OpenMAX™ Integration Layer Extension

NAL Unit Packaging

Version 1.0.0

June 2, 2010
Document version 1.0.0.0

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.

Contents

1 Overview

1.1 Introduction

This extension defines new functionality to support various formats of the Network Abstraction Layer (NAL) unit (NALU).

The ITU H.264\AVC specification specifies a provision for packaging video content into NALUs, a set of NAL units representing a coded picture.

These NALUs may be affixed with or without a Start Code Prefix (a unique byte sequence identifying the start of each NALU (ITU-T H.264\ISO 14496-10 Annex B Specification). In the case of NALUs without Start Codes, a different mechanism needs to be available to identify the start of NALUs, specifically for the case where multiple NALUs are contained in a single buffer.

There are no options available within the OpenMAX IL 1.1.2 specification for identifying the packaging options.

This standard extension introduces functionality that defines the various packaging formats and the ability to select the required format

1.2 Dependency

This extension is written against the wording of:

OpenMAX IL 1.1.2 Specification

Document Version 1.1.2.0

September 1, 2008

1.3 Extension Definitions

1.3.1 Index Definitions

Table 1: Extension Indices

OpenMAX IL Indices (OMX_IndexExt.h)	**Corresponding OpenMAX IL Structures**
`OMX_IndexParamNALStreamFormatSupported`	`OMX_NALSTREAMFORMATTYPE`
`OMX_IndexParamNALStreamFormat`	`OMX_NALSTREAMFORMATTYPE`
`OMX_IndexParamNALStreamFormatSelect`	`OMX_NALSTREAMFORMATTYPE`

1.3.2 Data Structure Definitions

4.3.xx OMX_NALSTREAMFORMATTYPE

The OMX_NALSTREAMFORMATTYPE structure is used to specify the NAL unit format and its associated size.

OMX_NALSTREAMFORMATTYPE is defined as follows.

```
typedef struct OMX_NALSTREAMFORMATTYPE{
    OMX_U32  nSize;
    OMX_VERSIONTYPE nVersion;
    OMX_U32  nPortIndex;
    OMX_NALUFORMATSTYPE  eNaluFormat;
} OMX_NALSTREAMFORMATTYPE;
```

4.3.xx.1 Parameters

The parameters for OMX_NALSTREAMFORMATTYPE are defined as follows.

- nPortIndex is the value containing the index of the port.
- eNaluFormat indicates the format of the NALU. Refer to Table 1 for a listing of the various formats. This parameter contains bit-mapped values as defined by Table 1.

  The default mode of operation shall be OMX_NaluFormatStartCodes.

NALU Format	**Description**
OMX_NaluFormatStartCodes	NALUs separated by Start Codes (ITU-T H.264\ISO 14496-10 Annex B)
OMX_NaluFormatOneNaluPerBuffer	One NALU per buffer. Multiple NALUs in the same buffer are forbidden
OMX_NaluFormatOneByteInterleavedLength	NALU separated by 1-byte interleaved length fields
OMX_NaluFormatTwoByteInterleavedLength	NALU separated by 2-byte interleaved length fields
OMX_NaluFormatFourByteInterleavedLength	NALU separated by 4-byte interleaved length fields

Table 1: NALU Formats

Payload Packaging Options for cases when Start Codes are not in use:

A buffer containing a single NAL unit appears as:

<NAL Size X bytes><NAL unit>

A buffer containing multiple NALs unit appears as:

<NAL Size X bytes><NAL unit><NAL Size X bytes><NAL unit>

<NAL Size X Bytes> is the number of bytes indicating the size of the NAL unit payload

4.3.xx.2 Functionality

In order for an OpenMAX IL Client to properly configure a component graph to consume the stream, it needs to be able to query the components to determine:

- The stream packaging format (Source Component – component emitting the NALU payload)
- The stream formats supported by the component consuming the NALU payload.

The determination of the NALU formatting shall be queried via the source components that will be emitting the stream content, for example Demuxer Components. These components though will only have access to this formatting information when it has been given the opportunity to parse the source content, typically achieved when in OMX_StateExecuting state. Utilizing the auto-detection support, the IL Client will be able to query this information after the component issues the OMX_EventPortFormatDetected event.

The IL Client shall use OMX_GetParam(`OMX_IndexParamNALStreamFormat`) on the source component's output port to query the native NALU packaging format within the embedded stream.

The IL Client shall use OMX_GetParam(`OMX_IndexParamNALStreamFormatSupported`) on the source component's output port to query the NALU packaging formats supported – the nNaluFormat parameter shall return all the formats supported or'ed together.

