OpenMAX™ Integration Layer Extension

Extended Callback Events

Version 1.0.0

May 18, 2010
Document version 1.0.0.0

Contents

1 Overview

1.1 Introduction

OpenMAX IL standard currently defines nine types of standard events:

- `OMX_EventCmdComplete`
- `OMX_EventError`
- `OMX_EventMark`
- `OMX_EventPortSettingsChanged`
- `OMX_EventBufferFlag`
- `OMX_EventResourcesAcquired`
- `OMX_EventComponentResumed`
- `OMX_EventDynamicResourcesAvailable`
- `OMX_EventPortFormatDetected`

None of the above is suitable for real-time events that signal changes in run-time controls (e.g., auto-focus events). Specifically, `OMX_EventPortSettingsChanged` is only described in the standard for situations that require the client to disable the corresponding port before taking any other action. Additionally, while the IL event enumerated type `OMX_EVENTTYPE` has a dedicated range for vendor extensions, IL clients have no way of knowing beforehand, if a component supports a given extension event.

This extension provides a solution by defining one new event type for signaling a change in any IL parameter or config.

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

The extension consists of three parts: a new event type, a new config index for enabling callback events, and a data structure to be used with the new index.

1.3.1 Event definition

This extension introduces a new event type `OMX_EventIndexSettingChanged` that can be used for signaling changes in IL params, configs, and vendor extensions. When receiving this event the event handler parameters are

`eEvent` (in OMX_CoreExt.h)	`nData1`	`nData2`	`pEventData`
`OMX_EventIndexSettingChanged`	Port index (may also be `OMX_ALL`)	Param or config index	`NULL`

This event signals a change in the parameter or config with index '`nData2`' in component port '`nData1`'. Naturally, the index in `nData2` may be an index that is specific to a regular vendor extension as well as one of the standard IL indices.

IL parameter and config data structures typically have multiple fields of parameters. If a component supports `OMX_EventIndexSettingChanged` for a particular index, it shall send the event to the IL client each time at least one of the fields the in corresponding struct changes value.

When receiving the event, the client should call `OMX_GetParameter()` or `OMX_GetConfig()` as appropriate to get the new settings.

1.3.2 Index definition

To avoid backwards compatibility problems, callbacks are always disabled when a component is first created. To enable or disable callbacks during the component lifecycle, the IL client uses a new extension index `OMX_IndexConfigCallbackRequest`, with a corresponding data structure `OMX_CONFIG_CALLBACKREQUESTTYPE`.

OpenMAX IL Index (in OMX_IndexExt.h)	**Corresponding OpenMAX IL Structures**
`OMX_IndexConfigCallbackRequest`	`OMX_CONFIG_CALLBACKREQUESTTYPE`

1.3.3 Data Structure Definition

`OMX_CONFIG_CALLBACKREQUESTTYPE` structure is used for enabling and disabling callbacks using the `OMX_EventIndexSettingChanged` event.

`OMX_CONFIG_CALLBACKREQUESTTYPE` is defined in OMX_CoreExt.h as follows.

```
typedef struct OMX_CONFIG_CALLBACKREQUESTTYPE {
    OMX_U32 nSize;
    OMX_VERSIONTYPE nVersion;
    OMX_U32 nPortIndex;
    OMX_INDEXTYPE nIndex;
    OMX_BOOL bEnable;
} OMX_CONFIG_CALLBACKREQUESTTYPE;
```

1.3.3.1 Parameters

The parameters for `OMX_CONFIG_CALLBACKREQUESTTYPE` are defined as follows.

- `nPortIndex` is the port the setting applies to (can be `OMX_ALL`).
- `nIndex` is the index the client changes the callback setting for.
- `bEnable` either enables (`OMX_TRUE`) or disables (`OMX_FALSE`) the callbacks; the default value is `OMX_FALSE`.

1.3.3.2 Functionality

Callback settings can be changed and queried using the standard `OMX_SetConfig()` and `OMX_GetConfig()` functions, respectively. Callback settings are fully independent of any other settings on the component, including component state.

Component implementations shall fail the `OMX_SetConfig()` call for `OMX_IndexConfigCallbackRequest` with `OMX_ErrorUnsupportedIndex` if it does not support callbacks at all, and with `OMX_ErrorUnsupportedSetting` if it does not support callbacks for the particular index, but supports callbacks for some other settings.