OpenMAX AL™ is an application-level multimedia playback and recording API for mobile embedded devices. It provides a device-independent, cross-platform interface for applications to access a device’s audio, video and imaging capabilities. OpenMAX AL is suitable for mobile embedded devices, including basic mobile phones and smart phones, PDAs, mobile digital music players, and other sophisticated media playback and recording devices.

### Object-Interface Mapping Table

This table describes the object-interface mapping and mandated objects per profile.

- The top row shows whether objects are mandated or optional in the profiles.
- The second row lists the objects available in OpenMAX AL.
- The left column shows the OpenMAX AL interfaces.
- The center columns indicate the object-interface mapping.
- The right column shows analogous interfaces in OpenSL ES when applicable.

#### Legend for Object-Interface Mapping Table

- **MP** Object mandated in (MP) Media Player or (MR) Media Player/Recorder profile.
- **MR** Object optional in (MP) Media Player or (MR) Media Player/Recorder profile.
- **[ ]** Applicable optional interface.
- **[ ]** Implicit and mandated interface.
- **[ ]** Mandated (explicit) interface.

- `n` Mandated (explicit) interface with optional methods, mandated only for...
  - 1 = time-based media content stored locally;
  - 2 = use cases with audio or video;
  - 3 = use cases with image;
  - 4 = use cases with audio.

#### Profiles

An OpenMAX AL profile is a defined subset of features of the same functional type collectively required on any implementation that claims to support that profile.

**Media Player:**  
This profile encapsulates media playback functionality including the ability to render audio, video and image data in one or more formats. This profile is appropriate for playback-only devices which do not include any support for capturing or recording media. Personal media players are good examples of devices that would use this profile.

**Media Player/Recorder:**  
This profile encapsulates all-inclusive media playback and recording functionality including the ability to capture as well as render audio, video and image data in one or more formats. High-end mobile phones are good examples of devices that would use this profile. This profile subsumes the Media Player profile.
### Functions

**xaCreateEngine** [6.1]

Initializes the engine object and gives the user a handle.

- **pEngine**
  - Pointer to the resulting engine object.
- **numOptions**
  - Number of elements in the options array.
- **pEngineOptions**
  - Array of optional configuration data.
- **numInterfaces**
  - Number of interfaces that the object is requested to support (not including implicit interfaces).
- **pinterfaceIds**
  - An array of numInterfaces interface IDs, which the object should support.
- **pInterfaceRequired**
  - Array of numInterfaces flags, each specifying whether the respective interface is required on the object or optional.

**xaQueryNumSupportedEngineInterfaces()** [6.2]

Queries the number of interfaces on an object.

- **pNumSupportedInterfaces**
  - Identifies the number of supported interfaces available.

**xaQuerySupportedEngineInterfaces()** [6.3]

Queries the number of supported interfaces on an object engine.

- **index**
  - Index used to enumerate available supported interfaces.
- **pInterfaceId**
  - Identifies the supported interface corresponding to the given index.

### Interfaces

**XAudioDecoderCapabilitiesItf** [8.2]

Queries the audio decode capabilities.

- **Object**: Engine

**GetAudioDecoders**

Retrieves available audio decoders.

**GetAudioDecoderCapabilities**

Queries for audio decoder capabilities.

**XAudioEncoderItf** [8.3]

Sets audio encoder capabilities of audio engine.

- **Object**: Media Recorder

**SetEncoderSettings**

Sets audio encoder parameters.

**GetEncoderSettings**

Get audio encoder settings.

**XAudioEncoderCapabilitiesItf** [8.4]

Queries audio encoding capabilities of audio engine.

- **Object**: Media Recorder

**GetAudioEncoders**

Queries supported audio encoders.

**GetAudioEncoderCapabilities**

Queries for the audio encoder’s capabilities.

**XAudioIODeviceCapabilitiesItf** [8.5]

Enumerates audio I/O devices and query capabilities of each available audio I/O device.

- **Object**: Engine

**GetAvailableAudioInputs**

Gets number and IDs of audio input devices.

**QueryAudioInputCapabilities**

Gets capabilities of specified audio input device.

**RegisterAvailableAudioInputsChangedCallback**

Sets/clears xAvailableAudioInputsChangedCallback() callback.

