

Macromedia Flash MX Updater for Flash Lite

User Guide

Version 1.0



Trademarks

Afterburner, AppletAce, Attain, Attain Enterprise Learning System, Attain Essentials, Attain Objects for Dreamweaver, Authorware, Authorware Attain, Authorware Interactive Studio, Authorware Star, Authorware Synergy, Backstage, Backstage Designer, Backstage Desktop Studio, Backstage Enterprise Studio, Backstage Internet Studio, Design in Motion, Director, Director Multimedia Studio, Doc Around the Clock, Dreamweaver, Dreamweaver Attain, Drumbear, Drumbear 2000, Extreme 3D, Fireworks, Flash, Fontographer, FreeHand, FreeHand Graphics Studio, Generator, Generator Developer's Studio, Generator Dynamic Graphics Server, Knowledge Objects, Knowledge Stream, Knowledge Track, Lingo, Live Effects, Macromedia, Macromedia Contribute, Macromedia M Logo & Design, Macromedia Flash, Macromedia Flash Lite, Macromedia swflite, Macromedia Xres, Macromind, Macromind Action, MAGIC, Mediamaker, Object Authoring, Power Applets, Priority Access, Roundtrip HTML, Scriptlets, SoundEdit, ShockRave, Shockmachine, Shockwave, Shockwave Remote, Shockwave Internet Studio, Showcase, Tools to Power Your Ideas, Universal Media, Virtuoso, Web Design 101, Whirlwind and Xtra are trademarks of Macromedia, Inc. and may be registered in the United States or in other jurisdictions including internationally. Other product names, logos, designs, titles, words or phrases mentioned within this publication may be trademarks, servicemarks, or tradenames of Macromedia, Inc. or other entities and may be registered in certain jurisdictions including internationally.

Third-Party Information

This guide contains links to third-party Web sites that are not under the control of Macromedia, and Macromedia is not responsible for the content on any linked site. If you access a third-party Web site mentioned in this guide, then you do so at your own risk. Macromedia provides these links only as a convenience, and the inclusion of the link does not imply that Macromedia endorses or accepts any responsibility for the content on those third-party sites.

Apple Disclaimer

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

Copyright © 2003 Macromedia, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Macromedia, Inc.

Acknowledgments

Director: Erick Vera

Producer: Barbara Nelson

Writing: Paul Goldman

Editing: Lisa Stanziano

Print Design and Production: Chris Basmajian, Caroline Branch

First Edition: January 2003

Macromedia, Inc.
600 Townsend St.
San Francisco, CA 94103

CONTENTS

| | |
|--|----|
| CHAPTER 1 | |
| Introduction | 5 |
| CHAPTER 2 | |
| Getting Started | 7 |
| Flash Lite features | 8 |
| CHAPTER 3 | |
| Developing Content for Flash Lite | 11 |
| Testing Flash Lite compatible content | 13 |
| Tutorial I—Creating a simple movie for Flash Lite (no sound) | 14 |
| Flash Lite sound | 14 |
| User configuration file—FlashLiteUser.txt. | 18 |
| Sound linkage identifier | 19 |
| Using linkage wildcards for batch sound substitution | 20 |
| Creating standard Flash 4 content | 20 |
| Sound substitution flowcharts | 21 |
| APPENDIX | |
| Warning/Error Messages | 23 |

CHAPTER 1

Introduction

Macromedia has created a new Flash Player version, called Macromedia® Flash™ Lite, that runs on a new class of consumer mobile devices. This format is designed to run optimally on devices with limited resources (memory, processor speed, display area). Content created for Flash Lite is most similar to Flash Player 4 content, with the primary exception of how sound is incorporated and played.

This user guide describes tools and guidelines for authors creating content for Macromedia Flash Lite. The next section describes in detail the considerations authors should have when creating Flash content that is run in mobile devices versus standard desktop and notebook devices.

About the Macromedia Flash MX Updater for Flash Lite

With the existing Macromedia Flash MX authoring tool, you can create movies for Flash Player versions 1 through 6, but Flash MX does not directly support creating content for Flash Lite. After you install the Flash Lite Updater, however, you can use the modified Flash MX to preview, publish, and validate content for Flash Lite.

Flash MX uses two versions of the Flash Player for authoring: the Flash MX test movie player and the stand-alone player. The test movie player is a plug-in to Flash MX that you can use to preview SWF movies during authoring. Whenever you select the Control > Test Movie or Control > Test Scene menu commands in Flash MX, the test movie player is started. You can use the stand-alone player to view an existing Flash SWF movie after it has been published.

To support Flash Lite, both the Flash MX test movie player and the Flash 6 stand-alone player are modified. When you preview SWF files using the test movie player, additional information/warning/error messages are displayed to let you know if there is any violation of Flash Lite rules and syntax. This helps you determine how to make the content Flash Lite compatible. Because Flash Lite supports only MIDI (Music Instrument Digital Interface) and MFi (Melody Format for i-Mode) as its native sound formats, a new utility—swflite.exe—enables you to insert MIDI or MFi sound directly into a SWF file.

These Flash Lite enhancements are available in the English and Japanese versions of Flash MX after you install the Flash MX Updater for Flash Lite.

CHAPTER 2

Getting Started

Installing the Macromedia Flash MX Updater for Flash Lite on your computer does the following:

- Installs some default configuration files
- Installs the `swflite.exe` (swflite on the Mac) utility
- Replaces the existing stand-alone Flash Player 6 for Flash MX
- Replaces the existing Flash MX test movie player

Note: Always check the Macromedia Flash Support Center at http://www.macromedia.com/go/flash_support, for newer updates to Flash MX.

