OpenSL ES 1.1 Quick Reference Card - Page 1

OpenSL ES™ is a royalty-free, cross-platform, hardware-accelerated audio API tuned for embedded systems. It provides a standardized, high-performance, low-latency method to access audio functionality for developers of native applications on embedded mobile multimedia devices, enabling straightforward cross-platform deployment of hardware and software audio capabilities, reducing implementation effort, and promoting the market for advanced audio.

- **[n]** refers to a section in the OpenSL ES 1.1 Specification at www.khronos.org/opensles
- **[n]** refers to a section for the analogous interface in the OpenMAX AL 1.1 Specification at www.khronos.org/openmax

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 OpenSL ES.
- The left column shows the OpenSL ES interfaces.
- The center columns indicate the object-interface mapping.
- The right column shows analogous interfaces in OpenMAX AL when applicable.

INTERFACE \ OBJECT / PROFILE	Engine			Audio Player			MIDI Player			Audio Recorder			Listener			3D Group			Output Mix			Vibra			LED Array			Metadata Extractor			OpenMAX AL
	P	M	G	P	M	G	P	M	G	P	M	G	P	M	G	P	M	G	P	M	G	P	M	G	P	M	G	P	M	G	
SLObjectItf [8.34]	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	[8.25]
SLDynamicInterfaceManagementItf [8.17]	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	[8.10]
SLEngineItf [8.21]	■	■	■																												[8.13]
SLEngineCapabilitiesItf [8.22]	■	■	■																												
SLThreadSyncItf [8.44]	■	■	■																												[8.36]
SLAudioIODeviceCapabilitiesItf [8.12]	■	■	■																												[8.5]
SLAudioDecoderCapabilitiesItf [8.9]	■	■	■																												[8.2]
SLAudioEncoderCapabilitiesItf [8.11]										■	■	■																			[8.4]
SLConfigExtensionsItf [8.15]	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	[8.8]
SLLEDArrayItf [8.25]																									■	■	■				[8.20]
SLVibraItf [8.45]																						■	■	■							[8.37]
SLPlayItf [8.37]				■	■	■	■	■	■																						[8.27]
SLRecordItf [8.42]										■	■	■																			[8.32]
SLAudioEncoderItf [8.10]										■	■	■																			[8.3]
SLPrefetchStatusItf [8.39]				■	■	■	■	■	■																						[8.29]
SLSeekItf [8.43]				A	■	■	A	■	■																						[8.33]
SLPlaybackRateItf [8.38]				■	■	■	■	■	■																						[8.28]
SLRatePitchItf [8.41]				■	■	1																									
SLPitchItf [8.36]				■	■	■	■	■	■																						
SLVolumeItf [8.48]				B	■	■	B	■	■	■	■	■							B	■	■										[8.42]
SLMuteSoloItf [8.33]				■	■	1																									
SLBufferQueueItf [8.14]						2			2	■	■	■																			
SLMIDIMessageItf [8.29]							■	■	■																						
SLMIDITimeItf [8.32]							C	■	■																						
SLMIDITempoItf [8.31]							■	■	■																						
SLMIDIMuteSoloItf [8.30]							■	■	■																						
SL3DCommitItf [8.2]	■	■	■																												
SL3DGroupingItf [8.4]				■	■	■	■	■	■																						
SL3DHintItf [8.5]				■	■	■	■	■	■							■	■	■													
SL3DLocationItf [8.6]				■	■	■	■	■	■				■	■	■	■	■	■													
SL3DSourceItf [8.8]				■	■	■	■	■	■							■	■	■													
SL3DDopplerItf [8.3]				■	■	■	■	■	■				■	■	■	■	■	■													
SL3DMacroscopicItf [8.7]				■	■	■	■	■	■							■	■	■													
SLEffectSendItf [8.20]				■	■	■	■	■	■																						
SLBassBoostItf [8.13]				■	■	■	■	■	■	■	■	■							■	■	■										
SLEqualizerItf [8.24]				■	■	■	■	■	■	■	■	■							■	■	■										[8.14]
SLPresetReverbItf [8.40]				■	■	■	■	■	■										■	■	■										
SLEnvironmentalReverbItf [8.23]				■	■	■	■	■	■										■	■	■										
SLVirtualizerItf [8.46]				■	■	■	■	■	■										■	■	■										
SLMetadataExtractionItf [8.26]				■	■	■	■	■	■																			■	■	■	[8.21]
SLMetadataMessageItf [8.27]				■	■	■	■	■	■																			■	■	■	[8.23]
SLMetadataTraversalItf [8.28]				■	■	■	■	■	■																			■	■	■	[8.24]
SLVisualizationItf [8.47]				■	■	■	■	■	■	■	■	■							■	■	■										
SLOutputMixItf [8.35]																			■	■	■										[8.26]
SLDynamicSourceChangeItf [8.18] (deprecated)				■	■	■	■	■	■	■	■	■																■	■	■	
SLDynamicSourceSinkChangeItf [8.19]				■	■	■	■	■	■	■	■	■																■	■	■	[8.12]
SLDeviceVolumeItf [8.16]	■	■	■																												[8.9]

Legend for Object Interface Mapping Table

- **P M G** — Object mandated in (P)hone, (M)usic, or (G)ame profile.
- **P M G** — Object optional in (P)hone, (M)usic, or (G)ame profile.
- Applicable optional interface
- Implicit and mandated interface
- Mandated (explicit) interface
- **A** — Arbitrary loop points are not mandated in this profile, only end-to-end looping is mandated.
- **B** — `{Set,Get,Enable,IsEnabled}StereoPosition()` are not mandated in this profile.
- **C** — SetLoopPoints() and GetLoopPoints() are not mandated in the profile.
- **1** — Explicit interface mandated for all players excluding those with Java Tone Sequences (JTS) data sources.
- **2** — Explicit interface mandated only where data source locator is a buffer queue (`SLDataLocator_{BufferQueue, MIDIBufferQueue}`).

OpenSL ES 1.1 Quick Reference Card - Page 2

Functions

Following are the functions available in OpenSL ES along with their parameters.

slCreateEngine() [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.

slQueryNumSupportedEngineInterfaces() [6.2]

Queries the number of available interfaces on an object.

pNumSupportedInterfaces	Identifies the number of supported interfaces available.

slQuerySupportedEngineInterfaces() [6.3]

Queries the supported interfaces on engine object.

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

Profiles

OpenSL ES is segmented into three profiles: Phone, Music and Game. Combinations of these three profiles are also possible—for example a full-featured game-and-music mobile phone would incorporate all three profiles. Recording functionality, which is commonly used for recording voice memos on mobile phones, is an optional feature and is not part of any profile.

Phone	This is the basic profile designed for the low-end or "basic" mobile phone market segment. This includes ringtone and alert tone playback (basic MIDI functionality), basic audio playback, and simple 2D audio games.
Music	This profile is designed for the music-centric mobile device market. Characteristics of such devices include high-quality audio, the ability to support multiple music audio codecs, and the ability to play music from local files. Some high-end devices could also support streaming audio from remote servers (although this is not mandated functionality in OpenSL ES). A mobile phone that has a built-in music player would incorporate both the Phone and Music profiles. A digital music-only mobile device would use only the Music profile.
Game	This profile is designed for the game-centric mobile device market. Characteristics of such devices include advanced MIDI functionality, and sophisticated audio capabilities such as 3D audio, audio effects, and the ability to handle buffers of audio. A mobile phone that offers sophisticated game-playing ability would incorporate both the Phone and Game profiles. A game-only device would use only the Game profile.