The IL Client shall use OMX_GetParam(`OMX_IndexParamNALStreamFormatSupported`) on the consumer component's input port to query the NALU packaging format supported – the nNaluFormat parameter shall return all the formats supported or'ed together.

In the case where a consumer component's input port does not support the NAL stream format selection, the responsibility of formatting the stream payload appropriate reverts to the source component. A source component shall support the ability to emit the NALU payload in either configurable option, however it is not mandated that the component shall support the ability to package multiple NALUs within a single buffer – although this is highly recommended.

Note: Configuring a source component to format the NALU payload in a format that is non-native to the stream's embedded format may incur a performance penalty.

The IL Client shall use OMX_SetParam(`OMX_IndexParamNALStreamFormatSelect`) on a source component's output port to configure it to the appropriate setting.

In the case where a consumer component's input port is capable of supporting the native NALU packaging format within the embedded stream but differs from the default OMX_NaluFormatStartCodes mode, the IL Client may alternatively configure the consumer component's input port instead of the source component's output port to consume the stream. The IL Client shall use OMX_SetParam(`OMX_IndexParamNALStreamFormatSelect`) on a consumer component's input port to configure it to the appropriate setting.

4.3.xx.3 Call Sequence Examples

This section provides various examples that may be encountered.

Figure 1 shows the case when a Video Decoder supports the NALU configuration support within the embedded stream.

Figure 2 shows the case when a Video Decoder does not support the NALU configuration within the stream, the IL Client configures the Demuxer to emit a format supported by the Video Decoder (e.g. NALU using Start Code - ITU-T H.264\ISO 14496-10 Annex B Specification).

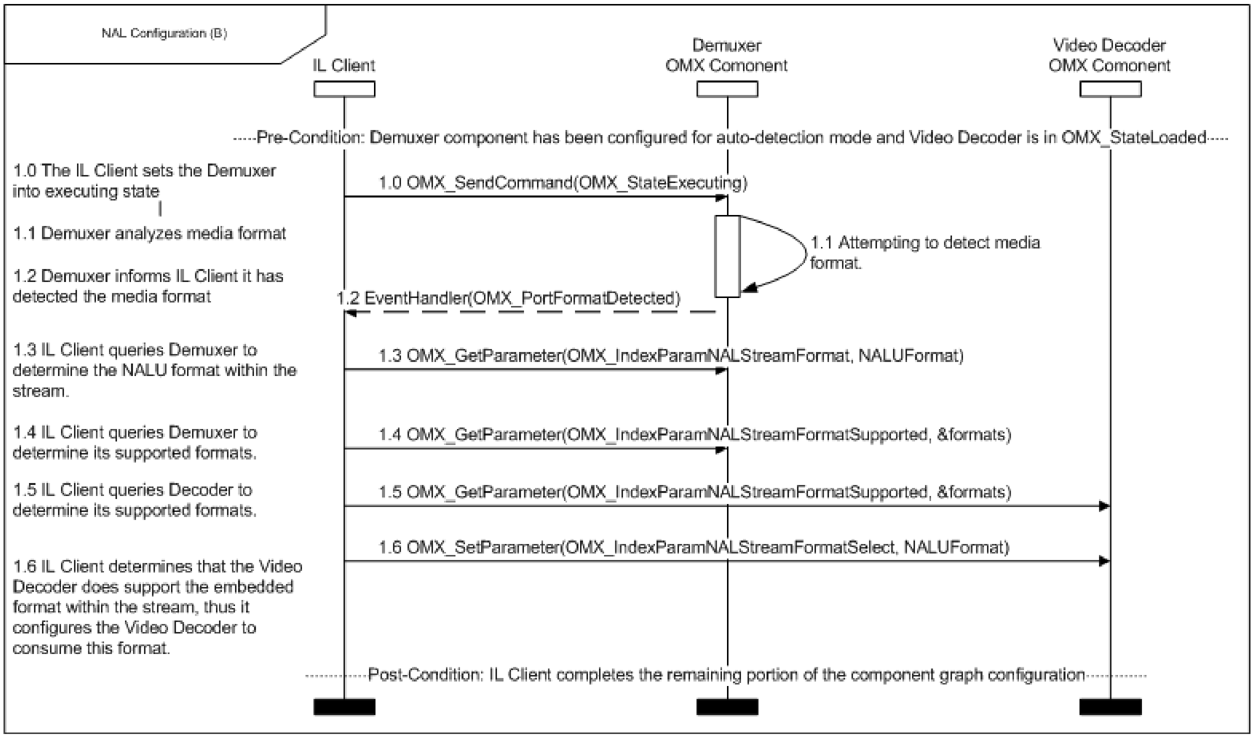
NAL Configuration (B)
IL Client
Demuxer OMX Comonent
Video Decoder OMX Comonent
Pre-Condition: Demuxer component has been configured for auto-detection mode and Video Decoder is in OMX_StateLoaded
1.0 The IL Client sets the Demuxer into executing state
1.0 OMX_SendCommand(OMX_StateExecuting)
1.1 Demuxer analyzes media format
1.1 Attempting to detect media format.
1.2 Demuxer informs IL Client it has detected the media format
1.2 EventHandler(OMX_PortFormatDetected)
1.3 IL Client queries Demuxer to determine the NALU format within the stream.
1.3 OMX_GetParameter(OMX_IndexParamNALStreamFormat, NALUFormat)
1.4 IL Client queries Demuxer to determine its supported formats.
1.4 OMX_GetParameter(OMX_IndexParamNALStreamFormatSupported, &formats)
1.5 IL Client queries Decoder to determine its supported formats.
1.5 OMX_GetParameter(OMX_IndexParamNALStreamFormatSupported, &formats)
1.6 OMX_SetParameter(OMX_IndexParamNALStreamFormatSelect, NALUFormat)
1.6 IL Client determines that the Video Decoder does support the embedded format within the stream, thus it configures the Video Decoder to consume this format.
Post-Condition: IL Client completes the remaining portion of the component graph configuration

