuxer().

21.42 flixengine2/on2types.h File Reference

21.42.1 Detailed Description

Reference module(s): Base Types.

Definition in file on2types.h.

```
#include <sys/types.h>
```

Defines

- #define OTC(str)

  a macro suitable for declaring a constant on2tc

- #define ON2TC

  printf format string suitable for printing an on2tc

- #define PRId64
- #define ON264

  printf format string suitable for printing an on2s64

- #define ON2API

  library calling convention/storage class attributes.

Typedefs

- typedef char int8_t
- typedef short int16_t
- typedef int int32_t
- typedef unsigned char uint8_t
- typedef unsigned short uint16_t
- typedef unsigned int uint32_t
- typedef int8_t on2s8
- typedef uint8_t on2u8
- typedef int16_t on2s16
- typedef uint16_t on2u16
- typedef int32_t on2s32
- typedef uint32_t on2u32
- typedef int32_t on2bool
- typedef char on2tc
- typedef long long on2s64
- typedef unsigned long long on2u64
- typedef on2s64 int64_t
- typedef on2u64 uint64_t

Enumerations

- enum _on2bool {
  on2false,
  on2true }
- enum on2sc {
  ON2_NOT_FOUND,
  ON2_BUFFER_EMPTY,
  ON2_BUFFER_FULL,
  ON2_CONNREFUSED,
  ON2_TIMEDOUT,
  ON2_WOULDBLOCK,
  ON2_NET_ERROR,
  ON2_INVALID_VERSION,
  ON2_INPROGRESS,
  ON2_NOT_SUPP,
  ON2_NO_MEM,
  ON2_INVALID_PARAMS,
  ON2_ERROR,
  ON2_OK,
  ON2_DONE }

  Common return type.

21.42.2 Define Documentation

21.42.2.1 #define ON2API

library calling convention/storage class attributes.

Specifies whether the function is imported through a dll or is from a static library.

Definition at line 192 of file on2types.h.

21.43 flixengine2/overlay_options.h File Reference

21.43.1 Detailed Description

Reference module(s): Overlay (Watermark).

Definition in file overlay_options.h.

Functions

Deprecated functions

- on2sc overlay_options_Reset (const FLIX2HANDLE flix)
  Resets the overlay options.
- on2sc overlay_options_GetUseOverlay (const FLIX2HANDLE flix, on2bool *pUseOverlay)
  Determines if an overlay is to be used.
- on2sc overlay_options_SetUseOverlay (FLIX2HANDLE flix, const on2bool bUseOverlay)
  Enables or disables overlay usage.
- on2sc overlay_options_GetOverlayPath (const FLIX2HANDLE flix, on2tc *pOverlayFilePath, int32_t *pLen)
  Returns the path to the overlay image file.
- on2sc overlay_options_SetOverlayPath (FLIX2HANDLE flix, const on2tc *pOverlayFilePath)
  Set the path to the overlay image file.
- on2sc overlay_options_GetMaskPixelXY (const FLIX2HANDLE flix, int32_t *pMaskPixelX, int32_t *pMaskPixelY)
  Return the X and Y coordinates of the mask pixel.
- on2sc overlay_options_SetMaskPixelXY (FLIX2HANDLE flix, int32_t maskPixelX, int32_t maskPixelY)
  Set the X and Y coordinates of the mask pixel.
- on2sc overlay_options_GetMaskPixelRGB (FLIX2HANDLE flix, uint8_t *pMaskPixelR, uint8_t *pMaskPixelG, uint8_t *pMaskPixelB)
  Return the Red, Green, and Blue component values of the mask pixel.
- on2sc overlay_options_SetMaskPixelRGB (FLIX2HANDLE flix, uint8_t maskPixelR, uint8_t maskPixelG, uint8_t maskPixelB)
  Set the Red, Green, and Blue component values of the mask pixel.
- on2sc overlay_options_GetOverlayPosition (FLIX2HANDLE flix, FE2_OverlayPositionMode *pMode, uint32_t *pX, uint32_t *pY)
  Return the overlay position.
- on2sc overlay_options_SetOverlayPosition (FLIX2HANDLE flix, FE2_OverlayPositionMode mode, uint32_t x, uint32_t y)
  Set the overlay position.

21.44 flixengine2/swf_options.h File Reference

21.44.1 Detailed Description

Reference module(s): SWF.

Definition in file swf_options.h.

Functions

Deprecated functions

- on2sc swf_options_Reset (FLIX2HANDLE flix)
  Resets the swf options.
- on2sc swf_options_GetEmbeddedUrl (FLIX2HANDLE flix, char *embeddedUrl, int32_t *len)
  Gets the embedded URL.
- on2sc swf_options_GetEmbeddedUrlTarget (FLIX2HANDLE flix, char *embeddedUrlTarget, int32_t *len)
  Gets the target of embedded URL.
- on2sc swf_options_SetEmbeddedUrl (FLIX2HANDLE flix, const char *embeddedUrl)
  Sets the embedded URL of the SWF or HTML that will be loaded when the user clicks on the video.
- on2sc swf_options_SetEmbeddedUrlTarget (FLIX2HANDLE flix, const char *embeddedUrlTarget)
  Sets the target of the embedded URL.
- on2sc swf_options_GetEmbeddedUrlType (FLIX2HANDLE flix, FE2_EmbeddedUrlType *embeddedUrlType)
  Gets the type of embedded URL.
- on2sc swf_options_SetEmbeddedUrlType (FLIX2HANDLE flix, const FE2_EmbeddedUrlType embeddedUrlType)
  Sets the type of embedded URL.
- on2sc swf_options_GetSwfFramerate (FLIX2HANDLE flix, int32_t *pSwfFramerate)
  Gets the SWF framerate.
- on2sc swf_options_SetSwfFramerate (FLIX2HANDLE flix, const int32_t swfFramerate)
  Sets the SWF framerate.
- on2sc swf_options_GetInsertBlankFrameOnStart (FLIX2HANDLE flix, on2bool *pInsertBlankFrameOnStart)
  Gets if the engine is to insert a blank frame as the first frame of the SWF or not.
- on2sc swf_options_SetInsertBlankFrameOnStart (FLIX2HANDLE flix, const on2bool insertBlankFrameOnStart)
  Tells the engine to insert a blank frame as the first frame of the SWF or not.
- on2sc swf_options_GetSwfFramerateAsDouble (FLIX2HANDLE flix, double *pSwfFramerate)
  Gets the SWF framerate as a double.

- on2sc swf_options_SetSwfFramerateAsDouble (FLIX2HANDLE flix, const double swfFramerate)
  Sets the SWF framerate as a double.
- on2sc swf_options_GetEnablePreloader (FLIX2HANDLE flix, on2bool *pEnablePreloader)
  Gets if a video preloader is enabled or disabled.
- on2sc swf_options_SetEnablePreloader (FLIX2HANDLE flix, const on2bool enablePreloader)
  Enables or disables the video preloader.
- on2sc swf_options_GetPercentToPreload (FLIX2HANDLE flix, int32_t *pPercentToPreload)
  Gets the percent of the SWF movie to preload before playback begins.
- on2sc swf_options_SetPercentToPreload (FLIX2HANDLE flix, const int32_t percentToPreload)
  Sets the percent of the SWF movie to preload before playback begins.
- on2sc swf_options_GetPreloaderType (FLIX2HANDLE flix, FE2_SwfPreloaderOptions *pPreloaderType)
  Gets the type of preloader.
- on2sc swf_options_SetPreloaderType (FLIX2HANDLE flix, const FE2_SwfPreloaderOptions preloaderType)
  Sets the type of preloader.
- on2sc swf_options_GetAdaptivePreloaderBufferFactor (FLIX2HANDLE flix, double *pPreloaderBufferFactor)
  Gets the adaptive preload buffer factor.
- on2sc swf_options_SetAdaptivePreloaderBufferFactor (FLIX2HANDLE flix, const double preloaderBufferFactor)
  Sets the adaptive preload buffer factor.
- on2sc swf_options_GetMovieOnEndOptions (FLIX2HANDLE flix, FE2_SwfOnEndOptions *pOnEndOptions)
  Gets the options for the end of the SWF.
- on2sc swf_options_SetMovieOnEndOptions (FLIX2HANDLE flix, const FE2_SwfOnEndOptions onEndOptions)
  Sets the options for the end of the SWF.
- on2sc swf_options_GetLoopCount (FLIX2HANDLE flix, int32_t *pLoopCount)
  Gets the number of times the SWF should loop.
- on2sc swf_options_SetLoopCount (FLIX2HANDLE flix, const int32_t loopCount)
  Sets the number of times the SWF should loop.
- on2sc swf_options_GetLoadMovieOnEndUrl (FLIX2HANDLE flix, char *pLoadMovieOnEndUrl, int32_t *len)
  Gets the URL of a SWF movie to load after the current movie ends.
- on2sc swf_options_SetLoadMovieOnEndUrl (FLIX2HANDLE flix, const char *loadMovieOnEndUrl)
  Sets the URL of a SWF movie to load after the current movie ends.

- on2sc swf_options_GetMovieOnStartOptions (FLIX2HANDLE flix, FE2_SwfOnStartOptions *pOnStartOptions)
  Gets the options for the start of the SWF.
- on2sc swf_options_SetMovieOnStartOptions (FLIX2HANDLE flix, const FE2_-SwfOnStartOptions onStartOptions)
  Sets the options for the start of the SWF.
- on2sc swf_options_GetWaitTimeToStart (FLIX2HANDLE flix, int32_t *pWaitTimeToStart)
  Gets the number of seconds to wait before playback begins.
- on2sc swf_options_SetWaitTimeToStart (FLIX2HANDLE flix, const int32_t waitTimeToStart)
  Sets the number of seconds to wait before playback begins.
- on2sc swf_options_AddVariable (FLIX2HANDLE flix, const char *name, const char *value)
  Adds a custom SWF variable as a name/value pair.
- on2sc swf_options_DeleteVariable (FLIX2HANDLE flix, const int32_t index)
  Deletes a custom SWF variable.
- on2sc swf_options_GetVariableCount (FLIX2HANDLE flix, int32_t *pVariableCount)
  Gets the number of custom SWF variables already added.
- on2sc swf_options_UpdateVariable (FLIX2HANDLE flix, const int32_t index, const char *name, const char *value)
  Updates an already existing SWF variable.

21.45 flixengine2/video_options.h File Reference

21.45.1 Detailed Description

Reference module(s): Video Encoding Options, Frame Rate, Scale.

Definition in file video_options.h.

Enumerations

- enum FE2_VideoCodec {
  CODEC_NULL,
  CODEC_H263,
  CODEC_SCREENVIDEO,
  CODEC_VP6,
  CODEC_VP6ALPHA }

  Output video codec types, influences quality/compatibility.

