XAudio2 BasicSound Sample

⊟ Collapse All

This sample demonstrates the XAudio2 API by showing you how to initialize the XAudio2 engine, create a mastering voice, and play sample files.

Sample Location

Windows

The full working source code for this sample is in:

`<Installed SDK Location>\Samples\C++\XAudio2\XAudio2BasicSound`

Xbox 360

The full working source code for this sample is in:

`<Installed XDK Location>\Source\Samples\Audio\XAudio2BasicSound`

Sample Description

The basic steps taken by the sample are as follows:

1. Initialize the XAudio2 engine by calling the XAudio2Create method.

    At this point, you can set some basic run-time parameters, and set the notification callback.

2. Create a mastering voice using the **IXAudio2::CreateMasteringVoice** method.

    This method controls the final mix format used for all audio processing within the application.

3. For Windows, create and play two WAV files.

    a. Locate the WAV file to load.

    b. Read in the WAV file and sample data using a sample helper class.

    c. Create a source voice based on the format of the loaded WAV file.

    By default, the source voice will be linked to the first created mastering voice, so you don't need a voice send list.

    d. To submit data to the source voice, create an XAUDIO2_BUFFER structure, specifying some play parameters.

    e. Submit data to the source voice by means of the function **IXAudio2SourceVoice::SubmitSourceBuffer**.

    Note that the sample data is not duplicated by XAUdio2, so the *pAudioData* buffer must remain available for the duration of play.

    f. Start playback of the source voice, and perform a simple loop to detect when the playback has completed.

    g. Clean up the source voice and associated sample data.

4. For Xbox 360, create and play two XMA2 files.

    The code uses the same logic as the WAV file processing, described in steps a-g above, because XMA2 uses a standard WAVEFORMATEX descriptor.

    ### Note

    Memory for the XMA2 data must be aligned to 2K in physical memory.

5. Clean up by releasing the XAudio2 engine.