Macromedia Flash MX Updater for Flash Lite

The Flash Lite Updater package installs the following files into existing Macromedia Flash MX folders:

- `AuthPlay.dll` (AuthPlayLib on the Mac) replaces the Flash MX test movie player.
- `SAFlashPlayer.exe` (SAFlashPlayer on the Mac) replaces the stand-alone player.
- `swflite.exe` (swflite on the Mac) is a new utility for batch processing of MIDI/MFi sound substitution
- `FlashLiteUser.txt` is an ASCII configuration file containing user-definable flags used by the test movie player and the `swflite.exe` utility for sound file substitution.
- `Devices.cfg` is a new binary configuration file; a system file describing the name, properties and features supported for each mobile device.

Installing the Flash Lite Updater

Before you install the Flash Lite Updater, make sure you have completed the following steps:

- Back up the current `AuthPlay.dll` file by renaming it **AuthPlay.dll.bak**. On Microsoft Windows 2000 and Windows XP, `AuthPlay.dll` can usually be found in the `C:\Documents and Settings\\Application Data\Macromedia\FX\MX\Configuration` directory. Older operating systems and Mac OS systems may use a different location for the Application Data directory.
- Back up the current `SAFlashPlayer.exe` file by renaming it **SAFlashPlayer.exe.bak**. On Microsoft Windows 2000 and Windows XP, the copy of `SAFlashPlayer.exe` can usually be found in the `C:\Program Files\Macromedia\FX\MX\Players` directory. Older operating systems and Mac OS systems may use a different location for this directory.

Along with the new Flash Lite features, the new updated players contain all of the features supported in Flash Player 6. To create Flash Player 6 content, you will not need to switch to the old players.

If you want to restore the Flash MX environment to its previous state and remove the Flash Lite enhancements, simply rename the `AuthPlay.dll.bak` and `SAFlashPlayer.exe.bak` files using their original names: `AuthPlay.dll` and `SAFlashPlayer.exe`. Flash MX also makes copies of the `AuthPlay.dll` file in each user's Application Data directory, and these files will need to be restored as well.

To install the Flash Lite Updater package on a Windows or a Macintosh computer:

1 Double-click one of the following installer programs:

- Windows English—FlashLiteInstaller.exe
- MacOS English—FlashLiteInstaller.hqx
- Windows Japanese—FlashLiteInstaller_ja.exe
- MacOS Japanese—FlashLiteInstaller_ja.hqx

2 Follow the onscreen instructions.

Flash Lite features

This section describes, at a high level, the features and restrictions for authoring a Flash Lite compatible movie. For different target devices there may be further feature restrictions or enhancements. Please consult the appropriate authoring guidelines document for details.

Navigation and key events

Macromedia Flash Lite reserves certain keys for navigation. The exact keys depend on the specific device. Consult the appropriate authoring guidelines document for details.

You can use ActionScript to retrieve input from other device keys. Again, the exact keys depend on the device. Consult your authoring guidelines document for details.

Text and fonts

Macromedia Flash Lite generally includes support for both device and embedded fonts. You can use embedded fonts to give you more control over the design of the movie, but this will increase the overall SWF file size. Using device text is generally limited to a single font style and size, depending on the device, but helps keep the SWF file size small.

For dynamic and input text fields, special text formatting is limited to justification (left, center, right) and color. The formatting options kerning and spacing, bold, and italic styles are not supported.

Flash Lite supports only single-line input text fields; there is no support for line-wrapping. Text is not edited in place, but in a separate page. Both Latin1-based and Shift-JIS character sets are supported, depending on the device.

Because most mobile devices do not support mouse-based navigation, it is not meaningful to set input text fields as selectable.

ActionScript functions and properties

Macromedia does not guarantee that all Flash 4 ActionScript and properties will run on all devices. Consult the appropriate authoring guidelines document for details.

The following general restrictions apply to using ActionScript when authoring for Flash Lite:

- You use the `add` command to concatenate strings.
- Button mouse events such as `dragOver`, `dragOut`, and `releaseOutside` cannot be used to trigger ActionScript attached to buttons. However, in addition to keypress events, the following events: `press`, `release`, `rollover`, `rollout`, can be used to trigger ActionScript when attached to buttons and accessed through key-based or focus-based navigation.
- Draggable movie clips (for example: `startDrag`, `stopDrag`, and `_dropTarget` properties) are not supported.
- URL encoding must be done manually using ActionScript. The `escape()` ActionScript function is not a Flash 4 function and is not available in Flash Lite.
- You use the `eq` operator to compare strings, and the `==` operator for numeric comparison.
- The `fscommand()` ActionScript function is partially supported. See your device's authoring guidelines document for details.
- Key events can only be attached to specific keys, depending on the device.
- The `MaxScroll` and `Scroll` text-scrolling properties are not supported.
- The default Quality level for Flash Lite during playback is medium and there is no support for bitmap smoothing.
- Sound functionality is limited to event sound.
- There is no synchronized audio (only event sound) so the `_soundBuffTime` property isn't supported.

Flash Lite sound

Macromedia Flash Lite does not support the standard Flash audio formats—Raw, ADPCM, and MP3. Instead, it supports MIDI and MFi (Melody Format for i-mode).

Only event sounds are supported. Looping of sounds is not supported.

See Chapter 3 “Developing Content for Flash Lite” for more information on developing with sound in Flash Lite.