**GetAvailableAudioOutputs**

Gets number and IDs of audio output devices.

**QueryAudioOutputCapabilities**

Gets the output capabilities.

**RegisterAvailableAudioOutputsChangedCallback**

Sets/clears xAvailableAudioOutputsChangedCallback() callback.

**RegisterDefaultDeviceIDMapChangedCallback**

Sets/clears xDefaultDeviceIDMapChangedCallback() callback.

**GetAssociatedAudioInputs**

Returns array of audio input devices physically associated with this I/O device.

**GetAssociatedAudioOutputs**

Returns array of audio output devices physically associated with this I/O device.

**GetDefaultAudioDevices**

Gets the number of audio devices currently mapped to the given default device ID.

**QuerySampleFormatSupported**

Gets an array of sample formats supported by the audio I/O device for the given sampling rate.

**XCameraItf** [8.6]

Queries and configures camera I/O device.

- **Object**: Camera

**RegisterCallback**

Sets callback for camera event notifications.

**SetFlashMode**

Sets the camera flash setting.

**XCameraCapabilitiesItf** [8.7]

Queries the camera I/O device capabilities.

- **Object**: Camera

**GetCameraCapabilities**

Queries the device capabilities.

**GetCameraRegionPatterns**

Queries focus region patterns.

**GetSupportedFocusManualSettings**

Gets the supported manual focus settings.

**GetSupportedISOOverlaySettings**

Queries the supported ISO overlay settings.

**GetSupportedApertureManualSettings**

Gets the supported manual aperture settings.

**GetSupportedShutterSpeedManualSettings**

Gets the supported manual shutter speeds.

**GetSupportedWhiteBalanceManualSettings**

Gets supported manual white balance settings.

**XCameraConfigItf** [8.6]

Queries and configures camera I/O device.

- **Object**: Camera

**RegisterCallback**

Sets callback for camera event notifications.

**SetFlashMode**

Sets the camera flash setting.

**XCameraConfigExtensionsItf** [8.8]

Sets and queries codec and non-codec configurations of the underlying media engine.

- **Object**: All

**SetConfiguration**

Sets configuration as key-value pair.

**GetConfiguration**

Gets config. setting as key-value pair.

**XDeviceVolumeItf** [8.9]

Manipulates I/O device-specific volumes.

- **Object**: Engine

**GetVolumeScale**

Gets the supported volume scale properties.

**SetVolume**

Sets the volume.

**GetVolume**

Gets the volume.

**XDynamicInterfaceManagementItf** [8.10]

Manages interface exposure on a realized object.

- **Object**: All

**AddInterface**

Exposes an interface on a realized object.

**RemoveInterface**

Removes dynamically exposed interface.

**ResumeInterface**

Resumes dynamically exposed interface.

**RegisterCallback**

Registers callback for an interface’s events.

**XDynamicSourceItf** [8.11]

Deprecated. Instead use XDynamicSourceSinkChangeItf.

**XDynamicSourceSinkChangeItf** [8.12]

Changes data source or sink during object lifetime.

- **Objects**: Media Player, Media Recorder, Metadata Extractor

**ChangeSource**

Changes a specified data source.

**ChangeSink**

Changes a specified data sink.

**RegisterSourceChangeCallback**

Sets or clears xSourceChangeCallback.

**RegisterSinkChangeCallback**

Sets or clears xSinkChangeCallback.

**XEngineItf** [8.13]

Exposes creation methods of all object types.

- **Object**: Engine

**CreateCameraDevice**

Creates a camera device.

**CreateRadioDevice**

Creates a radio device.

**CreateLEDDevice**

Creates an LED device.

**CreateVibraDevice**

Creates a vibrator device.

**CreateMediaPlayer**

Creates a media player.

**CreateMediaRecorder**

Creates a media recorder.

**CreateOutputMix**

Creates an output mix.

**CreateMetadataExtractor**

Creates a Metadata Extractor.

**CreateExtensionObject**

Creates an externally defined extension object.

**GetImplementationInfo**

Queries the OpenMAX AL implementation information.

**QuerySupportedProfiles**

Queries supported profiles.