Interfaces

Refer to the Object-Interview Mapping Table (page 1) to see the objects to which each interface applies.

SL3DCommitItf [8.2]

Controls 3D interface commit policy.

Object: Engine

Commit	Commits changes to 3D interfaces.
SetDeferred	[En/dis]ables deferred committing of 3D parameters.

SL3DDopplerItf [8.3]

Controls Doppler for listener, player, or 3D group.

Objects: Audio Player, Midi Player, Listener, 3D Group

SetVelocityCartesian	Sets the object's velocity using Cartesian coordinates.
SetVelocitySpherical	Sets the object's velocity using spherical coordinates.
GetVelocityCartesian	Gets the object's velocity.
SetDopplerFactor	Sets the object's Doppler factor.
GetDopplerFactor	Gets the object's Doppler factor.

SL3DGroupingItf [8.4]

Sets the player's 3D group.

Objects: Audio Player, Midi Player

Set3DGroup	Sets the 3D group for player, removing player from previous 3D group.
Get3DGroup	Gets the 3D group for the player.

SL3DHintItf [8.5]

Defines the rendering quality hint.

Objects: Audio Player, Midi Player, 3D Group

SetRenderHint	Sets 3D source's rendering quality hint.
GetRenderHint	Retrieves the rendering quality hint.

SL3DLocationItf [8.6]

Controls location and orientation.

Objects: Audio Player, Midi Player, Listener, 3D Group

SetLocationCartesian	Sets the 3D location using Cartesian coordinates.
SetLocationSpherical	Sets the 3D location using spherical coordinates.
Move	Moves the object.
GetLocationCartesian	Gets the 3D location using Cartesian coordinates.
SetOrientationVectors	Sets 3D orientation using vectors.
SetOrientationAngles	Sets 3D orientation using angles.
Rotate	Rotates the object.
GetOrientationVectors	Gets 3D orientation as vectors.

SL3DMacroscopicItf [8.7]

Controls the size of a 3D sound source.

Objects: Audio Player, Midi Player, 3D Group

SetSize	Sets size of 3D sound source.
GetSize	Gets size of 3D sound source.
SetOrientationAngles	Sets the 3D orientation of volume using angles.
SetOrientationVectors	Sets the 3D orientation of volume using vectors.
Rotate	Rotates macroscopic volume.
GetOrientationVectors	Gets the 3D orientation.

SL3DSourceItf [8.8]

Controls 3D parameters unique to 3D sources.

Objects: Audio Player, Midi Player, 3D Group

SetHeadRelative	Sets whether 3D source is head relative.
GetHeadRelative	Gets the 3D source's head relative state.
SetRolloffDistances	Sets the min/max rolloff distances.
GetRolloffDistances	Gets the min/max rolloff distances.
SetRolloffMaxDistanceMute	Sets mute policy beyond rolloff distance.
GetRolloffMaxDistanceMute	Gets mute policy beyond rolloff distance.
SetRolloffFactor	Sets the distance rolloff factor.
GetRolloffFactor	Gets the distance rolloff factor.
SetRoomRolloffFactor	Sets the room rolloff factor.
GetRoomRolloffFactor	Gets the distance room rolloff factor.
SetRolloffModel	Sets distance decay rolloff model.
GetRolloffModel	Gets the distance rolloff model.
SetCone	Sets the sound cones.
GetCone	Gets the sound cones.

SLAudioDecoderCapabilitiesItf [8.9]

Queries the engine decode capabilities.

Object: Engine

GetAudioDecoders	Retrieves the available audio decoders.
GetAudioDecoderCapabilities	Queries for the audio decoder capabilities.

SLAudioEncoderItf [8.10]

Sets audio encoder parameters.

Object: Audio Recorder

SetEncoderSettings	Set audio encoder settings.
GetEncoderSettings	Get audio encoder settings.

SLAudioEncoderCapabilitiesItf [8.11]

Queries audio encoding capabilities.

Object: Audio Recorder

GetAudioEncoders	Queries the supported audio encoders.
GetAudioEncoderCapabilities	Queries for the audio encoder's capabilities.

SLAudioIODeviceCapabilitiesItf [8.12]

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 the capabilities of the specified audio input device.
RegisterAvailableAudioInputsChangedCallback	Sets/clears slAvailableAudioInputsChangedCallback().
GetAvailableAudioOutputs	Gets the number and IDs of audio output devices.
QueryAudioOutputCapabilities	Gets the capabilities of an audio output device.
RegisterAvailableAudioOutputsChangedCallback	Sets/clears slAvailableAudioOutputsChangedCallback().
RegisterDefaultDeviceIDMapChangedCallback	Sets/clears slDefaultDeviceIDMapChangedCallback().
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.
QuerySampleFormatsSupported	Gets array of sample formats supported by the audio I/O device for the given sampling rate.

Interfaces continues >

Interfaces (continued)

SLBassBoostItf [8.13]

Controls bass boost functionality.

Objects: Audio Player, Midi Player, Audio Recorder, Output Mix

SetEnabled	Enables the effect.
IsEnabled	Gets the enabled status.
SetStrength	Sets the strength of the effect.
GetRoundedStrength	Gets the strength of the effect.
IsStrengthSupported	Indicates if setting strength is supported.

SLBufferQueueItf [8.14]

Streams audio data.

Objects: Audio Player, Midi Player, Audio Recorder

Enqueue	Adds a buffer to the queue.
Clear	Releases all queued buffers.
GetState	Returns state of buffer queue.
RegisterCallback	Sets callback function to be called on buffer completion.
SetCallbackEventsMask	[En/dis]ables notification of buffer queue events.
GetCallbackEventsMask	Queries the notification state of buffer queue events.

SLConfigExtensionsItf [8.15]

Sets/queries the configuration of the audio engine.

Objects: All

SetConfiguration	Sets configuration as a key-value pair.
GetConfiguration	Gets the configuration setting as a key-value pair.

SLDeviceVolumeItf [8.16]

Exposes controls for manipulating the volume of audio I/O devices.

Object: Engine

GetVolumeScale	Gets the volume scale properties.
SetVolume	Sets the volume level.
GetVolume	Gets the volume level.

SLDynamicInterfaceManagementItf [8.17]

Manages interface exposure on a created and realized object.

Objects: ALL

AddInterface	Asynchronously expose an interface.
RemoveInterface	Removes a dynamically exposed interface.
ResumeInterface	Resumes a dynamically exposed interface.
RegisterCallback	Callback for various interface errors or events.

SLDynamicSourceItf [8.18]

Deprecated. Instead use `SLDynamicSourceSinkChangeItf`.

SLDynamicSourceSinkChangeItf [8.19]

Object: Metadata Extractor

Changes data source or sink during object lifetime.

ChangeSource	Changes an audio data source.
ChangeSink	Changes a data sink.
RegisterSourceChangeCallback	Sets or clears slSourceChangeCallback.
RegisterSinkChangeCallback	Sets or clears slSinkChangeCallback.