Figure 1: NALU Formatting Supported By Video Decoder

The sequence starts with a Pre-Condition that the IL Client has configured the output port formats (e.g. OMX_IndexParamVideoPortFormat) of the Demuxer to auto detect.

The IL Client commands the Demuxer component to transition into executing state – Step 1.0.

The Demuxer reads and parses the media content until it is able to detect the media container format – Step 1.1.

The Demuxer component detects the media format and notifies the IL Client via an event callback – Step 1.2.

At the point, the Demuxer component is capable of determining the native NALU stream formatting within the embedded container. The IL Client queries this information from the Demuxer – Step 1.3.

IL Client queries the Demuxer to determine its supported formats (not a required step, shown for completeness) – Step 1.4.

IL Client queries the Video Decoder to determine its supported formats – Step 1.5.

The IL Client determines that the Video Decoder is capable of supporting the format within the stream. The IL Client configures the Video Decoder to consume this format – Step 1.6.

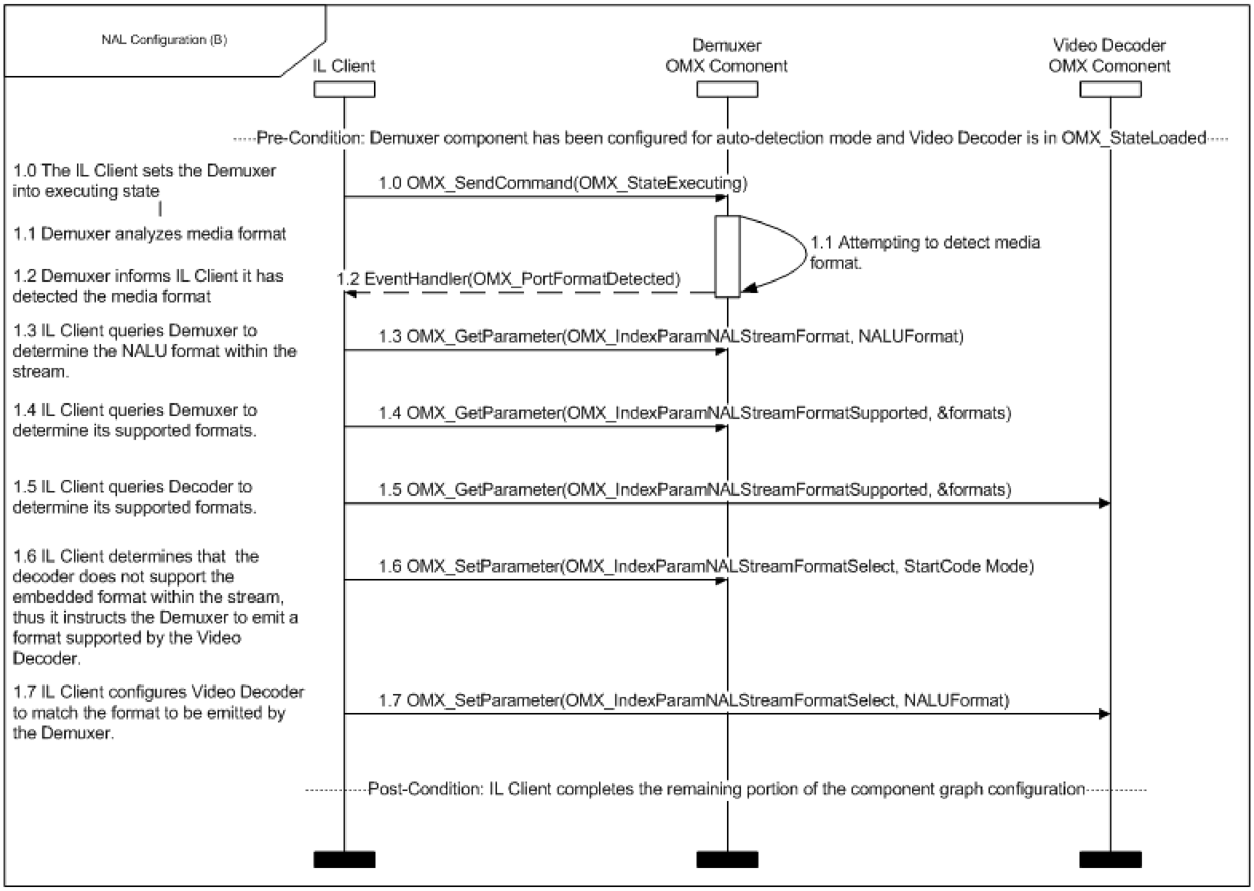
NAL Configuration (B)
IL Client
Demuxer OMX Comonent
Video Decoder OMX Comonent
Pre-Condition: Demuxer component has been configured for auto-detection mode and Video Decoder is in OMX_StateLoaded
1.0 The IL Client sets the Demuxer into executing state
1.0 OMX_SendCommand(OMX_StateExecuting)
1.1 Demuxer analyzes media format
1.1 Attempting to detect media format.
1.2 Demuxer informs IL Client it has detected the media format
1.2 EventHandler(OMX_PortFormatDetected)
1.3 IL Client queries Demuxer to determine the NALU format within the stream.
1.3 OMX_GetParameter(OMX_IndexParamNALStreamFormat, NALUFormat)
1.4 IL Client queries Demuxer to determine its supported formats.
1.4 OMX_GetParameter(OMX_IndexParamNALStreamFormatSupported, &formats)
1.5 IL Client queries Decoder to determine its supported formats.
