Using Sound with Mobile Devices

In this unit, you will import sound into your Flash Lite movie and control the sound on the Timeline.

Adobe

Objectives

After completing this unit, you should be able to:

- Import sound files into your Flash Lite application
- Test the sound capabilities of the target device
- Control sound using both the Timeline and a `Sound` object

Incorporating sound into a Flash Lite application

- Adding sound to a Flash Lite application can greatly enhance the user experience for an animation, game or application.
- Many mobile devices support sound and Flash Lite can test for this capability.
- Sound files may be either imported directly into the FLA, and compiled into the SWF when published. They may be externally loaded at runtime as well.

Importing sound

Most major audio file types may be imported by the Flash 8 Authoring tool. Internally, imported sounds are converted to the MP3 format.

To import a sound file into your Flash document Library

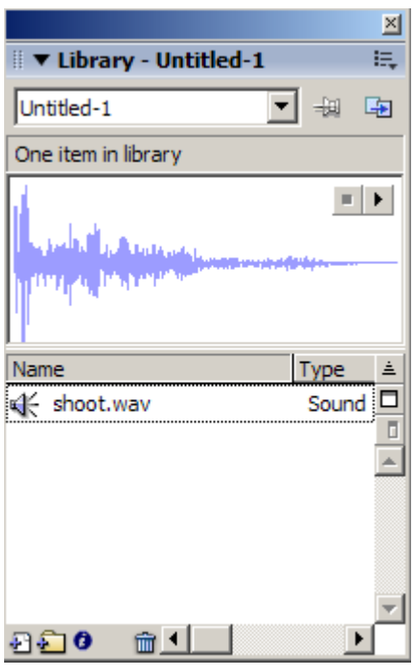
1.select File > Import > Import to Library

2.navigate to the sound file

3.press 'Open'

Importing sound

- Imported sound files appear in the Library with their original file name. However, this name can be changed, as imported sound files are embedded within the FLA.
- When the FLA is published, all sounds in use will be compiled into the SWF. If the original sound file changes, it must be re-imported.



Importing Sounds to the Library

Adobe

Importing sound

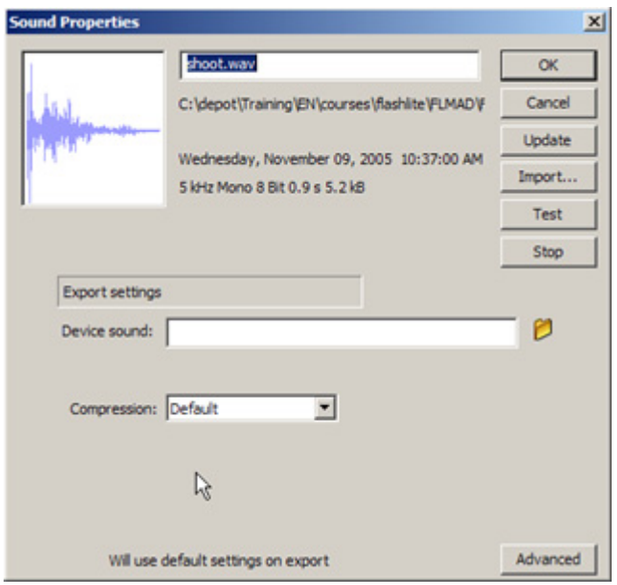
When you import a sound to the Library, the imported sound file:

- Is compressed to MP3 format by default.
- Is stored once and can be played over and over in its application.
- Adds file size to the Flash Lite document and application.
- May be added to a keyframe in a `Button` or `MovieClip` Timeline, including the main document timeline.
- Must be specifically exported for use by ActionScript.

Changing imported sound properties

Once you have imported a sound file, you can change its compression type in the `Sound Properties` dialog box. To open the Sound Properties dialog box:

- ▶double-click the sound file's icon in the Library, or
- ▶right-click over the sound file's Library icon, and select `Properties`



Changing Sound properties

Adobe

Compressing and exporting sounds

Sound compression can be set globally, or per sound file:

- ▶Defaults for stream or event sound: File > Publish > Publish Settings
- ▶Individual settings per sound file

Some cautions about compression:

- ▶Low frequency sound, such as bass sounds, compress extremely well.
- ▶Sounds such as hi-hat drums may cut off if you over-compress them.

Adobe

Individual compression

- ▶Individual sound compression may be set in the Sound Properties dialog in the Library. The advantage of setting the compression individually is that you can:
- ▶Cut the file size down for parts that compress quite well
- ▶Use less compression on more fidelity-sensitive sounds to get a better sound quality.

Set an individual file's compression:

1. Find the file in the Library
2. Right-click
3. Select the Properties dialog

- ▶The sound you produce will most likely be coming through a tiny monophonic speaker; therefore the sound quality will be only as good as the speaker allows. Ideally, you would have a sample of the actual device for testing.