SLEffectSendItf [8.20]

Controls a sound's contribution to aux effects.

Objects: Audio Player, MIDI Player

EnableEffectSend	[En/dis]ables player's contribution to an aux effect.
IsEnabled	Returns if output goes to an aux effect.
SetDirectLevel	Sets dry (direct) path level for a sound.
GetDirectLevel	Gets the player's direct path level.
SetSendLevel	Sets send path level for an aux effect.
GetSendLevel	Gets send path level for an aux effect.

SLEngineItf [8.21]

Exposes creation methods of all object types.

Object: Engine

CreateLEDDevice	Creates an LED device.
CreateVibraDevice	Creates a vibrator device.
CreateAudioPlayer	Creates an audio player.
CreateAudioRecorder	Creates an audio recorder.
CreateMidiPlayer	Creates a MIDI player.
CreateListener	Creates a listener.
Create3DGroup	Creates a 3D group.
CreateOutputMix	Creates an output mix.
CreateMetadataExtractor	Creates a Metadata Extractor.
CreateExtensionObject	Creates an object.
QueryNumSupportedInterfaces	Queries the number of supported interfaces.
QuerySupportedInterfaces	Queries supported interfaces.
QueryNumSupportedExtensions	Queries the number of supported extensions.
QuerySupportedExtension	Gets extension name by index.
IsExtensionSupported	Queries if extension is supported.

SLEngineCapabilitiesItf [8.22]

Queries engine capabilities, as different implementations support profiles with varying capabilities.

Object: Engine

QuerySupportedProfiles	Queries the supported profiles.
QueryAvailableVoices	Queries number of simultaneous free voices available.
QueryNumberOfMIDISynthesizers	Queries number of supported MIDI synthesizers.
QueryAPIVersion	Queries the OpenSL ES API version.
QueryLEDCapabilities	Queries LED array capabilities.
QueryVibraCapabilities	Queries vibra device capabilities.
IsThreadSafe	Gets the thread-safety status.

SLEnvironmentalReverbItf [8.23]

Controls properties in a global reverb environment.

Objects: Audio Player, MIDI Player, Output Mix

SetRoomLevel	Sets reverb master volume level.
GetRoomLevel	Gets reverb master volume level.
SetRoomHFLevel	Controls low-pass filter volume level.
GetRoomHFLevel	Gets the room HF level.
SetDecayTime	Sets reverb decay time.
GetDecayTime	Gets the decay time.
SetDecayHFRatio	Sets relative HF decay time.
GetDecayHFRatio	Gets the relative HF decay time.
SetReflectionsLevel	Sets early reflections volume level.
GetReflectionsLevel	Gets the reflections level.
SetReflectionsDelay	Sets the early reflections delay time.
GetReflectionsDelay	Gets the reflections delay.

SLEnvironmentalReverbItf continues >

SLEnvironmentalReverbItf (continued)

SetReverbLevel	Sets the late reverberation volume level.
GetReverbLevel	Gets the reverb level.
SetReverbDelay	Sets reverb delay length.
GetReverbDelay	Gets the reverb delay length.
SetDiffusion	Sets the late reverberation decay echo density.
GetDiffusion	Gets the level of diffusion.
SetDensity	Controls the late reverberation decay modal density.
GetDensity	Gets the density level.
SetEnvironmentalReverbProperties	Sets all environment reverb properties.
GetEnvironmentalReverbProperties	Gets all the environment's reverb properties.

SLEqualizerItf [8.24]

Manipulates equalizer settings.

Objects: Audio Player, Midi Player, Audio Recorder, Output Mix

SetEnabled	Enables the effect.
IsEnabled	Gets enabled status of the effect.
GetNumberOfBands	Gets number of equalizer bands.
GetBandLevelRange	Returns 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 affect a frequency the most.
GetCurrentPreset	Gets the current preset.
UsePreset	Sets the equalizer according to the given preset.
GetNumberOfPresets	Gets the number of presets supported.
GetPresetName	Gets the preset name based on the index.

SLLEDArrayItf [8.25]

Sets LED array state and color.

Object: LED Array

ActivateLEDArray	Activates or deactivates individual LEDs.
IsLEDArrayActivated	Returns the state of each LED.
SetColor	Sets color of an individual LED.
GetColor	Gets color of an individual LED.

SLMetadataExtractionItf [8.26]

Extracts metadata.

Objects: Audio Player, Midi Player, Metadata Extractor

GetItemCount	Returns the number of metadata items.
GetKeySize	Returns the byte size of a given metadata key.
GetKey	Returns metadata by key.
GetValueSize	Returns the byte size of a given metadata value.
GetValue	Returns metadata by value.
AddKeyFilter	Adds a filter for a specific key.
ClearKeyFilter	Clears the key filter.

SLMetadataMessageItf [8.27]

Sets metadata callbacks during playback.

Objects: Audio Player, Midi Player, Metadata Extractor

RegisterMetadataCallback	Sets or clears the metadata callback.

Interfaces continues >

Interfaces (continued)

SLMetadataTraversalItf [8.28]

Traverses a file's metadata.

Objects: Audio Player, Midi Player, Metadata Extractor

SetMode	Sets the metadata traversal mode.
GetChildCount	Returns the number of child nodes in scope.
GetChildMIMETypeSize	Returns a child's MIME size.
GetChildInfo	Returns information about a child.
SetActiveNode	Sets the scope to a child index.

SLMIDIMessageItf [8.29]

Sends MIDI messages and sets MIDI callbacks.

Objects: MIDI Player

SendMessage	Sends a MIDI message to a player.
RegisterMetaEventCallback	Sets or clears an SMF meta-event callback.
RegisterMIDIMessageCallback	Sets or clears a MIDI message callback.
AddMIDIMessageCallbackFilter	Adds a message type to a MIDI callback filter.
ClearMIDIMessageCallbackFilter	Clears MIDI message callback filter.

SLMIDIMuteSoloItf [8.30]

Mutes and solos MIDI channels and tracks.

Object: MIDI Player

SetChannelMute	Mutes or unmutes a MIDI channel.
GetChannelMute	Returns mute state of a MIDI channel.
SetChannelSolo	Solos or unsolos a MIDI channel.
GetChannelSolo	Returns solo status of a MIDI channel.
GetTrackCount	Returns number of MIDI tracks in player's SMF data.
SetTrackMute	Mutes or unmutes a MIDI track.
GetTrackMute	Returns mute status of a MIDI track.
SetTrackSolo	Solos or unsolos a MIDI track.
GetTrackSolo	Returns solo status of a MIDI track.

SLMIDITempoItf [8.31]

Manages the MIDI data's tempo.

Object: MIDI Player

SetTicksPerQuarterNote	Sets the player's tick resolution.
GetTicksPerQuarterNote	Returns the tick resolution.
SetMicrosecondsPerQuarterNote	Sets the player's tempo.
GetMicrosecondsPerQuarterNote	Returns the tempo.

SLMIDITimeItf [8.32]

Manages the MIDI data in time (ticks).

Object: MIDI Player

GetDuration	Returns the duration in MIDI ticks.
SetPosition	Sets a player's position in MIDI ticks.
GetPosition	Returns the position in MIDI ticks.
SetLoopPoints	Sets loop points in MIDI ticks.
GetLoopPoints	Returns loop points in MIDI ticks.