CHAPTER 3

Developing Content for Flash Lite

This chapter describes the process of developing content for Flash Lite and the issues to be aware of when authoring Flash Lite compatible movies. A number of tutorials are provided to help you accomplish the following tasks:

- Creating a simple Flash Lite compatible movie (no sound)
- Associating a MIDI sound file with an ActionScript sound symbol
- Using the swflite.exe (swflite on the Mac) utility for simple sound substitution

To author for Flash Lite (general process):

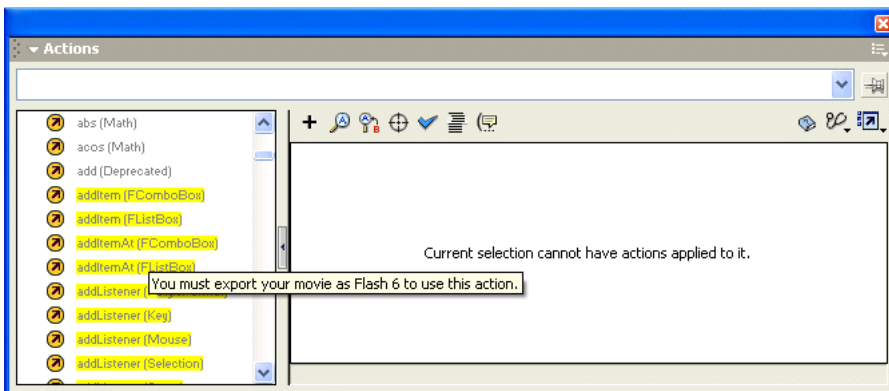
- 1 After you have installed the Macromedia Flash MX Updater for Flash Lite, launch Flash MX.
- 2 Configure Flash MX to author for Flash Lite:
 - Select File > Publish Settings and click the Flash tab.
 - In the Version pop-up menu, select Flash Player 4.
 - Click OK.

The modified Flash MX test movie player recognizes all Flash content, but specifically interprets Flash Player 4 content as Flash Lite. However, you can still preview standard Flash Player 4 content by modifying a flag in the user configuration file—FlashLiteUser.txt (see “Creating standard Flash 4 content” on page 20). For details on setting user configuration options see “User configuration file—FlashLiteUser.txt” on page 18.

After setting the Publish Settings to Flash Player 4, you can see what Flash 4 ActionScript is supported in the Actions window.

Note: Not all Flash 4 ActionScript is guaranteed to run on all devices. Consult your device’s authoring guidelines.

Open the Actions window (Window > Actions), click Index in the left pane and scroll down. ActionScript not supported in Flash 4 is highlighted in yellow. If you move your mouse over the highlighted text, a tooltip message appears letting you know what Flash version supports this ActionScript.



Example of supported and unsupported (highlighted yellow) Flash Player 4 ActionScript

- 3 Change the dimensions of the test movie player window to emulate the target device’s screen:
 - Select Modify > Document.
 - Change the dimension options to emulate the device’s screen; for example: 200 x 200 pixels.
 - Click OK.
- 4 Create your movie.

If you want to use event sound, see “Flash Lite sound” on page 14.

Note: Do not use unsupported Flash 4 ActionScript. Consult your device’s authoring guidelines document for detailed information on supported, unsupported, or partially supported ActionScript.

- 5 Select Control > Test Movie or Control > Test Scene to start the test movie player and test the movie. For more details on testing, see “Testing Flash Lite compatible content” on page 13.

An Output window appears containing information/error/warning messages. Carefully study all of the messages. Make changes as necessary and repeat this step.

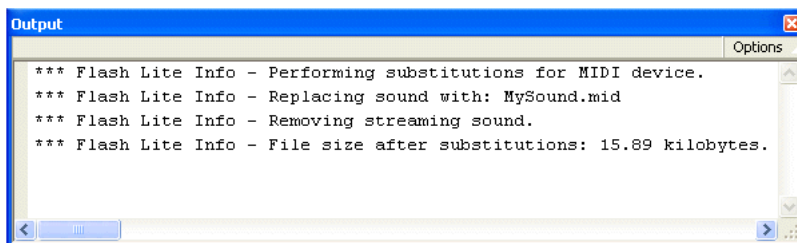
- 6 Export your finished movie (File > Export Movie) and close Macromedia Flash MX.

If the Flash Lite Updater is installed, the stand-alone player can play any version of a Flash movie including Flash Lite. However, it does not flag and report content that is not Flash Lite compatible.

The resulting SWF file is now ready to be uploaded to a distribution server for the consumer to view on the targeted device.

Testing Flash Lite compatible content

After the Flash Lite Updater is installed, the Flash MX test movie player recognizes any content published in Flash Player 4 as Flash Lite compatible content. When you select Control > Test Movie or Control > Test Scene, new information/warning/error messages specifically related to Flash Lite are displayed in a separate Output window:



Example of messages displayed in the modified test movie player Output window.

Whenever an unknown tag is encountered, error messages are displayed so the author can modify the content appropriately. Not all SWF content that is not Flash Lite compatible is flagged as being in error, such as invalid ActionScript and key input.

Since different handsets may have different size limitations, the Flash MX test movie player always informs the author about the size of the SWF movie each time it runs (see the last line in the graphic).

For a detailed explanation of all messages related to Flash Lite, see the appendix. This appendix lists all of the error messages that you might see when creating Flash Lite compatible content in general or for a specific deployment platform

Tutorial I—Creating a simple movie for Flash Lite (no sound)

In this tutorial you'll create a movie (without sound) that can run on a mobile device that is compatible with Flash Lite.

