



**OpenMAX™ Integration Layer
Extension**

VP8 and WebP Codec Component

Version 1.0.0

Copyright © 2012 The Khronos Group Inc.

February 14, 2012
Document version 1.0.0.0

Copyright © 2005-2012 The Khronos Group Inc. All Rights Reserved.

This specification is protected by copyright laws and contains material proprietary to the Khronos Group, Inc. It or any components may not be reproduced, republished, distributed, transmitted, displayed, broadcast, or otherwise exploited in any manner without the express prior written permission of the Khronos Group. You may use this specification for implementing the functionality therein, without altering or removing any trademark, copyright or other notice from the specification, but the receipt or possession of this specification does not convey any rights to reproduce, disclose, or distribute its contents, or to manufacture, use, or sell anything that it may describe, in whole or in part.

Khronos Group grants express permission to any current Promoter, Contributor or Adopter member of Khronos to copy and redistribute UNMODIFIED versions of this specification in any fashion, provided that NO CHARGE is made for the specification and the latest available update of the specification for any version of the API is used whenever possible. Such distributed specification may be reformatted AS LONG AS the contents of the specification are not changed in any way. The specification may be incorporated into a product that is sold as long as such product includes significant independent work developed by the seller. A link to the current version of this specification on the Khronos Group website should be included whenever possible with specification distributions.

Khronos Group makes no, and expressly disclaims any, representations or warranties, express or implied, regarding this specification, including, without limitation, any implied warranties of merchantability or fitness for a particular purpose or non-infringement of any intellectual property. Khronos Group makes no, and expressly disclaims any, warranties, express or implied, regarding the correctness, accuracy, completeness, timeliness, and reliability of the specification. Under no circumstances will the Khronos Group, or any of its Promoters, Contributors or Members or their respective partners, officers, directors, employees, agents or representatives be liable for any damages, whether direct, indirect, special or consequential damages for lost revenues, lost profits, or otherwise, arising from or in connection with these materials.

SAMPLE CODE and EXAMPLES, as identified herein, are expressly depicted herein with a “grey” watermark and are included for illustrative purposes only and are expressly outside of the Scope as defined in Attachment A - Khronos Group Intellectual Property (IP) Rights Policy of the Khronos Group Membership Agreement. A Member or Promoter Member shall have no obligation to grant any licenses under any Necessary Patent Claims covering SAMPLE CODE and EXAMPLES.

Khronos and OpenMAX are trademarks of the Khronos Group Inc. Bluetooth is a registered trademark of the Bluetooth Special Interest Group. RealAudio and RealVideo are registered trademarks of RealNetworks, Inc. Windows Media is a registered trademark of Microsoft Corporation.

Contents

1	INTRODUCTION	4
2	MOTIVATION	4
3	PROBLEM DESCRIPTION	4
3.1	VP8	4
3.2	WEBP	4
4	PROPOSED SOLUTION.....	4
4.1	VP8	4
4.2	WEBP	7
4.3	VP8 BUFFER PAYLOAD	8
4.4	MEDIA CONTAINERS	8
5	STANDARD COMPONENTS.....	8
5.1	VP8 DECODER COMPONENT	8
5.2	VP8 ENCODER COMPONENT	10
5.3	WEBP DECODER.....	12
5.4	WEBP ENCODER.....	13
6	REFERENCES	16

1 Introduction

This document describes a proposal for standardizing OpenMAX IL extensions for VP8 video codec [1] and WebP image format [2].

2 Motivation

VP8 is a video compression algorithm that was designed and developed by On2 Technologies. Google acquired On2 in February, 2010 and released the VP8 specification and reference implementations as open-source technology in May, 2010 as part of the WebM open media project [3]. WebP is an image format which compression algorithm is based on the intra-frame coding of the VP8 video format. Both VP8 and WebP are licensed under a BSD license and are entirely free of royalties.

To enable OpenMAX IL application developers to use and create VP8 and WebP content, Google proposes to define OpenMAX IL extensions for VP8 video codec and WebP image format.

The proposals describe extensions to the 1.x OpenMAX IL specification.

3 Problem Description

3.1 VP8

There are no video coding types, parameters, and indexes defined for VP8 in the OpenMAX IL specification.

3.2 WebP

There are no image coding types defined for WebP in the OpenMAX IL specification.