SLMuteSoloItf [8.33]

Manages channel mute and solo status.

Object: Audio Player

SetChannelMute	Mutes or unmutes a channel.
GetChannelMute	Retrieves a channel's mute status.
SetChannelSolo	[En/dis]ables soloing of a channel.
GetChannelSolo	Retrieves a channel's soloed state.
GetNumChannels	Retrieves the number of audio channels.

SLObjectItf [8.34]

Provides essential utility methods for all objects.

Objects: ALL

Realize	Transitions Unrealized to Realized state.
Resume	Transitions Suspended to Realized state.
GetState	Retrieves the current object state.
GetInterface	Obtains the object's exposed interface.
RegisterCallback	Callback for error or async completion.
AbortAsyncOperation	Aborts asynchronous call in progress.
Destroy	Destroys the object.
SetPriority	Set the object's priority.
GetPriority	Gets the object's priority.
SetLossOfControlInterfaces	Sets/unsets loss of control functionality.

SLOutputMixItf [8.35]

Manages an output mix object.

Object: Output Mix

GetDestinationOutputDeviceIDs	Gets the destination device IDs.
RegisterDeviceChangeCallback	Callback for changes to the output device IDs.
ReRoute	Changes the specified set of output devices.

SLPitchItf [8.36]

Controls sound pitch shift.

Objects: Audio Player, MIDI Player

SetPitch	Sets a player's pitch shift.
GetPitch	Gets the player's pitch shift.
GetPitchCapabilities	Retrieves pitch shifting capabilities.

SLPlayItf [8.37]

Controls an object's playback state.

Objects: Audio Player, MIDI Player

SetPlayState	Transitions into the given play state.
GetPlayState	Gets the player's current play state.
GetDuration	Gets the duration of the current content.
GetPosition	Returns the relative position of the playback head.
RegisterCallback	Sets the playback callback function.
SetCallbackEventsMask	[En/dis]ables notification of playback events.
GetCallbackEventsMask	Queries the notification state of playback events.
SetMarkerPosition	Sets the position of the playback marker.
ClearMarkerPosition	Clears marker.
GetMarkerPosition	Queries the position of playback marker.
SetPositionUpdatePeriod	Sets the position notification interval.
GetPositionUpdatePeriod	Queries the position notification interval.

SLPlaybackRateItf [8.38]

Gets and sets the playback rate.

Objects: Audio Player, MIDI Player

SetRate	Sets the rate of presentation.
GetRate	Gets the rate of presentation.
SetPropertyConstraints	Sets the rate property constraints.
GetProperties	Gets the current properties.
GetCapabilitiesOfRate	Gets the capabilities of the specified rate.
GetRateRange	Retrieves the ranges of rates supported.

SLPrefetchStatusItf [8.39]

Queries the prefetch status of a player.

Objects: Audio Player, MIDI Player

GetPrefetchStatus	Gets the player's current prefetch status.
GetFillLevel	Queries the fill level of the prefetch.
RegisterCallback	Sets the prefetch callback function.
SetCallbackEventsMask	Sets the notification state of the prefetch events.
GetCallbackEventsMask	Queries the notification state of the prefetch events.
SetFillUpdatePeriod	Sets the notification period for fill level updates.
GetFillUpdatePeriod	Queries the notification period for fill level updates.
GetError	Retrieves the last error code.

SLPresetReverbItf [8.40]

Configures the global reverb with a preset.

Objects: Audio Player, MIDI Player, Output Mix

SetPreset	Enables a preset on the global reverb.
GetPreset	Gets the current global reverb preset.

SLRatePitchItf [8.41]

Controls the sound playback rate.

Object: Audio Recorder

SetRate	Sets a player's rate.
GetRate	Gets the player's rate.
GetRatePitchCapabilities	Retrieves the player's rate pitch capabilities.

SLRecordItf [8.42]

Controls the recording state of an object.

Object: Audio Recorder

SetRecordState	Transitions into the given record state.
GetRecordState	Gets the recorder's record state.
SetDurationLimit	Sets the duration of current content.
GetPosition	Returns the relative position of the recording head.
RegisterCallback	Registers the record callback function.
SetCallbackEventsMask	Sets the notification state of record events.
GetCallbackEventsMask	Queries the notification state of record events.
SetMarkerPosition	Sets the position of the recording marker.
ClearMarkerPosition	Clears marker.
GetMarkerPosition	Queries the position of the recording marker.
SetPositionUpdatePeriod	Sets the position notification interval.
GetPositionUpdatePeriod	Queries the position update interval.

SLSeekItf [8.43]

Manages a playback head's position and looping.

Objects: Audio Player, MIDI Player

SetPosition	Sets the position of the playback head.
SetLoop	Sets looping parameters.
GetLoop	Query looping parameters.

SLThreadSyncItf [8.44]

Thread control.

Object: Engine

EnterCriticalSection	Transitions the engine into critical section state.
ExitCriticalSection	Transitions into non-critical section state.

Interfaces continues >

Interfaces (continued)

SLVibraItf [8.45]

Controls the Vibra I/O device.

Object: Vibra

Vibrate	Activates or deactivates vibration.
IsVibrating	Returns whether the I/O device is vibrating.
SetFrequency	Sets the vibration frequency.
GetFrequency	Returns the vibration frequency.
SetIntensity	Sets the vibration intensity.
GetIntensity	Returns the vibration intensity.

SLVirtualizerItf [8.46]

Controls the audio virtualizer.

Objects: Audio Player, MIDI Player, Output Mix

SetEnabled	Enables the effect.
IsEnabled	Gets enabled status of the effect.
SetStrength	Sets the strength of the effect.
GetRoundedStrength	Gets the strength of the effect.
IsStrengthSupported	Query support of setting strength.

SLVisualizationItf [8.47]

Gets data for visualization.

Objects: Audio Player, MIDI Player, Audio Recorder, Output Mix

RegisterVisualizationCallback	Sets or clears the slVisualizationCallback.
GetMaxRate	Gets the maximum supported rate.

SLVolumeItf [8.48]

Volume control.

Objects: Audio Player, MIDI Player, Audio Recorder, Output Mix

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

Structures

SLAudioCodecDescriptor [9.1.1]

Query the audio codec capabilities.

SLAudioEncoderSettings [9.1.2]

Set the audio encoding parameters.

SLAudioInputDescriptor [9.1.3]

Return the description of input device capabilities.

SLAudioOutputDescriptor [9.1.4]

Return the description of output device capabilities.

SLBufferQueueState [9.1.5]

Number of buffers and index of current buffer.

SLDataFormat_MIME [9.1.6]

MIME parameters.

SLDataFormat_PCM [9.1.7]

Deprecated. Instead use `SLDataFormat_PCM_EX`.

SLDataFormat_PCM_EX [9.1.8]

PCM parameters.

SLDataLocator_Address [9.1.9]

A data locator for a memory address.

SLDataLocator_IODevice [9.1.10]

A data locator for an I/O Device.

SLDataLocator_BufferQueue [9.1.11]

A data locator for a buffer queue.

SLDataLocator_ContentPipe [9.1.12]

A data locator for a content pipe.

