

Autodesk® Scaleform®

Scaleform Audio Support Overview

This document outlines the audio support included with Scaleform 4.0 and higher versions.

Author: Vladislav Merker
Version: 1.01
Last Edited: June 19, 2012

Copyright Notice

Autodesk® Scaleform® 4.2

© 2012 Autodesk, Inc. All rights reserved. Except as otherwise permitted by Autodesk, Inc., this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 123D, 3ds Max, Algor, Alias, AliasStudio, ATC, AUGI, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk Homestyler, Autodesk Intent, Autodesk Inventor, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSketch, AutoSnap, AutoTrack, Backburner, Backdraft, Beast, Beast (design/logo) Built with ObjectARX (design/logo), Burn, Buzzsaw, CAiCE, CFdesign, Civil 3D, Cleaner, Cleaner Central, ClearScale, Colour Warper, Combustion, Communication Specification, Constructware, Content Explorer, Creative Bridge, Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignKids, DesignProf, DesignServer, DesignStudio, Design Web Format, Discreet, DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWFx, DXF, Ecotect, Evolver, Exposure, Extending the Design Team, Face Robot, FBX, Fempro, Fire, Flame, Flare, Flint, FMDesktop, Freewheel, GDX Driver, Green Building Studio, Heads-up Design, Heidi, Homestyler, HumanIK, i-drop, ImageModeler, iMOUT, Incinerator, Inferno, Instructables, Instructables (stylized robot design/logo), Inventor, Inventor LT, Kynapse, Kynogon, LandXplorer, Lustre, MatchMover, Maya, Mechanical Desktop, MIMI, Moldflow, Moldflow Plastics Advisers, Moldflow Plastics Insight, Moondust, MotionBuilder, Movimento, MPA, MPA (design/logo), MPI (design/logo), MPX, MPX (design/logo), Mudbox, Multi-Master Editing, Navisworks, ObjectARX, ObjectDBX, Opticore, Pipeplus, Pixlr, Pixlr-o-matic, PolarSnap, Powered with Autodesk Technology, Productstream, ProMaterials, RasterDWG, RealDWG, Real-time Roto, Recognize, Render Queue, Retimer, Reveal, Revit, RiverCAD, Robot, Scaleform, Scaleform GfX, Showcase, Show Me, ShowMotion, SketchBook, Smoke, Softimage, Sparks, SteeringWheels, Stitcher, Stone, StormNET, Tinkerbox, ToolClip, Topobase, Toxik, TrustedDWG, T-Splines, U-Vis, ViewCube, Visual, Visual LISP, Vtour, WaterNetworks, Wire, Wiretap, WiretapCentral, XSI.

All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

How to Contact Autodesk Scaleform:

Document	Scaleform Audio Support Overview
Address	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
Website	www.scaleform.com
Email	info@scaleform.com
Direct	(301) 446-3200
Fax	(301) 446-3199

Table of Contents

1	Introduction.....	1
2	Sound System Initialization.....	2
3	Video Sound System Initialization	3
4	Sound Event Interface	4
5	Video Sound System Interfaces.....	5
6	Sample Source Code.....	7
7	Additional Resources	7

1 Introduction

Autodesk® Scaleform® 4.0 and higher provides full audio support for playing Flash® sound files through a custom sound renderer interface based on the cross-platform FMOD™ sound library. Embedded audio files and external files loaded from ActionScript can be played in Scaleform. The audio files can be of various formats such as WAV, MP3 etc. Sounds from video files can also be played with the default sound renderer. Additionally, developers can utilize their own sound library using the sound renderer interface implemented in Scaleform. Please note that it will be the customers' responsibility to obtain appropriate FMOD and MP3 licenses.

Using embedded Flash audio may be more suitable for casual games or games which are primarily Flash based. It is also useful for prototyping screens and allows artists to quickly iterate on the look and feel of a User Interface (UI) without requiring a game engine or final audio pipeline in place. For larger game development, a more common solution is to play audio through triggered events and handle the events using the game's native audio engine. This allows the same audio pipeline, sound engineers, file packing and resource management system to be used for both UI audio and in-game sounds. Scaleform allows for this type of solution by providing a standard event-based, Flash sound interface for FMOD/ Designer and Wwise®/SoundFrame, see the section below on Sound Event Interface for more details.