- enum FE2_CuePointType {
  CUE_EVENT,
  CUE_NAVIGATION }

  Cue point type for use with video_options_AddFLVCuePoint().

Functions

- on2sc video_options_Reset (FLIX2HANDLE flix)

  Reset the video options to their defaults.

- on2sc video_options_Validate (const FLIX2HANDLE flix)

  Ensure the current video settings are valid.

- on2sc video_options_GetImageQuality (const FLIX2HANDLE flix, int32_t *lpImageQuality)

  Get the current image quality factor.

- on2sc video_options_SetImageQuality (FLIX2HANDLE flix, const int32_t lImageQuality)

  Set the image quality factor.

- on2sc video_options_GetKeyframeInterval (const FLIX2HANDLE flix, int32_t *lpKeyframeInterval)

  Get the current interval between keyfames.

- on2sc video_options_SetKeyframeInterval (FLIX2HANDLE flix, const int32_t lKeyframeInterval)

  Set the interval between keyfames.

- on2sc video_options_GetKeyframeIntervalType (const FLIX2HANDLE flix, FE2_VideoKeyframeTypes *pKeyframeIntervalType)

Get the current keyframe interval type.

- on2sc video_options_SetKeyframeIntervalType (FLIX2HANDLE flix, const FE2_VideoKeyframeTypes keyframeIntervalType)

  Set the keyframe interval type.

- on2sc video_options_GetMaximumBitrate (const FLIX2HANDLE flix, int32_t *lpMaximumBitrate)

  Get the current maximum bitrate target.

- on2sc video_options_SetMaximumBitrate (FLIX2HANDLE flix, const int32_t lMaximumBitrate)

  Set the maximum bitrate target.

- on2sc video_options_GetRateControlType (const FLIX2HANDLE flix, FE2_VideoBitrateControls *pRateControlType)

  Get the current rate control type.

- on2sc video_options_SetRateControlType (FLIX2HANDLE flix, const FE2_VideoBitrateControls rateControlType)

  Set the rate control type.

- on2sc video_options_GetSwfHeight (const FLIX2HANDLE flix, int32_t *lpSwfHeight)

  Gets the SWF height.

- on2sc video_options_SetSwfHeight (FLIX2HANDLE flix, const int32_t lSwfHeight)

  Sets the SWF height.

- on2sc video_options_GetSwfWidth (const FLIX2HANDLE flix, int32_t *lpSwfWidth)

  Gets the SWF width.

- on2sc video_options_SetSwfWidth (FLIX2HANDLE flix, const int32_t lSwfWidth)

  Sets the SWF width.

- on2sc video_options_GetUseCustomSwfDimensions (const FLIX2HANDLE flix, on2bool *bpUseCustomSwfDimensions)

  Determine if the engine is using the SWF width and height for the SWF.

- on2sc video_options_SetUseCustomSwfDimensions (FLIX2HANDLE flix, const on2bool bUseCustomSwfDimensions)

  Tells the engine to use the SWF width and SWF height for the SWF.

- on2sc video_options_GetUseMaximumBitrate (const FLIX2HANDLE flix, on2bool *bpUseMaximumBitrate)

  Determine if the maximum bitrate target will be used.

- on2sc video_options_SetUseMaximumBitrate (FLIX2HANDLE flix, const on2bool bUseMaximumBitrate)

  Enable/disable the maximum bitrate target value.

- on2sc video_options_GetSwfFramerate (const FLIX2HANDLE flix, int32_t *pSwfFramerate)

Gets the SWF framerate.

- on2sc video_options_SetSwfFramerate (FLIX2HANDLE flix, const int32_t swfFramerate)

  Sets the SWF framerate.

- on2sc video_options_GetVideoCodec (const FLIX2HANDLE flix, FE2_VideoCodec *pVideoCodec)

  Determine which video codec will be used.

- on2sc video_options_SetVideoCodec (FLIX2HANDLE flix, const FE2_VideoCodec videoCodec)

  Select the video codec to be used.

- on2sc video_options_GetAlphaPercentage (const FLIX2HANDLE flix, int32_t *percentage)

  Gets the percentage of the video bitrate to be used to encode the alpha channel.

- on2sc video_options_SetAlphaPercentage (FLIX2HANDLE flix, const int32_t percentage)

  Sets the percentage of the video bitrate to be used to encode the alpha channel.

- on2sc video_options_GetSourceHeight (const FLIX2HANDLE flix, int32_t *height)

  Get the height of the source video.

- on2sc video_options_GetSourceWidth (const FLIX2HANDLE flix, int32_t *width)

  Get the width of the source video.

- on2sc video_options_AddFLVCuePoint (FLIX2HANDLE flix, const char *pName, const double time, const FE2_CuePointType type)

  Adds a cue point to the FLV.

- on2sc video_options_AddFLVCuePointParameter (FLIX2HANDLE flix, const char *pCuePointName, const char *pName, const char *pValue)

  Adds a name/value parameter to an already added cue point.

- on2sc video_options_GetCompressMode (const FLIX2HANDLE flix, FE2_CompressMode *mode)

  Get the current VP6 compression mode.

- on2sc video_options_SetCompressMode (FLIX2HANDLE flix, FE2_CompressMode mode)

  Set the VP6 compression mode.

Deprecated functions

- on2sc video_options_GetImageHeight (const FLIX2HANDLE flix, int32_t *lpImageHeight)

  Get the current scaled image height.

- on2sc video_options_SetImageHeight (FLIX2HANDLE flix, const int32_t lImageHeight)

  Set the scaled image height.

- on2sc video_options_GetImageWidth (const FLIX2HANDLE flix, int32_t *lpImageWidth)

  Get the current scaled image width.

- on2sc video_options_SetImageWidth (FLIX2HANDLE flix, const int32_t lImageWidth)
  Set the scaled image width.
- on2sc video_options_GetUseSourceDimensions (const FLIX2HANDLE flix, on2bool *bpUseSourceDimensions)
  Determine if the source's dimensions will be used instead of the scaled dimensions.
- on2sc video_options_SetUseSourceDimensions (FLIX2HANDLE flix, const on2bool bUseSourceDimensions)
  Switch between source and scaled dimensions.

Deprecated functions

- on2sc video_options_GetDeinterlace (const FLIX2HANDLE flix, on2bool *lpDeinterlace)
  Determine if the deinterlace filter is enabled.
- on2sc video_options_SetDeinterlace (FLIX2HANDLE flix, const on2bool lDeinterlace)
  Enable/disable the deinterlace filter.

Deprecated functions

- on2sc video_options_GetVideoFramerate (const FLIX2HANDLE flix, int32_t *lpVideoFramerate)
  Get the current video framerate.
- on2sc video_options_SetVideoFramerate (FLIX2HANDLE flix, const int32_t lVideoFramerate)
  Set the video framerate.
- on2sc video_options_GetVideoFramerateAsDouble (const FLIX2HANDLE flix, double *p_fps)
  Get the current video framerate.
- on2sc video_options_SetVideoFramerateAsDouble (FLIX2HANDLE flix, const double fps)
  Set the video framerate.
- on2sc video_options_GetUseSourceFramerate (FLIX2HANDLE flix, on2bool *bpUseSourceFramerate)
  Determine if the source's framerate will be used instead of the modified framerate.
- on2sc video_options_SetUseSourceFramerate (FLIX2HANDLE flix, const on2bool bUseSourceFramerate)
  Switch between source and scaled framerate.
- on2sc video_options_GetDecimateValue (const FLIX2HANDLE flix, uint32_t *pValue)
  Get the current decimation of the video framerate.
- on2sc video_options_SetDecimateValue (FLIX2HANDLE flix, const uint32_t value)
  Set the decimation of the video framerate.

21.46 mainpage.dox File Reference

21.47 samples.dox File Reference

Index

_on2bool
basetypes, 448
3GPP - FFmpeg, 403
3GPP2 - FFmpeg, 405

AAC, 228
AAC+, 230
about.dox, 469
AMR_NB - FFmpeg, 232
Audio Codecs, 227
Audio Encoding Options, 213
Audio Filters, 280
audio_codecs
FE2_CODEC_PCM, 227
audio_options_GetBitrate
audopts_deprecated, 216
audio_options_GetFlvAudioFormat
audopts_deprecated, 217
audio_options_GetSamplingrate
audopts_deprecated, 217
audio_options_GetStereo
audopts_deprecated, 218
audio_options_Reset
audopts_deprecated, 218
audio_options_SetBitrate
audopts_deprecated, 219
audio_options_SetFlvAudioFormat
audopts_deprecated, 219
audio_options_SetSamplingrate
audopts_deprecated, 220
audio_options_SetStereo
audopts_deprecated, 221
audio_options_Validate
audopts_deprecated, 221
audopts
Bitrate112kbps, 214
Bitrate128kbps, 214
Bitrate144kbps, 214
Bitrate160kbps, 214
Bitrate16kbps, 214
Bitrate192kbps, 214
Bitrate224kbps, 214
Bitrate24kbps, 214
Bitrate256kbps, 214
Bitrate320kbps, 214
Bitrate32kbps, 214
Bitrate40kbps, 214
Bitrate48kbps, 214
Bitrate56kbps, 214
Bitrate64kbps, 214
Bitrate80kbps, 214
Bitrate8kbps, 214
Bitrate96kbps, 214
FE2_AudioBitrates, 214
audopts_deprecated
FlvAudioMp3, 216
FlvAudioUncompressed, 216
Hertz11025, 216
Hertz22050, 216
Hertz44100, 216
audopts_deprecated
audio_options_GetBitrate, 216
audio_options_GetFlvAudioFormat, 217
audio_options_GetSamplingrate, 217
audio_options_GetStereo, 218
audio_options_Reset, 218
audio_options_SetBitrate, 219
audio_options_SetFlvAudioFormat, 219
audio_options_SetSamplingrate, 220
audio_options_SetStereo, 221
audio_options_Validate, 221
FE2_AudioSamplingrates, 216
FE2_FlvAudioFormat, 216

Base Types, 445
BASE_H264PROFILE
codec_h264, 240
basetypes
_on2bool, 448
int16_t, 446
int32_t, 446
int64_t, 446
int8_t, 447
ON264, 446
ON2_BUFFER_EMPTY, 448
ON2_BUFFER_FULL, 448
ON2_CONNREFUSED, 448
ON2_DONE, 449
ON2_ERROR, 449
ON2_INPROGRESS, 448