To create a Flash Lite compatible movie:

- 1 In Flash MX, create a new document and name it **tutorial fla**.
- 2 Select **Modify > Document**. Change the dimension options to emulate the device's screen (for example, 200 x 200 pixels). Click **OK**.
- 3 Select **File > Publish Settings**, then click the **Flash** tab. In the **Version** pop-up menu, select **Flash Player 4**. Click **OK**.
- 4 Select **Window > Common Libraries > Buttons**. Select a button and drag it to the Stage.
- 5 If the button is not selected on the Stage, click it once. In the **Actions** panel (select **Window > Actions** if the panel is not visible), type the following:

```
on (release)
{
    getURL("http://www.macromedia.com", _top);
}
```

- 6 Select **Control > Test Movie**.

Flash MX starts the test movie player. You can now interact with the movie. When you click the button, a browser window opens at <http://www.macromedia.com>.

- 7 Select **File > Publish** to save the SWF file as **tutorial.swf**. Since this file does not contain any sound, it is ready to run on the device's Flash player.

Note: This procedure is no different from one producing standard Flash Player 4 content. The real differences arise when you include sound in your movie.

Flash Lite sound

Flash Lite does not support standard Flash audio formats—Raw, ADPCM, or MP3. Instead, it supports MIDI and MFi (Melody Format for i-mode) formats. Since Flash MX does not natively support MIDI or MFi, you must temporarily substitute a sound in a recognized format such as MP3. To indicate the actual corresponding MIDI or MFi sound file, you associate it with an **ActionScript** sound symbol.

Note: Sound substitution for the Flash MX test movie player occurs automatically. Sound substitution for a device's Flash Player requires using the `swflite.exe` utility.

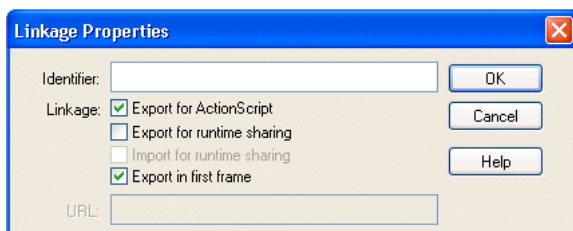
Tutorial II—Adding sound to your Flash Lite compatible movie

The following tutorial illustrates a simple case of associating an MIDI sound file with an **ActionScript** sound symbol so the Flash MX test movie player (not the device's Flash Player) can recognize and play it.

To associate a MIDI sound file with an ActionScript sound symbol:

- 1 Create a MIDI sound file named **MySound.mid**.
- 2 In Flash MX, create a new file and name it **FlashLiteSound fla**. Save it in the same directory as **MySound.mid**.
- 3 Select **File > Publish Settings > Flash**. In the **Version** pop-up list, select **Flash Player 4**. Click **OK**.

- 4 Select Window > Common Libraries > Buttons. Select a button and drag it to the Stage.
- 5 Double-click the new button. The Timeline should change to edit the button and display frames named Up, Over, Down, and Hit.
- 6 Select Insert > Layer to create a new layer. Select Modify > Layer and change the name of the layer to **Sound**.
- 7 Select the Down frame in the Sound layer and select Insert > Keyframe.
- 8 Select Window > Common Libraries > Sounds to open the Sounds Library window.
- 9 Select Window > Library to open the current document's library window.
- 10 Select a sound in the Sounds Library window and drag it to the document library window.
- 11 Associate the sound with the keyframe by doing the following:
 - Select the sound from the Sound pop-up menu in the Properties inspector. (Select Windows > Properties to display the Properties inspector if it is not already visible.)
 - Select Event from the Sync pop-up menu.
- 12 Associate the sound with the MySound.mid file by doing the following:
 - Select Window > Library and find the sound that you added earlier. Select the sound and right-click it to open the context menu. Select Linkage from the context menu. The Linkage Properties dialog box appears:



- Select the Export for ActionScript option.
- In the Identifier text box, enter the following text:

subst:MySound.mid

The prefix tag *subst*: indicates that the text identifies a sound substitution.

Note: Wildcards can be used in the filename; this is a more powerful, but more involved sound substitution procedure. For more details on using wildcards, see "Sound linkage identifier" on page 19.

- Click OK.
- 13 Select Control > Test Movie to start the Flash MX test movie player. Along with the test movie player window, the Output window displays information/warning/error messages related to Flash Lite. In the Output window, along with other messages, you should see the following line indicating sound substitution has taken place:


```
*** Flash Lite Info - Replacing sound with MySound.mid
```
 - 14 Click in the test movie player window. Since most Flash Lite compatible devices do not have a mouse, you should press Tab until the focus is on the button. Press the Enter key (on PC) or Return key (on Mac.) You should hear the sound from the MIDI file you created in Step 1.

The swflite.exe utility

The new Flash MX test movie player automatically substitutes MIDI/MFi sound files that have been associated with sound symbols. However, to properly publish a Flash Lite compatible movie, you must use a new utility—swflite.exe (swflite on the Mac).

After the Flash movie has been published or exported from Flash MX, to insert the MIDI or MFi sound(s) into the resulting SWF file, you must explicitly run the swflite.exe utility on the published SWF movie to insert sounds. After you run the swflite.exe utility, the resulting SWF file contains Flash Lite compatible sound. This movie can be played in the new stand-alone player, or a Flash Lite compatible mobile device.

The swflite.exe utility copies an existing published SWF movie and scans it for sound data with linkage identifiers that begin with the prefix *subst*. If found, the sound data is removed from the SWF file and replaced with either a MIDI or MFi sound file. The swflite.exe utility uses parameters found in FlashLiteUser.txt to derive the name of the MIDI or MFi sound file used for substitution. The copied file is then renamed to indicate the type of sound files it plays.