4 Proposed Solution

4.1 VP8

A new coding type **OMX_VIDEO_CodingVP8** needs to be defined to refer to the VP8 coded stream format [1, p.93-105]. A standard extension or index needs to be defined to configure the OpenMAX IL VP8 component. Also a parameter structure needs to be defined for the extension or index.

The proposal is to use extension “**OMX.Khronos.index.param.vp8**” or index **OMX_IndexParamVideoVp8**, whichever is appropriate, for configuring an OpenMAX IL VP8 component and **OMX_VIDEO_PARAM_VP8TYPE** as the parameter structure associated with the extension or index.

```
typedef struct OMX_VIDEO_PARAM_VP8TYPE {  
    OMX_U32 nSize;
```

```

OMX_VERSIONTYPE nVersion;
OMX_U32 nPortIndex;
OMX_VIDEO_VP8PROFILETYPE eProfile;
OMX_VIDEO_VP8LEVELTYPE eLevel;
OMX_U32 nDCTPartitions;
OMX_BOOL bErrorResilientMode;
} OMX_VIDEO_PARAM_VP8TYPE;

```

The parameters for OMX_VIDEO_PARAM_VP8TYPE are defined as follows.

- nPortIndex is the read-only value containing the index of the port.
- eProfile is the profile used for the types of VP8 encoding or decoding that are supported.

Table 1 shows the possible VP8 video profile types in OMX_VIDEO_VP8PROFILETYPE.

Table 1: Supported VP8 Profile Types

Field Name	VP8 Profile Descriptions
OMX_VIDEO_VP8ProfileUnknown	Unknown, unused or not required profile
OMX_VIDEO_VP8ProfileMain	VP8 Main profile
OMX_VIDEO_VP8ProfileMax	Maximum value

- eLevel is the level used for VP8 encoding or decoding. Table 2 shows the possible VP8 video level types in OMX_VIDEO_VP8LEVELTYPE.

Table 2: Supported VP8 Level Types

Field Name	VP8 Level Descriptions
OMX_VIDEO_VP8LevelUnknown	Unknown, unused or not required setting
OMX_VIDEO_VP8Level_Version0	VP8 Level “Version 0”
OMX_VIDEO_VP8Level_Version1	VP8 Level “Version 1”
OMX_VIDEO_VP8Level_Version2	VP8 Level “Version 2”
OMX_VIDEO_VP8Level_Version3	VP8 Level “Version 3”
OMX_VIDEO_VP8Level_VersionMax	Maximum value

In VP8 certain decoding tools are enabled or disabled based on the eLevel and higher level means less decoding complexity. Table 3 shows which decoding tools are enabled or disabled.

Table 3: Decoding Tools Usage Based on VP8 level

Level	Reconstruction Filter	Loop Filter
“Version 0”	Bicubic	Normal
“Version 1”	Bilinear	Simple
“Version 2”	Bilinear	None
“Version 3”	None	None

- nDCTPartitions specifies the number of DCT coefficient data partitions within a compressed frame. Using more than 1 partition may allow more effective multi-threaded decoding. Table 4 shows the possible values for nDCTPartitions.

Table 4: nDCTPartitions Values

Value	Description
0	1 DCT residual partition
1	2 DCT residual partitions
2	4 DCT residual partitions

3	8 DCT residual partitions
---	---------------------------

• `bErrorResilientMode` is a Boolean value used to indicate if error resilient mode is enabled. This mode prevents cumulative probability updates and is used in video telephony.

VP8 uses two encoding concepts:

1) Frame coding type.

There are only two types of frames in VP8, intraframes (key frames, I-frames) and interframes (prediction frames, P-frames). Frame coding type is controlled with **OMX_CONFIG_INTRAREFRESHVOPTYPE** structure.

2) Reference frame buffers.

VP8 uses three reference frame buffers called immediately previous frame, golden frame and alternate frame to predict blocks in an interframe. Every key frame is automatically golden frame and alternate frame. Optionally any interframe may replace the most recent golden frame and/or alternate frame.