ON2_INVALID_PARAMS, 449
ON2_INVALID_VERSION, 448
ON2_NET_ERROR, 448
ON2_NO_MEM, 448
ON2_NOT_FOUND, 448
ON2_NOT_SUPP, 448
ON2_OK, 449
ON2_TIMEDOUT, 448
ON2_WOULDBLOCK, 448
on2bool, 447
on2false, 448
on2s16, 447
on2s32, 447
on2s64, 447
on2s8, 447
on2sc, 448
ON2TC, 446
on2tc, 447
on2true, 448
on2u16, 447
on2u32, 447
on2u64, 447
on2u8, 447
OTC, 446
PRId64, 446
uint16_t, 448
uint32_t, 448
uint64_t, 448
uint8_t, 448
binding_deviations.dox, 470
Bitrate112kbps
audopts, 214
Bitrate128kbps
audopts, 214
Bitrate144kbps
audopts, 214
Bitrate160kbps
audopts, 214
Bitrate16kbps
audopts, 214
Bitrate192kbps
audopts, 214
Bitrate224kbps
audopts, 214
Bitrate24kbps
audopts, 214
Bitrate256kbps
audopts, 214
Bitrate320kbps
audopts, 214
Bitrate32kbps
audopts, 214
Bitrate40kbps
audopts, 214
Bitrate48kbps
audopts, 214
Bitrate56kbps
audopts, 214
Bitrate64kbps
audopts, 214
Bitrate80kbps
audopts, 214
Bitrate8kbps
audopts, 214
Bitrate96kbps
audopts, 214
Blur, 295
BLUR_GAUSS
filter_blur, 297
BLUR_LOWPASS
filter_blur, 296
blurfilter
filter_blur, 296
blurfilter_t
filter_blur, 296
Brightness/Contrast/Hue/Saturation, 284

CBR_1PASSControl
codecparam_common, 225
CBR_2PASSControl
codecparam_common, 225
Codec Manipulation, 379
CODEC_H263
vidopts_deprecated, 457
codec_h264
BASE_H264PROFILE, 240
HIGH_H264PROFILE, 240
MAIN_H264PROFILE, 240
codec_lame
LAME_ABR, 243
LAME_CBR, 243
LAME_VBR_mtrh, 243
LAME_VBR_rh, 243
CODEC_NULL
vidopts_deprecated, 457
CODEC_SCREENVIDEO
vidopts_deprecated, 457
CODEC_VP6
vidopts_deprecated, 457
codec_vp6
COMPRESSMODE_BEST, 254
COMPRESSMODE_GOOD, 254
VP6_E, 254
VP6_S, 254
CODEC_VP6ALPHA
vidopts_deprecated, 457
codec_aac
FE2_AAC_BITRATE, 229

FE2_CODEC_AAC, 229
codec_aacplus
FE2_AACPLUS_BITRATE, 231
FE2_AACPLUS_PARAMETRIC_STEREO, 231
FE2_CODEC_AACPLUS, 231
codec_amrlavc
FE2_AMR_BITRATE, 233
FE2_CODEC_AMR_NB, 233
codec_h263
FE2_CODEC_H263, 234
FE2_CODEC_H263_BASELINE, 234
FE2_H263_BITRATE, 235
FE2_H263_KFFREQ, 235
FE2_H263_KFINTTYPE, 235
FE2_H263_MAX_Q, 236
FE2_H263_MIN_Q, 236
FE2_H263_RC_MODE, 236
codec_h264
FE2_CODEC_H264, 238
FE2_H264_B_FRAME_RATE, 238
FE2_H264_BITRATE, 238
FE2_H264_KFFREQ, 239
FE2_H264_KFINTTYPE, 239
FE2_H264_PROFILE, 239
FE2_H264_RC_MODE, 240
FE2_H264_SPEED, 240
h264profile, 240
h264profile_t, 240
codec_lame
FE2_CODEC_LAME, 242
FE2_LAME_BITRATE, 242
FE2_LAME_CHANNELS, 242
FE2_LAME_QUALITY, 242
FE2_LAME_RC_MODE, 243
lame_rcmode, 243
lame_rcmode_t, 243
codec_vorbislavc
FE2_CODEC_VORBIS, 244
FE2_VORBIS_BITRATE, 244
codec_vp6
FE2_CODEC_VP6, 248
FE2_CompressMode, 254
FE2_VP6_2PASS_MAX_SECTION, 248
FE2_VP6_2PASS_MIN_SECTION, 248
FE2_VP6_BITRATE, 248
FE2_VP6_CONCURRENCY, 248
FE2_VP6_CXMODE, 249
FE2_VP6_KFFREQ, 250
FE2_VP6_KFINTTYPE, 250
FE2_VP6_MAX_Q, 250
FE2_VP6_MIN_Q, 250
FE2_VP6_NOISE_REDUCTION, 251
FE2_VP6_PROFILE, 251
FE2_VP6_RC_MODE, 251
FE2_VP6_SHARPNESS, 251
FE2_VP6_STREAM_MAX_BUFFER, 252
FE2_VP6_STREAM_OPTIMAL_BUFFER, 252
FE2_VP6_STREAM_PEAK_BITRATE, 252
FE2_VP6_STREAM_PREBUFFER, 252
FE2_VP6_TEMPORAL_DOWN_-WATERMARK, 253
FE2_VP6_TEMPORAL_RESAMPLING, 253
FE2_VP6_UNDERSHOOT_PCT, 253
vp6profile, 254
vp6profile_t, 254
codec_vp6a
FE2_CODEC_VP6ALPHA, 257
FE2_VP6A_2PASS_MAX_SECTION, 257
FE2_VP6A_2PASS_MIN_SECTION, 257
FE2_VP6A_ALPHA_BITRATE, 257
FE2_VP6A_ALPHA_MAX_Q, 257
FE2_VP6A_ALPHA_MIN_Q, 258
FE2_VP6A_ALPHA_NOISE_REDUCTION, 258
FE2_VP6A_ALPHA_SHARPNESS, 258
FE2_VP6A_BITRATE, 258
FE2_VP6A_CXMODE, 258
FE2_VP6A_KFFREQ, 259
FE2_VP6A_KFINTTYPE, 259
FE2_VP6A_MAX_Q, 259
FE2_VP6A_MIN_Q, 260
FE2_VP6A_NOISE_REDUCTION, 260
FE2_VP6A_RC_MODE, 260
FE2_VP6A_SHARPNESS, 260
FE2_VP6A_STREAM_MAX_BUFFER, 261
FE2_VP6A_STREAM_OPTIMAL_BUFFER, 261
FE2_VP6A_STREAM_PEAK_BITRATE, 261
FE2_VP6A_STREAM_PREBUFFER, 262
FE2_VP6A_TEMPORAL_DOWN_-WATERMARK, 262
FE2_VP6A_TEMPORAL_RESAMPLING, 262
FE2_VP6A_UNDERSHOOT_PCT, 263
codec_vp8
FE2_CODEC_VP8, 266
FE2_VP8_2PASS_MAX_SECTION, 266
FE2_VP8_2PASS_MIN_SECTION, 266
FE2_VP8_ALTREF, 266
FE2_VP8_AR_MAX_FRAMES, 267
FE2_VP8_AR_STRENGTH, 267
FE2_VP8_AR_TYPE, 267
FE2_VP8_BITRATE, 267
FE2_VP8_CXMODE, 267
FE2_VP8_DROP_THRESH, 268

FE2_VP8_KFFREQ, 268
FE2_VP8_KFINTTYPE, 268
FE2_VP8_LAG, 268
FE2_VP8_MAX_Q, 269
FE2_VP8_MB_STATIC_THRESHOLD, 269
FE2_VP8_MIN_Q, 269
FE2_VP8_NOISE_REDUCTION, 269
FE2_VP8_OVERSHOOT_PCT, 270
FE2_VP8_PROFILE, 270
FE2_VP8_RC_MODE, 270
FE2_VP8_SHARPNESS, 270
FE2_VP8_STREAM_INITIAL_BUFFER, 271
FE2_VP8_STREAM_MAX_BUFFER, 271
FE2_VP8_STREAM_OPTIMAL_BUFFER, 271
FE2_VP8_THREADS, 271
FE2_VP8_TOKEN_PARTITIONS, 272
FE2_VP8_UNDERSHOOT_PCT, 272
codecfunc
Flix2_AddCodec, 379
Flix2_CodecGetParam, 380
Flix2_CodecGetParamAsStr, 380
Flix2_CodecSetParam, 381
Flix2_CodecSetParamAsStr, 381
Flix2_RemoveCodec, 382
codecparam_common
CBR_1PASSControl, 225
CBR_2PASSControl, 225
FIXED_KEYFRAMES, 225
MAX_KEYFRAMES, 225
VBR_1PASSControl, 225
VBR_2PASSControl, 225
codecparam_common
FE2_CODECPARAM_BITRATE, 224
FE2_VCODECPARAM_KFFREQ, 224
FE2_VCODECPARAM_KFINTTYPE, 225
FE2_VCODECPARAM_RC_MODE, 225
FE2_VideoBitrateControls, 225
FE2_VideoKeyframeTypes, 225
Codecs, 223
Common Codec Parameters, 224
COMPRESSMODE_BEST
codec_vp6, 254
COMPRESSMODE_GOOD
codec_vp6, 254
corelib
Flix2_Deinit, 371
Flix2_GetLogLevel, 371
Flix2_GetLogPath, 372
Flix2_Init, 372
Flix2_SetLogLevel, 372
Flix2_SetLogPath, 373
Crop, 298
CUE_EVENT
vidopts_deprecated, 457
CUE_NAVIGATION
vidopts_deprecated, 457
Cut, 303