**QueryNumSupportedInterfaces**

Queries number of supported interfaces.

**QuerySupportedInterfaces**

Queries supported interfaces.

**QueryNumSupportedExtensions**

Queries number of supported extensions.

**QuerySupportedExtension**

Gets extension name by index.
Interfaces (continued)

**XAEengineItf** [8.14]

Manipulates equalizer settings.

- **Objects**: Media Player, Media Recorder, Camera
- **SetEnabled** Enables the effect.
- **IsEnabled** Gets the enabled status of effect.
- **GetNumberOfBands** Gets number of frequency bands.
- **GetBandLevelRange** Returns the min/max band levels.
- **SetBandLevel** Sets a band's gain level.
- **GetBandLevel** Gets a band's gain level.
- **GetCenterFreq** Gets a band's center frequency.
- **GetBandFreqRange** Gets a band's frequency range.
- **GetBand** Gets the band that affects a frequency the most.
- **GetCurrentPreset** Gets the current preset.
- **UsePreset** Sets equalizer according to the given preset.
- **GetNumberOfPresets** Gets number of presets supported.
- **GetPresetName** Gets preset name based on index.

**XAEImageControlItf** [8.15]

Adjusts image or video content.

- **Objects**: Media Player, Media Recorder, Camera
- **SetBrightness** Sets the brightness level.
- **GetBrightness** Gets the brightness level.
- **SetContrast** Sets the contrast level.
- **GetContrast** Gets the contrast level.
- **SetGamma** Sets the gamma level.
- **GetGamma** Gets the gamma level.
- **SetImageSettings** Disables an image effect.
- **GetImageSettings** Queries if extension is supported.
- **QueryColorFormats** Queries supported color formats.
- **GetSupportedGammaSettings** Gets the current preset.
- **GetSupportedKeysCount** Queries number of fixed keys or encodings.
- **GetKeySize** Returns metadata byte size by index.
- **GetKey** Returns a XAMetadataInfo structure & associated data referenced by the structure for a supported key.
- **SetFreeKeysEncoding** Sets or clears the metadata encoding.
- **InsertMetadataItem** Inserts/overwrites metadata.
- **GetMetadataString** Returns the metadata string.
- **GetState** Retries the current object state.
- **Resume** Transitions Unrealized to Realized state.
- **Realize** Transitions into given play state.
- **GetFreqRangeProperties** Sets position of playback marker.
- **GetMarkerPosition** Clears marker.
- **GetNumberOfItf** [8.14] Returns the state of each LED.
- **GetColor** Returns color of an individual LED.

**XAEObjectItf** [8.17]

Manages image and video effects.

- **Objects**: Media Player, Media Recorder, Camera
- **SetEnabled** Enables the effect.
- **IsEnabled** Gets the enabled status of effect.
- **GetNumberOfBands** Gets number of frequency bands.
- **GetBandLevelRange** Returns the min/max band levels.
- **SetBandLevel** Sets a band's gain level.
- **GetBandLevel** Gets a band's gain level.
- **GetCenterFreq** Gets a band's center frequency.
- **GetBandFreqRange** Gets a band's frequency range.
- **GetBand** Gets the band that affects a frequency the most.
- **GetCurrentPreset** Gets the current preset.
- **UsePreset** Sets equalizer according to the given preset.
- **GetNumberOfPresets** Gets number of presets supported.
- **GetPresetName** Gets preset name based on index.
Interfaces (continued)

XARadiotif [8.31]

Accesses RDS and RBDS features.