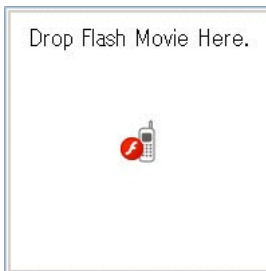
For a visual representation of the sound substitution process, see “Sound substitution flowcharts” on page 21.

Tutorial III—Using the swflite.exe utility for sound substitution

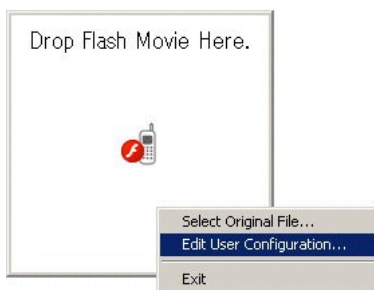
This tutorial uses the swflite.exe utility to create the same output as the previous tutorial. However, instead of viewing the results in the test movie player when you are done, the output from the swflite.exe utility is viewed in a stand-alone player.

To use swflite.exe for simple sound substitution:

- 1 In Flash MX, open the FlashLiteSound.fla file that you created in the previous tutorial.
- 2 Select File > Export Movie to create a published SWF file and save it as **FlashLiteSound.swf**.
- 3 Double-click the swflite.exe file. It should be in the same folder as the Flash MX program files. The swflite.exe application window appears:

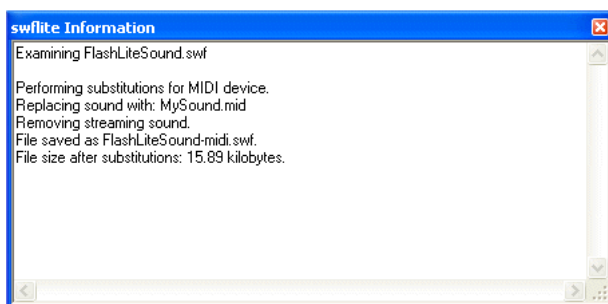


- 4 Right-click in the swflite.exe application window and select Edit User Configuration.



FlashLiteUser.txt appears in Notepad or some other text editor.

- 5 Edit the configuration file so that it only contains the following line:
`device_type=MIDI;`
- 6 Drag the FlashLiteSound.swf file onto the swflite.exe application window. The swflite.exe utility processes the file and messages appear in a separate window indicating that the sound substitution has taken place:



The swflite.exe utility makes a copy of the original published file, FlashLiteSound.swf, and modifies it to play MIDI files (in this case, one MIDI file—MySound.mid). The swflite.exe utility also renames the copied file to FlashLiteSound-midi.swf to indicate it plays MIDI files.

- 7 Right-click in the swflite.exe application window and select Exit to close the swflite.exe application.
- 8 Run FlashLiteSound-midi.swf in a stand-alone player. The selected library sound has been replaced with MySound.mid.

User configuration file–FlashLiteUser.txt

The previous tutorial demonstrated how the user configuration file can be used for elementary sound substitution. Now we'll discuss how to use the configuration file in more detail.

FlashLiteUser.txt is a simple ASCII text file consisting of one or more lines of text. A device's configuration takes the following format:

```
device_type=<device>;soundfile_wildcard=<wildcard
character>;soundfile_dir=<directory path>; output_suffix=<suffix>;
```

For example:

```
device_type=MIDI; soundfile_wildcard=m; soundfile_dir=C:\SWF\Music\MIDI;
output_dir=C:\SWF; output_suffix=-m.swf;
```

Each configuration attribute/value pair ends with a semicolon (;). The semicolon is necessary even when a carriage-return/line-feed (CR/LF) follows an attribute value. You can add CR/LFs to the configuration file to ease viewing and editing but they are ignored during processing. You can specify a value for an attribute more than once for a single device type but only the last attribute value is used.

A valid sound file must meet the following criteria:

- The file must exist and be readable.
- The file type must match the device's capabilities (for example: if the device supports MIDI, the file must be MIDI).
- If the sound file has device-specific extensions, all of the extensions must be valid for the device.

The following table lists the device attributes that you can set for each device in the FlashLiteUser.txt file.

| Device Configuration Attribute | Description |
|--------------------------------|--|
| <i>device_type</i> | The device type specifies the sound capabilities of the device and a Flash Player compatibility mode. Two base types are currently supported: MIDI and Standard (for Flash 4, not Flash Lite). For devices that play other sound formats than MIDI, consult your device's authoring guidelines. |
| <i>soundfile_wildcard</i> | The string used to replace the asterisk in the export identifier. If this tag is not defined, the default is the <i>device_type</i> string. For example, MIDI |
| <i>soundfile_dir</i> | The directory where device sound files are located. If not defined or overridden by the sound linkage identifier, the swflite.exe utility looks for the files in the same directory as the original file. |
| <i>output_suffix</i> | The suffix added to the device-specific SWF file if one is created. If omitted, the entry in the device configuration file is used, usually <i>device-name.swf</i> ; for example: MIDI.swf. |
| <i>output_dir</i> | The directory where the device-specific file is stored. If not defined, the file is stored in the same directory as the original file. |

User configuration file examples

The following examples demonstrate how the user configuration file can be used for sound substitution.

Example 1

```
device_type=MIDI;
```

The Flash MX test movie player plays a SWF file that uses MIDI files. If more device types are specified, the result is the same—the test movie player uses only the first device specified in the configuration file. Because the `soundfile_dir` attribute is omitted, sound files are searched for in the same directory as the original file.

Note: Since the test movie player creates and plays a SWF file dynamically, the `output_suffix` and `output_dir` attributes are irrelevant and hence, ignored.