Deinterlace, 281
DEINTERLACE_1_2_1_BLUR
filter_adaptive_deinterlace, 282
DEINTERLACE_ADAPTIVE
filter_adaptive_deinterlace, 282
DEINTERLACE_DROP_FIELD
filter_adaptive_deinterlace, 282
DEINTERLACE_NONE
filter_adaptive_deinterlace, 282
deinterlacemode
filter_adaptive_deinterlace, 282
deintmode_t
filter_adaptive_deinterlace, 282
Denoise, 309
Deprecated, 215, 393, 455

editor_options_GetBrightness
filter_bchs, 287
editor_options_GetContrast
filter_bchs, 287
editor_options_GetCrop
filter_crop, 300
editor_options_GetCropBounds
filter_crop, 300
editor_options_GetCutStartTime
filter_cut, 305
editor_options_GetCutStopTime
filter_cut, 305
editor_options_GetHue
filter_bchs, 287
editor_options_GetSaturation
filter_bchs, 288
editor_options_GetUseBrightness
filter_bchs, 288
editor_options_GetUseContrast
filter_bchs, 289
editor_options_GetUseCut
filter_cut, 306
editor_options_GetUseHue
filter_bchs, 289
editor_options_GetUseSaturation
filter_bchs, 290
editor_options_Reset
filterfunc_deprecated, 393
editor_options_SetBrightness
filter_bchs, 290
editor_options_SetContrast
filter_bchs, 290

editor_options_SetCrop
filter_crop, 301
editor_options_SetCropBounds
filter_crop, 301
editor_options_SetCutStartTime
filter_cut, 306
editor_options_SetCutStopTime
filter_cut, 307
editor_options_SetHue
filter_bchs, 291
editor_options_SetSaturation
filter_bchs, 291
editor_options_SetUseBrightness
filter_bchs, 292
editor_options_SetUseContrast
filter_bchs, 292
editor_options_SetUseCut
filter_cut, 308
editor_options_SetUseHue
filter_bchs, 293
editor_options_SetUseSaturation
filter_bchs, 293
editor_options_Validate
filterfunc_deprecated, 393
EmbeddedUrlIsGetUrl
muxer_swf, 420
EmbeddedUrlIsLoadMovie
muxer_swf, 420
Encoding Statistics, 273
encoding_status_GetAverageBitrate
encstatus, 273
encoding_status_GetAverageFramesize
encstatus, 274
encoding_status_GetElapsedTime
encstatus, 274
encoding_status_GetEndTime
encstatus, 274
encoding_status_GetMaximumFramesize
encstatus, 275
encoding_status_GetMinimumFramesize
encstatus, 275
encoding_status_GetStartTime
encstatus, 276
encoding_status_GetTotalFrames
encstatus, 276
encoding_status_PercentComplete
encstatus, 276
EncStateIdle
mainengine, 357
EncStateQueued
mainengine, 357
EncStateRunning
mainengine, 357
encstatus
encoding_status_GetAverageBitrate, 273
encoding_status_GetAverageFramesize, 274
encoding_status_GetElapsedTime, 274
encoding_status_GetEndTime, 274
encoding_status_GetMaximumFramesize, 275
encoding_status_GetMinimumFramesize, 275
encoding_status_GetStartTime, 276
encoding_status_GetTotalFrames, 276
encoding_status_PercentComplete, 276
ErrEncodeA
mainengine, 357
ErrEncodeV
mainengine, 357
ErrFileDecode
mainengine, 357
ErrFileDecodeA
mainengine, 357
ErrFileDecodeV
mainengine, 357
ErrFileIO
mainengine, 357
ErrFileOpen
mainengine, 357
ErrNone
mainengine, 357
ErrSys
mainengine, 357
ExportFLVVideo
mainengine, 358
ExportSWF3To6Video
mainengine, 358
ExportSWF7PlusVideo
mainengine, 358
ExportSWFVectorVideo
mainengine, 358

FE2_OVERLAY_POS_MODE_BOTLEFT
filter_overlay, 327
FE2_OVERLAY_POS_MODE_BOTRIGHT
filter_overlay, 327
FE2_OVERLAY_POS_MODE_CENTER
filter_overlay, 327
FE2_OVERLAY_POS_MODE_TOPLEFT
filter_overlay, 327
FE2_OVERLAY_POS_MODE_TOPRIGHT
filter_overlay, 327
FE2_OVERLAY_POS_MODE_XY
filter_overlay, 327
FE2_PNGEX_CP_ALL
filter_pngexporter, 343
FE2_PNGEX_CP_EVENT
filter_pngexporter, 343
FE2_PNGEX_CP_NAV
filter_pngexporter, 343

FE2_3G2_FASTSTART
muxer_3g2lavf, 405
FE2_3GP_FASTSTART
muxer_3gplavf, 403
FE2_AAC_BITRATE
codec_aac, 229
FE2_AACPLUS_BITRATE
codec_aacplus, 231
FE2_AACPLUS_PARAMETRIC_STEREO
codec_aacplus, 231
FE2_ADAPTIVE_DEINTERLACE_MODE
filter_adaptive_deinterlace, 282
FE2_AMR_BITRATE
codec_amrlavc, 233
FE2_AudioBitrates
audopts, 214
FE2_AudioSamplingrates
audopts_deprecated, 216
FE2_BCHS_BRIGHTNESS
filter_bchs, 285
FE2_BCHS_CONTRAST
filter_bchs, 286
FE2_BCHS_HUE
filter_bchs, 286
FE2_BCHS_SATURATION
filter_bchs, 286
FE2_BLUR_FILTER
filter_blur, 296
FE2_BLUR_MASKSIZE
filter_blur, 296
FE2_CODEC_AAC
codec_aac, 229
FE2_CODEC_AACPLUS
codec_aacplus, 231
FE2_CODEC_AMR_NB
codec_amrlavc, 233
FE2_CODEC_H263
codec_h263, 234
FE2_CODEC_H263_BASELINE
codec_h263, 234
FE2_CODEC_H264
codec_h264, 238
FE2_CODEC_LAME
codec_lame, 242
FE2_CODEC_PCM
audio_codecs, 227
FE2_CODEC_VORBIS
codec_vorbislavc, 244
FE2_CODEC_VP6
codec_vp6, 248
FE2_CODEC_VP6ALPHA
codec_vp6a, 257
FE2_CODEC_VP8
codec_vp8, 266
FE2_CODECPARAM_BITRATE
codecparam_common, 224
FE2_CompressMode
codec_vp6, 254
FE2_CROP_BOTTOM
filter_crop, 299
FE2_CROP_LEFT
filter_crop, 299
FE2_CROP_RIGHT
filter_crop, 299
FE2_CROP_TOP
filter_crop, 299
FE2_CuePointType
vidopts_deprecated, 457
FE2_CUT_START_SEC
filter_cut, 304
FE2_CUT_STOP_SEC
filter_cut, 304
FE2_CUT_USE_SEEK
filter_cut, 304
FE2_DENOISE_NOISE_LEVEL
filter_denoise, 309
FE2_EmbeddedUrlType
muxer_swf, 420
FE2_EncState
mainengine, 357
FE2_errno
mainengine, 357
FE2_ExportedVideoType
mainengine, 357
FE2_FILTER_ADAPTIVE_DEINTERLACE
filter_adaptive_deinterlace, 282
FE2_FILTER_BCHS
filter_bchs, 286
FE2_FILTER_BLUR
filter_blur, 296
FE2_FILTER_CROP
filter_crop, 299
FE2_FILTER_CUT
filter_cut, 304
FE2_FILTER_DENOISE
filter_denoise, 309
FE2_FILTER_FRAMERATE
filter_framerate, 311
FE2_FILTER_HIGHPASS
filter_highpass, 316
FE2_FILTER_LOWPASS
filter_lowpass, 318
FE2_FILTER_MIRROR
filter_mirror, 320
FE2_FILTER_OVERLAY
filter_overlay, 324
FE2_FILTER_PNGEX
filter_pngexporter, 338

FE2_FILTER_RESAMPLE
filter_resample, 344
FE2_FILTER_ROTATE
filter_rotate, 345
FE2_FILTER_SCALE
filter_scale, 347
FE2_FILTER_SHARPEN
filter_sharpen, 353
FE2_FLV_CUEPT_EVENT
muxer_flv, 397
FE2_FLV_CUEPT_NAV
muxer_flv, 397
FE2_FLV_CUEPT_PARAM
muxer_flv, 397
FE2_FLV_METADATA_DISABLE
muxer_flv, 398
FE2_FLV_METADATA_ENABLE
muxer_flv, 398
FE2_FlvAudioFormat
audopts_deprecated, 216
FE2_FRAMERATE_DECIMATE
filter_framerate, 311
FE2_FRAMERATE_FPS
filter_framerate, 311
FE2_FrameType
frameserver, 388
FE2_FXM_CUEPT_EVENT
muxer_fxm, 401
FE2_FXM_CUEPT_NAV
muxer_fxm, 401
FE2_FXM_CUEPT_PARAM
muxer_fxm, 401
FE2_FXM_METADATA_DISABLE
muxer_fxm, 402
FE2_FXM_METADATA_ENABLE
muxer_fxm, 402
FE2_H263_BITRATE
codec_h263, 235
FE2_H263_KFFREQ
codec_h263, 235
FE2_H263_KFINTTYPE
codec_h263, 235
FE2_H263_MAX_Q
codec_h263, 236
FE2_H263_MIN_Q
codec_h263, 236
FE2_H263_RC_MODE
codec_h263, 236
FE2_H264_B_FRAME_RATE
codec_h264, 238
FE2_H264_BITRATE
codec_h264, 238
FE2_H264_KFFREQ
codec_h264, 239
FE2_H264_KFINTTYPE
codec_h264, 239
FE2_H264_PROFILE
codec_h264, 239
FE2_H264_RC_MODE
codec_h264, 240
FE2_H264_SPEED
codec_h264, 240
FE2_HIGHPASS_CUTOFF
filter_highpass, 316
FE2_HIGHPASS_Q
filter_highpass, 316
FE2_ISOMEDIA_FASTSTART
isomedia.h, 520
FE2_LAME_BITRATE
codec_lame, 242
FE2_LAME_CHANNELS
codec_lame, 242
FE2_LAME_QUALITY
codec_lame, 242
FE2_LAME_RC_MODE
codec_lame, 243
FE2_LOWPASS_CUTOFF
filter_lowpass, 318
FE2_LOWPASS_Q
filter_lowpass, 318
FE2_MIRROR_HORIZONTAL
filter_mirror, 320
FE2_MIRROR_VERTICAL
filter_mirror, 321
FE2_MOV_FASTSTART
muxer_movlavf, 408
FE2_MP4_FASTSTART
muxer_mp4lavf, 409
FE2_MUXER_3G2
muxer_3g2lavf, 406
FE2_MUXER_3GP
muxer_3gplavf, 404
FE2_MUXER_FLV
muxer_flv, 398
FE2_MUXER_FXM
muxer_fxm, 402
FE2_MUXER_MOV
muxer_movlavf, 408
FE2_MUXER_MP4
muxer_mp4lavf, 410
FE2_MUXER_SWF
muxer_swf, 416
FE2_MUXER_WEBM
muxer_webmlavf, 444
FE2_OVERLAY_FILE
filter_overlay, 324
FE2_OVERLAY_MASK_B
filter_overlay, 325