A new structure needs to be defined to allow flexible usage of the VP8 reference frame buffers while encoding is in progress. The proposal is to use index

OMX_IndexConfigVideoVp8ReferenceFrame and the **OMX_VIDEO_VP8REFERENCEFRAMETYPE** structure to control this behavior.

```
typedef struct OMX_VIDEO_VP8REFERENCEFRAMETYPE {
    OMX_U32 nSize;
    OMX_VERSIONTYPE nVersion;
    OMX_U32 nPortIndex;
    OMX_BOOL bPreviousFrameRefresh;
    OMX_BOOL bGoldenFrameRefresh;
    OMX_BOOL bAlternateFrameRefresh;
    OMX_BOOL bUsePreviousFrame;
    OMX_BOOL bUseGoldenFrame;
    OMX_BOOL bUseAlternateFrame;
} OMX_VIDEO_VP8REFERENCEFRAMETYPE ;
```

The parameters for **OMX_VIDEO_VP8REFERENCEFRAMETYPE** are defined as follows.

- `nPortIndex` is the read-only value containing the index of the port.
- `bPreviousFrameRefresh` is a Boolean value used to indicate if the next frame is used to refresh (update) the immediately previous frame.
- `bGoldenFrameRefresh` is a Boolean value used to indicate if the next frame is to be encoded as a golden reference frame.
- `bAlternateFrameRefresh` is a Boolean value used to indicate if the next frame is to be encoded as an alternate reference frame.
- `bUsePreviousFrame` is a Boolean value used to indicate if the immediately previous frame should be used for prediction.
- `bUseGoldenFrame` is a Boolean value used to indicate if the golden reference frame should be used for prediction.
- `bUseAlternateFrame` is a Boolean value used to indicate if the alternate reference frame should be used for prediction.

Table 5: Possible Ways to Refresh Reference Frames

bPrevious FrameRefresh	bGolden FrameRefresh	bAlternate FrameRefresh	Effect on Coding
OMX_FALSE	OMX_FALSE	OMX_FALSE	Droppable frame. Usable for temporal scalability, as no future frames will use this frame as a reference. Frame is droppable if <code>bErrorResilientMode = OMX_TRUE</code> .
OMX_FALSE	OMX_FALSE	OMX_TRUE	Alternate reference frame is updated by this frame.
OMX_FALSE	OMX_TRUE	OMX_FALSE	Golden reference frame is updated by this frame.
OMX_FALSE	OMX_TRUE	OMX_TRUE	Alternate reference frame and golden reference frame are updated by this frame.
OMX_TRUE	OMX_FALSE	OMX_FALSE	Immediately previous frame is updated by this frame.
OMX_TRUE	OMX_FALSE	OMX_TRUE	Immediately previous frame and alternate reference frame are updated by this frame.
OMX_TRUE	OMX_TRUE	OMX_FALSE	Immediately previous frame and golden reference frame are updated by this frame.
OMX_TRUE	OMX_TRUE	OMX_TRUE	Immediately previous frame, golden reference frame, and alternate reference frame are updated by this frame.

The `OMX_VIDEO_VP8REFERENCEFRAMEINFOTYPE` structure is used to report the VP8 reference frame type while video decoding is in progress. The proposal is to use index `OMX_IndexConfigVideoVp8ReferenceFrameType` and the `OMX_VIDEO_VP8REFERENCEFRAMEINFOTYPE` structure to control this behavior.

```
typedef struct OMX_VIDEO_VP8REFERENCEFRAMEINFOTYPE {
    OMX_U32 nSize;
    OMX_VERSIONTYPE nVersion;
    OMX_U32 nPortIndex;
    OMX_BOOL bIsIntraFrame;
    OMX_BOOL bIsGoldenOrAlternateFrame;
} OMX_VIDEO_VP8REFERENCEFRAMEINFOTYPE ;
```

The parameters for `OMX_VIDEO_VP8REFERENCEFRAMEINFOTYPE` are defined as follows.

- `nPortIndex` is the read-only value containing the index of the port.
- `bIsIntraFrame` is a Boolean value used to indicate if the frame is an Intra frame.
- `bIsGoldenOrAlternateFrame` is a Boolean value used to indicate if the frame is a golden frame or an alternate frame.

The parameter may only be used to query the reference frame type at any time that the component is in the `OMX_StateExecuting` state.

4.2 WebP

A new image compression format `OMX_IMAGE_CodingWEBP` needs to be defined to refer to the WebP coded image format.

Table 6: Supported Image Compression Formats

Field Name	Compression Format Description	Reference to Standard
OMX_IMAGE_CodingWEBP	WebP image format	WEBP

The **OMX_IMAGE_PARAM_QFACTORTYPE** structure from the OpenMAX IL specification (v1.1.2) can be used to configure the quality factor for WebP compression.