SLDataLocator_MediaObject [9.1.13]

A data locator for a media object.

SLDataLocator_MIDIBufferQueue [9.1.14]

A data locator for a MIDI buffer queue.

SLDataLocator_Null [9.1.15]

A null data locator used in conjuction with `SLDynamicSourceSinkChangeItf`.

SLDataLocator_OutputMix [9.1.16]

A data locator for an output mix.

SLDataLocator_URI [9.1.17]

A data locator for a URI.

SLDataSink [9.1.18]

A data sink by locator and format.

SLDataSource [9.1.19]

A data source by locator and format.

SLEngineOption [9.1.20]

Specify different options during engine creation.

SLEnvironmentalReverbSettings [9.1.21]

Store environmental reverb settings.

SLHSL [9.1.22]

A color defined in HSL color space.

SLInterfaceID [9.1.23]

The interface ID type.

SLLEDDescriptor [9.1.24]

Represents the capabilities of the LED array I/O Device.

SLMetadataInfo [9.1.25]

A key or a value from a metadata item key/value pair.

SLVec3D [9.1.26]

Coordinates in Cartesian form.

SLVibraDescriptor [9.1.27]

Capabilities of the Vibra I/O device.

Macros

SL_3DHINT_* [9.2.1]

Defines the importance of a 3D source or group.

OFF, QUALITY_LOWEST, QUALITY_LOW, QUALITY_MEDIUM, QUALITY_HIGH, QUALITY_HIGHEST

SL_AUDIOCODEC_* [9.2.2]

The audio encoding type.

PCM, MP3, AMR, AMRWB, AMRWBPLUS, AAC, WMA, REAL, VORBIS

SL_AUDIOPROFILE_* and SL_AUDIOMODE_* [9.2.3]

Audio profiles and modes.

SL_AUDIOSTREAMFORMAT_UNDEFINED

PCM Profiles and Modes
SL_AUDIOPROFILE_PCM

RealAudio Profiles and Levels
SL_AUDIOPROFILE_REALAUDIO,
SL_AUDIOMODE_REALAUDIO_G2,
SL_AUDIOMODE_REALAUDIO_8,
SL_AUDIOMODE_REALAUDIO_10,
SL_AUDIOMODE_REALAUDIO_SURROUND

MP3 Profiles and Modes
SL_AUDIOPROFILE_MPEG1_L3,
SL_AUDIOPROFILE_MPEG2_L3,
SL_AUDIOPROFILE_MPEG25_L3,
SL_AUDIOCHANMODE_MP3_MONO,
SL_AUDIOCHANMODE_MP3_ STEREO,
SL_AUDIOCHANMODE_MP3_JOINTSTEREO,
SL_AUDIOCHANMODE_MP3_ DUAL

AMR Profiles and Modes
SL_AUDIOPROFILE_AMR,
SL_AUDIOSTREAMFORMAT_CONFORMANCE,
SL_AUDIOSTREAMFORMAT_IF1,
SL_AUDIOSTREAMFORMAT_IF2,
SL_AUDIOSTREAMFORMAT_FSF,
SL_AUDIOSTREAMFORMAT_RTPPAYLOAD,
SL_AUDIOSTREAMFORMAT_ITU

AMR-WB Profiles and Modes
SL_AUDIOPROFILE_AMRWB

AMR-WB+ Profiles and Modes
SL_AUDIOPROFILE_AMRWBPLUS

AAC Profiles and Modes
SL_AUDIOPROFILE_AAC_AAC
SL_AUDIOMODE_AAC_{LC, SSR, LTP, HE, HE_PS, HE_MPS, MAIN, SCALABLE, ERLC, LD}
SL_AUDIOSTREAMFORMAT_MP4{ADTS, LOAS, LATM}
SL_AUDIOSTREAMFORMAT_ {MP2ADTS, ADIF, MP4FF, RAW}

Vorbis Profiles and Levels
SL_AUDIOPROFILE_VORBIS
SL_AUDIOMODE_VORBIS

Windows Media Audio Profiles and Modes
SL_AUDIOPROFILE_WMA{7, 8, 9, 10}
SL_AUDIOMODE_WMA_LEVEL{1, 2, 3, 4}
SL_AUDIOMODE_WMAPRO_LEVELM{0, 1, 2, 3}

SL_API [9.2.4]

A function prototype declaration macro.

SL_API

SLAPIENTRY [9.2.5]

An API entry point macro.

SLAPIENTRY

SL_BOOLEAN_* [9.2.6]

Canonical values for boolean type.

FALSE, TRUE

SL_BUFFERQUEUEEVENT_* [9.2.7]

Flags for a buffer queue event.

PROCESSED, UNREALIZED, CLEARED, STOPPED, ERROR, CONTENT_END

SL_BYTEORDER_* [9.2.8]

The byte order of 16- or 32-bit data.

BIGENDIAN, LITTLEENDIAN, NATIVE

SL_CHARACTERENCODING_* [9.2.9]

Metadata character encoding.

UNKNOWN, BINARY, ASCII, BIG5, CODEPAGE1252, GB2312, HZGB2312, GB12345, GB18030, GBK, IMAPUTF7, ISO2022JP, ISO2022JP1, ISO88591, ISO885910, ISO8859913, ISO885914, ISO885915, ISO88592, ISO88593, ISO88594, ISO88595, ISO88596, ISO88597, ISO88598, ISO88599, ISOEUCJP, SHIFTJIS, SMS7BIT, UTF7, UTF8, JAVACONFORMANTUTF8, UTF16BE, UTF16LE

Macros continues >

Macros (Continued)

SL_CONTAINERTYPE_* [9.2.10]
The data source or sink container type.
UNSPECIFIED, RAW, ASF, AVI, BMP, JPG, JPG2000, M4A, MP3, MP4, MPEG_ES, MPEG_PS, MPEG_TS, QT, XMF_0, XMF_1, XMF_2, XMF_3, WAV, XMF_GENERIC, AMR, AAC, 3GPP, 3GA, RM, DMF, SMF, MOBILE_DLS, OGG

SL_DATAFORMAT_* [9.2.11]
The possible data formats.
MIME, PCM, RESERVED3, PCM_EX

SL_DATALOCATOR_* [9.2.12]
The possible data locators.
NULL, URI, ADDRESS, IODEVICE, OUTPUTMIX, RESERVED5, BUFFERQUEUE, MIDIBUFFERQUEUE, MEDIAOBJECT, CONTENTPIPE

SL_DEFAULTDEVICEID_* [9.2.13]
Default device IDs.
AUDIOINPUT, AUDIOOUTPUT, LED, VIBRA, RESERVED1

SL_DEVICECONNECTION_* [9.2.14]
Types of I/O device connections.
INTEGRATED, ATTACHED_{WIRED, WIRELESS}, NETWORK

SL_DEVICELOCATION_* [9.2.15]
I/O device locations.
HANDSET, HEADSET, CARKIT, DOCK, REMOTE

SL_DEVICESCOPE_* [9.2.16]
I/O device scopes.
UNKNOWN, ENVIRONMENT, USER

SL_DYNAMIC_ITF_EVENT_* [9.2.17]
Dynamic interface events.
RUNTIME_ERROR, ASYNC_TERMINATION, RESOURCES_{AVAILABLE, LOST, LOST_PERMANENTLY}