FE2_OVERLAY_MASK_G
filter_overlay, 325
FE2_OVERLAY_MASK_R
filter_overlay, 325
FE2_OVERLAY_MASK_RGB
filter_overlay, 325
FE2_OVERLAY_MASK_X
filter_overlay, 326
FE2_OVERLAY_MASK_XY
filter_overlay, 326
FE2_OVERLAY_MASK_Y
filter_overlay, 326
FE2_OVERLAY_POS
filter_overlay, 326
FE2_OVERLAY_POS_X
filter_overlay, 327
FE2_OVERLAY_POS_Y
filter_overlay, 327
FE2_OverlayPositionMode
filter_overlay, 327
FE2_PNGEX_AUTO_EXPORT_COUNT
filter_pngexporter, 338
FE2_PNGEX_AUTO_EXPORT_END_TIME
filter_pngexporter, 339
FE2_PNGEX_AUTO_EXPORT_RANDOM_-PERIOD
filter_pngexporter, 339
FE2_PNGEX_AUTO_EXPORT_START_TIME
filter_pngexporter, 339
FE2_PNGEX_COMPRESSION_LEVEL
filter_pngexporter, 340
FE2_PNGEX_ENABLE_ALPHA
filter_pngexporter, 340
FE2_PNGEX_EXPORT_CUE_POINTS
filter_pngexporter, 340
FE2_PNGEX_EXPORT_FIRST_FRAME_PNG
filter_pngexporter, 341
FE2_PNGEX_EXPORT_INTERVAL
filter_pngexporter, 341
FE2_PNGEX_EXPORT_TIME_STRING
filter_pngexporter, 341
FE2_PNGEX_FILENAME_PREFIX
filter_pngexporter, 342
FE2_PNGEX_FILENAME_SUFFIX
filter_pngexporter, 342
FE2_PNGEX_HEIGHT
filter_pngexporter, 342
FE2_PNGEX_OUTPUT_DIRECTORY
filter_pngexporter, 342
FE2_PNGEX_WIDTH
filter_pngexporter, 343
FE2_PNGExCuePtMode
filter_pngexporter, 343
FE2_RESAMPLE_CHANNELS
filter_resample, 344
FE2_RESAMPLE_RATE
filter_resample, 344
FE2_ROTATE_ANGLE
filter_rotate, 345
FE2_SCALE_HEIGHT
filter_scale, 347
FE2_SCALE_WIDTH
filter_scale, 348
FE2_SWF_ADAPTIVE_PRELOAD_BUFFER_-FACTOR
muxer_swf, 416
FE2_SWF_ADD_VARIABLE
muxer_swf, 416
FE2_SWF_DELETE_VARIABLE
muxer_swf, 417
FE2_SWF_EMBEDDED_URL
muxer_swf, 417
FE2_SWF_EMBEDDED_URL_TARGET
muxer_swf, 417
FE2_SWF_EMBEDDED_URL_TYPE
muxer_swf, 417
FE2_SWF_FIXED_PRELOAD_PCT
muxer_swf, 418
FE2_SWF_FRAMERATE
muxer_swf, 418
FE2_SWF_HEIGHT
muxer_swf, 418
FE2_SWF_LOOP_COUNT
muxer_swf, 418
FE2_SWF_ON_END_OPTION
muxer_swf, 419
FE2_SWF_ON_END_URL
muxer_swf, 419
FE2_SWF_ON_START_OPTION
muxer_swf, 419
FE2_SWF_PRELOAD_TYPE
muxer_swf, 419
FE2_SWF_START_BLANK_FRAME
muxer_swf, 419
FE2_SWF_START_WAIT_SEC
muxer_swf, 420
FE2_SWF_WIDTH
muxer_swf, 420
FE2_SwfOnEndOptions
muxer_swf, 420
FE2_SwfOnStartOptions
muxer_swf, 421
FE2_SwfPreloaderOptions
muxer_swf, 421
FE2_VCODECPARAM_KFFREQ
codecparam_common, 224
FE2_VCODECPARAM_KFINTTYPE
codecparam_common, 225

FE2_VCODECPARAM_RC_MODE
codecparam_common, 225
FE2_VideoBitrateControls
codecparam_common, 225
FE2_VideoCodec
vidopts_deprecated, 457
FE2_VideoFormat
frameserver, 388
FE2_VideoKeyframeTypes
codecparam_common, 225
FE2_VORBIS_BITRATE
codec_vorbislavc, 244
FE2_VP6_2PASS_MAX_SECTION
codec_vp6, 248
FE2_VP6_2PASS_MIN_SECTION
codec_vp6, 248
FE2_VP6_BITRATE
codec_vp6, 248
FE2_VP6_CONCURRENCY
codec_vp6, 248
FE2_VP6_CXMODE
codec_vp6, 249
FE2_VP6_KFFREQ
codec_vp6, 250
FE2_VP6_KFINTTYPE
codec_vp6, 250
FE2_VP6_MAX_Q
codec_vp6, 250
FE2_VP6_MIN_Q
codec_vp6, 250
FE2_VP6_NOISE_REDUCTION
codec_vp6, 251
FE2_VP6_PROFILE
codec_vp6, 251
FE2_VP6_RC_MODE
codec_vp6, 251
FE2_VP6_SHARPNESS
codec_vp6, 251
FE2_VP6_STREAM_MAX_BUFFER
codec_vp6, 252
FE2_VP6_STREAM_OPTIMAL_BUFFER
codec_vp6, 252
FE2_VP6_STREAM_PEAK_BITRATE
codec_vp6, 252
FE2_VP6_STREAM_PREBUFFER
codec_vp6, 252
FE2_VP6_TEMPORAL_DOWN_WATERMARK
codec_vp6, 253
FE2_VP6_TEMPORAL_RESAMPLING
codec_vp6, 253
FE2_VP6_UNDERSHOOT_PCT
codec_vp6, 253
FE2_VP6A_2PASS_MAX_SECTION
codec_vp6a, 257
FE2_VP6A_2PASS_MIN_SECTION
codec_vp6a, 257
FE2_VP6A_ALPHA_BITRATE
codec_vp6a, 257
FE2_VP6A_ALPHA_MAX_Q
codec_vp6a, 257
FE2_VP6A_ALPHA_MIN_Q
codec_vp6a, 258
FE2_VP6A_ALPHA_NOISE_REDUCTION
codec_vp6a, 258
FE2_VP6A_ALPHA_SHARPNESS
codec_vp6a, 258
FE2_VP6A_BITRATE
codec_vp6a, 258
FE2_VP6A_CXMODE
codec_vp6a, 258
FE2_VP6A_KFFREQ
codec_vp6a, 259
FE2_VP6A_KFINTTYPE
codec_vp6a, 259
FE2_VP6A_MAX_Q
codec_vp6a, 259
FE2_VP6A_MIN_Q
codec_vp6a, 260
FE2_VP6A_NOISE_REDUCTION
codec_vp6a, 260
FE2_VP6A_RC_MODE
codec_vp6a, 260
FE2_VP6A_SHARPNESS
codec_vp6a, 260
FE2_VP6A_STREAM_MAX_BUFFER
codec_vp6a, 261
FE2_VP6A_STREAM_OPTIMAL_BUFFER
codec_vp6a, 261
FE2_VP6A_STREAM_PEAK_BITRATE
codec_vp6a, 261
FE2_VP6A_STREAM_PREBUFFER
codec_vp6a, 262
FE2_VP6A_TEMPORAL_DOWN_-
WATERMARK
codec_vp6a, 262
FE2_VP6A_TEMPORAL_RESAMPLING
codec_vp6a, 262
FE2_VP6A_UNDERSHOOT_PCT
codec_vp6a, 263
FE2_VP8_2PASS_MAX_SECTION
codec_vp8, 266
FE2_VP8_2PASS_MIN_SECTION
codec_vp8, 266
FE2_VP8_ALTREF
codec_vp8, 266
FE2_VP8_AR_MAX_FRAMES
codec_vp8, 267
FE2_VP8_AR_STRENGTH

codec_vp8, 267
FE2_VP8_AR_TYPE
codec_vp8, 267
FE2_VP8_BITRATE
codec_vp8, 267
FE2_VP8_CXMODE
codec_vp8, 267
FE2_VP8_DROP_THRESH
codec_vp8, 268
FE2_VP8_KFFREQ
codec_vp8, 268
FE2_VP8_KFINTTYPE
codec_vp8, 268
FE2_VP8_LAG
codec_vp8, 268
FE2_VP8_MAX_Q
codec_vp8, 269
FE2_VP8_MB_STATIC_THRESHOLD
codec_vp8, 269
FE2_VP8_MIN_Q
codec_vp8, 269
FE2_VP8_NOISE_REDUCTION
codec_vp8, 269
FE2_VP8_OVERSHOOT_PCT
codec_vp8, 270
FE2_VP8_PROFILE
codec_vp8, 270
FE2_VP8_RC_MODE
codec_vp8, 270
FE2_VP8_SHARPNESS
codec_vp8, 270
FE2_VP8_STREAM_INITIAL_BUFFER
codec_vp8, 271
FE2_VP8_STREAM_MAX_BUFFER
codec_vp8, 271
FE2_VP8_STREAM_OPTIMAL_BUFFER
codec_vp8, 271
FE2_VP8_THREADS
codec_vp8, 271
FE2_VP8_TOKEN_PARTITIONS
codec_vp8, 272
FE2_VP8_UNDERSHOOT_PCT
codec_vp8, 272
Filter Manipulation, 375
filter_adaptive_deinterlace
DEINTERLACE_1_2_1_BLUR, 282
DEINTERLACE_ADAPTIVE, 282
DEINTERLACE_DROP_FIELD, 282
DEINTERLACE_NONE, 282
filter_blur
BLUR_GAUSS, 297
BLUR_LOWPASS, 296
MASK_3x3, 297
MASK_5x5, 297
filter_overlay
FE2_OVERLAY_POS_MODE_BOTLEFT, 327
FE2_OVERLAY_POS_MODE_BOTRIGHT, 327
FE2_OVERLAY_POS_MODE_CENTER, 327
FE2_OVERLAY_POS_MODE_TOPLEFT, 327
FE2_OVERLAY_POS_MODE_TOPRIGHT, 327
FE2_OVERLAY_POS_MODE_XY, 327
filter_pngexporter
FE2_PNGEX_CP_ALL, 343
FE2_PNGEX_CP_EVENT, 343
FE2_PNGEX_CP_NAV, 343
filter_adaptive_deinterlace
deinterlacemode, 282
deintmode_t, 282
FE2_ADAPTIVE_DEINTERLACE_MODE, 282
FE2_FILTER_ADAPTIVE_DEINTERLACE, 282
video_options_GetDeinterlace, 282
video_options_SetDeinterlace, 283
filter_bchs
editor_options_GetBrightness, 287
editor_options_GetContrast, 287
editor_options_GetHue, 287
editor_options_GetSaturation, 288
editor_options_GetUseBrightness, 288
editor_options_GetUseContrast, 289
editor_options_GetUseHue, 289
editor_options_GetUseSaturation, 290
editor_options_SetBrightness, 290
editor_options_SetContrast, 290
editor_options_SetHue, 291
editor_options_SetSaturation, 291
editor_options_SetUseBrightness, 292
editor_options_SetUseContrast, 292
editor_options_SetUseHue, 293
editor_options_SetUseSaturation, 293
FE2_BCHS_BRIGHTNESS, 285
FE2_BCHS_CONTRAST, 286
FE2_BCHS_HUE, 286
FE2_BCHS_SATURATION, 286
FE2_FILTER_BCHS, 286
filter_blur
blurfilter, 296
blurfilter_t, 296
FE2_BLUR_FILTER, 296
FE2_BLUR_MASKSIZE, 296
FE2_FILTER_BLUR, 296
masksiz, 297