1.5 OMX_GetParameter(OMX_IndexParamNALStreamFormatSupported, &formats)
1.6 IL Client determines that the decoder does not support the embedded format within the stream, thus it instructs the Demuxer to emit a format supported by the Video Decoder.
1.6 OMX_SetParameter(OMX_IndexParamNALStreamFormatSelect, StartCode Mode)
1.7 IL Client configures Video Decoder to match the format to be emitted by the Demuxer.
1.7 OMX_SetParameter(OMX_IndexParamNALStreamFormatSelect, NALUFormat)
Post-Condition: IL Client completes the remaining portion of the component graph configuration

Figure 2: NALU Formatting Not Supported By Video Decoder

The sequence starts with a Pre-Condition that the IL Client has configured the output port formats (e.g. OMX_IndexParamVideoPortFormat) of the Demuxer to auto detect.

The IL Client commands the Demuxer component to transition into executing state – Step 1.0.

The Demuxer reads and parses the media content until it is able to detect the media container format – Step 1.1.

The Demuxer component detects the media format and notifies the IL Client via an event callback – Step 1.2.

At the point, the Demuxer component is capable of determining the native NALU stream formatting within the embedded container. The IL Client queries this information from the Demuxer – Step 1.3.

IL Client queries the Demuxer to determine its supported formats (not a required step, shown for completeness) – Step 1.4.

IL Client queries the Video Decoder to determine its supported formats – Step 1.5.

The IL Client determines that the Video Decoder does not support the format within the stream. The IL Client configures the Demuxer to emit a format supported by the Video Decoder – Step 1.6.

The IL Client configures the Video Decoder to consume the format to be emitted by the Demuxer – Step 1.7.

2 References

[1] RTP Payload Format for H.264 Video, RFC 3984
http://www.ietf.org/rfc/rfc3984.txt?number=3984