

Using ADOBE® FLASH® MEDIA LIVE ENCODER 3

© 2008 Adobe Systems Incorporated. All rights reserved.

Using Adobe® Flash® Media Live Encoder 3 for Windows®

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Adobe Premiere, and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

This work is licensed under the Creative Commons Attribution Non-Commercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/us/>

MPEG Layer-3 audio coding technology licensed from Fraunhofer IIS and Thomson.

Flash Media Encoder video is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved. <http://www.on2.com>

Speech compression and decompression technology licensed from Nellymoser, Inc. (www.nellymoser.com).

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

Chapter 1: Getting started

Flash Media Live Encoder basics	1
Encoding and streaming	5
Resources	7

Chapter 2: Configuration and automation

Configuring encoding options	8
Automating startup tasks	21
Configuring Flash Media Server authentication	24

Chapter 3: Working with timecode

About timecode	27
Configuring Flash Media Live Encoder for timecode	27
Using ActionScript to get timecode information	28

Index	29
--------------------	----

Chapter 1: Getting started

Adobe® Flash® Media Live Encoder is a software program for capturing, encoding, and streaming audio and video to Adobe® Flash® Media Server or the Adobe® Flash® Video Streaming Service.

Flash Media Live Encoder is designed for technical audio/video producers who seek a convenient and simplified workflow. It provides a unified interface to audio and video capture devices, basic editing tools, fine control over encoding parameters, and live broadcast using Flash Media Server or Flash Video Streaming Service providers. Flash Media Live Encoder can also be run from a command-line interface, making it possible to set up continuously running encoding sessions and integrate them with existing automated systems.

For more information about Flash Media Live Encoder and related products and services, see the following sites:

Flash Media Live Encoder www.adobe.com/go/learn_fme_main_en

Flash Media Server www.adobe.com/go/fms

Flash Media Solution Provider Program www.adobe.com/go/fmsp

Flash Video Streaming Service www.adobe.com/go/fvss

Flash Media Live Encoder basics

What's new in Flash Media Live Encoder 3

Flash Media Live Encoder 3 includes the following new features:

Multi-bit rate encoding Encode up to three streams at multiple bit rates and multiple output video sizes simultaneously. Multi-bit rate encoding is applicable for video streams, with or without audio.

Digital video recording Lets you start and stop media recording on the server within the encoding environment. When media recording is performed on the server, users can seek and pause within a live video stream. Digital video recording requires Flash Media Interactive Server 3.5. For more information, see the *Flash Media Server Developer Guide*.

Auto Adjust > Degrade quality The previous version of Flash Media Live Encoder, Flash Media Encoder 2.5, included a frame dropping capability to ensure optimal delivery of bits, even under suboptimal network conditions. In Flash Media Live Encoder 3, you can automatically adjust encoding quality by degrading quality or by dropping frames.

Embed system time Embed the system time as timecode. This feature is useful if you use a capture device that does not generate timecode information.

Use parameters in filenames and stream names Include parameters, such as bit rate and others, to generate meaningful stream names and filenames.

Chunking of output files Save an encoded media stream in chunks, instead of a single large file. You can create chunks by limiting the size and duration of output files.

New presets New presets for multi-bit rate encoding.

Support for new devices For a list of supported devices, see www.adobe.com/go/FMLE_DeviceMatrix.

Multi-core processing for VP6 Encoding is shared on all available processors on your encoding computer, which enables faster multi-bit rate encoding of VP6.

Updated user interface The user interface has been streamlined and updated to make it easier and more intuitive to use.

About GUI and command-line modes

You can run Flash Media Live Encoder from the graphical user interface (GUI) or from the command line. Some operations are available in only one of the two modes, and some are available in both modes.

Start the Flash Media Live Encoder GUI

❖ Do any one of the following:

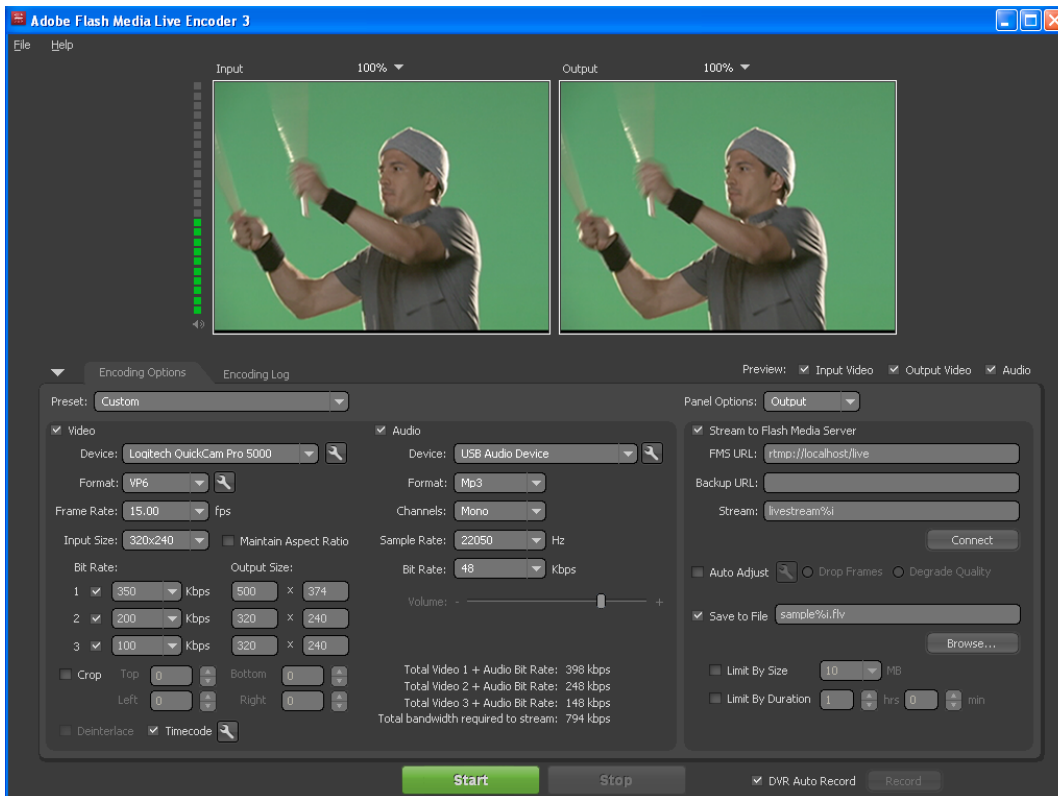
- Select Start > All Programs > Adobe > Adobe Flash Media Live Encoder 3.
- From a command prompt in your Flash Media Live Encoder installation directory (typically C:\Program Files\Adobe\Flash Media Live Encoder 3), enter the following:

```
FMLEcmd /g [/p [path\]profile.xml]
```

The `/g` parameter starts Flash Media Live Encoder in GUI mode. For information on the optional `/p` parameter, see [“Start Flash Media Live Encoder in command-line mode”](#) on page 4.

About the Flash Media Live Encoder GUI

The Flash Media Live Encoder GUI includes a preview of input video, output video, and the audio level. Before encoding, you can select the input/output size to view in the Video Preview window. During encoding, if you are encoding multiple streams simultaneously, you can also select which stream to view, using the Output Preview control below the output window. The control panel in the GUI is where you configure encoding options, view and configure encoding logs, and start and stop encoding sessions.



Flash Media Live Encoder 3 GUI, before encoding begins.

Previewing video and audio

The Preview panel in the Flash Media Live Encoder GUI includes the following:

- An audio level meter, indicating input audio volume, and a mute button
- An Input display of the video currently being captured
- An Output display, which reflects current video encoding settings. If you are encoding multiple streams simultaneously, you can select the desired stream to view. This control appears below the output window during encoding.

Note: Before encoding starts, the Output display reflects only crop and resize settings.

By default, the Preview panel is active and appears at actual output size.

Show or hide input video, output video, and audio

❖ Select or deselect the following Preview options:

- Input Video
- Output Video
- Audio

Mute or unmute the audio

❖ Click the speaker icon  below the audio level meter.

Adjust the zoom level

- ❖ Select a percentage from the pop-up menu on the upper right of either the Input or Output preview.

Note: Zoom level does not affect encoding or the stream that viewers see.

Show or hide the control panel

- ❖ Click the triangle next to the Encoding Options tab.