masksiz_t, 296
filter_crop
editor_options_GetCrop, 300
editor_options_GetCropBounds, 300
editor_options_SetCrop, 301
editor_options_SetCropBounds, 301
FE2_CROP_BOTTOM, 299
FE2_CROP_LEFT, 299
FE2_CROP_RIGHT, 299
FE2_CROP_TOP, 299
FE2_FILTER_CROP, 299
filter_cut
editor_options_GetCutStartTime, 305
editor_options_GetCutStopTime, 305
editor_options_GetUseCut, 306
editor_options_SetCutStartTime, 306
editor_options_SetCutStopTime, 307
editor_options_SetUseCut, 308
FE2_CUT_START_SEC, 304
FE2_CUT_STOP_SEC, 304
FE2_CUT_USE_SEEK, 304
FE2_FILTER_CUT, 304
filter_denoise
FE2_DENOISE_NOISE_LEVEL, 309
FE2_FILTER_DENOISE, 309
filter_framerate
FE2_FILTER_FRAMERATE, 311
FE2_FRAMERATE_DECIMATE, 311
FE2_FRAMERATE_FPS, 311
video_options_GetDecimateValue, 311
video_options_GetUseSourceFramerate, 312
video_options_GetVideoFramerate, 312
video_options_GetVideoFramerateAsDouble, 313
video_options_SetDecimateValue, 313
video_options_SetUseSourceFramerate, 314
video_options_SetVideoFramerate, 314
video_options_SetVideoFramerateAsDouble, 315
filter_highpass
FE2_FILTER_HIGHPASS, 316
FE2_HIGHPASS_CUTOFF, 316
FE2_HIGHPASS_Q, 316
filter_lowpass
FE2_FILTER_LOWPASS, 318
FE2_LOWPASS_CUTOFF, 318
FE2_LOWPASS_Q, 318
filter_mirror
FE2_FILTER_MIRROR, 320
FE2_MIRROR_HORIZONTAL, 320
FE2_MIRROR_VERTICAL, 321
filter_overlay
FE2_FILTER_OVERLAY, 324
FE2_OVERLAY_FILE, 324
FE2_OVERLAY_MASK_B, 325
FE2_OVERLAY_MASK_G, 325
FE2_OVERLAY_MASK_R, 325
FE2_OVERLAY_MASK_RGB, 325
FE2_OVERLAY_MASK_X, 326
FE2_OVERLAY_MASK_XY, 326
FE2_OVERLAY_MASK_Y, 326
FE2_OVERLAY_POS, 326
FE2_OVERLAY_POS_X, 327
FE2_OVERLAY_POS_Y, 327
FE2_OverlayPositionMode, 327
overlay_options_GetMaskPixelRGB, 328
overlay_options_GetMaskPixelXY, 328
overlay_options_GetOverlayPath, 329
overlay_options_GetOverlayPosition, 329
overlay_options_GetUseOverlay, 330
overlay_options_Reset, 331
overlay_options_SetMaskPixelRGB, 331
overlay_options_SetMaskPixelXY, 332
overlay_options_SetOverlayPath, 333
overlay_options_SetOverlayPosition, 333
overlay_options_SetUseOverlay, 334
filter_pngexporter
FE2_FILTER_PNGEX, 338
FE2_PNGEX_AUTO_EXPORT_COUNT, 338
FE2_PNGEX_AUTO_EXPORT_END_-TIME, 339
FE2_PNGEX_AUTO_EXPORT_-RANDOM_PERIOD, 339
FE2_PNGEX_AUTO_EXPORT_START_-TIME, 339
FE2_PNGEX_COMPRESSION_LEVEL, 340
FE2_PNGEX_ENABLE_ALPHA, 340
FE2_PNGEX_EXPORT_CUE_POINTS, 340
FE2_PNGEX_EXPORT_FIRST_FRAME_-PNG, 341
FE2_PNGEX_EXPORT_INTERVAL, 341
FE2_PNGEX_EXPORT_TIME_STRING, 341
FE2_PNGEX_FILENAME_PREFIX, 342
FE2_PNGEX_FILENAME_SUFFIX, 342
FE2_PNGEX_HEIGHT, 342
FE2_PNGEX_OUTPUT_DIRECTORY, 342
FE2_PNGEX_WIDTH, 343
FE2_PNGExCuePtMode, 343
filter_resample
FE2_FILTER_RESAMPLE, 344
FE2_RESAMPLE_CHANNELS, 344
FE2_RESAMPLE_RATE, 344
filter_rotate
FE2_FILTER_ROTATE, 345
FE2_ROTATE_ANGLE, 345
filter_scale

FE2_FILTER_SCALE, 347
FE2_SCALE_HEIGHT, 347
FE2_SCALE_WIDTH, 348
video_options_GetImageHeight, 348
video_options_GetImageWidth, 349
video_options_GetUseSourceDimensions, 349
video_options_SetImageHeight, 350
video_options_SetImageWidth, 350
video_options_SetUseSourceDimensions, 351
filter_sharpen
FE2_FILTER_SHARPEN, 353
filterfunc
Flix2_AddFilter, 375
Flix2_FilterGetParam, 376
Flix2_FilterGetParamAsStr, 376
Flix2_FilterSetParam, 377
Flix2_FilterSetParamAsStr, 377
Flix2_RemoveFilter, 378
filterfunc_deprecated
editor_options_Reset, 393
editor_options_Validate, 393
Filters, 278
FIXED_KEYFRAMES
codecparam_common, 225
Flix Engine API, 354
Flix Engine Core Library, 371
Flix2_AddCodec
codecfunc, 379
Flix2_AddFilter
filterfunc, 375
Flix2_AddMuxer
muxerfunc, 383
Flix2_CodecGetParam
codecfunc, 380
Flix2_CodecGetParamAsStr
codecfunc, 380
Flix2_CodecSetParam
codecfunc, 381
Flix2_CodecSetParamAsStr
codecfunc, 381
Flix2_Copyright
mainengine, 358
Flix2_Create
mainengine, 358
Flix2_CreateEx
mainengine, 359
Flix2_Deinit
corelib, 371
Flix2_Destroy
mainengine, 359
Flix2_Encode
mainengine, 360
Flix2_EncodeFrame
frameserver, 389
Flix2_Errno
mainengine, 360
Flix2_FilterGetParam
filterfunc, 376
Flix2_FilterGetParamAsStr
filterfunc, 376
Flix2_FilterSetParam
filterfunc, 377
Flix2_FilterSetParamAsStr
filterfunc, 377
Flix2_GetEncoderState
mainengine, 361
Flix2_GetExportAudio
mainengine, 361
Flix2_GetExportVideo
mainengine, 362
Flix2_GetExportVideoType
mainengine, 362
Flix2_GetFrameServer
frameserver, 390
Flix2_GetInputFile
mainengine, 363
Flix2_GetLogLevel
corelib, 371
Flix2_GetLogPath
corelib, 372
Flix2_GetOutputFile
mainengine, 363
Flix2_GetOverwriteExistingFiles
mainengine, 364
Flix2_GetSourceDuration
mainengine, 364
Flix2_Init
corelib, 372
Flix2_IsEncoderRunning
mainengine, 365
Flix2_MuxerGetParam
muxerfunc, 384
Flix2_MuxerGetParamAsStr
muxerfunc, 384
Flix2_MuxerSetParam
muxerfunc, 385
Flix2_MuxerSetParamAsStr
muxerfunc, 385
Flix2_RemoveCodec
codecfunc, 382
Flix2_RemoveFilter
filterfunc, 378
Flix2_RemoveMuxer
muxerfunc, 386
Flix2_Reset
mainengine, 365
Flix2_SetAudioProperties
frameserver, 390

Flix2_SetExportAudio
mainengine, 366
Flix2_SetExportVideo
mainengine, 366
Flix2_SetExportVideoType
mainengine, 367
Flix2_SetFrameServer
frameserver, 391
Flix2_SetInputFile
mainengine, 367
Flix2_SetLogLevel
corelib, 372
Flix2_SetLogPath
corelib, 373
Flix2_SetOutputFile
mainengine, 368
Flix2_SetOverwriteExistingFiles
mainengine, 368
Flix2_SetVideoProperties
frameserver, 391
Flix2_StopEncoding
mainengine, 369
Flix2_Validate
mainengine, 369
Flix2_Version
mainengine, 370
FLIX2HANDLE
flixengine2.h, 512
FLIX2PLGNHANDLE
flixengine2.h, 512
flixengine2.h
FLIX2HANDLE, 512
FLIX2PLGNHANDLE, 512
FLIXENGINE_API, 511
FLIXENGINE_VERSION_CHIEF, 511
FLIXENGINE_VERSION_EXTRA, 511
FLIXENGINE_VERSION_MAJOR, 511
FLIXENGINE_VERSION_MINOR, 512
FLIXENGINE_VERSION_PATCH, 512
FLIXENGINE_VERSION_STR, 512
flixengine2/audio_options.h, 471
flixengine2/codec_constants.h, 473
flixengine2/codecs/aac.h, 474
flixengine2/codecs/amr.h, 475
flixengine2/codecs/codec_common.h, 476
flixengine2/codecs/h263.h, 477
flixengine2/codecs/h264.h, 478
flixengine2/codecs/lame.h, 479
flixengine2/codecs/vorbis.h, 480
flixengine2/codecs/vp6.h, 481
flixengine2/codecs/vp6_alpha.h, 483
flixengine2/codecs/vp8.h, 485
flixengine2/encoding_status.h, 487
flixengine2/filter_constants.h, 488
flixengine2/filters/adaptive_deinterlace.h, 489
flixengine2/filters/bchs.h, 490
flixengine2/filters/blur.h, 491
flixengine2/filters/crop.h, 492
flixengine2/filters/cut.h, 493
flixengine2/filters/denoise.h, 494
flixengine2/filters/framerate.h, 495
flixengine2/filters/highpass.h, 496
flixengine2/filters/lowpass.h, 497
flixengine2/filters/mirror.h, 498
flixengine2/filters/overlay.h, 499
flixengine2/filters/png_export.h, 501
flixengine2/filters/resample.h, 503
flixengine2/filters/rotate.h, 504
flixengine2/filters/scale.h, 505
flixengine2/filters/sharpen.h, 506
flixengine2/flixengine2.h, 507
flixengine2/fs_options.h, 513
flixengine2/media_editor_options.h, 514
flixengine2/muxer_constants.h, 516
flixengine2/muxers/flv.h, 517
flixengine2/muxers/fxm.h, 519
flixengine2/muxers/isomedia.h, 520
flixengine2/muxers/swf.h, 522
flixengine2/muxers/webm.h, 524
flixengine2/on2types.h, 525
flixengine2/overlay_options.h, 527
flixengine2/swf_options.h, 528
flixengine2/video_options.h, 531
FLIXENGINE_API
flixengine2.h, 511
FLIXENGINE_VERSION_CHIEF
flixengine2.h, 511
FLIXENGINE_VERSION_EXTRA
flixengine2.h, 511
FLIXENGINE_VERSION_MAJOR
flixengine2.h, 511
FLIXENGINE_VERSION_MINOR
flixengine2.h, 512
FLIXENGINE_VERSION_PATCH
flixengine2.h, 512
FLIXENGINE_VERSION_STR
flixengine2.h, 512
FLV, 395
flv_metadata
muxer_flv, 398
FlvAudioMp3
audopts_deprecated, 216
FlvAudioUncompressed
audopts_deprecated, 216
flvmetadata_t
muxer_flv, 398
Frame Rate, 310
Frame Server API, 387

frameserver
FE2_FrameType, 388
FE2_VideoFormat, 388
Flix2_EncodeFrame, 389
Flix2_GetFrameServer, 390
Flix2_SetAudioProperties, 390
Flix2_SetFrameServer, 391
Flix2_SetVideoProperties, 391
FrameTypeAudio, 388
FrameTypeVideo, 388
VideoFormatBGRA, 389
VideoFormatYV12, 389
FrameTypeAudio
frameserver, 388
FrameTypeVideo
frameserver, 388
FXM, 400
fxmmetadata_t
muxer_fxm, 402