Object: Radio

QueryRDS Signal Returns the RDS reception status.
GetProgrammeServiceName Gets the current Programme Service name (PS).
GetRadioText Gets the current Radio Text (RT).
GetProgrammeType Gets the current Programme Type (PTY) as a number.
GetProgrammeTypeString Gets the current PTY as a String.
GetProgrammeIdentificationCode Gets the current Programme Identification Code (PI).
GetClockTime Gets current Clock Time & date (CT).
GetTrafficAnnouncement Gets the status of Traffic Announcement (TA).
GetTrafficProgramme Gets the status of the Traffic Programme (TP) switch.
SeekByProgrammeType Seeks for a given PTY.
SeekTrafficAnnouncement Seeks for a TA.
SeekTrafficProgramme Seeks for a TP.
SetAutomaticSwitching Sets/clears the Automatic Switching (AS) switch.
GetAutomaticSwitching Gets the Automatic Switching (AS) state.
SetAutomaticTrafficAnnouncement Enables/disables TA automatic switching.
GetAutomaticTrafficAnnouncement Gets TA automatic switching state.
GetODA Group Gets ODA data by async callback.
SubscribeODAGroup Subscribes the given ODA group.
UnsubscribeODAGroup Unsubscribes the given ODA group.
ListODAGroupSubscriptions Gets subscribed ODA groups.
RegisterODADebugRegisterSets clears the xaaRadioCallback().
RegisterRDSItfSets/clears the xaaRadioCallback().

XARecordItf [8.33]

Manages playback position and looping.

Object: Media Recorder

SetPosition Sets the position of the playback head.
SetLoop Sets loop parameters.
GetLoop Query loop parameters.
InitiateSnapshot Sets parameters prior to TakeSnapshot().
TakeSnapshot Async take and store snapshot(s).
CancelSnapshot Cancels an ongoing shooting session.
ReleaseBuffers Releases a buffer.
GetMaxPicsPer Burst Gets the max. pictures per burst.
GetBufFPS Range Gets possible min/max burst rates.
SetShutterFeedback Enables/disables shutter feedback.
GetShutterFeedback Queries if shutter feedback enabled.

XASnapshotItf [8.34]

Controls a camera device.

Object: Media Recorder

QueryMediaContainerInformation Queries media container information.
QueryStreamType Queries the stream domain.
QueryStreamsInformation Queries information about the stream.
QueryStreamName Queries the stream name.
RegisterStreamName ChangeCallback Callback for stream change events.
QueryActiveStreams Returns the active state for all streams.
SetActiveStream Sets the active state for all streams.
SetDestinationRectangle Defines destination rectangle.
SetSourceRectangle Defines source rectangle.
SetScaleOptions Sets the scaling options.
SetMirror Sets the mirroring options.
Commit Commits changes since last Commit().

XASubscriptionItf [8.35]

Queries a stream's properties.

Objects: Media Player, Metadata Extractor

QueryMediaContainerInformation Queries media container information.
QueryStreamType Queries the stream domain.
QueryStreamsInformation Queries information about the stream.
QueryStreamName Queries the stream name.
RegisterStreamName ChangeCallback Callback for stream change events.
QueryActiveStreams Returns the active state for all streams.
SetActiveStream Sets the active state for all streams.
SetDestinationRectangle Defines destination rectangle.
SetSourceRectangle Defines source rectangle.
SetScaleOptions Sets the scaling options.
SetMirror Sets the mirroring options.
Commit Commits changes since last Commit().

XAVolumeItf [8.42]

Manages audio volume of the object.

Objects: Media Player, Media Recorder, Output Mix

SetVolumeLevel Sets the volume level.
GetVolumeLevel Gets the volume level.
GetMaxVolumeLevel Gets maximum supported level.
SetMute Mutes or unmutes object.
GetMute Gets the mute state.
EnableStereoPosition Enables/disables stereo positioning effect.
IsEnabledStereoPosition Returns the stereo positioning enabled state.
SetStereoPosition Sets stereo position (pan/balance).
GetStereoPosition Gets stereo position setting.

Structures

XAAudioCodeDescriptor [9.1.1]

Audio codec capabilities.

XAAudioEncoderSettings [9.1.2]

Set the audio encoding parameters.

XAAudioInputDescriptor [9.1.3]

Return the description of audio input device capabilities.

XAAudioOutputDescriptor [9.1.4]

Return the description of audio output device capabilities.

XAAudioStreamInformation [9.1.5]

Audio stream information.

XACameraDescriptor [9.1.6]

Query the camera capabilities.

XADeviceFormat_PCM [9.1.8]

Deprecated. Instead use XADeviceFormat_PCM_EX.

XADeviceFormat_PCM_EX [9.1.9]

Describes audio PCM parameters.

XADeviceFormat_RawImage [9.1.10]

Describes the raw image format.