Start Flash Media Live Encoder in command-line mode

- ❖ From a command prompt in your Flash Media Live Encoder installation directory, enter the following:

```
FMLEcmd [/p [path\]profile.xml] [/l [path\]log_file] [/ap userID:password]
[/ab userID:password] [/t dd:hh:mm] [/r] /d [/s fml_e_uid] [/u fml_e_uid]
[/dvrstart [fml_e_uid] | /dvrstop fml_e_uid | /dvrstate fml_e_uid]
```

Parameter	Description
/h	Shows a list of all parameters and their usages.
/p [path\profile.xml]	(Optional) Specifies the path and filename of an XML profile to use. If you do not use this parameter, Flash Media Live Encoder uses the default profile, startup.xml. For more information, see “Saving encoding profiles” on page 20.
/g	Starts Flash Media Live Encoder in GUI mode. This option cannot be used with any parameters except for /p.
/l	(Optional) Specifies the path and filename of an encoding log file. The settings in the log file override the settings specified in the currently loaded profile. For more information, see “Using the Encoding Log” on page 6.
/ap userID:password	(Required if the primary server requires authentication) Specifies an authenticating user ID and password for the primary server. Because the password is shown in plain text, take care when entering your password.
/ab userID:password	(Required if the backup server requires authentication) Specifies an authenticating user ID and password for the backup server. Because the password is shown in plain text, take care when entering your password.
/t dd:hh:mm	(Optional) Instructs Flash Media Live Encoder to restart after the time interval specified by dd:hh:mm. For more information, see “Scheduling automated maintenance restarts” on page 22.
/r	(Optional) Registers the encoding session to restart at an operating system relaunch after a system closure or crash. For more information, see “Configuring Flash Media Live Encoder to launch at system startup” on page 23.
/d	(Recommended) Validate the encoding profile and, if any values are invalid, use the default values instead.
/s fml_e_uid	Stops the Flash Media Live Encoder session (instance) specified by the identifier fml_e_uid. Prints a list of all running sessions if the identifier is invalid or unspecified.
/u fml_e_uid	Un-register the Flash Media Live Encoder session (instance) specified by the identifier fml_e_uid. Unregistered sessions do not start automatically when the operating system is launched. Prints a list of all registered sessions if the identifier is invalid or unspecified.
/dvrstart [fml_e_uid] /dvrstop fml_e_uid /dvrstate fml_e_uid	Controls recording during streaming. For more information on these options, see “Start or stop digital video recording from the command line” on page 6.

Flash Media Live Encoder launches without displaying the GUI and immediately starts an encoding session.

Encoding and streaming

Start and stop encoding from the GUI

1 To start Digital Video Recording as soon as encoding begins, select DVR Auto Record, next to the Start/Stop buttons at the bottom of the window.

2 Click the Start button in the Flash Media Live Encoder window.

Note: If both the Stream To Flash Media Server and Save To File options in the Encoding Options tabbed panel are deselected, the Start button is unavailable.

3 If an authentication dialog box appears, enter your user ID and password.

4 To start Digital Video Recording manually, click Record.

5 To stop encoding, click Stop.

Note: Although the encoding stops, the server connection persists until you click Disconnect. For information on connecting to and disconnecting from the server, see “Connect and disconnect” on page 17.

Starting and stopping encoding sessions from the command line

When you start Flash Media Live Encoder from the command line, it automatically starts encoding at the same time. For encoding sessions that were started from the command line, you can show a list of any running sessions and stop an encoding session.

Show a list of encoding sessions that were started from the command line

❖ From a command prompt in your Flash Media Live Encoder installation directory, enter the following:

```
FMLEcmd /s
```

Stop an encoding session that was started from the command line

❖ From a command prompt in your Flash Media Live Encoder installation directory, enter the following:

```
FMLEcmd /s fmle_UID
```

When the session is streaming to a server, use the format *fms_URL+stream_name* for *fmle_UID*. For example, to stop an encoding session that is running on the server *fms.myserver.com* with the application named *live*, and the stream named *livestream*, you would enter the following:

```
FMLEcmd /s rtmp://fms.myserver.com/live+livestream
```

Note: When using the /s parameter, the stream_name portion of fmle_UID cannot contain a plus sign (+), because + is used as a delimiter in fmle_UID.

When the session is being saved to file but is not streaming to a server, use the filepath and filename for *fmle_UID*. For example:

```
FMLEcmd /s "C:\Documents and Settings\abc\My Documents\My Videos\sample.flv"
```

Note: When encoding multiple streams, use the first stream name or filename for the stream name or filename portion of fmle_uid.

Start or stop digital video recording from the command line

Flash Media Interactive Server 3.5 and later supports DVR media players. When a publisher and server-side DVR application exist, DVR media players let users rewind and pause a live stream and resume playback from the paused location. Flash Media Live Encoder supports this functionality by letting you start and stop recording within the encoding environment. To enable DVR functionality, record the live stream to the server as it streams.

To manage recording from the command-line, use the options in the following table. The *fmle_uid* parameter specifies the encoding session streaming to a Flash Media Server instance. The value of *fmle_uid* is `fms_URL+stream_name`. For example, an encoding session that is running on the server `fms.myserver.com` with the application named `live` and the stream named `livestream` has the following *fmle_uid*: `rtmp://fms.myserver.com/live+livestream`.

Option	Description
<code>/dvrstart [fmle_uid]</code>	<p>Immediately starts recording of the encoding session specified by the <i>fmle_uid</i> parameter.</p> <p>The <i>fmle_uid</i> parameter is optional. When <i>fmle_uid</i> is not used, a new encoding session is launched. Recording begins as soon as encoding begins.</p> <p>When <i>fmle_uid</i> is used, Flash Media Live Encoder issues the start command to the specified session, which starts recording on the Flash Media Server. If an invalid <i>fmle_uid</i> value is used, Flash Media Live Encoder displays all registered sessions.</p>
<code>/dvrstop fmle_uid</code>	<p>Immediately stops recording of the encoding session specified by the <i>fmle_uid</i> parameter.</p> <p>If an invalid or no <i>fmle_uid</i> is used, Flash Media Live Encoder displays all registered sessions.</p>
<code>/dvrstate fmle_uid</code>	<p>Returns the state of the encoding session specified by <i>fmle_uid</i>. The state consists of whether recording is on or off and the duration of the recording.</p> <p>If an invalid or no <i>fmle_uid</i> is used, Flash Media Live Encoder displays all registered sessions.</p>

DVR functionality is applicable only when streaming media to Flash Media Interactive Server 3.5 and later. DVR operations are performed on both primary and backup (if applicable) servers and for all streams in multi-bit rate encoding.

Using the Encoding Log

The Encoding Log provides detailed information on encoding sessions and is an essential tool for troubleshooting. If you are trying to resolve an issue with Flash Media Live Encoder, support personnel may request that you examine the screen log or submit a log file. With the Encoding Log, you can view and save logs of encoding sessions and view status information about sessions.

In the GUI, the Encoding Log tab appears automatically when encoding starts. During encoding, logged events and encoding status messages appear in real time.

The log section on the left side of the panel lists important encoding events, file system events, and error messages as they occur. Each event and message includes a time stamp.

The Statistics section on the right side of the panel shows information on what the video and audio encoders are actually achieving, given the current settings and encoding environment. On the Encoding tab, you can see the current (for the last 1 second) encoding time, actual bit rates and frame rates, and whether any video frames are being dropped. You can also view the averages achieved for the current encoding session. This information can help you refine your encoding settings for best results. The Publishing tab shows information about the bandwidth at which Flash Media Live Encoder is publishing to the server, the RTMP buffer length, and the number of frames dropped by automatic adjustment (if the Auto Adjust option is enabled).

Note: To set the limit for the RTMP buffer at which point auto adjustment is triggered, see the Auto Adjust option under “Configure video encoding options” on page 9.

Enable logging from the GUI

- 1 Click the Encoding Log tab.
- 2 Select Log To File.
- 3 By default, logs are saved in your My Documents\My Videos folder. To save the log in a different folder, click Log Directory and select a folder or create a new one.