H263 - FFmpeg, 234
H264, 237
h264profile
codec_h264, 240
h264profile_t
codec_h264, 240
Hertz11025
audopts_deprecated, 216
Hertz22050
audopts_deprecated, 216
Hertz44100
audopts_deprecated, 216
HIGH_H264PROFILE
codec_h264, 240
Highpass, 316

int16_t
basetypes, 446
int32_t
basetypes, 446
int64_t
basetypes, 446
int8_t
basetypes, 447
isomedia.h
FE2_ISOMEDIA_FASTSTART, 520

LAME_ABR
codec_lame, 243
LAME_CBR
codec_lame, 243
LAME_VBR_mtrh
codec_lame, 243
LAME_VBR_rh
codec_lame, 243
lame_rcmode
codec_lame, 243
lame_rcmode_t
codec_lame, 243
libvorbis - FFmpeg, 244
Lowpass, 318

Main Engine Interface, 355
MAIN_H264PROFILE
codec_h264, 240
mainengine
EncStateIdle, 357
EncStateQueued, 357
EncStateRunning, 357
ErrEncodeA, 357
ErrEncodeV, 357
ErrFileDecode, 357
ErrFileDecodeA, 357
ErrFileDecodeV, 357
ErrFileIO, 357
ErrFileOpen, 357
ErrNone, 357
ErrSys, 357
ExportFLVVideo, 358
ExportSWF3To6Video, 358
ExportSWF7PlusVideo, 358
ExportSWFVectorVideo, 358
FE2_EncState, 357
FE2_errno, 357
FE2_ExportedVideoType, 357
Flix2_Copyright, 358
Flix2_Create, 358
Flix2_CreateEx, 359
Flix2_Destroy, 359
Flix2_Encode, 360
Flix2_Errno, 360
Flix2_GetEncoderState, 361
Flix2_GetExportAudio, 361
Flix2_GetExportVideo, 362
Flix2_GetExportVideoType, 362
Flix2_GetInputFile, 363
Flix2_GetOutputFile, 363
Flix2_GetOverwriteExistingFiles, 364
Flix2_GetSourceDuration, 364
Flix2_IsEncoderRunning, 365
Flix2_Reset, 365
Flix2_SetExportAudio, 366
Flix2_SetExportVideo, 366
Flix2_SetExportVideoType, 367
Flix2_SetInputFile, 367
Flix2_SetOutputFile, 368
Flix2_SetOverwriteExistingFiles, 368
Flix2_StopEncoding, 369

Flix2_Validate, 369
Flix2_Version, 370
mainpage.dox, 535
MASK_3x3
filter_blur, 297
MASK_5x5
filter_blur, 297
masksiz
filter_blur, 297
masksiz_t
filter_blur, 296
MAX_KEYFRAMES
codecparam_common, 225
MD_AUDIO_CODECID
muxer_flv, 399
MD_AUDIO_DATARATE
muxer_flv, 399
MD_AUDIO_SIZE
muxer_flv, 399
MD_CANSEEKTOEND
muxer_flv, 399
MD_DATASIZE
muxer_flv, 399
MD_DURATION
muxer_flv, 399
MD_FRAMERATE
muxer_flv, 399
MD_HEIGHT
muxer_flv, 399
MD_KEYFRAMES
muxer_flv, 399
MD_LASTKEYFRAMELOCATION
muxer_flv, 399
MD_LASTKEYFRAMETIMESTAMP
muxer_flv, 399
MD_LASTTIMESTAMP
muxer_flv, 399
MD_VIDEO_CODECID
muxer_flv, 399
MD_VIDEO_DATARATE
muxer_flv, 399
MD_VIDEO_SIZE
muxer_flv, 399
MD_WIDTH
muxer_flv, 399
Mirror, 320
MOV - FFmpeg, 407
MP3 - LAME, 241
MP4 - FFmpeg, 409
Muxer Manipulation, 383
muxer_flv
MD_AUDIO_CODECID, 399
MD_AUDIO_DATARATE, 399
MD_AUDIO_SIZE, 399
MD_CANSEEKTOEND, 399
MD_DATASIZE, 399
MD_DURATION, 399
MD_FRAMERATE, 399
MD_HEIGHT, 399
MD_KEYFRAMES, 399
MD_LASTKEYFRAMELOCATION, 399
MD_LASTKEYFRAMETIMESTAMP, 399
MD_LASTTIMESTAMP, 399
MD_VIDEO_CODECID, 399
MD_VIDEO_DATARATE, 399
MD_VIDEO_SIZE, 399
MD_WIDTH, 399
muxer_swf
EmbeddedUrlIsGetUrl, 420
EmbeddedUrlIsLoadMovie, 420
SwfAdaptivePreloader, 421
SwfFixedPreloader, 421
SwfOnMovieEndLoadMovie, 421
SwfOnMovieEndLoop, 421
SwfOnMovieEndNothing, 420
SwfOnMovieEndSTOP, 420
SwfOnMovieEndUnload, 421
SwfOnMovieStartAutomatically, 421
SwfOnMovieStartEmbedSTOP, 421
SwfOnMovieStartOnClick, 421
SwfOnMovieStartWait, 421
SwfPreloaderNone, 421
muxer_3g2lavf
FE2_3G2_FASTSTART, 405
FE2_MUXER_3G2, 406
muxer_3gplavf
FE2_3GP_FASTSTART, 403
FE2_MUXER_3GP, 404
muxer_flv
FE2_FLV_CUEPT_EVENT, 397
FE2_FLV_CUEPT_NAV, 397
FE2_FLV_CUEPT_PARAM, 397
FE2_FLV_METADATA_DISABLE, 398
FE2_FLV_METADATA_ENABLE, 398
FE2_MUXER_FLV, 398
flv_metadata, 398
flvmetadata_t, 398
muxer_fxm
FE2_FXM_CUEPT_EVENT, 401
FE2_FXM_CUEPT_NAV, 401
FE2_FXM_CUEPT_PARAM, 401
FE2_FXM_METADATA_DISABLE, 402
FE2_FXM_METADATA_ENABLE, 402
FE2_MUXER_FXM, 402
fxmmetadata_t, 402
muxer_movlavf
FE2_MOV_FASTSTART, 408
FE2_MUXER_MOV, 408

muxer_mp4lavf
FE2_MP4_FASTSTART, 409
FE2_MUXER_MP4, 410
muxer_swf
FE2_EmbeddedUrlType, 420
FE2_MUXER_SWF, 416
FE2_SWF_ADAPTIVE_PRELOAD_-BUFFER_FACTOR, 416
FE2_SWF_ADD_VARIABLE, 416
FE2_SWF_DELETE_VARIABLE, 417
FE2_SWF_EMBEDDED_URL, 417
FE2_SWF_EMBEDDED_URL_TARGET, 417
FE2_SWF_EMBEDDED_URL_TYPE, 417
FE2_SWF_FIXED_PRELOAD_PCT, 418
FE2_SWF_FRAMERATE, 418
FE2_SWF_HEIGHT, 418
FE2_SWF_LOOP_COUNT, 418
FE2_SWF_ON_END_OPTION, 419
FE2_SWF_ON_END_URL, 419
FE2_SWF_ON_START_OPTION, 419
FE2_SWF_PRELOAD_TYPE, 419
FE2_SWF_START_BLANK_FRAME, 419
FE2_SWF_START_WAIT_SEC, 420
FE2_SWF_WIDTH, 420
FE2_SwfOnEndOptions, 420
FE2_SwfOnStartOptions, 421
FE2_SwfPreloaderOptions, 421
swf_options_AddVariable, 421
swf_options_DeleteVariable, 422
swf_options_-GetAdaptivePreloaderBufferFactor, 422
swf_options_GetEmbeddedUrl, 423
swf_options_GetEmbeddedUrlTarget, 424
swf_options_GetEmbeddedUrlType, 425
swf_options_GetEnablePreloader, 425
swf_options_GetInsertBlankFrameOnStart, 426
swf_options_GetLoadMovieOnEndUrl, 426
swf_options_GetLoopCount, 427
swf_options_GetMovieOnEndOptions, 427
swf_options_GetMovieOnStartOptions, 428
swf_options_GetPercentToPreload, 429
swf_options_GetPreloaderType, 429
swf_options_GetSwfFramerate, 430
swf_options_GetSwfFramerateAsDouble, 430
swf_options_GetVariableCount, 431
swf_options_GetWaitTimeToStart, 431
swf_options_Reset, 432
swf_options_-SetAdaptivePreloaderBufferFactor, 432
swf_options_SetEmbeddedUrl, 433
swf_options_SetEmbeddedUrlTarget, 434
swf_options_SetEmbeddedUrlType, 435
swf_options_SetEnablePreloader, 436
swf_options_SetInsertBlankFrameOnStart, 436
swf_options_SetLoadMovieOnEndUrl, 437
swf_options_SetLoopCount, 437
swf_options_SetMovieOnEndOptions, 438
swf_options_SetMovieOnStartOptions, 439
swf_options_SetPercentToPreload, 439
swf_options_SetPreloaderType, 440
swf_options_SetSwfFramerate, 441
swf_options_SetSwfFramerateAsDouble, 441
swf_options_SetWaitTimeToStart, 442
swf_options_UpdateVariable, 442
muxer_webmlavf
FE2_MUXER_WEBM, 444
muxerfunc
Flix2_AddMuxer, 383
Flix2_MuxerGetParam, 384
Flix2_MuxerGetParamAsStr, 384
Flix2_MuxerSetParam, 385
Flix2_MuxerSetParamAsStr, 385
Flix2_RemoveMuxer, 386
Muxers, 394