XADataFormat_MIME [9.1.7]

Describes a MIME type.

XADataFormat_PCM [9.1.8]

Deprecated. Instead use XADeviceFormat_PCM_EX.
Structures (continued)

XADataSink [9.1.19]
A data sink by locator and format.

XADataSource [9.1.20]
A data source by locator and format.

XAMetadataInfo [9.1.30]
A key or a value from a metadata item key/value pair.

RealAudio Profiles and Levels
XA_AUDIOPROFILE_REALAUDIO
XA_AUDIOMODE_REALAUDIO|(G2, 8, 10, SURROUND)

Vorbis Profiles and Levels
XA_AUDIOPROFILE_VORBIS, XA_AUDIOMODE_VORBIS

XA_BOOLEAN [9.2.5]
Canonical values for Boolean type.
FALSE, TRUE

XA_BYTEORDER [*] [9.2.6]
The byte order of 16-, 32-, or 64-bit data.
BIGENDIAN, LITTLEENDIAN, NATIVE

XA_CAMERA_FOCUSMODESTATUS_* [9.2.14]
Camera focus mode status.
OFF, REQUEST, REACHED, UNABLEToreach, LOST

XA_CAMERA_FOCUSMODE_* [9.2.13]
Camera focus mode.
MANUAL, AUTO, CENTROID, CONTINUOUS_AUTO, CONTINUOUS_CENTROID

XA_CAMERA_FOCUSMODESTATUS_* [9.2.15]
Camera iso sensitivity.
MANUAL, AUTO

XA_CAMERA_LOCK_* [9.2.16]
Locks for camera settings.
AUTOFOCUS, AUTOEXPOSURE, AUTOWHITEBALANCE

XA_CAMERA_METERINGMODE_* [9.2.17]
Camera metering mode.
AVERAGE, SPOT, MATRIX

XA_CAMERA_WHITEBALANCEMODE_* [9.2.18]
White balance settings.
AUTOMATIC, INCANDESCENT, FLUORESCENT, NORMAL, SLOW, FASTEST

XA_CAMERA_WHITEBALANCEMODE_* [9.2.19]
White balance settings.
AUTOMATIC, INCANDESCENT, FLUORESCENT, NORMAL, SLOW, FASTEST

XACameraItf::SetZoom

Macros

XA_API [9.2.1]
A platform-specific macro to declare OpenMAX AL function prototypes.

XAAPIENTRY [9.2.2]
A system-dependent API entry point macro to indicate the required calling conventions for global functions.

XA_AUDIOCODEC_* [9.2.3]
The audio encoding type.
PCM, MP3, AMR, AMRWB, AMRWBPLUS, AAC, WMA, REAL, VORBIS

XADATASTREAMFORMAT_* and XA_AUDIOMODE_* [9.2.4]
Audio profiles and modes.
XADATASTREAMFORMAT_UNDEFINED

PC M Profiles and Modes
XA_AUDIOPROFILE_PCM

MP3 Profiles and Modes
XA_AUDIOPROFILE_MP3{1, 2, 25, 13},
XA_AUDIOCHANN ED_MP3_MONO,
XA_AUDIOCHANN ED_MP3_STEREO,
XA_AUDIOCHANN ED_MP3_JOINTSTereo,
XA_AUDIOCHANN ED_MP3_DUAL

AMR Profiles and Modes
XA_AUDIOPROFILE_AMR

XRTPProfiles and Modes
XA_AUDIOPROFILE_RTP_PAYLOAD

AMR-WB Profiles and Modes
XA_AUDIOPROFILE_AMRWB

AACProfiles and Modes
XA_AUDIOPROFILE_AAC_AAC

Windows Media Audio Profiles and Modes
XA_AUDIOPROFILE_WMAM(7, 8, 9, 10)
XA_AUDIOPROFILE_WMALVE(1, 2, 3, 4)
XA_AUDIOPROFILE_WMALVE0(1, 2, 3)

XAEngineOption [9.1.21]
Engine creation options.

XAFocusPointPosition [9.1.22]
Camera focus region.

XAImageCodecDescriptor [9.1.24]
Image codec capabilities.

XAlmageSettings [9.1.25]
Image encoding parameters.