When logging is enabled, Flash Media Live Encoder creates a new log file each time you start encoding. The log filename has the following format:

```
fmle_session_yyyymmdd_hhmmss.log
```

The date and time are the local system date and time. For example, the name of a log file for a Flash Media Live Encoder session that began at 10:30 on 5 November 2009 would be as follows:

```
fmle_session_20091105_103000.log
```

Enable logging from the command line

- ❖ From a command prompt in your Flash Media Live Encoder installation directory, start Flash Media Live Encoder using the `/l` parameter in the following format:

```
FMLEcmd /l [path\]log_file
```

Here *path* (optional) and *log_file* are the path and filename of the log file you want to create. If *path* is not specified, the log file is saved in your My Documents\My Videos folder.

For example:

```
FMLEcmd /p c:\Profiles\my_profile.xml /l c:\Logs\my_log.txt
```

Note: If you include the `/l` parameter, the log you specify overrides the log preference specified in the XML profile. For more information on XML profiles, see “[Saving encoding profiles](#)” on page 20.

Resources

Flash Media Live Encoder product site This site, at www.adobe.com/go/learn_fme_main_en, provides information about system requirements, contains links to release notes, and lets you download Flash Media Live Encoder.

Flash Media Live Encoder Forums This site, at www.adobe.com/go/fme_forum, provides forums for Flash Media Live Encoder users to share information and ideas.

Flash Media Server Support Center This site, at www.adobe.com/go/flashmediaserver_support_en, provides Tech Notes and the latest information about Adobe® Flash® Media Server.

Flash Media Server Developer Center This site, at www.adobe.com/go/flashmediaserver_desdev_en, provides articles, tips, and samples for creating Flash Media Server applications.

Flash Developer Center This site, at www.adobe.com/go/learn_flv_devnet_en, provides articles, tips, and samples for working with video in Flash.

Feature Request/Bug Report Form This site, at www.adobe.com/go/wish, provides a form to request new features or suggest modifications to existing features, including reports of bugs.

Chapter 2: Configuration and automation

Configuring encoding options

You access most options and controls in Flash Media Live Encoder on the Encoding Options tabbed panel, which appears by default when you start Flash Media Live Encoder. This panel includes options for adjusting video, audio, and output settings.

If no capture device is connected to the system when you start the application, Video and Audio are deselected in the Encoding Options panel and encoding options are grayed out. To activate the options, connect a device and select Video and Audio.

Select a preset bandwidth option

Flash Media Live Encoder includes the following encoding preset options. The presets specify video and audio settings that are appropriate for your bandwidth. Low, medium, and high bandwidth presets are available for VP6 video. Medium and high bandwidth presets are available for H.264 video. Presets for encoding multi-bit rate streams are also available.

Preset option	Video format	Video bit rate (Kbps)	Size (pixels)	Frame rate (fps)	Audio format	Audio channel	Sample rate (Hz)	Audio bit rate (Kbps)
Low bandwidth (150 Kbps)	VP6	100	176 x 144	15	Nellymoser	Mono	11025	22
Medium bandwidth (300 Kbps)	VP6	200	320 x 240	15	MP3	Mono	22050	48
High bandwidth (800 Kbps)	VP6	650	320 x 240	30	MP3	Stereo	44100	128
Medium bandwidth (300 Kbps)	H.264 Main profile, level 2.1	200	320 x 240	20	MP3	Mono	22050	48
High bandwidth (800 Kbps)	H.264 Main profile, level 3.1	650	320 x 240	30	MP3	Stereo	44100	128
Multi-bit rate - 2 streams	VP6	300; 1000	320 x 240; 640 x 480	25	MP3	Stereo	44100	128
Multi-bit rate - 3 streams	VP6	150; 300; 650	176 x 144; 320 x 240; 640 x 480	25	MP3	Stereo	44100	128
Multi-bit rate - 2 streams	H.264 Base profile, level 3.1	300; 700	320 x 240; 640 x 480	25	MP3	Stereo	44100	128
Multi-bit rate - 3 streams	H.264 Base profile, level 3.1	150; 300; 650	176 x 144; 320 x 240; 640 x 480	25	MP3	Stereo	44100	128

- ❖ Choose an option from the Preset menu. Make your selection based on the bandwidth that will be available at broadcast time.

Note: If your capture device does not produce video or audio in the preset values, Flash Media Live Encoder selects the closest available value.


About custom bandwidth options

An additional option in the Preset menu, Custom, is selected when you manually adjust any settings. If you are using custom settings when you quit Flash Media Live Encoder, the next time you start the program, it applies your custom settings. If you choose a preset option and exit the program, however, any custom settings are lost. For information on saving encoding settings for easy retrieval, see “Saving encoding profiles” on page 20.

Configure video encoding options

- ❖ On the Video side of the Encoding Options tabbed panel, configure any of the following encoding settings:

Video When deselected, the encoding session does not include video.

Device Specifies the video capture device to use. The menu lists any video devices that are attached to the system and that are compatible with Microsoft® DirectShow®. Click the settings icon  next to a selected device to configure its settings. For a list of compatible capture devices, see www.adobe.com/go/FMLE_DeviceMatrix.

Format Specifies the video codec to use: On2 VP6 or H.264.

The following video/audio combinations are supported:

Codec	Nellymoser	MP3	AAC*	HE-AAC*
H.264		X	X	X
VP6	X	X		
*AAC and HE-AAC are available when you install the MainConcept AAC Encoder plug-in.				

Note: To add support for AAC and HE-AAC, purchase the MainConcept AAC Encoder plug-in for Adobe Flash Media Live Encoder. See the [MainConcept website](#).

See also “[Configure settings for VP6 video](#)” on page 12 and “[Configure settings for H.264 video](#)” on page 13.

Note: If you encode H.264 files, the output files are fragmented. You must flatten the output files before you can edit them in a video editing tool such as Adobe® Premiere® Pro, or before you can play them in Adobe® Media Player. Adobe provides a command-line flattening tool for this purpose; see http://www.adobe.com/go/learn_fmle_tools.

Input size Specifies the video dimensions for the input video, in pixels.

Frame rate Specifies the target frame rate in frames per second (fps).

Bit rate Specifies the video bit rate in kilobytes per second (Kbps). Select a bit rate or enter a value. For VP6, you can enter a value from 0 to 20000. For H.264, the value depends on the selected level and ranges from 1 to 20000 Kbps.


You can specify a bit rate for a single stream or multiple bit rates to encode up to three streams simultaneously. The audio settings must be the same for all streams.

Note: When encoding multiple-bit rate streams, you can select which bit-rate stream to show in the Output window, after you click Start.

Output size Specifies the video dimensions for final output. Setting the output size is useful if your video capture device doesn’t provide the output dimensions you require, or if you crop the input video and want to restore specific output dimensions. To avoid distortion when you adjust the value of either the width or the height, select Maintain Aspect Ratio. See “[Resize video output](#)” on page 11.

Crop When selected, activates controls for cropping input video, to create focus or draw attention to a section of the image. See “[Crop video output](#)” on page 11.

Deinterlace When selected, activates the video deinterlacing filter, which may improve video quality by applying a simple vertical blur. This option is available only when the input video height is greater than 480 pixels and the input video width is greater than 576 pixels. The deinterlacing filter is applied when encoding starts.

Timecode When selected, embeds timecode in the video stream. You can choose to embed either the system time as timecode or device timecode. To configure timecode settings, click the settings icon  next to the Timecode box. See “[Configure timecode settings](#)” on page 11.

Configure multi-bit rate encoding

- 1 Under the Preset menu, select an option for multi-bit streaming for either VP6 or H.264 video, if desired. Otherwise, enter your own custom encoding values.

Under Bit Rate and Output Size, the values for each stream are numbered 1-2 or 1-3. The numeric value is the index of the encoded stream. You can use this index value, represented as %i, in the stream name and filename as a differentiator. Other parameters are available as well.

- 2 Under Output, specify a stream name and filename.

When encoding multi-bit rate streams, you can either use semicolons to separate unique stream names or you can use a single stream name with parameters. For the filename, you can use parameters to differentiate the streams.

Note: For the stream name, if you use both semicolons and parameters, the parameters are ignored.

- 3 If desired, chunk the output file into smaller files by size and duration.
- 4 Click Start. After encoding starts, you can preview streams 1, 2, or 3 in the output window by clicking the menu near the bottom of the output window.