4.3 VP8 Buffer Payload

VP8 bitstream format does not contain start-codes or other means of identifying frame boundaries. Therefore **OMX_BUFFERFLAG_ENDOFFRAME** flag should be used to identify the end of frame.

4.4 Media Containers

OpenMAX IL specification v1.2 defines a new structure called **OMX_MEDIACONTAINER_INFOTYPE** which identifies the media container format. New media container format types **OMX_FORMAT_WEBM** and **OMX_FORMAT_WEBP** need to be defined to refer the WebM and WebP media containers.

Table 7: Media Container Formats

OMX_MEDIACONTAINER_FORMATTYPE Enumerated Value	Description
OMX_FORMAT_WEBM	WebM file
OMX_FORMAT_WEBP	WebP file

5 Standard Components

Standard components for VP8 and WebP codecs can also be specified for standardizing the behavior of an OpenMAX component (encoder/decoder) of the above mentioned codecs.

The behavior to be defined is similar to other video encoder/decoder standard components, except for additional configurations specific to the particular codec.

5.1 VP8 Decoder Component

Name	video_decoder.vp8			
Description	Decodes the given compressed video stream into an uncompressed video stream.			
Ports	Index	Domain	Direction	Description
	VPB+0	video	input	Consumes compressed video content.
	VPB+1	video	output	Produces uncompressed raw video.

Port Index	VPB+0		
Description	Consumes compressed video content.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	Specify/query the video port settings. nFrameWidth = 176 nFrameHeight = 144

			<p>nBitRate = 64000</p> <p>xFrameRate = 15</p> <p>eCompressionFormat = OMX_VIDEO_CodingVP8</p> <p>eColorFormat = OMX_COLOR_FormatUnused</p>
	OMX_IndexParamVideoPortFormat	r/w	<p>Specify/query the video format.</p> <p>eCompressionFormat = OMX_VIDEO_CodingVP8</p> <p>eColorFormat = OMX_COLOR_FormatUnused</p>
	OMX_IndexParamVideoVP8	r	<p>eProfile = OMX_VIDEO_VP8ProfileMain</p> <p>eLevel = OMX_VIDEO_VP8Level_Version0</p>
	OMX_IndexParamVideoProfileLevel QuerySupported	r	Query supported profile/level pair by index.
	OMX_IndexParamVideoProfileLevel Current	r	Query current profile/level pair.

Port Index	VPB+1		
Description	Produces uncompressed raw video.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	<p>Specify/query the video port settings.</p> <p>nFrameWidth = 176</p> <p>nFrameHeight = 144</p> <p>eCompressionFormat = OMX_VIDEO_CodingUnused</p> <p>eColorFormat = At least one of the following: OMX_COLOR_FormatYUV420Planar OMX_COLOR_FormatYUV420PackedPlanar OMX_COLOR_FormatYUV420SemiPlanar OMX_COLOR_FormatYUV420PackedSemiPlanar OMX_COLOR_FormatYVU420Planar OMX_COLOR_FormatYVU420PackedPlanar OMX_COLOR_FormatYVU420SemiPlanar OMX_COLOR_FormatYVU420PackedSemiPlanar</p>
	OMX_IndexParamVideoPortFormat	r/w	<p>Specify/query the video format.</p> <p>eCompressionFormat = OMX_VIDEO_CodingUnused</p>

			<p>eColorFormat = At least one of the following:</p> <p><i>OMX_COLOR_FormatYUV420Planar</i> <i>OMX_COLOR_FormatYUV420PackedPlanar</i> <i>OMX_COLOR_FormatYUV420SemiPlanar</i> <i>OMX_COLOR_FormatYUV420PackedSemiPlanar</i> <i>OMX_COLOR_FormatYVU420Planar</i> <i>OMX_COLOR_FormatYVU420PackedPlanar</i> <i>OMX_COLOR_FormatYVU420SemiPlanar</i> <i>OMX_COLOR_FormatYVU420PackedSemiPlanar</i></p>
--	--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.2 VP8 Encoder Component

Name	video_encoder.vp8			
Description	Encodes the given uncompressed video stream into a compressed format.			
Ports	Index	Domain	Direction	Description
	VPB+0	video	input	Consumes the uncompressed raw video content.
	VPB+1	video	output	Produces compressed video.