XAlmageStreamInformation [9.1.26]
Image stream information.

XInterfaceID [9.1.27]
The interface ID type.

XALEDDescriptor [9.1.28]
Represents the capabilities of the LED array I/O Device.

XAMediaContainerInformation [9.1.29]
Media container information.

XMDataSourceInfo [9.1.30]
A data source by locator and format.

XAMIDISStreamInformation [9.1.31]
MIDI stream information.

XANativeHandle [9.1.32]
Opaque handle to a display or window.

XARectangle [9.1.33]
Specifies a rectangle.

XATimedTextStreamInformation [9.1.34]
Timed text stream information.

XAVendorStreamInformation [9.1.35]
Vendor-specific stream information.

XAVideoCodecDescriptor [9.1.37]
Video codec capabilities.

XAVideoSettings [9.1.38]
Video encoding parameters.

XAVideoStreamInformation [9.1.39]
Video stream information.

XAVideoStreamInformation [9.1.39]
Vendor-specific stream information.

©2011 Khronos Group - Rev. 0111
www.khronos.org/openmax
Macros (continued)

XA_COLORFORMAT_* [9.2.23]
The data source or sink container type.

XA_DATAFORMAT_[9.2.24]
The possible data formats.

MIME, RAWIMAGE, PCM_EX

XA_DATALOCATOR_[9.2.25]
The possible data locators.

NULL, URI, ADDRESS, IODEVICE, OUTPUTMIX, NATIVEDisplay, RESERVED6, RESERVED7, MEDIAOBDIENGINE, CONTENTYPE

XA_DEFAULTDEVICEID_* [9.2.26]
Default device IDs.

AUDIOPROPERTY, AUDIOPROPERTY, LED, VIBRA, CAMERA

XA_DEVICECONNECTIO_[9.2.27]
Types of I/O device connections.

INTEGRATED, ATTACHED_ [WIRE, WIRELESS, NETWORK

XA_DEVICELOCATION_* [9.2.28]
I/O device locations.

HANDSET, HEADSET, CARKIT, DOCK, REMOTE

XA_DEVICESCOPE_* [9.2.29]
I/O device scope.

UNKNOW, ENVIRONMENT, USER

XA_DOMAINTYPE_* [9.2.30]
Functional domain.

AUDIO, VIDEO, IMAGE, TIMEDTEXT, MIDI, VENDOR, UNKNOWN

XA_DYNAMIC_IFEVENT_* [9.2.31]
Dynamic interface events.

RUNTIME_ERROR, ASYNC_TERMINATION, RESOURCES_AVALAIBLE, LOVE, LOSE, PERMANENTLY

XA_ENGINEOPTION_* [9.2.32]
Engine object creation options (see xACreateEngine()).

THREADSAFE, LOSSOFCONTROL

XA_EQUALIZER [9.2.33]
Undefined equalizer setting.

XEQUALIZER_UNDEINIFIED

XA_FOCUSPOINTS_* [9.2.34]
Camera focus point pattern.

ONE, THREE_3X1, FIVE_CROSS3, SEVEN_CROSS, NINE_3X3, ELEVEN,CROSS, TWELVE_3X4, TWELVE_3X6, SIXTEEN_SQUARE, CUSTOM

XA_FREQUENCY_ [9.2.35]
Frequency range and modulation.

FMUERDAMERICA, FMJAPAN, AMLW, AMMW, AMSW

XA_IMAGECODEC_* [9.2.36]
Image encoding format.

JPEG, GIF, BMP, PNG, TIFF, RAW

XA_IMAGE EFFECT_* [9.2.37]
The image effect type.

MONOCROME, NEGATIVE, SEPIA, EMBOS, PAINTBRUSH, SOLARIZE, CARTOON

XA_IDEOICE_* [9.2.38]
I/O device sources and sinks.

AUDIOPROPERTY, LEDDARAY, VIBRA, CAMERA, RADIO, AUDIOPROPERTY

XA_METADATA_FILTER_* [9.2.39]
Bit-masks for metadata filtering criteria.

KEY, LANG, ENCODING

XA_METADATAATRAVERSALMODE_* [9.2.40]
Method of traversing metadata.