2 Sound System Initialization

To initialize the Scaleform sound system, an instance of the `Gfx::Audio` state class should be set on the `Gfx::Loader` object. The purpose of this state is to provide the sound renderer object, `Sound::SoundRenderer` and the synchronization parameters for playing Flash streaming sounds. The `Sound::SoundRenderer` class is an abstract C++ interface, which should be implemented in the game to produce the sound. Scaleform provides the default implementation, which is based on the FMOD cross-platform sound library and can be used on all supported platforms.

Example:

```
SoundRenderer* pSoundRenderer = GetSoundRenderer();
if (pSoundRenderer)
{
    // Sound renderer and parameters for playing SWF streaming sounds
    // are provided to the loader by Gfx::Audio. These parameters
    // control when and how the synchronization between SWF frames
    // and sound frames should be done.
    Ptr<Gfx::Audio> pAudioState = *new Audio(pSoundRenderer);
    loader.SetAudio(pAudioState);
    pSoundRenderer->SetMasterVolume(SoundVolume);
    pSoundRenderer->Mute(IsMute);
}
```

This sound system initialization is only required for embedded Flash audio support. If you are using the Sound Event Interface instead, with Wwise or your own sound engine, this step is not required.

Please refer to the source code of Scaleform Player and to the Flash sound example for details.

3 Video Sound System Initialization

If a video file has audio data, then an instance of the `GFX::Video::VideoSoundSystem` interface should be set on the `GFX::Video::Video` object for initializing the video playback system. Scaleform provides implementations of this interface on each supported platform based on `Sound::SoundRenderer` interface.

Example:

```
SoundRenderer* pSoundRenderer = GetSoundRenderer();
if (pSoundRenderer)
{
    // Setting a video sound system instance which is based on
    // SoundRenderer interface
    pVideo->SetSoundSystem(pSoundRenderer);
}
```

Example:

```
// Setting a video sound system instance which is specific to
// a particular platform
pVideo->SetSoundSystem(
    Ptr<Video::VideoSoundSystem>(*new Video::VideoSoundSystemDX8(0)));
```

Please note that the `SetSoundSystem()` method should be called only once per an instance of `GFX::Video::Video`.

4 Sound Event Interface

Scaleform 3.2 adds an event-based, Flash sound interface for FMOD/ Designer and Wwise®/SoundFrame (Sound::SoundEvent::PostEvent and Sound::SoundEvent::SetParam). This new interface allows sound playback to be easily triggered from within ActionScript and played back using the Scaleform Flash player or a game engine sound system.

Four implementations are included in Scaleform distribution:

1. Sound::SoundEventFMOD - based on FMOD Ex API.
2. Sound::SoundEventFMODDesigner - based on FMOD EventSystem API and supports the FMOD Designer tool.
3. Sound::SoundEventWwise - based on Wwise API.
4. Sound::SoundEventWwiseSF - supports the Wwise SoundFrame tool protocol.

Example:

```
#include "Sound/Sound_SoundEventWwise.h"
Ptr<SoundEvent> soundEvent = *new SoundEventWwiseSF;

soundEvent->PostEvent("Sound_PostEvent", "Play_Hello");
soundEvent->SetParam("Sound_SetParam", "Enable_Effect", 85);
```

Please refer to the source code of Scaleform Player, to the implementation of sound event interfaces and to Flash sound example for details.

5 Video Sound System Interfaces

GfX::Video::VideoSoundSystem is an abstract interface providing sound support for video (GfX::Video::Video) playback; developers can substitute the video sound implementation by making their own version of this class. Before playing video, an instance of this class needs to be created and installed with Video::SetSoundSystem(). Typically, a platform-specific implementation can be used to avoid implementing this interface.