Port Index	VPB+0		
Description	Consumes the uncompressed raw video content.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	<p>Specify/query the video port settings.</p> <p>nFrameWidth = 176</p> <p>nFrameHeight = 144</p> <p>eCompressionFormat = <i>OMX_VIDEO_CodingUnused</i></p> <p>eColorFormat = At least one of the following:</p> <p><i>OMX_COLOR_FormatYUV420Planar</i> <i>OMX_COLOR_FormatYUV420PackedPlanar</i> <i>OMX_COLOR_FormatYUV420SemiPlanar</i> <i>OMX_COLOR_FormatYUV420PackedSemiPlanar</i> <i>OMX_COLOR_FormatYVU420Planar</i> <i>OMX_COLOR_FormatYVU420PackedPlanar</i> <i>OMX_COLOR_FormatYVU420SemiPlanar</i> <i>OMX_COLOR_FormatYVU420PackedSemiPlanar</i></p>
	OMX_IndexParamVideoPortFormat	r/w	<p>Specify/query the video format.</p> <p>eCompressionFormat = <i>OMX_VIDEO_CodingUnused</i></p>

			<p>eColorFormat = At least one of the following:</p> <p><i>OMX_COLOR_FormatYUV420Planar</i> <i>OMX_COLOR_FormatYUV420PackedPlanar</i> <i>OMX_COLOR_FormatYUV420SemiPlanar</i> <i>OMX_COLOR_FormatYUV420PackedSemiPlanar</i> <i>OMX_COLOR_FormatYVU420Planar</i> <i>OMX_COLOR_FormatYVU420PackedPlanar</i> <i>OMX_COLOR_FormatYVU420SemiPlanar</i> <i>OMX_COLOR_FormatYVU420PackedSemiPlanar</i></p>
--	--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Port Index	VPB+1		
Description	Produces compressed video.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	Specify/query the video port settings. nFrameWidth = 176 nFrameHeight = 144 nBitRate = 64000 xFrameRate = 15 eCompressionFormat = <i>OMX_VIDEO_CodingVP8</i> eColorFormat = <i>OMX_COLOR_FormatUnused</i>
	OMX_IndexParamVideoPortFormat	r/w	Specify/query the video format. eCompressionFormat = <i>OMX_VIDEO_CodingVP8</i> eColorFormat = <i>OMX_COLOR_FormatUnused</i>
	OMX_IndexParamVideoBitrate	r/w	eControlRate = <i>OMX_Video_ControlRateConstant</i> <i>OMX_Video_ControlRateDisable</i> <i>OMX_Video_ControlRateVariable</i> nTargetBitrate = 64000
OMX_IndexParamVideoVP8	r/w	eProfile = <i>OMX_VIDEO_VP8ProfileMain</i> eLevel = <i>OMX_VIDEO_VP8Level_Version0</i> nDCTPartitions = 0 to 3 bErrorResilientMode =	

			<i>OMX_FALSE</i> <i>OMX_TRUE</i>
	OMX_IndexConfigVideoFramerate	r/w	Specify/query target framerate <i>xFrameRate = 15</i>
	OMX_IndexConfigVideoBitrate	r/w	Specify/query target bitrate <i>nBitRate = 64000</i>
	OMX_IndexParamVideoProfileLevel QuerySupported	r	Query supported profile/level pair by index.
	OMX_IndexParamVideoProfileLevel Current	r/w	Specify/query current profile/level pair.

5.3 WebP Decoder

Name	Image_decoder.WEBP			
Description	Decodes the given compressed image data stream into an uncompressed image data stream.			
Ports	Index	Domain	Direction	Description
	IPB+0	image	input	Accepts encoded image data.
	IPB+1	image	output	Emits decoded image data.

Port Index	IPB+0		
Description	Accepts encoded image data.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	Specify/query the image port settings. <i>nFrameWidth = 640</i> <i>nFrameHeight = 480</i> <i>eCompressionFormat =</i> <i>OMX_IMAGE_CodingWEBP</i> <i>eColorFormat =</i> <i>OMX_COLOR_FormatUnused</i>
	OMX_IndexParamImagePortFormat	r/w	Specify/query the image format. <i>eCompressionFormat =</i> <i>OMX_IMAGE_CodingWEBP</i> <i>eColorFormat =</i> <i>OMX_COLOR_FormatUnused</i>

Port Index	IPB+1		
Description	Emits decoded image data.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	Specify/query the image port settings. <i>nFrameWidth = 640</i> <i>nFrameHeight = 480</i> <i>eCompressionFormat =</i>