SL_ENGINEOPTION_* [9.2.18]
Engine object creation options (see `slCreateEngine()`).
THREADSAFE, LOSSOFCONTROL, MAJORVERSION, MINORVERSION, STEPVERSION

SL_EQUALIZER [9.2.19]
Undefined equalizer setting.
SL_EQUALIZER_UNDEFINED

SL_I3DL2_ENVIRONMENT_PRESET_* [9.2.20]
I3DL2 environmentl reverb settings.
DEFAULT, GENERIC, PADDEDCELL, ROOM, BATHROOM, LIVINGROOM, STONEROOM, AUDITORIUM, CONCERTHALL, CAVE, ARENA, HANGAR, CARPETEDHALLWAY, HALLWAY, STONECORRIDOR, ALLEY, FOREST, CITY, MOUNTAINS, QUARRY, PLAIN, PARKINGLOT, SEWERPIPE, UNDERWATER, SMALLROOM, MEDIUMROOM, LARGEROOM, MEDIUMHALL, LARGEHALL, PLATE

SL_IODEVICE_* [9.2.21]
I/O device sources and sinks.
AUDIOINPUT, AUDIOOUTPUT, LEDARRAY, VIBRA, RESERVED4, RESERVED5

SL_METADATA_FILTER_* [9.2.22]
Bit-masks for metadata filtering criteria.
KEY, LANG, ENCODING

SL_METADATATRAVERSALMODE_* [9.2.23]
Method of traversing metadata.
ALL, NODE

SL_MIDIMESSAGETYPE_* [9.2.24]
Filtering MIDI messages.
NOTE_ON_OFF, POLY_PRESSURE, CONTROL_CHANGE, PROGRAM_CHANGE, CHANNEL_PRESSURE, PITCH_BEND, SYSTEM_MESSAGE

SL_MILLIBEL_* [9.2.25]
Limit values for millibel units.
MAX, MIN

SL_MILLIHERTZ_MAX [9.2.26]
Limit value for milliHertz unit.
SL_MILLIHERTZ_MAX

SL_MILLIMETER_MAX [9.2.27]
Limit value for millimeter unit.
SL_MILLIMETER_MAX

SL_NODE_PARENT [9.2.28]
The current scope of the node's parent.
SL_NODE_PARENT

SL_NODETYPE_* [9.2.29]
The type of a node.
UNSPECIFIED, AUDIO, VIDEO, IMAGE

SL_OBJECT_EVENT_* [9.2.30]
Object event notifications.
RUNTIME_ERROR, ASYNC_TERMINATION, RESOURCES_LOST, RESOURCES_AVAILABLE, ITF_CONTROL_TAKEN, ITF_CONTROL_RETURNED, ITF_PARAMETERS_CHANGED

SL_OBJECT_STATE_* [9.2.31]
Object states.
UNREALIZED, REALIZED, SUSPENDED

SL_OBJECTID_* [9.2.32]
Object type identifiers.
ENGINE, LEDDEVICE, VIBRADEVICE, AUDIOPLAYER, AUDIORECORDER, MIDIPLAYER, LISTENER, 3DGROUP, OUTPUTMIX, METADATAEXTRACTOR

SL_PCM_REPRESENTATION_* [9.2.33]
PCM data type.
SIGNED_INT, UNSIGNED_INT, FLOAT

SL_PCMSAMPLEFORMAT_* [9.2.34]
Audio device sample formats.
FIXED_8, FIXED_16, FIXED_20, FIXED_24, FIXED_28, FIXED_32, FIXED_64

SL_PLAYEVENT_* [9.2.35]
Play events.
HEADATEND, HEADATMARKER, HEADATNEWPOS, HEADMOVING, HEADSTALLED, DURATIONUPDATED

SL_PLAYSTATE_* [9.2.36]
Playback state.
STOPPED, PAUSED, PLAYING

SL_PREFETCHEVENT_* [9.2.37]
Prefetch related events.
STATUSCHANGE, FILLLEVELCHANGE, ERROR, ERROR_UNRECOVERABLE

SL_PREFETCHSTATUS_* [9.2.38]
Player's prefetch status.
UNDERFLOW, SUFFICIENTDATA, OVERFLOW

SL_PRIORITY_* [9.2.39]
Priority levels.
LOWEST, VERYLOW, LOW, BELOWNORMAL, NORMAL, ABOVENORMAL, HIGH, VERYHIGH, HIGHEST

SL_PROFILES_* [9.2.40]
The OpenSL ES API profiles.
PHONE, MUSIC, GAME

SL_RATECONTROLMODE_* [9.2.41]
Rate control mode.
CONSTANTBITRATE, VARIABLEBITRATE

SL_RATEPROP_* [9.2.42]
Object rate-related properties.
RESERVED1, RESERVED2, SILENTAUDIO, STAGGEREDAUDIO, NOPITCHCORAUDIO, PITCHCORAUDIO

SL_RECORDEVENT_* [9.2.43]
Record events.
HEADATLIMIT, HEADATMARKER, HEADATNEWPOS, HEADMOVING, HEADSTALLED, BUFFER_FULL, BUFFERQUEUE_STARVED

SL_RECORDSTATE_* [9.2.44]
Object recording state.
STOPPED, PAUSED, RECORDING

SL_REVERBPRESET_* [9.2.45]
I3DL2 reverb presets.
NONE, SMALLROOM, MEDIUMROOM, LARGEROOM, MEDIUMHALL, LARGEHALL, PLATE

SL_RESULT_* [9.2.46]
Method return values.
SUCCESS, PRECONDITIONS_VIOLATED, PARAMETER_INVALID, MEMORY_FAILURE, RESOURCE_ERROR, RESOURCE_LOST, IO_ERROR, BUFFER_INSUFFICIENT, CONTENT_CORRUPTED, CONTENT_UNSUPPORTED, CONTENT_NOT_FOUND, PERMISSION_DENIED, FEATURE_UNSUPPORTED, INTERNAL_ERROR, UNKNOWN_ERROR, OPERATION_ABORTED, CONTROL_LOST, READONLY, ENGINEOPTION_UNSUPPORTED, SOURCE_SINK_INCOMPATIBLE

SL_ROLLOFFMODEL_* [9.2.47]
Rolloff distance models.
EXPONENTIAL, LINEAR

SL_SAMPLINGRATE_* [9.2.48]
Common audio sampling rates.
8, 11_025, 12, 16, 22_05, 24, 32, 44_1, 48, 64, 88_2, 96, 192

SL_SEEKMODE_* [9.2.49]
Seek modes.
FAST, ACCURATE

SL_SPEAKER_* [9.2.50]
Speaker locations used when specifying channel mask.
FRONT_{LEFT, RIGHT, CENTER}, LOW_FREQUENCY, BACK_{LEFT, RIGHT, CENTER}, FRONT_LEFT_OF_CENTER, FRONT_RIGHT_OF_CENTER, SIDE_{LEFT, RIGHT}, TOP_CENTER, TOP_FRONT_{LEFT, CENTER, RIGHT}, TOP_BACK_{LEFT, CENTER, RIGHT}