Because the `output_suffix` attribute is omitted, `swflite.exe` uses `-<device_type>.swf` as the output file suffix (`-MIDI.swf`). Because only one device type is specified, using `swflite.exe` to process a file named `test.swf` creates only one output file: `test-MIDI.swf`. Because the `output_dir` attribute is omitted, the SWF file is stored in the same directory as the original file. Because the `soundfile_dir` attribute is omitted, sound files are searched for in the same directory as the original file.

Example 2

```
device_type=MIDI;output_suffix=-myMIDI.swf; soundfile_dir=/music/MyMIDI;  
output_dir=C:/MyFlashLiteMovies;
```

In this case, the Flash MX test movie player plays a SWF file that uses MIDI files. The SWF file's sound files are searched for in the `/music/MyMIDI` directory.

`Swflite.exe` outputs only one SWF file because only one device type is specified. The new SWF file has the suffix `-myMIDI.swf`. For example, launching a file named `test.swf` generates the file `test-myMIDI.swf`. The output file's sound files are searched for in the `/music/MyMIDI` directory. The output file is stored in the `C:/MyFlashLiteMovies` directory.

Sound linkage identifier

The linkage identifier string used in previous tutorials has the following format:

```
subst:<filename or partial filename with wildcard>.mid
```

The filename or partial filename can take the following forms:

- A simple filename—for example, `MySound.mid`
- A filename with a relative path—for example, `music/MySound-*.mid`
- A filename with an absolute path—for example, `C:/MyMusic/MySound.mid`

You can use an asterisk (“*”) wildcard and you can place it anywhere in the string. Asterisks can be used for batch sound processing with the `swflite.exe` utility. For more details on batch sound processing, see “Using linkage wildcards for batch sound substitution” on page 20.

If the file extension is not specified, the substitution process searches for a file without an extension. If that file cannot be found, the default sound file extension for the device is appended to the file name; for example—`.mid`, and the file is searched for again. For example, `subst:newsound` resolves to `newsound` or `newsound.mid` if no file named `newsound` exists and `.mid` is the default extension for the device's sound file type.

Linkage identifier examples

These linkage identifier examples assume that the configuration file contains one device type—MIDI—and the file suffix for MIDI files is .mid.

- `subst:newsound-*.mid`—replaces the sound with newsound-MIDI.mid.
- `subst:newsound.mid`—replaces the sound with newsound.mid.
- `subst:music/newsound-*`—replaces the sound with music/newsound-MIDI or music/newsound-MIDI.mid.

Using linkage wildcards for batch sound substitution

You can use the wildcard portion of the linkage identifier filename with `swflite.exe` to create multiple SWF files, each targeted to a specific device. When testing a movie, the Flash MX test movie player, when it encounters wildcards, uses only the first device type specified in the configuration file for sound substitution.

The `swflite.exe` utility, on the other hand, uses each device type specified in the configuration file (unless commented out) for sound substitution and generates a SWF file for each. Typically, a partial filename with a wildcard defines the base filename (for example, `MySong-*.mid`) and is resolved to a complete filename during the substitution process (for example, `MySong-devicetype1.mid`, `MySong-devicetype2.mid`, and so on).

For example, given a file named `test.swf` and the following substitution string for one sound:

```
subst:MySound-*.mid
```

And the following configuration file:

```
device_type=MIDI; output_suffix=-midi.swf;  
device_type=Device1; output_suffix=-Device1.swf;  
device_type=Device2; output_suffix=-Device2.swf;  
device_type=Device3; output_suffix=-Device3.swf;
```

The `swflite.exe` utility creates four SWF files:

- `test-midi.swf`
- `test-Device1.swf`
- `test-Device2.swf`
- `test-Device3.swf`

Note: The last three device type values used here—`Device1`, `Device2` and `Device3`—are not real device type values, but are presented simply for demonstration. Consult your device's authoring guidelines document for your device's real configuration values.

Creating standard Flash 4 content

You can also create standard Flash 4 content. By default, when configured to publish Flash 4 content (File > Publish Settings > Flash Tab > Version=Flash 4 Player), Flash MX creates content for Flash Lite. However, you can override this in the `FlashLiteUser.txt` file. Enter the following line in `FlashLiteUser.txt`:

```
device_type=Standard;
```

