alDeleteBuffers

Description

This function deletes one or more buffers, freeing the resources used by the buffer.

alIsBuffer

Description

alBuffer3i

Description

This function sets a floating point property of a buffer.

```
void alBuffer3i(
```

alGetBuffer3i

Source Functions

Properties

Property	Data Type	Description
AL_PITCH	f, fv	

alSourcefv

Description

This function sets a floating point-vector property of a source.

```
void alSourcefv(
      ALuint source,
      ALenum param,
      ALfloat *values
);
```

Parameters

source

alSourcei

Description

alSourceiv

Description

This function sets an integer property of a source.

```
void alSourceiv(
      ALuint source,
      ALenum param,
      ALint *values
);
```

Parameters

source

alGetSourcefv

Description

alGetSourcei

Description

This function retrieves an integer property of a source.

```
void alGetSourcei(
      ALuint source,
      ALenum pname,
      ALint *value
);
```

alGetSource3i

Description

This function retrieves an integer property of a source.

```
void alGetSource3i(
      ALuint source,
      ALenum param,
      ALint *v1,
      ALint *v2,
      ALint *v3
);
```

Parameters

source	source name whose attribute is being retrieved
pname	the name of the attribute to retrieve:

alGetSourceiv

Description

This function retrieves an integer property of a source.

alSourcePausev

Description

This function pauses a set of sources.

```
void alSourcePausev(
      ALsizei n,
      ALuint *sources
);
```

Parameters

n	the number of sources to be paused
sources	a pointer to an array of sources to be paused

Possible Error States

State	***Description***
AL_INVALID_VALUE	The value pointer given is not valid.

alSourceStop

Description

alSourceStopv

Description

This function stops a set of sources.

```
void alSourceStopv(
      ALsizei n,
      ALuint *sources
);
```

Parameters

n	the number of sources to stop

alSourceRewind

alSourceRewindv

Description

alSourceQueueBuffers

Description

alSourceUnqueueBuffers

Description

This function unqueues a set of buffers attac

Listener Functions

Properties

Property	Data Type	Description
AL_GAIN	f, fv	“master gain” value should be positive
AL_POSITION	fv, 3f, iv, 3i	

alListener3f

Description

alListenerfv

Description

alListener3i

Description

alListeneriv

Description

This function sets an integer property of the listener.

```
void alListeneriv(
      ALenum param,
      ALint *values
);
```

Parameters

param

alGetListener3f

alGetListenerfv

alGetListeneri

Description

This function retrieves an integer property of the listener.

alGetListener3i

Description

This function retrieves an integer property of the listener.

alGetListeneriv

Description

This function retrieves an integer property of the listener.

alEnable

Description

alDisable

Description

alIsEnabled

Description

alGetDouble

Description

This function returns a double precision floating point OpenAL state.

```
Aldouble alGetDouble(
      ALenum  param
);
```

Parameters

param

alGetFloat

alGetInteger

Description

This function returns an integer OpenAL state.

```
Alint alGetInteger(
      ALenum  param
);
```

Parameters

alGetBooleanv

alGetDoublev

Description

This function retrieves a double precision floating point OpenAL state.

alGetFloatv

Description

This function retrieves a floating point OpenAL state.

```
void alGetFloatv(
```

alGetIntegerv

Description

This function retrieves an integer OpenAL state.

alGetString

alDistanceModel

Remarks

The default distance model in OpenAL is AL_INVERSE_DISTANCE_CLAMPED.

alDopplerFactor

Description

alSpeedOfSound

Description

This function selects the speed of sound for use in Doppler calculations.

Error Functions

Error Codes

alGetError

Description

This function returns the current error state and then clears the error state.

```
ALenum alGetError(ALvoid);
```

Parameters

None

alIsExtensionPresent

Description

Context Management Functions

Properties

Property	Data Type	Description
ALC_FREQUENCY	I	output frequency
ALC_MONO_SOURCES	I	

alcCreateContext

Description

alcProcessContext

Description

alcGetContextsDevice

Description

Context Error Functions

Context Device Functions

Functions

alcIsExtensionPresent

Description

alcGetProcAddress

Description

This function retrieves the address of a specified context extension function.

alcGetEnumValue

Description

This function retrieves the enum value for a specified enumeration name.

```
ALCenum alcGetEnumValue(
      ALCdevice *device,
      const ALCchar *enumName
);
```