SL_TIME [9.2.51]
Out of range playback time.
SL_TIME_UNKNOWN

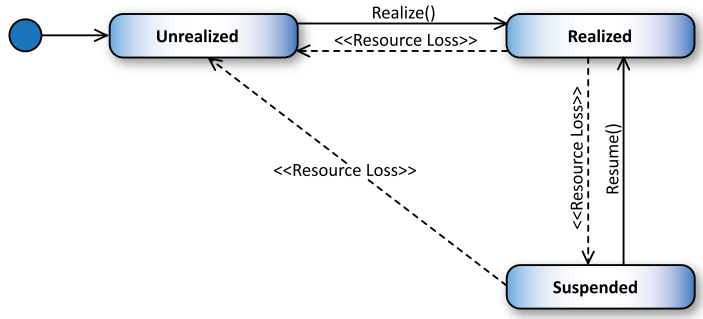
SL_VOICETYPE_* [9.2.52]
Voice types.
2D_AUDIO, MIDI, 3D_AUDIO, 3D_MIDIOUTPUT

Object State Diagram [3.1.1]

This diagram illustrates the object states and state transitions. When the application destroys an object, the object implicitly transitions through the Unrealized state. During the transition, it frees its resources and makes them available to other objects. Every object maintains a state machine with the following states:

- **Unrealized (initial state)**: The object is alive but has not yet allocated any resources. It is not usable, and its interfaces' methods cannot be called.
- **Realized**: The object's resources are allocated and the object is usable.
- **Suspended (optional state)**: The object has fewer resources than required to be usable, but it maintains the state it was in at the time of suspension. The system has the option of putting an object either in the Suspended state or the Unrealized state when resources are insufficient.

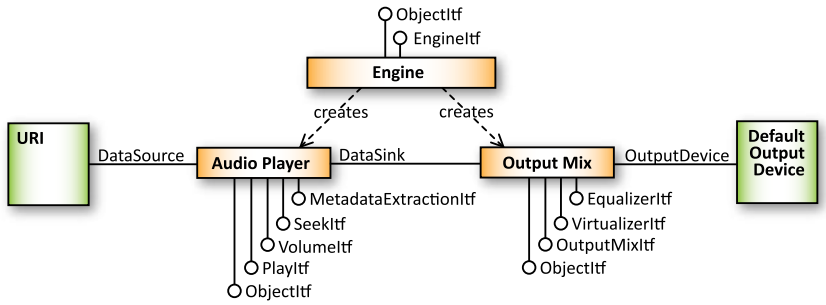
----> System-induced transitions ——> Client-induced transitions



Use Case: Sampled Audio Playback [4.6.1]

This example illustrates basic audio playback.

An Audio Player is created using the `SLEngineItf` interface of the Engine object. Upon creation, the Audio Player is associated with an Output Mix, created using the `SLEngineItf` interface, for audio output. The data source of the Audio Player is also set during creation. The data source is a URI pointing to an audio file on the local file system. The Output Mix is by default associated with the system-dependent default output device.

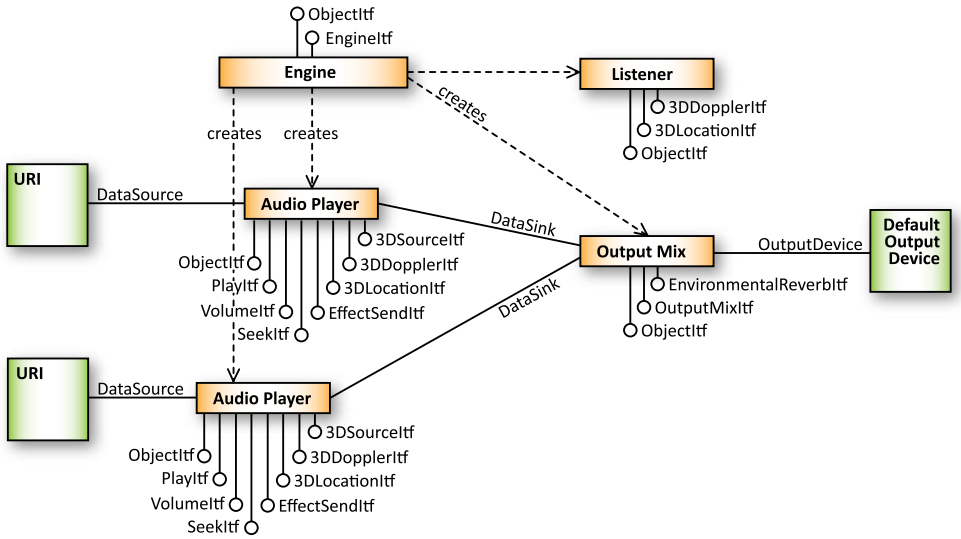


Use Case: 3D Audio [4.6.3]

This example illustrates positional 3D audio rendering using two sampled audio players simultaneously.

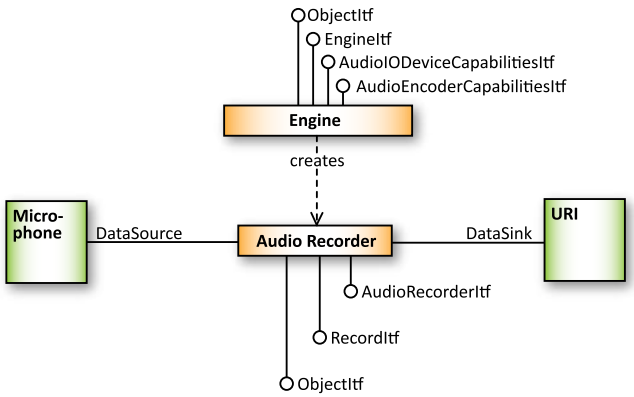
Both Audio Player objects are created using the `SLEngineItf` interface of the Engine object. Upon creation, both Audio Players are associated with the same Output Mix for audio output. Requesting the `SL3DLocationItf` interfaces from the Audio Players upon their creation causes them to be rendered as 3D sources.

The virtual listener is controlled with a Listener object which is created using the `SLEngineItf` interface of the engine object. The reverberation of the virtual acoustical space is controlled by the `SLEnvironmentalReverbItf` interface of the Output Mix. The `SLEffectSendItf` interfaces are exposed on the Audio Players to feed the audio signals to the reverberator of the Output Mix.



Use Case: Recording Audio [4.6.4]

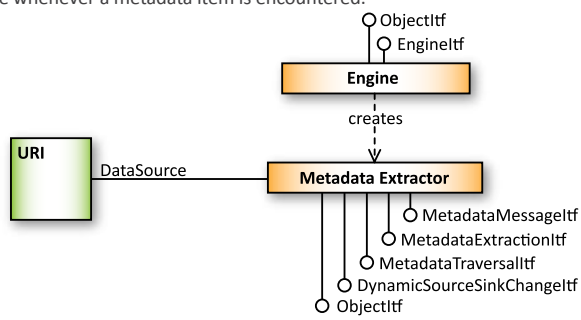
This example illustrates basic audio recording. Recording audio is handled by an Audio Recorder object, created using the `SLEngineItf` interface of the engine object. Upon creation, it is associated with an audio data source, in this case a microphone. The data sink of the Audio Recorder is a URI pointing to an audio file in the local file system to which the audio will be recorded.