See also

[“Use parameters in filenames and stream names”](#) on page 15

[“Configure output options”](#) on page 14

Crop video output

- 1 In the Encoding Options tabbed panel, select the Crop option.
- 2 To specify the crop area, do any of the following:
 - Drag any of the square handles that appear at the edges and corners of the Input video preview.
 - Enter a pixel value in any of the Top, Bottom, Left, and Right boxes.
 - Click the Up or Down arrow next to a pixel value box.
- 3 To move the crop selection box to another position, place the pointer inside the bounding box and drag.


Note: If overlay is disabled on your graphics card, the square handles and bounding box are unavailable.

Resize video output

- 1 In the Encoding Options tabbed panel, go to the Output Size section.
- 2 To maintain the width and height ratio when adjusting the value of one dimension, select Maintain Aspect Ratio.
- 3 Enter a pixel value in the Width or Height box.

Note: Although PC-based players may accommodate unexpected video sizes and dimensions, specialty players (such as cell phone players) have stricter requirements.

Configure timecode settings

- 1 In the Encoding Options tabbed panel, select the Timecode option.
- 2 Click the settings icon  next to the Timecode option.
- 3 In the Timecode Settings dialog box, configure any of the following settings:

Frame Interval Specifies the frame interval at which timecode is embedded into the video stream. The default is 15.

Embed System Time as Timecode Embeds the system time as AMF data tags from the encoding computer. Use this option if you want embedded timecode but your capture device cannot generate it.

The system date is embedded in the `<sd>` tag in the format `dd-mm-yyyy` (day-month-year). The system time is embedded in the `<st>` tag in the format `hh:mm:ss.ms` (hour:minute:second:millisecond). The hour is in 24-hour clock format.

Device Timecode Embeds the timecode that is generated by the capture device in the stream. If the selected video device is not capable of generating timecode, this option is not available.

Note: If the selected video device is not a FireWire device, you may need to provide a DLL file in the appropriate format for that device to embed a timecode. Install the DLL in the FMLEInstallDir\Timecode folder (typically C:\Program Files\Adobe\Flash Media Live Encoder 3\TimeCode).

Vertical Line Number Specifies the vertical line from where the device timecode is extracted.

Burn Device Timecode When selected, the external video device burns a timecode on the video image, and the timecode is visible in the Input and Output preview.


Note: If the selected video device can generate timecode but cannot burn timecode on the video, the Burn Timecode option is unavailable.

Position Specifies where the device timecode is burned, by row and column. If the encoding device does not support burning of timecode, this option is disabled.

Timecode settings are stored in the XML profile. For information on XML profiles, see “[Saving encoding profiles](#)” on page 20. For more information on using timecode, see “[Working with timecode](#)” on page 27.

Configure settings for VP6 video

Important: These options pertain only to video encoded with the VP6 codec. Use these options carefully. The settings affect the way that Flash Media Live Encoder uses your computer’s processor and memory. These settings persist until the next time you readjust them.

- 1 With VP6 selected, click the settings icon  next to the Video Format box.
- 2 In the Advanced Encoder Settings dialog box, configure any of the following settings:

Keyframe Frequency Specifies the minimum frequency for inserting keyframes in the output video stream. Keyframes are encoded without reference to any other frames in the stream, and they provide a recovery point if the viewer's connection is dropped or rebuffered.

Quality Specifies image quality. Depending on the nature of the content (such as static or dynamic) and processor capability, this setting can affect the output, as noted in each setting. The Best Quality setting attempts to retain image sharpness but is processor-intensive; if CPU resources become exhausted while encoding, Flash Media Live Encoder compensates by dropping frames. In general, the first setting in this list that your system can support without dropping frames is best. To confirm that frames aren’t being dropped, monitor the Status display while encoding. For more information on the Status display, see “[Using the Encoding Log](#)” on page 6.

Note: This setting may depend on the CPU Usage setting. For example, if you choose Best Quality but set CPU Usage to Very Low, Flash Media Live Encoder drops frames.

Noise Reduction The encoder attempts to blur what it perceives as unwanted artifacts in the video stream. This setting specifies how aggressively to apply the technique.

Note: Noise reduction is a matter of personal preference; do not adjust this setting unless you determine that there is a quality advantage.

Datarate Window Specifies how much data rate fluctuation to allow while encoding.


Flash Media Live Encoder can recognize similarities between adjacent frames. It allocates fewer bandwidth resources to encoding sequences that have fewer changes in the frames (such as a sitting interview using a fixed camera), reserving the resources for sequences with more motion (such as a sports event). With a larger Datarate Window value, Flash Media Live Encoder holds more data in its buffer—it can process more video frames, resulting in better compression. If you believe that the viewers may have suboptimal network connections, you may want to reduce this value. Conversely, for encoding streams that are saved to file for local playback only (that is, streams that are never transmitted over a network), use the Huge setting.

The best way to determine the practical effects of this setting is through real-world testing of the viewer experience.

CPU Usage Specifies the CPU resources to allocate to the video encoder. You achieve best encoding results from a dedicated machine. Special circumstances may require that you reserve some processing power for a second task.

Configure settings for H.264 video

Important: These options pertain only to video encoded with the H.264 codec. Use these options carefully. The settings affect the way that Flash Media Live Encoder uses the CPU and memory. These settings persist until the next time you readjust them.

- 1 With H.264 selected, click the settings icon  next to the Video Format box.
- 2 In the Advanced Encoder Settings dialog box, configure any of the following settings:

Profile Select either the Baseline or Main profile.

The Baseline profile is commonly used in videoconferencing, mobile applications, or other applications which operate with limited processing power, storage capacity, and bandwidth. The Main profile is appropriate for applications that broadcast media, such as network broadcasting, and requires 1.5% more processing power than the Baseline profile.

Level Flash Media Live Encoder supports all available levels. Here are some examples of the resolution and frame rate achieved with different levels:

Type	Level	Maximum resolution (width x height)	Frame rate (frames per second)
Standard definition	1.3	352 x 288	30 fps
Standard definition	3	720 x 576	25 fps
High definition	3.1	1280 x 720	30 fps
High definition	5.1	1920 x 1088	120.5 fps

Encoding with the H.264 codec is CPU-intensive. Adobe recommends that you select a level and video size that enables CPU usage to remain below 75% during encoding.


For more information on H.264 profiles and levels, see commonly available resources about the H.264 video codec, such as the ISO/IEC 14496-10 standard on the [ISO website](#) and [Wikipedia](#).

Keyframe frequency The interval at which to insert keyframes. The default is 5 seconds, which means a keyframe is inserted every 5 seconds.

Configure audio encoding options

- ❖ On the Audio side of the Encoding Options tabbed panel, configure any of the following encoding settings:

Audio When deselected, the encoding session does not include audio.

Device Lists any audio devices that are attached to the system and are DirectShow compatible. Click the settings icon  next to the device to configure the device's settings.

Format Specifies the audio format. Flash Media Live Encoder includes support for Nellymoser and MP3.

Note: To add support for AAC and HE-AAC, purchase the MainConcept AAC Encoder plug-in for Adobe Flash Media Live Encoder. See the [MainConcept website](#).

The following video/audio combinations are supported:

Codec	Nellymoser	MP3	AAC*	HE-AAC*
H.264		X	X	X
VP6	X	X		
*AAC and HE-AAC are available when you install the MainConcept AAC Encoder plug-in.				

Channels Specifies the channel option, mono or stereo. If you have the MainConcept AAC Encoder plug-in, parametric stereo is also available.

Sample Rate Specifies the audio sample rate, as determined by the audio format and sample rate that you select.

Bit Rate Specifies the audio bit rate in Kbps, as determined by the audio format and sample rate that you select.

Volume Specifies the input audio volume level.

Configure output options

You can use the output options to stream live video to a server, save your broadcast to a file, or both.

- 1 From the Panel Options menu on the right side of the Encoding Options tabbed panel, select Output.
- 2 Configure any of the following options:

Stream To Flash Media Server When selected, enables streaming to the specified servers when encoding starts.

FMS URL Specifies the fully qualified URL, or *RTMP string*, of the primary Flash Media Server computer. The URL must begin with `rtmp://` or `rtmpt://` and include your server domain name and the streaming application to use. You can optionally include a specific application instance and port number. For example:

```
rtmp://fms.myserver.com:80/live/livestream
```

Here, `fms.myserver.com` is the Flash Media Server computer host name, `80` is the port number to connect to (optional), `live` is the name of an application on the host (required), and `livestream` is the instance of the application to connect to (optional).

For more information on RTMP strings, see the Flash Media Server documentation.