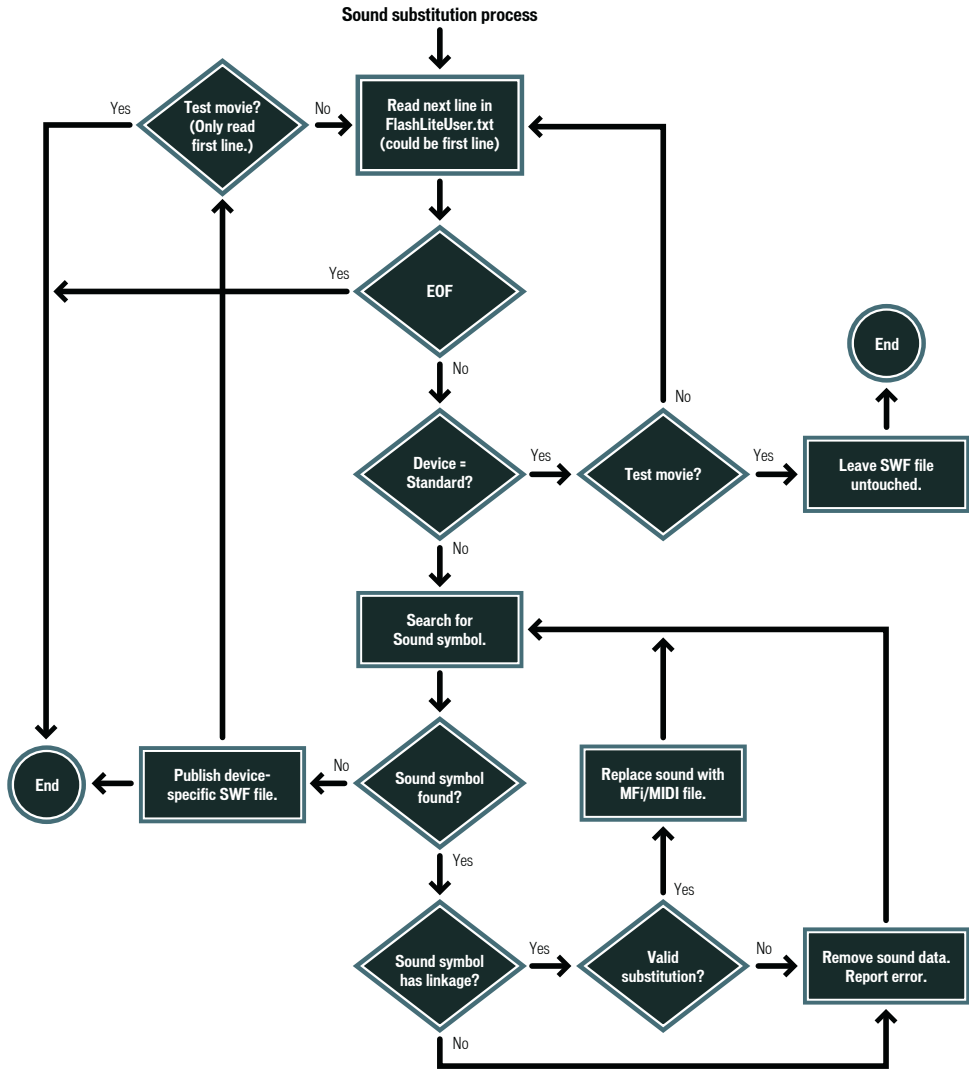
When the Flash MX test movie player or `swflite.exe` encounters this line, no substitution takes place and the output file is run in standard Flash 4 mode.

Sound substitution flowcharts

How sound substitution occurs in Flash Lite is probably the most complicated concept for developers to grasp. The following flowcharts are intended to supplement your understanding of how sound substitution occurs. The flowcharts are divided into the following three parts:

- Test movie sound substitution occurs when you select File > Test Movie.
- Swflite sound substitution presents the swflite.exe utility process.
- Sound substitution process is the process common to both test movie and the swflite.exe utility.





APPENDIX

Warning/Error Messages

This appendix lists all the possible error messages that you might see when creating content for Flash Lite in general. They may not all apply for a specific deployment platform.

| Message Identifier | Message | Explanation |
|--------------------|--|---|
| SWFS001 | Examining <SWF filename> | swflite.exe is parsing the input SWF file and examining it for sounds and invalid or restricted ActionScript commands. |
| SWFS002 | Config file does not exist or cannot be read. | The swflite user configuration or device configuration file does not exist or cannot be read. |
| SWFS003 | Config file has invalid format. | The swflite user configuration file is not in a valid format. swflite was unable to completely parse the file. |
| SWFS004 | Performing substitutions for <device type> device> | Message displayed by swflite to indicate the target device type for the device-specific SWF file being generated. This is an informational message for the convenience of the content developer. |
| SWFS005 | Unable to read file | swflite was unable to read the selected SWF file. The file may be corrupted. |
| SWFS006 | Invalid file format | The file passed to swflite is not a valid Macromedia Flash SWF file. |
| SWFS007 | File not in Flash Lite format. | The file passed to swflite is a valid Macromedia Flash SWF file, but was published in the wrong format. |
| SWFS008 | Unable to parse export tag: | swflite detected an export tag associated with a sound that was not in the expected format: subst:filename or subst:partial filename with wildcard. |
| SWFS009 | No sounds found. No substitutions necessary. | swflite did not find any sounds in the original SWF file. The file will not be modified and no device-specific files will be generated. |
| SWFS011 | Sound file <sound filename> not in correct format for device - removing sound. | During sound substitution, swflite detected that the selected sound file is not in the correct format for the device, so the sound will be removed from the device-specific SWF file. For example, a MIDI or MP3 file was selected for a device that only supports MFi. |
| SWFS012 | Removing streaming sound. | swflite has detected streaming sounds embedded in the input SWF file and is removing them. Streaming sound is not supported by Flash Lite . |
| SWFS013 | Replacing sound with: <sound filename> | swflite has located a valid device-specific sound file and is replacing the embedded sound with the device-specific sound. |