Use Case: Reading Metadata [4.6.5]

This example illustrates reading the metadata from a file without playback. A Metadata Extractor object reads the metadata of an audio file without allocating resources for audio playback. The Metadata Extractor object is created using the `SLEngineItf` interface of the engine object and the data source is set. The data source is a URI pointing to an audio file in the local file system. The `SLMetadataExtractionItf` and `SLMetadataTraversalItf` interfaces are used for reading and traversing the metadata from the file.

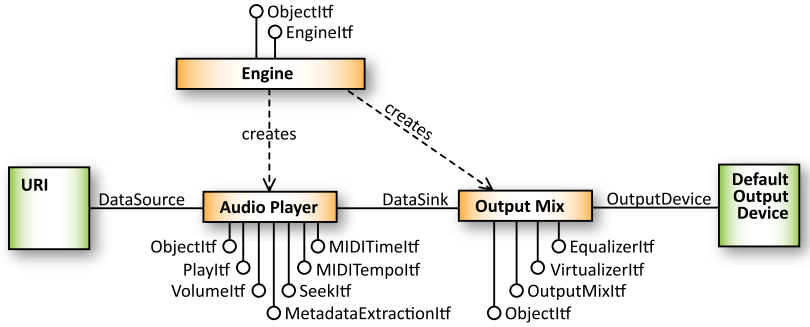
The Metadata Extractor supports the `SLDynamicSourceSinkChangeItf` interface which can be used to change the data source. Therefore metadata can be extracted from multiple files (in series) without creating a new Metadata Extractor object for every single file. The `SLMetadataMessageItf` interface is used to set callbacks that execute whenever a metadata item is encountered.



Use Case: MIDI Playback [4.6.2]

This example illustrates the use of OpenSL ES objects in a typical audio player use case using a MIDI Player object for audio playback.

The MIDI Player is created using the `SLEngineItf` interface of the Engine object. Upon creation, the MIDI Player is associated with an Output Mix for audio output. The data source of the MIDI Player is also set during creation. The Output Mix is by default associated with the system-dependent default output device.

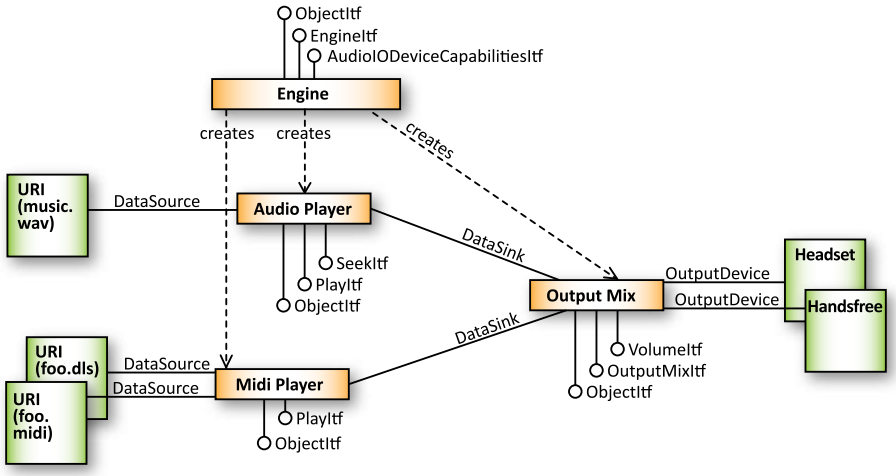


Use Case: Music Interrupted By Message [C3]

This example illustrates how to handle music playback on a headset when a message notification is played.

An incoming message alert requires MIDI to begin. The music playing on the headset is stopped after storing the music play position. A new MIDI player is created to generate a message alert from a file, directing the audio to a different IODevice output, the phone handsfree speaker. When the MIDI player has completed playing, the MIDI player is destroyed and music playback is resumed on the headset from where it was stopped.

See sample code to accompany this illustration in the specification [C.3.2].

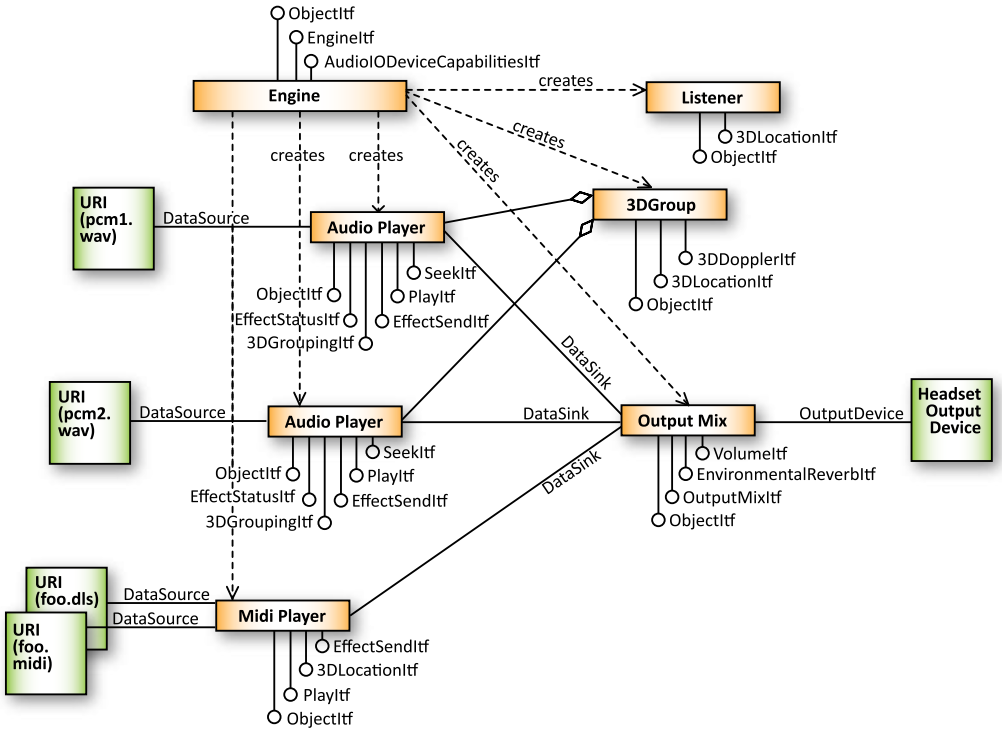


Use Case: Grouping 3D Movement [C4]

This example illustrates a combining a stationary 3D-positioned MIDI sound source with two moving PCM sound sources.

The two PCM sources, the sounds of a car engine and siren, are part of the same moving object and a 3DGroup is used to control their location as one. The listener is stationary, looking forward, as the moving sounds travel at a speed of 50kph from the left to the right. The location of the listener is set using the listener object. The 3DGroup object is used to control the location and Doppler of the car as it passes the listener. The environmental reverb and master volume are controlled using the Output mix object.

See sample code to accompany this illustration in the specification [C.4.2].



OpenSL ES

KHRONOS GROUP

The Khronos Group is an industry consortium creating open standards for the authoring and acceleration of parallel computing, graphics and dynamic media on a wide variety of platforms and devices. See www.khronos.org to learn more about the Khronos Group. OpenSL ES is a trademark of Khronos Group Inc.

Reference card production by Miller & Mattson www.millermattson.com