Backup URL Specifies the fully qualified URL of a backup server and application (optional). If you specify a backup server, Flash Media Live Encoder streams to both servers.

Stream Specifies a name for the video stream. If you are encoding multiple-bit rate streams, you can either separate each unique stream name with a semicolon (;) or use parameters to create stream names. For more information on using parameters, see “[Use parameters in filenames and stream names](#)” on page 15.

If you do not use parameters for naming, be sure that the number of unique stream names you provide, separated by a semicolon, match the number of streams being encoded.

Note: If you use both semicolons and parameters in stream names, parameters will not be applied. The characters that represent parameters are treated as literal characters in the stream name.

Auto Adjust When required by network limitations, Flash Media Live Encoder automatically adjusts the quality of video by either degrading video quality or dropping frames. For more information, see “[Automatically adjust video quality](#)” on page 16

Save To File When selected, saves the output to a file. Click Browse to specify a different filename and folder. You can use parameters in filenames. For more information, see “[Use parameters in filenames and stream names](#)” on page 15.

You can limit the size or duration of output files to chunk large encoding sessions into multiple files. For more information, see “[Save files in chunks](#)” on page 16.

The following table shows to which file format media is saved, depending on the codecs that are used:

Codec	Nellymoser	mp3	AAC/HE-AAC	Video only
H.264	N/A	F4V	F4V	F4V
VP6	FLV	FLV	N/A	FLV
Audio only	FLV	FLV	F4V	N/A

***Note:** If you use a filename that exists in the selected folder, Flash Media Live Encoder renames the existing file so that the newly saved file does not overwrite it. For example, if `sample.flv` exists, Flash Media Live Encoder renames it `sample.0.flv` and saves your new file as `sample.flv`. If both `sample.flv` and `sample.0.flv` exist, `sample.flv` is renamed `sample.1.flv`, and so on. If the wrong filename extension is used, Flash Media Live Encoder changes it to the correct filename extension.*

Use parameters in filenames and stream names

When saving encoded streams to file, you can use parameters in filenames and stream names. During multi-bit rate encoding, it is especially helpful to include the bit rate or other information in the name to distinguish the content.

***Note:** For stream names, you can use either parameters or separate unique stream names with a semicolon, but not both together. If you use both parameters and semicolons, Flash Media Live Encoder treats the characters used to represent parameters as literal characters in the stream name.*

To use parameters, enter an optional identifier (name) of your choice, followed by one or more parameters.

Parameter	Description
%i	The index of the bit rate. This value is meaningful when encoding streams at multiple bit rates. The first index value is 1. Flash Media Live Encoder supports up to three streams at a time, so the index values are 1, 2, and 3. Adobe recommends that you include the index parameter in the stream name. The index parameter can differentiate streams/files that otherwise have identical values for bit rate, sample rate, and so on.
%v	The video bit rate.
%f	The output frame size.
%a	The audio bit rate.
%s	The audio sample rate.
%b	The total bit rate (that is, %v+%a).

You can use multiple parameters in a single name. When using multiple parameters, if desired, you can separate parameters with underscores to make the resulting names easier to read. For example, say that you want to create an FLV file whose name reflects the video bit rate of 156 Kbps and the output frame rate of 24 fps. In the Save To File box, enter `myFLVFile_%v_%f`. The resulting filename is `myFLVFile_156_24`.

***Note:** While it is not required to enter an identifier, such as the identifier `myFLVFile` used in the example, an identifier of your choosing can help differentiate files and streams.*

Save files in chunks

For long encoding sessions, you can *chunk* a stream into several files by limiting the size or duration of each output file. Specify the desired size or duration for each chunk in the Output panel of the Encoding Options tab. If you limit by both size and duration, Flash Media Live Encoder chunks the file when one of the limits is reached (whichever limit is reached first). Files are chunked at keyframes to ensure a smoother transition between chunks.

When chunking files, Flash Media Live Encoder appends the name that you enter in the Save To File box (or that you specify with parameters) with a date and time stamp. The date and time stamp indicates the system time when the chunking of that file occurred. Each chunked file has a different date and time stamp.

The date is in the form YYYYMMDD. The time is in the form HHMMSS, using a 24-hour clock. The date and time are separated by an underscore. The final filename is in the format *name.date_time*.

Note: The calculation of the file size or duration might not be 100% accurate. Do not use the entered values for any purpose other than to chunk files.

Automatically adjust video quality

Note: Automatic adjustment is applicable only to single-bit-rate encoding and to streaming using Flash Media Server. It is not applicable when encoding multi-bit rate streams or when saving to a file only.

Flash Media Live Encoder monitors the amount of data in the RTMP buffer and compares it to the maximum buffer size specified in the encoding profile.

Sometimes, the rate at which Flash Media Live Encoder streams media to Flash Media Server exceeds the rate at which the network can handle the data. This scenario happens if network latency is high or network bandwidth is lowered for some reason. Excess data is stored in the RTMP buffer. If the buffer value exceeds the maximum size specified in Auto Adjust settings, Flash Media Live Encoder can automatically adjust the video quality to ensure uninterrupted streaming.

Automatic adjustment applies only to video streams, with or without audio. It does not apply to audio-only streams.

During an encoding session, you can enable or disable automatic adjustment in the user interface but not in command-line mode. In command-line mode, enable or disable automatic adjustment before you start the encoding session.

To turn on automatic adjustment, select the Auto Adjust option in the Output panel. You can choose to degrade video quality or drop video frames.

Degrade quality

This setting degrades the quality of the video by reducing the bit rate until data can be streamed without exceeding the specified RTMP buffer size. This feature is not enabled by default.

Click the settings icon to configure the following quality settings:

- Maximum buffer size. Once this value is exceeded, Flash Media Live Encoder enacts quality degradation. For higher bandwidth, use a higher number. For lower bandwidth, use a lower number.
- Minimum video bit rate. The minimum acceptable bit rate at which to stream media. If network conditions deteriorate to the point that a lower bit rate is required to stream video, frame dropping is enabled.
- Preserve per-frame quality. Reduces the frame rate to as low as 50% of the specified value.

Both the published stream and the saved output file reflect any quality degradation that takes place. Once the RTMP buffer value no longer exceeds the maximum value, Flash Media Live Encoder reapplies the values that were originally set in the encoding profile, in steps and as the network conditions allow.

Drop frames

This setting drops video frames from the RTMP buffer until data can stream without exceeding the RTMP buffer size. When Auto Adjust is enabled, Drop Frames is enabled by default.

Click the settings icon to configure the maximum buffer size. Once this value is exceeded, Flash Media Live Encoder starts dropping frames. For higher bandwidth, use a higher number. For lower bandwidth, use a lower number.

Dropping frames affects streaming video only. Subscribers usually notice when frames are dropped because the video appears jerky or appears to stutter.

Flash Media Live Encoder drops frames at intervals, rather than all at once. It drops backward-facing video frames between keyframes. For VP6 video, I-frames are dropped. For H.264 video, I-frames, B-frames, and P-frames may be dropped. If necessary, keyframes (VP6) or IDR frames (H.264) may also be dropped.

When the RTMP buffer size returns to a normal value, Flash Media Live Encoder stops dropping video frames.

Connect and disconnect

- 1 To connect to the URLs specified in FMS URL and Backup URL (if used), click the Connect button in the Encoding Options tabbed panel.

Note: If the Stream To Flash Media Server option is deselected, the Connect button is unavailable.

- 2 If the server requires authentication, enter your user ID and password in the dialog box that appears.

Clicking the Connect button establishes a connection to the URLs specified in FMS URL and Backup URL (if used) and changes the button label to Disconnect.

- 3 To disconnect, click Disconnect.

For information on how to start and stop encoding, see “[Encoding and streaming](#)” on page 5.

Recording live streams for DVR media players

Flash Media Interactive Server 3.5 adds support for DVR media players. When a publisher and server-side DVR application exist, DVR media players let users rewind and pause a live stream and resume playback from the paused location. Flash Media Live Encoder supports this functionality by letting you start and stop recording within the encoding environment.

To enable DVR functionality, record the live stream to the server as it streams. You can start and stop recording from Flash Media Live Encoder in the following ways:

- In the GUI:
 - Start recording when encoding begins. Select the DVR Auto Record option, next to the Start/Stop encoding buttons. You can pause and restart recording at any time during the session.
 - Start recording after encoding begins. Click the Record button during the session to start recording.
- On the command line. See “[Start or stop digital video recording from the command line](#)” on page 6.