ON264
basetypes, 446
ON2_BUFFER_EMPTY
basetypes, 448
ON2_BUFFER_FULL
basetypes, 448
ON2_CONNREFUSED
basetypes, 448
ON2_DONE
basetypes, 449
ON2_ERROR
basetypes, 449
ON2_INPROGRESS
basetypes, 448
ON2_INVALID_PARAMS
basetypes, 449
ON2_INVALID_VERSION
basetypes, 448
ON2_NET_ERROR
basetypes, 448
ON2_NO_MEM
basetypes, 448
ON2_NOT_FOUND
basetypes, 448
ON2_NOT_SUPP
basetypes, 448
ON2_OK
basetypes, 449

ON2_TIMEDOUT
basetypes, 448
ON2_WOULDBLOCK
basetypes, 448
ON2API
on2types.h, 526
on2bool
basetypes, 447
on2false
basetypes, 448
on2s16
basetypes, 447
on2s32
basetypes, 447
on2s64
basetypes, 447
on2s8
basetypes, 447
on2sc
basetypes, 448
ON2TC
basetypes, 446
on2tc
basetypes, 447
on2true
basetypes, 448
on2types.h
ON2API, 526
on2u16
basetypes, 447
on2u32
basetypes, 447
on2u64
basetypes, 447
on2u8
basetypes, 447
OTC
basetypes, 446
Overlay (Watermark), 322
overlay_options_GetMaskPixelRGB
filter_overlay, 328
overlay_options_GetMaskPixelXY
filter_overlay, 328
overlay_options_GetOverlayPath
filter_overlay, 329
overlay_options_GetOverlayPosition
filter_overlay, 329
overlay_options_GetUseOverlay
filter_overlay, 330
overlay_options_Reset
filter_overlay, 331
overlay_options_SetMaskPixelRGB
filter_overlay, 331
overlay_options_SetMaskPixelXY
filter_overlay, 332
overlay_options_SetOverlayPath
filter_overlay, 333
overlay_options_SetOverlayPosition
filter_overlay, 333
overlay_options_SetUseOverlay
filter_overlay, 334

PNG Image Export (Thumbnail), 336
PRId64
basetypes, 446

Resample, 344
Rotate, 345

samples.dox, 536
Scale, 346
Sharpen, 353
SWF, 411
swf_options_AddVariable
muxer_swf, 421
swf_options_DeleteVariable
muxer_swf, 422
swf_options_GetAdaptivePreloaderBufferFactor
muxer_swf, 422
swf_options_GetEmbeddedUrl
muxer_swf, 423
swf_options_GetEmbeddedUrlTarget
muxer_swf, 424
swf_options_GetEmbeddedUrlType
muxer_swf, 425
swf_options_GetEnablePreloader
muxer_swf, 425
swf_options_GetInsertBlankFrameOnStart
muxer_swf, 426
swf_options_GetLoadMovieOnEndUrl
muxer_swf, 426
swf_options_GetLoopCount
muxer_swf, 427
swf_options_GetMovieOnEndOptions
muxer_swf, 427
swf_options_GetMovieOnStartOptions
muxer_swf, 428
swf_options_GetPercentToPreload
muxer_swf, 429
swf_options_GetPreloaderType
muxer_swf, 429
swf_options_GetSwfFramerate
muxer_swf, 430
swf_options_GetSwfFramerateAsDouble
muxer_swf, 430
swf_options_GetVariableCount
muxer_swf, 431
swf_options_GetWaitTimeToStart

muxer_swf, 431
swf_options_Reset
muxer_swf, 432
swf_options_SetAdaptivePreloaderBufferFactor
muxer_swf, 432
swf_options_SetEmbeddedUrl
muxer_swf, 433
swf_options_SetEmbeddedUrlTarget
muxer_swf, 434
swf_options_SetEmbeddedUrlType
muxer_swf, 435
swf_options_SetEnablePreloader
muxer_swf, 436
swf_options_SetInsertBlankFrameOnStart
muxer_swf, 436
swf_options_SetLoadMovieOnEndUrl
muxer_swf, 437
swf_options_SetLoopCount
muxer_swf, 437
swf_options_SetMovieOnEndOptions
muxer_swf, 438
swf_options_SetMovieOnStartOptions
muxer_swf, 439
swf_options_SetPercentToPreload
muxer_swf, 439
swf_options_SetPreloaderType
muxer_swf, 440
swf_options_SetSwfFramerate
muxer_swf, 441
swf_options_SetSwfFramerateAsDouble
muxer_swf, 441
swf_options_SetWaitTimeToStart
muxer_swf, 442
swf_options_UpdateVariable
muxer_swf, 442
SwfAdaptivePreloader
muxer_swf, 421
SwfFixedPreloader
muxer_swf, 421
SwfOnMovieEndLoadMovie
muxer_swf, 421
SwfOnMovieEndLoop
muxer_swf, 421
SwfOnMovieEndNothing
muxer_swf, 420
SwfOnMovieEndSTOP
muxer_swf, 420
SwfOnMovieEndUnload
muxer_swf, 421
SwfOnMovieStartAutomatically
muxer_swf, 421
SwfOnMovieStartEmbedSTOP
muxer_swf, 421
SwfOnMovieStartOnClick
muxer_swf, 421
SwfOnMovieStartWait
muxer_swf, 421
SwfPreloaderNone
muxer_swf, 421

uint16_t
basetypes, 448
uint32_t
basetypes, 448
uint64_t
basetypes, 448
uint8_t
basetypes, 448

VBR_1PASSControl
codecparam_common, 225
VBR_2PASSControl
codecparam_common, 225
Video Codecs, 226
Video Encoding Options, 450
Video Filters, 279
video_options_AddFLVCuePoint
vidopts_deprecated, 457
video_options_AddFLVCuePointParameter
vidopts_deprecated, 458
video_options_GetAlphaPercentage
vidopts_deprecated, 459
video_options_GetCompressMode
vidopts_deprecated, 459
video_options_GetDecimateValue
filter_framerate, 311
video_options_GetDeinterlace
filter_adaptive_deinterlace, 282
video_options_GetImageHeight
filter_scale, 348
video_options_GetImageQuality
vidopts_deprecated, 460
video_options_GetImageWidth
filter_scale, 349
video_options_GetKeyframeInterval
vidopts_deprecated, 460
video_options_GetKeyframeIntervalType
vidopts_deprecated, 460
video_options_GetMaximumBitrate
vidopts_deprecated, 461
video_options_GetRateControlType
vidopts_deprecated, 461
video_options_GetSourceHeight
vidopts, 450
video_options_GetSourceWidth
vidopts, 451
video_options_GetSwfFramerate
vidopts_deprecated, 462

video_options_GetSwfHeight
vidopts, 451
video_options_GetSwfWidth
vidopts, 451
video_options_GetUseCustomSwfDimensions
vidopts, 452
video_options_GetUseMaximumBitrate
vidopts_deprecated, 462
video_options_GetUseSourceDimensions
filter_scale, 349
video_options_GetUseSourceFramerate
filter_framerate, 312
video_options_GetVideoCodec
vidopts_deprecated, 463
video_options_GetVideoFramerate
filter_framerate, 312
video_options_GetVideoFramerateAsDouble
filter_framerate, 313
video_options_Reset
vidopts, 452
video_options_SetAlphaPercentage
vidopts_deprecated, 463
video_options_SetCompressMode
vidopts_deprecated, 464
video_options_SetDecimateValue
filter_framerate, 313
video_options_SetDeinterlace
filter_adaptive_deinterlace, 283
video_options_SetImageHeight
filter_scale, 350
video_options_SetImageQuality
vidopts_deprecated, 464
video_options_SetImageWidth
filter_scale, 350
video_options_SetKeyframeInterval
vidopts_deprecated, 465
video_options_SetKeyframeIntervalType
vidopts_deprecated, 465
video_options_SetMaximumBitrate
vidopts_deprecated, 466
video_options_SetRateControlType
vidopts_deprecated, 467
video_options_SetSwfFramerate
vidopts_deprecated, 467
video_options_SetSwfHeight
vidopts, 452
video_options_SetSwfWidth
vidopts, 453
video_options_SetUseCustomSwfDimensions
vidopts, 453
video_options_SetUseMaximumBitrate
vidopts_deprecated, 468
video_options_SetUseSourceDimensions
filter_scale, 351
video_options_SetUseSourceFramerate
filter_framerate, 314
video_options_SetVideoCodec
vidopts_deprecated, 468
video_options_SetVideoFramerate
filter_framerate, 314
video_options_SetVideoFramerateAsDouble
filter_framerate, 315
video_options_Validate
vidopts, 454
VideoFormatBGRA
frameserver, 389
VideoFormatYV12
frameserver, 389
vidopts
video_options_GetSourceHeight, 450
video_options_GetSourceWidth, 451
video_options_GetSwfHeight, 451
video_options_GetSwfWidth, 451
video_options_-
GetUseCustomSwfDimensions, 452
video_options_Reset, 452
video_options_SetSwfHeight, 452
video_options_SetSwfWidth, 453
video_options_-
SetUseCustomSwfDimensions, 453
video_options_Validate, 454
vidopts_deprecated
CODEC_H263, 457
CODEC_NULL, 457
CODEC_SCREENVIDEO, 457
CODEC_VP6, 457
CODEC_VP6ALPHA, 457
CUE_EVENT, 457
CUE_NAVIGATION, 457
vidopts_deprecated
FE2_CuePointType, 457
FE2_VideoCodec, 457
video_options_AddFLVCuePoint, 457
video_options_AddFLVCuePointParameter,
458
video_options_GetAlphaPercentage, 459
video_options_GetCompressMode, 459
video_options_GetImageQuality, 460
video_options_GetKeyframeInterval, 460
video_options_GetKeyframeIntervalType, 460
video_options_GetMaximumBitrate, 461
video_options_GetRateControlType, 461
video_options_GetSwfFramerate, 462
video_options_GetUseMaximumBitrate, 462
video_options_GetVideoCodec, 463
video_options_SetAlphaPercentage, 463
video_options_SetCompressMode, 464
video_options_SetImageQuality, 464

video_options_SetKeyframeInterval, 465
video_options_SetKeyframeIntervalType, 465
video_options_SetMaximumBitrate, 466
video_options_SetRateControlType, 467
video_options_SetSwfFramerate, 467
video_options_SetUseMaximumBitrate, 468
video_options_SetVideoCodec, 468
VP6, 245
VP6 with Alpha, 255
VP6_E
codec_vp6, 254
VP6_S
codec_vp6, 254
vp6profile
codec_vp6, 254
vp6profile_t
codec_vp6, 254
VP8, 264

WebM - FFmpeg, 444