Default compression

- Default sound compression may be set from File > Publish Settings > Flash.
- A global export setting of ADPCM, 11kHz, 4 bit, Mono works fine for most scenarios. It provides a good balance of sound quality, for tiny speakers, and minimal file size.
- With Flash Lite and this technique, ADPCM is your only choice of exporting format for seamless looping.
  - Unlike desktop Flash productions, MP3 exported sound will not play seamlessly in Flash Lite.
  - MP3 compression in Flash Lite leaves a tiny gap of silence at each loop point so that the sound drops out for a split second as the sounds loop.
  - Playing the SWF file on your desktop will not reveal this problem until you load it into your mobile device.

Exporting sound files for use with ActionScript

Imported sounds must be exported for use in ActionScript, so they will be compiled into the SWF.

To export a sound file from your Library for ActionScript:

1. Right-click over the sound in the Library.
2. Select Linkage.
3. Check Export for ActionScript.
4. Leave the default Identifier value
5. Click OK.

Using the Sound Class

- You must create a `Sound` object for each sound you wish to play using ActionScript.
- A `Sound` object can play either an imported sound, or load an external MP3 sound file at runtime, depending on the capabilities of the device.

Using the attachSound() method

- Once the `Sound` object is created, you use the `attachSound()` method to attach an exported sound from the Library.

```
var instanceName:Sound = new Sound();
instanceName.attachSound(identifier);
```

- Once the sound is attached to the `Sound` object, you use the `start()` and `stop()` methods to play the sound and stop its playing. To create audio feedback for events, start your sound in an appropriate event handler.

Using the attachSound() method, code example

```
var startGame_btn:Button;
var sndBoing_snd:Sound = new Sound();
sndBoing_snd.attachSound("boing.wav");

startGame_btn.onRelease = function():Void
{
 sndBoing_snd.start();  // plays boing.wav
}
```

Using the loadSound()method

- The Flash Lite player can also load and play external MP3 files. You specify the URL to the MP3 file as a parameter to the `loadSound()` method. When the `onLoad` event fires, your sound is loaded and ready to play.

```
var sound:Sound = new Sound();
var soundURL:String = "/sounds/sound.wav";
sound.loadSound(soundURL, isStreaming);
sound.onLoad = function():Void
{
 sound.start();
}
```

- If the `isStreaming` parameter is `true`, the sound will begin to play as it loads. If it is `false`, the sound will not play until the `start()` method is called.

Testing for Sound Capabilities

You can programmatically test for the sound capabilities on a mobile device using the following:

- Use the `true` or `false` value of the `System.capabilities.hasAudio` property to determine whether audio is available on the device.
- Use `fscommand2("GetMaxVolumeLevel")` to determine the maximum sound level.
- Use `fscommand2("GetVolumeLevel")` to determine the current volume level.
- Use the `true` or `false` value of the `System.capabilities.hasCompoundSound` property to determine whether Flash Lite can process compound sound data.
- Use the `true` or `false` value of the `System.capabilities.hasStreamingAudio` global property to determine whether Flash Lite can provide streaming sound.

Returning the maximum volume of the device

The following example sets a variable called `maxvolume` to the maximum volume level of the device:

```
maxvolume = fscommand2("GetMaxVolumeLevel");
trace (maxvolume);
```

Returning the current volume of the device

The following example assigns the current volume level to the `volume` variable:

```
volume = fscommand2("GetVolumeLevel");
trace (volume);                                    // output: 50
```

Using the System.capabilities values

The `System.capabilities` properties return `true` or `false`, based on whether Flash Lite has a capability in the current device.

```
if (System.capabilities.hasAudio == true)
{
    soundTrack.play();
}
else
{
    delete soundTrack;
}
```


Adobe

Walkthrough 1: Loading External Sounds

In this walkthrough, you will perform the following tasks:

- Create a `Sound` object
- Get the sound location from the loaded XML
- Load the audio

Summary

- The `Sound` class allows Flash Lite to play audio files.
- Sounds may be loaded in two ways:
  - Import to FLA, export for ActionScript, attach with `attachSound(linkageID)` method.
  - Load at runtime using the`loadSound(url, isStreaming)` method.
- Sound compression values may be controlled two ways:
  - Default: settings in File > Publish Settings > Flash will be applied to all sounds by default.
  - Per Sound: settings made, per sound, in the Sound Properties dialog will override the default settings.
- Current device sound levels can be checked using `fscommand2("GetVolumeLevel")` and `fscommand2("GetMaxVolumeLevel")`.
- The ability of a device to support sound can be determined using the `System.capabilities.hasAudio`, `System.capabilities.hasStreamingAudio`, and `System.capabilities.hasCompoundSound` properties.
- The `start()` method causes the sound currently loaded or attached to a `Sound` object to play.

Summary

- The `stop()` method causes all sounds to stop playing. If the sound is an attached sound, it may be stopped specifically by passing its Identifier to the `stop()` method.