DVR functionality is applicable only when streaming media to Flash Media Interactive Server 3.5. You need to add server-side ActionScript to handle the recording and the correct client-side ActionScript to your media player.

DVR operations are performed on both primary and backup (if applicable) servers and for all streams in multi-bit rate encoding.

See also

[Flash Media Server DVR application](#)

[Adding DVR features to live video](#)

About metadata

In a media player application, metadata shows the viewer information about the stream, provides searchable keywords, and injects useful, human-readable identifiers into the binary stream. Flash Media Live Encoder handles metadata differently depending on the file encoding:

- For FLV files, Flash Media Live Encoder embeds metadata elements in the streams and media files that it creates.
- For F4V files, Flash Media Live Encoder embeds only custom metadata—metadata that you add through the GUI—in the file. It sends standard metadata with the stream only.

Flash Media Live Encoder embeds the following custom metadata fields for FLV and F4V files, which you can populate with information for your media stream:

- author
- copyright
- description
- keywords
- rating
- title

Standard metadata differs slightly for FLV files and F4V files, and for whether the media is a file or a live stream; see “[FLV metadata](#)” on page 18 and “[F4V metadata](#)” on page 19.

You can create your own metadata fields and delete any existing fields. See “[Add metadata values](#)” on page 19 and “[Create and delete metadata fields](#)” on page 20.

***Note:** Metadata is not available in FLV or F4V files if Flash Media Live Encoder terminates abnormally during an encoding session.*

FLV metadata

- audiochannels
- audiocodecid
- audiodatarate
- audiodevice
- audioinputvolume
- audiosamplerate
- creationdate
- duration (media files only)
- fmleversion (Flash Media Live Encoder version) (media files only)
- framerate
- height
- lastkeyframetimestamp (media files only)

- lasttimestamp (media files only)
- presetname
- videocodecid
- videodatarate
- videodevice
- videokeyframe_frequency
- width

F4V metadata

Standard metadata for F4V files is sent with the encoded stream only. It is not embedded in the F4V file. Custom metadata—metadata fields that you add through the Flash Media Live Encoder GUI—is embedded in the F4V file. The custom metadata fields are author, copyright, description, keywords, rating, and title. Standard metadata fields are as follows:

- audiochannels
- audiocodecid
- audiodatarate
- audiodevice
- audioinputvolume
- audiosamplerate
- avcprofile
- avclevel
- creationdate
- framerate
- height
- presetname
- videocodecid
- videodatarate
- videodevice
- videokeyframe_frequency
- width

Add metadata values

- 1 From the Panel Options menu on the right side of the Encoding Options tabbed panel, select Metadata.
- 2 Add values to any of the metadata fields. For example, to record a copyright date of 2010, enter 2010 in the Copyright metadata field.

Create and delete metadata fields

Any custom metadata fields that you add through the Flash Media Live Encoder GUI are embedded in an F4V file. Standard F4V metadata is not embedded.

- 1 From the Panel Options menu on the right side of the Encoding Options tabbed panel, select Metadata.
- 2 To add a new metadata field to the list, click the + icon and enter a field name. Do not add field names that are already included in the standard metadata set.
- 3 To delete an existing field from the list, select the field and click the - icon.

Saving encoding profiles

When you close Flash Media Live Encoder after running it from the GUI, all of the settings from that session are saved in the default encoding profile, `startup.xml`. The next time you start Flash Media Live Encoder—either from the GUI or from the command line without the `/p` parameter—Flash Media Live Encoder validates the profile and applies the settings in the `startup.xml` file. If the values of any settings aren't valid, Flash Media Live Encoder uses the default values for those settings instead.

The `startup.xml` file is stored in `Drive:\Documents and Settings\username\Application Data\Adobe\Flash Media Live Encoder 3`. For example, if your default Microsoft® Windows® drive is `C:\` and your username is `abc`, your `startup.xml` file is stored in `C:\Documents and Settings\abc\Application Data\Adobe\Flash Media Live Encoder 3`.

Note: To get the default encoding profile set by Adobe, close Flash Media Live Encoder, delete or rename `startup.xml`, and restart the application. Flash Media Live Encoder launches with the default values for encoding and creates a new `startup.xml` file. If desired, you can save this file under another name, such as `default.xml`, for later use.

Once you determine encoding and streaming settings that work well in your environment, it's a good idea to save them in a custom profile for easy retrieval. This is especially important if you will be running encoding sessions from both the GUI and the command line. You can configure encoding and streaming settings using the GUI, and then save your settings as a custom profile to use from the command line.


For example, imagine that you're running one encoding session in GUI mode and another in command-line mode. You make some changes in the Encoding Options and close the GUI instance. Flash Media Live Encoder updates the `startup.xml` profile to reflect the changed settings. If you stop the command-line encoding session and then restart it without specifying a custom profile, the new session uses the updated `startup.xml` profile from the GUI session. Thus, to avoid unexpected changes to your encoding profile, you may want to first configure settings in the GUI, save them as a custom profile, and then use the custom profile whenever you run Flash Media Live Encoder from the command line.

Note: When using the command line, use the `/d` parameter to validate and correct a profile.

Flash Media Live Encoder saves the following settings in an XML profile:


- Video and audio encoding settings, except for hardware device properties controlled by Windows or another application
- Flash Media Server connection settings, such as server URLs, stream name, and automatic startup and reconnection
- File archiving settings, including filename and path and chunking settings
- Metadata elements and values
- Preview and zoom settings
- Logging settings

Flash Media Live Encoder does *not* save the following settings in a profile:

- Hardware device properties. Flash Media Live Encoder provides access to the properties dialog boxes of your video and audio devices, but it cannot prevent or reverse changes made by other applications. To verify that your preferred device settings are in place, click the setting icon  next to the Device box on the Encoding Options tab.
- Deselected settings. For example, if you resize the video output but deselect the Resize option before saving a profile, the resized dimensions are not saved.

Save a profile

- 1 Select File > Save Profile.
- 2 In the Save As dialog box, enter a descriptive name for the file. Do not name the file startup.xml, because this name is reserved for storing settings from the last session.
- 3 Click Save.

 *If you use multiple Flash applications, servers, or content distribution network providers, it's a good idea to save a separate profile for each environment.*

Open a profile from the GUI

- 1 Disconnect from Flash Media Server, if applicable.
- 2 Select File > Open Profile.
- 3 Select a profile from the Open dialog box.

If any values in the profile are invalid, Flash Media Live Encoder uses the default values instead. To see which values were changed, check the Encoding Log tab.

Open a profile from the command line

- ❖ From a command prompt in your Flash Media Live Encoder installation directory, start Flash Media Live Encoder using the `FMLEcmd` command and the `/p` option, in the following format:

```
FMLEcmd /p [path to profilename.xml] /d
```

The *path* (optional) and *profilename* parameters are the path and filename of the XML profile that you want to open. The `/d` option validates that profile.

For example:

```
FMLEcmd /p c:\Profiles\my_profile.xml /d
```

If you do not specify a profile in the command line, Flash Media Live Encoder loads the default profile, startup.xml. If any values in the profile are invalid, Flash Media Live Encoder uses the default values instead. To see which values were changed, check the Encoding Log tab.

Automating startup tasks

Flash Media Live Encoder includes several settings and strategies that you can use to make sure that your session streams continuously, 24 hours a day, 7 days a week.

Specify reconnection settings

If the network connection is lost during an encoding session, by default Flash Media Live Encoder attempts to reconnect to the server every 5 seconds until the connection is restored. When the network connection is restored, Flash Media Live Encoder resumes encoding. You can adjust the number of reconnection attempts and the interval between attempts in an XML profile.

- ❖ Open the XML profile that you want to change in an XML or text editor, and add the following lines:

```
<reconnectinterval>  
<attempts>num_attempts</attempts>  
<interval>interval</interval>  
</reconnectinterval>
```

Here *num_attempts* is the number of times that Flash Media Live Encoder should attempt to reconnect and *interval* is the number of seconds that Flash Media Live Encoder should wait between reconnection attempts. You can use integers or decimal values in either parameter, but in *num_attempts*, decimal values are truncated. For both parameters, zero and negative values mean that Flash Media Live Encoder will not attempt to reconnect. In this case, the log file includes a message indicating that Flash Media Live Encoder did not attempt reconnection.