Parameters

device	a pointer to the device to be queried
enumName	a null terminated string describing the enum value

Possible Error States

State	***Description***

alcGetIntegerv

Description

This function returns integers related to the context.

```
void alcGetIntegerv(
      ALCdevice *device,
      ALCenum param,
      ALCsizei size,
      ALCint *data
);
```

Parameters

Context Capture Functions

Functions

alcCaptureOpenDevice
alcCaptureCloseDevice
alcCaptureStart

alcCaptureOpenDevice

Description

This function opens a capture device by name.

```
ALCdevice * alcCaptureOpenDevice(
      const ALCchar *devicename,
      ALCuint frequency,
```

alcCaptureCloseDevice

alcCaptureStart

Description

This function begins a capture operation.

```
void alcCaptureStart(
      ALCdevice *device
);
```

Parameters

device	a pointer to a capture device

Possible Error States

State

alcCaptureStop

Description

This function stops a capture operation.

alcCaptureSamples

Description

This function completes a capture operation, and does not block.

```
void alcCaptureSamples(
```

ALC and AL Function Lists

Functions new to OpenAL 1.1 are *italicized* and **boldface**.

ALC Functions

```
alcCreateContext
alcMakeContextCurrent
alcProcessContext
alcSuspendContext
alcDestroyContext
alcGetCurrentContext
alcGetContextsDevice
alcOpenDevice
alcCloseDevice
alcGetError
alcIsExtensionPresent
alcGetProcAddress
alcGetEnumValue
alcGetString
alcGetIntegerv
alcCaptureOpenDevice
alcCaptureCloseDevice
alcCaptureStart
alcCaptureStop
alcCaptureSamples
```

AL Functions

```
alEnable
alDisable
alGetBooleanv
alGetIntegerv
alGetFloatv
alGetDoublev
alGetBoolean
```

alGetIntegerv

Enumeration Extension

The Enumeration Extension enables the application developer to retrieve a list of device strings identifying the different OpenAL rendering and capture devices present on the user's PC. The OpenAL router takes care of querying the user's system to find valid device implementations. Any of the strings returned by the enumeration

Use with alcIsExtensionPresent

Creative Labs' Extensions to OpenAL

Creative has introduced a number of extensions to OpenAL, many of which take advantage of the unique features of their soundcards. The "Enumerate All" extension is similar to the Core OpenAL "Enumeration Extension" but is extended to cover all available soundcards (and audio end-points in Windows Vista). The "X-RAM" extension allows a developer to utilize on-board audio RAM for storing OpenAL buffers. The "

Enumerate All Extension

The Enumerate All Extension enables the application developer to retrieve a complete list of device strings identifying all the available O

ALC_ENUMERATE_ALL_EXT

Use with alcIsExtensionPresent to detect if the Enumerate ALl etensionP is avail6(_able.)]TJETEMC /P /4MCID 15>>BD

X-RAM

With the introduction of the Sound Blaster X-Fi™ series of audio cards, Creative has launched a range of products that include on-board RAM. 'X-RAM' is provided on the top-end Sound Blaster X-Fi solutions (Sound Blaster X-Fi Fatal1ty™ FPS and Sound Blaster X-Fi Elite Pro). These

If a future alBufferData call is made on a buffer in automatic mode, the driver will assume that the application is using the AL Buffer for streaming (requiring regular updates to the audio data in the buffer), and the sample data will be moved from X-RAM to host memory. If there is not enough system memory then the AL error AL_OUT_OF_MEMORY will be set as per the OpenAL 1.0 specification.

Manual Mode - Hardware (AL_STORAGE _HARDWARE)

The final step in preparing an application to use X-RAM functionality is to query for the values of the X-RAM enumerations using alGetEnumValue. AL_EAX_RAM_SIZE and AL_EAX_RAM_FREE are used with alGetInteger to retrieve the total amount of X-RAM and the amount of free X-RAM. AL_STORAGE_AUTOMATIC

Multi-Channel Buffers

The multi-channel extension provides a mechanism to play multi-channel data via OpenAL. A variety of formats are supported. Multi-channel buffers can be attached or queued on a source.

Sample 1, front center speaker
Sample 1, low frequency speaker
Sample 1, back left speaker
Sample 1, back right speaker