Following are the platform-specific sound system interfaces included in GfX::Video distribution:

- Video::VideoSoundSystemDX8 – DirectSound for Windows.
- Video::VideoSoundSystemXA2 – XAudio2 for Windows and Xbox360.
- Video::VideoSoundSystemPS3 – MultiStream for PS3.
- Video::VideoSoundSystemWii – Wii system sound.
- Video::VideoSoundSystemFMOD – sound interface based on FMOD.
- Video::VideoSoundSystemWwise – sound interface based on Wwise.

In the current version of Scaleform, video sound support is decoupled from the embedded Flash sound playback, allowing video to be used without requiring the general sound engine. To make this work, video is supported by an independent GfX::Video::VideoSoundSystem class that is separate from Sound::SoundRenderer used in the rest of Scaleform. This means that to get video sound support, Video::VideoSoundSystem and Video::VideoSound class only need to be implemented, which are much simpler than Sound::SoundRenderer. Note that if you already have Sound::SoundRenderer implementation, you can use it directly to initialize Video::Video, as it provides a superset of functionality. In some cases, the two implementations can be mixed (helpful if a custom video sound class provides better streaming support than the general sound engine).

Example:

```
#include "Video/Video_VideoSoundSystemXA2.h"
pVideo->SetSoundSystem(
    Ptr<Video::VideoSoundSystem>(*new VideoSoundSystemXA2(0, 0)));
```

For video sound support, Video::VideoSoundSystem and Video::VideoSound need to be implemented. Typically there is only one instance of VideoSoundSystem installed during video initialization. VideoSoundSystem exposes a single method, Create, used to create VideoSound objects representing independent video sound streams. Scaleform will call this function every time a new video is opened (there can be multiple videos playing at the same time). After each VideoSound object is created, Scaleform will call its various functions to instruct it to Start and Stop audio output. The actual sound data for the stream is obtained through polling of the VideoSound::PCMStream passed

to the given sound. Polling is typically done by a separate thread maintained by the VideoSoundSystem to service its active sounds.

Please note that the current implementation of VideoSoundSystemWwise based on Audio Input plug-in is available as part of Wwise SDK since v2009.2.1 build 3271. Audiokinetic provides the full source code and the Visual Studio solutions/projects of this plug-in can be found at *SDK\samples\Plugins\AkAudioInput*. Please refer to Wwise documentation for details. Scaleform Video distribution does not include any part of the Wwise SDK which should be installed separately.

Example:

```
#include " Video/Video_VideoSoundSystemWwise.h"

Ptr<Video::VideoSoundSystem> wwiseSound =
    Ptr<Video::VideoSoundSystem>(*new VideoSoundSystemWwise());
pVideo->SetSoundSystem(wwiseSound);
wwiseSound->Update();
```

Please refer to the source code of Scaleform Player and to the implementation of video sound system interfaces for details. Also, see the [Getting Started with Video](#) document for information on playing videos.

6 Sample Source Code

Please refer to the following files for more details on implementing the sound renderer.

- Scaleform Player source code, FMOD/Wwise initialization:
 - Apps\Samples\FxPlayer\FxPlayer.cpp
 - Apps\Samples\Common\FxSoundFMOD.cpp
 - Apps\Samples\Common\FxSoundWwise.cpp
- Sound renderer implementation based on FMOD:
 - Src\Sound\Sound_SoundRendererFMOD.h
 - Src\Sound\Sound_SoundRendererFMOD.cpp
- Sound event interfaces FMOD/Designer and Wwise/SoundFrame:
 - Src\Sound\Sound_SoundEventFMOD.h
 - Src\Sound\Sound_SoundEventWwise.h
- Video sound system interfaces implementation:
 - Src\Video\Video_VideoSoundSystem*.cpp
- Flash sound example:
 - Bin\Data\AS2\Samples\Audio_Demo_AS2 fla
 - Bin\Data\AS3\Samples\AudioDemo_AS3 fla

7 Additional Resources

Please refer to the below documents for additional information on the CRI Movie codec for playing video files, the ActionScript extensions used in playing the sound files, and the options for exporting the SWF into desired formats.

- [Adobe Flash ActionScript 2.0 Language Reference](#)
- [Scaleform - Flash Support Overview](#)
- [Getting Started with Video](#)
- [CRI Movie Encoder Command Line Tool Overview](#)
- [GFxExport Reference Guide](#)