Scheduling automated maintenance restarts

Flash Media Live Encoder is certified to run continuously for 7 days. If you plan to run it for longer than 7 days, it's a good idea to schedule an automated maintenance restart at regular intervals. Once Flash Media Live Encoder has run continuously for the amount of time you specify, it automatically stops, restarts, and continues encoding to the same session.

Schedule an automated maintenance restart from the XML profile

- ❖ Open the XML profile in an XML or text editor, and add the following lines:

```
<restartinterval>  
<days>dd</days>  
<hours>hh</hours>  
<minutes>mm</minutes>  
</restartinterval>
```

Schedule an automated maintenance restart from the command line

- ❖ From a command prompt in your Flash Media Live Encoder installation directory, start Flash Media Live Encoder using the `/t` option in the following format:

```
FMLEcmd /t dd:hh:mm
```

For example, with the following command, Flash Media Live Encoder restarts after 4 days, 10 hours, and 30 minutes:

```
FMLEcmd /p c:\Profiles\my_profile.xml /ap myid:mypassword /t 04:10:30
```

If the restart interval is specified in the XML profile as well as on the command line, the command-line setting overrides the XML profile setting.

Configuring Flash Media Live Encoder to launch at system startup

Imagine that you intend to run an encoding session continuously, but during the session the computer on which Flash Media Live Encoder is running crashes. As soon as that computer restarts, you'll want Flash Media Live Encoder to restart and continue encoding in the same session. You can configure Flash Media Live Encoder to restart automatically at system startup by registering the encoding session. Any crash-related errors are noted in the log file when the encoding session restarts. For more information on log files, see "Using the Encoding Log" on page 6.

Note: The registration of encoding sessions with the operating system are stored in a file called *OsLaunch.dat*. Restricted users cannot edit this file if it was created by another user. For example, when a user registers an encoding session, that user creates the *OsLaunch.dat* file. A restricted user cannot then register or unregister an encoding session on the same computer because they do not have permissions to edit *OsLaunch.dat*.

Register an encoding session

- 1 If the Flash Media Live Encoder Launcher is not already included in your Windows Scheduled Tasks, add it:
 - a In Windows, select Start > All Programs > Accessories > System Tools > Scheduled Tasks.
 - b In the Scheduled Tasks window, double-click Add Scheduled Task.
 - c In the Scheduled Task Wizard, click Next, and then browse to your Flash Media Live Encoder installation directory and open FMELauncher.exe.
 - d Select When My Computer Starts and click Next.
 - e Enter your user credentials, click Next, and then click Finish.
- 2 From a command prompt in your Flash Media Live Encoder installation directory, start Flash Media Live Encoder using `/r` as the last option in the command. For example:

```
FMLEcmd /p c:\Profiles\my_profile.xml /l c:\my_log.txt /r
```

If authentication is required, use the `/ap` and `/ab` parameters and specify your user credentials.

Any encoding sessions that are registered for relaunch are noted in the *OsLaunch.dat* file in *Drive:\Documents and Settings\All Users\Application Data\Adobe\Flash Media Live Encoder 3*. The Flash Media Live Encoder Launcher checks this file at system startup and launches the session that has a registered entry.

Show a list of all registered encoding sessions

- ❖ From a command prompt in your Flash Media Live Encoder installation directory, enter the following:

```
FMLEcmd /u
```

Unregister an encoding session

- ❖ From a command prompt in your Flash Media Live Encoder installation directory, enter the following:

```
FMLEcmd /u fmle_UID
```

If the session is streaming to a server, use the format *fms_URL+stream_name* for *fmle_UID*. For example, to unregister an encoding session that is running on the server *fms.myserver.com* with a Flash application named *live*, and the stream *livestream*, you would enter the following:

```
FMLEcmd /u rtmp://fms.myserver.com/live+livestream
```

Note: When using the `/u` option, the *stream_name* portion of *fmle_UID* cannot contain a plus sign (+), because + is used as a delimiter in *fmle_UID*.

If the session is being saved to a file but is not streaming to a server, use the *FLV* or *F4V* path and filename for *fmle_UID*. For example:

```
FMLEcmd /u "C:\Documents and Settings\abc\My Documents\My Videos\sample.flv"
```

The entry for the encoding session that you specify is removed from the OsLaunch.dat file in *Drive:\Documents and Settings\All Users\Application Data\Adobe\Flash Media Live Encoder 3*. At the next system startup, the encoding session does not restart.

Configuring Flash Media Server authentication

If you administer a Flash Media Server computer, you may want to require publishers who connect to your server with Flash Media Live Encoder to authenticate with approved credentials. To do this, you can either install and configure the Adobe Flash Media Live Encoder 3 Authentication Add-In, or create and configure your own authentication module. Using either of these options, you create accounts for users who connect to a server from Flash Media Live Encoder.

Configure the Adobe Flash Media Live Encoder 3 Authentication Add-in

1 Download the add-in from www.adobe.com/go/learn_fms_authaddin_en.

2 Install the add-in on your Flash Media Server computer.

The add-in installs the Server Access Adapter and the Flash Media Live Encoder User Management utility, which you can use to add, update, remove, and check user passwords.

3 From a command prompt, change to the following directory:

```
FlashMediaServerInstallDir\conf
```

4 To show a list of commands for the user management utility, enter the following:

```
users
```

Authentication interfaces

The Authentication Add-in contains the following interfaces. With this release, all interface names have been changed from “IFme” to “IFmle”. A summary of the interfaces follows; for more information, see the sample application included with the Authentication Add-in.

IFmleAccessAdaptor The interface of a Client Access Adaptor.

Command	Description
IFmleAccessAdaptor()	Constructor
getVersion(int& iMajor, int& iMinor, int& iMicro)	Gets the module version. The version number contains three integers, representing the major, minor, and micro version numbers, respectively.
getDescription()	Gets the module description.
process(IFmleConnection* pCon)	Processes a connection event.

IFmleAccessContainer The interface to the adaptor’s container (represents Flash Media Live Encoder).

Command	Description
IFmleAccessContainer()	Constructor
getVersion()	Gets the container (Flash Media Live Encoder) version.
getDescription()	Gets the server description string.
isHeadless()	Indicates whether Flash Media Live Encoder is operating through the command line, that is, without the user interface launched.
onComplete(IFmleConnection* pCon)	Called by the adaptor when it completes the processing of a connect event.

IFmleConnection The interface used to request a connection. To request that the access adaptor process a connection, provide an instance of this class to the access adaptor. This object is allocated by Flash Media Live Encoder. It is cleaned up once a connection is established or terminated. It carries all the information necessary for the access adaptor to perform authentication and allows the URL to be prepared for connect/reconnect.

Command	Description
getState()	Gets the state of the Connection event.
enum State	The state of the Connection event. Possible values are FMLE_CONNECT, FMLE_REJECTED, FMLE_SUCCESS.
getURL()	Gets the RTMP URL of the connection that is being made with Flash Media Server.
getRejectInfo()	Gets the Reject Message received from Flash Media Server.
setURL(const char* sURL)	Sets the RTMP URL. This URL is used to make a connection to Flash Media Server. The length of the URL cannot exceed STRING_VALUE_BUFFER_LEN, or 1024 characters.
setAction(Action a, const char* sReason)	Takes one of two possible actions, FMLE_CONTINUE or FMLE_ABORT.FMLE_CONTINUE: Perform this action if you want Flash Media Live Encoder to try to connect to Flash Media Server after configuration changes have been made. You can explain the changes in the sReason parameter.FMLE_ABORT: Cancel the connection process to Flash Media Server.
enum Action	The type of Action. Possible values are: FMLE_ABORT, FMLE_CONTINUE
getAdaptorSpecific()	Gets the value passed in the /ap or /ab parameter in command-line mode. This function is applicable only when Flash Media Live Encoder is used in command-line mode.

Configure a custom authentication module

- 1 Write a custom authentication module that includes a client-side adaptor DLL and a server-side adaptor DLL. Write the adaptors according to the behaviors and guidelines discussed in the following sections.
- 2 On the computer running Flash Media Live Encoder, copy your client-side DLL to the *FMLEInstallDir*\AccessModule folder (typically, C:\Program Files\Adobe\Flash Media Live Encoder 3\AccessModule).
- 3 Install the server-side DLL on the computer that is running Flash Media Server.