			<p><i>OMX_IMAGE_CodingUnused</i></p> <p>eColorFormat = At least one of the following: <i>OMX_COLOR_FormatYUV420Planar</i> <i>OMX_COLOR_FormatYUV420PackedPlanar</i> <i>OMX_COLOR_FormatYUV420SemiPlanar</i> <i>OMX_COLOR_FormatYUV420PackedSemiPlanar</i> <i>OMX_COLOR_FormatYVU420Planar</i> <i>OMX_COLOR_FormatYVU420PackedPlanar</i> <i>OMX_COLOR_FormatYVU420SemiPlanar</i> <i>OMX_COLOR_FormatYVU420PackedSemiPlanar</i></p>
	OMX_IndexParamVideoPortFormat	r/w	<p>Specify/query the image format. eCompressionFormat = <i>OMX_IMAGE_CodingUnused</i></p> <p>eColorFormat = At least one of the following: <i>OMX_COLOR_FormatYUV420Planar</i> <i>OMX_COLOR_FormatYUV420PackedPlanar</i> <i>OMX_COLOR_FormatYUV420SemiPlanar</i> <i>OMX_COLOR_FormatYUV420PackedSemiPlanar</i> <i>OMX_COLOR_FormatYVU420Planar</i> <i>OMX_COLOR_FormatYVU420PackedPlanar</i> <i>OMX_COLOR_FormatYVU420SemiPlanar</i> <i>OMX_COLOR_FormatYVU420PackedSemiPlanar</i></p>

5.4 WebP Encoder

Name	Image_encoder.WEBP			
Description	Encodes the given image data stream into a compressed format.			
Ports	Index	Domain	Direction	Description
	IPB+0	image	input	Accepts image data for encoding.
	IPB+1	image	output	Emits compressed image data.

Port Index	IPB+0		
Description	Accepts image data for encoding.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	Specify/query the image port settings. nFrameWidth = 640 nFrameHeight = 480

			<p>eCompressionFormat = OMX_IMAGE_CodingUnused</p> <p>eColorFormat = At least one of the following: OMX_COLOR_FormatYUV420Planar OMX_COLOR_FormatYUV420PackedPlanar OMX_COLOR_FormatYUV420SemiPlanar OMX_COLOR_FormatYUV420PackedSemiPlanar OMX_COLOR_FormatYVU420Planar OMX_COLOR_FormatYVU420PackedPlanar OMX_COLOR_FormatYVU420SemiPlanar OMX_COLOR_FormatYVU420PackedSemiPlanar</p>
	OMX_IndexParamImagePortFormat	r/w	<p>Specify/query the video format.</p> <p>eCompressionFormat = OMX_IMAGE_CodingUnused</p> <p>eColorFormat = At least one of the following: OMX_COLOR_FormatYUV420Planar OMX_COLOR_FormatYUV420PackedPlanar OMX_COLOR_FormatYUV420SemiPlanar OMX_COLOR_FormatYUV420PackedSemiPlanar OMX_COLOR_FormatYVU420Planar OMX_COLOR_FormatYVU420PackedPlanar OMX_COLOR_FormatYVU420SemiPlanar OMX_COLOR_FormatYVU420PackedSemiPlanar</p>

Port Index	IPB+1		
Description	Emits compressed image data.		
Required Parameters/ Configs	Index	Access	Description
	OMX_IndexParamPortDefinition	r/w	<p>Specify/query the image port settings.</p> <p>nFrameWidth = 640 (same as input)</p> <p>nFrameHeight = 480 (same as input)</p> <p>eCompressionFormat = OMX_IMAGE_CodingWEBP</p> <p>eColorFormat = OMX_COLOR_FormatUnused</p>
	OMX_IndexParamVideoPortFormat	r/w	<p>Specify/query the video format.</p> <p>eCompressionFormat = OMX_IMAGE_CodingWEBP</p>

			eColorFormat = <i>OMX_COLOR_FormatUnused</i>
--	--	--	-------------------------------------------------

6 References

- [1] **VP8 Video Bitstream Guide**. Revised February 4, 2011.
(<http://www.webmproject.org/media/pdf/vp8-bitstream.pdf>)
- [2] **WebP Image Format** (<http://code.google.com/speed/webp/>)
- [3] **WebM Project** (<http://www.webmproject.org/>)