ALL, NODE

XA_MIDI BANK_* [9.2.41]
MIDI instrument bank(s) used.

DEVICE, CUSTOM

XA_MIDIBANK_INDEX_* [9.2.42]
Value for unknown MIDI stream attribute.

XAMIDI_UNKNOWN

XA_MILLIBEL_* [9.2.43]
Limit values for millibels unit.

MIN, MAX

XA_MILLIBELMAX_* [9.2.44]
Limit value for millibels unit.

XAMILLIBEL_MAX

XA_MILLIMETER_MAX_* [9.2.45]
Limit value for millimeter unit.

XAMILLIMETER_MAX

XA_NODE_PARENT_* [9.2.46]
Used to set the current scope to the node's parent.

XANODE_PARENT

XA_NODESTATE_* [9.2.47]
The type of a node.

UNSPECIFIED, AUDIO, VIDEO, IMAGE

XAOBJECTEVENT_* [9.2.48]
Object event notifications.

RUNITIME_ERROR, ASYNC_TERMINATION, RESOURCES_AVALAIBLE, LOVE, LOSE, PERMANENTLY

ITF_CONTROL_TAKEN, RETURNED, ITF_PARAMETERS_CHANGED

XAOBJECTSTATE_* [9.2.49]
Object states.

UNREALIZED, REALIZED, SUSPENDED

XAOBJECTID_* [9.2.50]
Object type identifiers.

ENGINE, LEDDEVICE, VIBRDEVICE, MEDIAPLAYER, MEDIA EaRECODER, RADIODEVICE, OUTPUTMIX, META DATAEXTRACTOR, CAMERADevice

XAOBJECTORIENTATION_* [9.2.51]
Device orientation relative to the user.

UNKNOWN, OUTWARDS, INWARDS

XAPCLMSAMPLEFORMAT_* [9.2.52]
PCM data type.

SIGNED_INT, UNSIGNED_INT, FLOAT

XAPCMSAMPLEFORMAT_* [9.2.53]
PCM data sample formats.

FIXED_8, FIXED_16, FIXED_20, FIXED_24, FIXED_32, FIXED_64

XAPLAYEVENT_* [9.2.54]
Play events.

HEADATEND, HEADATMARKER, HEADATNEWS, HEADATMOVING, HEADATSTALLED, DURATIONUPDATE

XAPLAYSTATE_* [9.2.55]
Playback state.

STOPPED, PAUSED, PLAYING

XAPREFETCHEVENT_* [9.2.56]
Prefetch related events.

STATUSCHANGE, FILELEVELCHANGE

XAPREFETCHSTATUS_* [9.2.57]
Player's prefetch status.

UNDERFLOW, SUFFICIENTDATA, OVERFLOW

XAPRIORITY_* [9.2.58]
Priority levels.

NONE, LOWEST, VERYLOW, LOW, BELOWNORMAL, NORMAL, ABOVE NORMAL, HIGH, VERYHIGH, HIGHEST

XAPROFILE_* [9.2.59]
The OpenMAX AL API profiles.

MEDIA_PLAYER, MEDIA_PLAYER_RECORDER, PLUS_MIDI

XRADIO EVENT_* [9.2.60]
Events for xaradioSimpleback().

ANTENNA_STATUS_CHANGED, FREQUENCY_CHANGED, FREQUENCY_RANGE_CHANGED, PRESET_CHANGED, SEEK_COMPLETED, STEREO_STATUS_CHANGED, SIGNAL_STRENGTH_CHANGED, SQUELCH_CHANGED, FREQUENCY_ERROR, FREQUENCY_RANGE_ERROR

XRATETIMELIMIT_* [9.2.61]
Rate control mode.

CONSTANTBITRATE, VARIABLEBITRATE

XRATETRACE_* [9.2.62]
Object rate-related properties.

STAGGEREDVIDEO, SMOOTHVIDEO, SILENTAUDIO, STAGGEREDAUDIO, NORTHCAPAUDI0, PITCHCRAUDIO

XRDS_INCREMENT_* [9.2.63]
RDS field change event.

PI, PTY, PS, RT, RT_PLUS, CT, TA, TP, ALARM

XRDS_MODE_* [9.2.64]
RDS Program Type code (PTY).