Client and server adaptor interaction

The user agent for Flash Media Live Encoder 3 is FMLE/3.0. If you are using a custom server authentication module, change the header in the connect message to 'FMLE/3.0'. Alternatively, use the Adobe Flash Media Live Encoder 3 Authentication Add-in.

Note: Previous versions of the Authentication Add-in are not supported for Flash Media Live Encoder 3. If you used the Authentication Add-in with previous versions of Flash Media Live Encoder, update to the version included with Flash Media Live Encoder 3.

In every transaction, the server verifies the parameters appended in the query string that the client sends. The server-side adaptor identifies itself by prefixing the following signature substring in each rejection reason string:

```
' [ authmod=authmod_name ] :'
```

Initially, Flash Media Live Encoder identifies the server by its `authmod` signature and loads the corresponding client DLL from the `FMLEInstallDir\AccessModule` folder. For example, if the value of `authmod_name` is `myauth`, then Flash Media Live Encoder loads the `myauth.dll` file. Subsequent interactions between client and server rely on the `authmod` signature. The server's response determines whether to abort or continue the authentication process. Flash Media Live Encoder verifies the `authmod` signature and acts as a medium for client-server interaction. If there is any discrepancy in the signature, Flash Media Live Encoder takes the appropriate action.

Authentication requirements

When authenticating, users must provide a username, password, and RTMP string.

Flash Media Live Encoder can publish to multiple servers. If the servers have different authentication credentials, Flash Media Live Encoder may need to prompt for passwords more than once. To ensure that users enter the correct user ID and password, make sure that the password prompt indicates which server is being connected to.

Chapter 3: Working with timecode

When you work with a compatible capture device, you can easily configure Adobe Flash Media Live Encoder to embed timecode into video streams. Client-side applications can get the timecode information by using ActionScript®.

About timecode

The Society of Motion Picture and Television Engineers (SMPTE) establishes a set of cooperating standards that are used to label individual frames of video and film with pertinent information. These standards are referred to as *SMPTE timecode*. SMPTE timecode consists of binary-coded strings of 80 bits that contain information about the hour, minute, second, and frame (expressed as *HH:MM:SS:FF*), the type of timecode (non-drop or drop-frame), and 32 user-definable bits.

Flash Media Live Encoder can extract SMPTE timecode that is generated by compatible devices, embed it into the encoded video stream, and pass it to Flash Media Server. You can save that stream locally as an FLV file or F4V file. Subscribing clients that connect to the published stream can get the embedded timecode information.

In addition, Flash Media Live Encoder 3 can embed the system time from the encoding computer as AMF data tags. The system time can substitute for a timecode if your capture device doesn't generate timecode.

Supported timecode devices

Flash Media Live Encoder does not generate SMPTE timecode, but it can extract timecode. Use a compatible capture device and timecode DLL to generate timecode information. Flash Media Live Encoder can extract timecode from devices that use an MSDV driver. Your Flash Media Live Encoder installation includes a timecode DLL for MSDV devices.

See also

[“Configure timecode settings”](#) on page 11

[List of supported devices](#)

Supported timecode formats

For timecode generated by a capture device, Flash Media Live Encoder can extract timecode in the following formats:

Vertical Interval Timecode (VITC) is recorded into the vertical blanking interval (VBI) of the video signal.

Burnt-in Timecode (BITC) burns numbers into the video image, making the timecode human-readable.

See also

[“Configure timecode settings”](#) on page 11

Configuring Flash Media Live Encoder for timecode

Enable the timecode feature by selecting the Timecode option in the Encoding Options tabbed panel.

The device timecode option is enabled if 1) the capture device can generate timecode and can burn timecode values in video frames, and 2) the correct DLL is present in the *[RootInstall]/Flash Media Live Encoder 3/TimeCode* directory.

If you don't have a device that can generate timecode, you can embed the system time as timecode.

See also

[“Configure video encoding options”](#) on page 9

[“Configure timecode settings”](#) on page 11

Using ActionScript to get timecode information

Flash Media Live Encoder contains a special built-in handler, `onFI`, that subscribing clients can use in their ActionScript code to access timecode information. The following client-side ActionScript code shows how to get timecode information using the `onFI` handler. The object `ns` is the NetStream object. You can get timecode and system date and time information, if timecode and system date and time were embedded in the stream, by accessing the `tc`, `sd`, and `st` properties of the info object that is passed as an argument to `onFI()`:

```
ns.onFI = function(infoObj) {
    var timecode:String; // Timecode generated by capture device
    var systemDate:String; // System date "sd" embedded by Flash Media Live Encoder
    var systemTime:String; // System time "st" embedded by Flash Media Live Encoder

    for( i in infoObj)
    {
        if(i == "tc")
            timecode = infoObj.tc; //string formatted hh:mm:ss:ff
    }
    {
        if(i == "sd")
            systemDate = infoObj.sd //string formatted as dd-mm-yy
    }
    {
        if(i == "st")
            systemTime = infoObj.st //string formatted as hh:mm:ss.ms
    }
}
```

Index

A

- Adobe Flash Media Encoder
 - forums 7
 - GUI overview 2
 - installation directory 2
 - new features 1
 - starting 2, 4
- Adobe Flash Media Encoder 2
 - Authentication Add-in 24
- Adobe Flash Media Server resources 1, 7
- Adobe Flash Video Streaming Service 1
- audio
 - bit rate 14
 - device 13
 - encoding options 13
 - format 13
 - muting 3
 - previewing 3
 - sample rate 14
 - volume 14
- authentication
 - command-line parameters 4
 - configuring 24
 - from the GUI 5
- AutoAdjust 14, 17

B

- backup server
 - specifying 14
- bandwidth options
 - custom 9
 - preset 8, 9
- bit rate
 - audio 14
 - preset bandwidth option 9
 - video 10

C

- command-line options
 - about 2
- connecting to servers 17
- control panel, showing and hiding 4
- CPU usage, specifying 13
- cropping video output 10, 11

D

- datarate window, specifying 12
- deinterlace filter, enabling 10
- disconnecting from servers 17
- documentation
 - additional resources 7

E

- encoding options
 - about 8
 - audio 13
 - custom 9
 - H.264 13
 - preset 8, 9
 - timecode 11
 - video 9, 12, 13
 - VP6 12
- encoding profiles. See profiles.
- encoding sessions
 - logging 6
 - reconnection settings, specifying 22
 - registering 23
 - restarting 22, 23
 - starting 4, 5
 - stopping 5
 - unregistering 23

F

- file output, configuring 14
- FMELauncher.exe 23
- FMS URL, specifying 14
- FMS. See Adobe Flash Media Server
- forums 7
- frame dropping, automatic 14, 17
- frame interval, specifying 11
- frame rate, preset bandwidth option 9
- FVSS. See Flash Video Streaming Service

H

- H.264 encoding options 13

I

- input video, previewing 3

L

- logging
 - about 6
 - enabling 7
 - file format 7
 - statistics 6

M

- metadata
 - about 18
 - adding values 19
 - creating and deleting fields 20
 - default fields 18
- muting audio 3

O

- output video, previewing 3

P

- previewing video and audio 3
- profiles
 - about 20
 - custom 20
 - default 20
 - opening 21
 - reconnection settings, specifying 22
 - restart settings, specifying 22
 - saving 21
 - timecode settings 12

R

- reconnection settings, specifying 22
- registering an encoding session 23
- resizing video output 10, 11
- restarting encoding sessions
 - about 22
 - scheduling 22
- RTMP string 14

S

- sample rate, specifying 14
- servers, connecting and disconnecting 17
- startup tasks, automating 21
- startup.xml 20

T

timecode

- about 27
- advanced options 11
- Burn Timecode option 12
- client-side ActionScript 28
- compatible devices 27
- embedding 10

U

- unregistering an encoding session 23

V

- vertical line number, specifying 12

video

- advanced options 12, 13
 - bit rate 10
 - CPU usage 13
 - cropping output 10, 11
 - datarate window 12
 - deinterlacing 10
 - device 9
 - dimensions 10
 - encoding options 9
 - frame rate 10
 - keyframe frequency 12
 - noise reduction 12
 - previewing 3
 - quality 12
 - resizing output 10, 11
 - size 9
 - timecode options 10, 11
- volume, specifying 14
 - VP6 encoding options 12

X

- XML profiles. See profiles.

Z

- zoom level, adjusting 4