SDK Developer Reference for VP8

Media SDK API Version 1.29

LEGAL DISCLAIMER

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web Site.

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2007-2019, Intel Corporation. All Rights reserved.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Table of Contents

SDK Developer Reference for VP8 1
Table of Contents 4
Overview 5
- Document Conventions 5
- Acronyms and Abbreviations 5

Architecture & Programming Guide 5
- Decoding Procedure 5
- Encoding Procedure 5

Structure Reference 5
- mfxExtVP8CodingOption 5

Enumerator Reference 6
- CodecProfile 6
- ExtendedBufferID 6

Overview

Intel® Media Software Development Kit – SDK is a software development library that exposes the media acceleration capabilities of Intel platforms for decoding, encoding and video processing. The API library covers a wide range of Intel platforms.

This document describes the SDK extension to support VP8 video codec.

Document Conventions

The SDK API uses the Verdana typeface for normal prose. With the exception of section headings and the table of contents, all code-related items appear in the `Courier New` typeface. Examples relevant to this document are `mfxStatus` and `MFXInit`. Hyperlinks appear in underlined boldface, such as `mfxStatus`.

Acronyms and Abbreviations

SDK	Intel® Media Software Development Kit – SDK
`API`	Application Programming Interface

Architecture & Programming Guide

SDK extension for VP8 requires the application to use an additional include file `mfxvp8.h`, in addition to the regular SDK include files. No additional library is needed at the link time.

```
Include these files:
#include "mfxvideo.h"    /* SDK functions in C */
#include "mfxvideo++.h"    /* Optional for C++ development */
#include "mfxvp8.h"        /* VP8 development */
Link to this library:
    libmfx.lib            /* The SDK dispatcher library */
```

The SDK extends the codec identifier `MFX_CODEC_VP8` for VP8 processing.

Decoding Procedure

The application should use the same decoding procedure that described in the *SDK Developer Reference*. The only difference is in partitioning of input bitstream. Unlike other supported by SDK decoders, VP8 can accept only complete frame as input and application should provide it accompanied by `MFX_BITSTREAM_COMPLETE_FRAME` flag.

Encoding Procedure

The application should use the same encoding procedure that described in the *SDK Developer Reference*.

Structure Reference

mfxExtVP8CodingOption

Definition

```
typedef struct {
    mfxExtBuffer    Header;

    mfxU16   Version;
    mfxU16   EnableMultipleSegments;
    mfxU16   LoopFilterType;
    mfxU16   LoopFilterLevel[4];
    mfxU16   SharpnessLevel;
    mfxU16   NumTokenPartitions;
    mfxI16   LoopFilterRefTypeDelta[4];
    mfxI16   LoopFilterMbModeDelta[4];
    mfxI16   SegmentQPDelta[4];
    mfxI16   CoeffTypeQPDelta[5];
    mfxU16   WriteIVFHeaders;
    mfxU32   NumFramesForIVFHeader;
    mfxU16   reserved[223];
} mfxExtVP8CodingOption;
```

Description

This `mfxExtVP8CodingOption` structure describes VP8 encoder configuration parameters.

Members

`Header.BufferId`	Must be set to `MFX_EXTBUFF_VP8_CODING_OPTION`.
`Version`	Determines the bitstream version. Corresponds to the same VP8 syntax element in `frame_tag`.
`EnableMultipleSegments`	Set this option to ON, to enable segmentation. This is tri-state option. See the `CodingOptionValue` enumerator for values of this option in the *SDK Developer Reference* for details
`LoopFilterType`	Selecting the type of filter (normal or simple). Corresponds to VP8 syntax element `filter_type`.
`LoopFilterLevel`	Controls the filter strength. Corresponds to VP8 syntax element `loop_filter_level`.
`SharpnessLevel`	Controls the filter sensitivity. Corresponds to VP8 syntax element `sharpness_level`.
`NumTokenPartitions`	Specifies number of token partitions in the coded frame.

LoopFilterRefTypeDelta	Loop filter level delta for reference type (intra, last, golden, altref).
LoopFilterMbModeDelta	Loop filter level delta for MB modes.
SegmentQPDelta	QP delta for segment.
CoeffTypeQPDelta	QP delta for coefficient type (YDC, Y2AC, Y2DC, UVAC, UVDC).
WriteIVFHeaders	Set this option to ON, to enable insertion of IVF container headers into bitstream. This is tri-state option. See the CodingOptionValue enumerator for values of this option in the *SDK Developer Reference* for details
NumFramesForIVFHeader	Specifies number of frames for IVF header when WriteIVFHeaders is ON.

Change History

This structure is available since SDK API 1.12.

Enumerator Reference

CodecProfile

Description

The CodecProfile enumerator is extended to support VP8 profiles. See the *SDK Developer Reference* for additional profile definitions.

Name/Description

MFX_PROFILE_VP8_0, MFX_PROFILE_VP8_1, MFX_PROFILE_VP8_2, MFX_PROFILE_VP8_3	VP8 profiles

Change History

This enumerator is available since SDK API 1.0. SDK API 1.12 added VP8 profiles.

ExtendedBufferID

Description

The ExtendedBufferID enumerator is extended to add VP8 support. See the *SDK Developer Reference* for additional definitions.

Name/Description

MFX_EXTBUFF_VP8_CODING_OPTION	This extended buffer describes VP8 encoder configuration parameters. See the mfxExtVP8CodingOption structure for details. The application can attach this buffer to the mfxVideoParam structure for encoding initialization.

Change History

This enumerator is available since SDK API 1.0.

SDK API 1.12 adds MFX_EXTBUFF_VP8_CODING_OPTION.