RDSPTY, NONE, NEWS, CURRENTAFFAIRS, INFORMATION, SPORT, EDUCATION, DRAMA, CULTURE, SCIENCE, VARIETY, POMPUS, ROCKMUSI0, EASILYLISTENING, LIGHTCLASSICAL, SERIOUSCLASSICAL, OTHERMUSI0, WEAPPY, FINANCE, CHILDRENSPROGRAMS, SOCIALAFFAIRS, RELIGION, PHONEIN, TRAVEL, LEISURE, JAZZMUSI0, COUNTRYMUSI0, NATIONALMUSI0, OLDIESMUSI0, FOLKMUSI0, DOCUMENTARY, ALARMTEST, ALARM

XRDSPTY_* [9.2.65]
Object rate-related properties.

STAGGEREDVIDEO, SMOOTHVIDEO, SILENTAUDIO, STAGGEREDAUDI0, NORTHCAPAUDI0, PITCHCRAUDIO
Use Case: Audio and Video Playback [4.7.1]

This illustrates the use of the Media Player object for audio and video playback. The support for this use case is mandated in all profiles.

A Media Player is created using the XAEngineItf interface of the engine object. Upon creation, the Media Player is associated with an Output Mix, created using the XAEngineItf interface, for audio output and with a native display handle for video output. The data source of the Media Player is also set during creation. The data source could be, for example, a URI pointing to a video file in the local file system. The Output Mix is by default associated with the system-dependent default output device.
Use Case: Video Camera [4.7.5]

This illustrates the use of the Media Recorder object for recording and a Media Player for the viewfinder. The support for this use case is mandated only in the Media Player/Recorder profile.

The Media Recorder and Media Player objects are created using the XAEngineItf interface of the engine object. Upon creation, both objects are associated with the same Camera object, created using the XACameraItf interface. The audio data source of the Media Recorder is set to be a microphone (an audio input device). The data sink for the Media Player is a native window or display handle (as it was in the previous video playback use case). The data sink of the Media Recorder can be a URI pointing to a video file in the local file system where the data will be recorded.

Use Case: Still Camera [4.7.6]

The still camera use case is similar to the video camera use case except the Media Recorder exposes different interfaces. The support for this use case is mandated only in the Media Player/Recorder profile.

The Media Recorder object provides the XASnapshotItf interface for still image capture and XAImageEncoderItf for the image encoder settings (instead of the XARecordItf and XAVideoEncoderItf interfaces respectively).

Use Case: Radio Playback [4.7.7]

This illustrates the use of the Media Player object for radio playback. The support for this use case is optional in all profiles since support for Radio I/O device object is optional.

As always, the Media Player is created using the XAEngineItf interface of the engine object. Upon creation, the Media Player is associated with an Output Mix, created using the XAEngineItf interface, for audio output. By default, OpenMAX AL automatically associates the Output Mix with the system-dependent default output device. During the creation, the Radio I/O device, created using the XAEngineItf interface, is set as the data source.

Use Case: Recording Audio [4.7.3]

This illustrates the use of the Media Recorder object for recording audio. The support for this use case is mandated only in the Media Player/Recorder profile.

The Audio Recorder is created using the XAEngineItf interface of the engine object. Upon creation, it is associated with an audio data source, which can be, for example, a microphone (an audio input device). The data sink of the Media Recorder can be a URI pointing to an audio file in the local file system on which the audio will be recorded.

Use Case: Reading Metadata [4.7.8]

A Metadata Extractor object will read the metadata of a media file without allocating resources for audio playback. The support for this use case is mandated in all profiles.

The Metadata Extractor object is created using XAEngineItf interface of the engine object and, upon creation, its data source is set. The data source is typically a URI pointing to a media file in the local file system. However, the Metadata Extractor supports the XADynamicSourceSinkChangeItf interface which can be used to change the data source dynamically. Therefore metadata from multiple files (in series) can be extracted without creating a new Metadata Extractor object for every single file. The XAMetadataExtractionItf and XAMetadataTraversalItf interfaces are used for actually reading and traversing the metadata from a file. The XAMetadataMessageItf interface is used to set callbacks that execute whenever a metadata item is encountered.