| Message Identifier | Message | Explanation |
|---------------------------|---|--|
| SWFS014 | Removing sound with no or invalid export tag. | swflite has found a sound in the original input SWF file with no export tag or an invalid export tag. In these cases the embedded sound is removed from the generated device-specific SWF file. |
| SWFS015 | Unable to substitute sound with file: <sound filename> -- removing sound. | swflite was unable to substitute the sound information from the specified sound file for the sound in the original input SWF file, so the sound will be removed from the device-specific SWF file. |
| SWFS016 | Detected loadMovie - will be ignored. | swflite detected that the SWF movie contains a loadMovie ActionScript command, which is not supported by the specified device's Flash Player. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS017 | Detected loadVariables - will be ignored. | swflite detected that the SWF movie contains a loadVariables ActionScript command, which is not supported by the specified device's Flash Player. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS018 | Detected GetURL - restrictions may apply. | swflite detected that the SWF movie contains a GetURL ActionScript command, which has some runtime restrictions when played by the specified device's Flash Player. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS019 | startDrag action not supported. | swflite detected that the SWF movie contains a startDrag ActionScript command, which is not supported by Flash Lite. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS020 | stopDrag action not supported. | swflite detected that the SWF movie contains a stopDrag ActionScript command, which is not supported by Flash Lite. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS021 | _droptarget property not supported. | swflite detected that the SWF movie contains a getProperty or setProperty ActionScript command referring to the _droptarget property, which is not supported by Flash Lite. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS022 | _highquality property not fully supported. | swflite detected that the SWF movie contains a getProperty or setProperty ActionScript command referring to the _highquality property, which is not fully supported by Flash Lite. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS023 | _soundbuftime property not supported. | swflite detected that the SWF movie contains a getProperty or setProperty ActionScript command referring to the _soundbuftime property, which is not supported by Flash Lite. No modifications will be made to the device-specific SWF file - this is just a warning. |

| Message Identifier | Message | Explanation |
|---------------------------|--|---|
| SWFS024 | scroll property not supported. | swflite detected that the SWF movie contains an ActionScript reference to the scroll property, which is not supported by the specified device's Flash Player. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS025 | maxscroll property not supported. | swflite detected that the SWF movie contains an ActionScript reference to the maxscroll property, which is not supported by the specified device's Flash Player. No modifications will be made to the device-specific SWF file - this is just a warning. |
| SWFS026 | Unable to write file | swflite was unable to write the output file. The <code>output_suffix</code> or <code>output_dir</code> directives in the user configuration file may be invalid, or the output directory may not be writable. |
| SWFS027 | File saved as <filename> | Message displayed by swflite to indicate the name it is using for the device-specific SWF file. |
| SWFS028 | File size after substitution: <nnn> kilobytes | Message displayed by swflite to indicate the size of the device-specific SWF file after substitution/removal of sounds. This is an informational message for the convenience of the content developer. |
| SWFS029 | File too large for device -- limit is <nnn> kilobytes. | Message displayed by swflite to indicate the size of the device-specific SWF file after substitution/removal of sounds exceeds the size limit of the target device. This is an informational message for the convenience of the content developer - the file will not be truncated or otherwise modified. |
| SWFS030 | Failed -- unable to substitute or remove sounds. | swflite was unable to process the original SWF file or create the device-specific SWF file. |
| SWFS032 | Detected FSComamnd - will be ignored. | swflite detected that the SWF movie contains a FSCommand ActionScript command, which is not supported by the specified device's Flash Player. No modifications will be made to the device-specific SWF file - this is just a warning. |
| FTPE001 | the key will not be processed: <key> keycode: <nnn> | While testing the movie, a key was pressed that is not accepted by the specified device's Flash Player - it will be ignored. |
| FTPA002 | FSCommand is ignored. | While testing the movie, a FSCommand ActionScript command was encountered. This command is not supported by the specified device's Flash Player and will be ignored. |
| FTPA003 | loadVariables is ignored. | While testing the movie, a <code>loadVariables</code> ActionScript command was encountered. This command is not supported by the specified device's Flash Player and will be ignored. |
| FTPA004 | loadMovie is ignored. | While testing the movie, a <code>loadMovie</code> ActionScript command was encountered. This command is not supported by the specified device's Flash Player and will be ignored. |

| Message Identifier | Message | Explanation |
|--------------------|--|--|
| FTPA005 | The call to GetURL for <URL> was ignored because there was more than one request per keypress. | While testing the movie, multiple ActionScript <code>getURL</code> commands were called during a keypress event. The specified device's Flash Player only allows one <code>getURL</code> command per keypress, so only the first command will be processed - the others will be ignored. |
| FTPA006 | The call to GetURL for <URL> was ignored because it was not associated with a keypress. | While testing the movie, a <code>getURL</code> ActionScript command was encountered outside of a keypress event. The specified device's Flash Player only allows <code>getURL</code> commands to be handled during a keypress event. Calls to <code>getURL</code> outside of a keypress event will be ignored. |
| FTPA007 | <code>getProperty</code> or <code>setProperty</code> not supported for: <property name> | While testing the movie, a <code>getProperty</code> or <code>setProperty</code> ActionScript command was encountered for a property that is not supported by the specified device's Flash Player. The command will be ignored. |
| FTPA008 | <code>getProperty</code> or <code>setProperty</code> not fully supported for: <property name> | While testing the movie, a <code>getProperty</code> or <code>setProperty</code> ActionScript command was encountered for a property that is not supported by the specified device's Flash Player. The command will be performed, but the results might not be as expected. |
| FTPA009 | <code>startDrag</code> and <code>stopDrag</code> are not supported. | While testing the movie, a <code>startDrag</code> or <code>stopDrag</code> ActionScript command was encountered. These commands are not supported by Flash Lite and will be ignored. |
| FTPS010 | Streaming Sound is unsupported. | While testing the movie, a streaming sound was encountered. Streaming sound is not supported by Flash Lite - it will be ignored. |
| FTPS011 | Only a single sound can be played at a time (no mixing). | While testing the movie, a sound was started while another sound was already playing. Flash Lite does not support sound mixing, so the first sound will be stopped to allow the second sound to play. |
| FTPS012 | Event sound was ignored because it was not associated with a keypress. | While testing the movie, an event was encountered outside of a keypress event. The specified device's Flash Player only allows event sounds to be handled during keypress events. Event sounds outside of a keypress event will be ignored. |