

Overview

Sothink Video Encoder Engine for Adobe Flash (Linux Version) provides you a complete and creative video hosting solution for video uploading and sharing, especially a perfect solution for building your own video hosting site just like YouTube and many other popular video sharing web sites. It does best to meet your personal or business video hosting needs, and can be applied to Web servers, blogs, forums and other interactive areas.

This Video to Flash converter can convert almost all popular formats of video to Flash Video (FLV) format. You can do advanced video editing, batch conversion, quality control and other tasks with CGI or any other server-side scripting language. To help you to build your own video hosting site, we also provide full source code of a live demo site written in PHP with detailed SDK documents and tutorials.

What's Next

- [Learn how to install and control the video encoder engine](#)
- [Learn how to build a video hosting site with PHP](#)
- [Learn how to create a customized Flash Video player for your Web site](#)

Overview

This chapter tells you how to use the encoder in console mode and control the properties of converted Flash Video such as dimension, frame rate, key frame interval, bit rate, etc. via command-line parameters.

Table of Content

- [System Requirements](#)
- [Installation](#)
- [Usage](#)
- [Samples](#)
- [Command line parameters](#)
- [Program output & exit code](#)
- Appendix
 - [MP3 Audio settings](#)
 - [AAC Audio settings](#)
 - [Glossary](#)

System Requirements

- **Supported operating systems**

- Red Hat Enterprise Linux 4/5
- Ubuntu
- Debian
- CentOS
- Fedora

- **Hardware requirements**

- 2.4GHz Intel Pentium 4 processor (dual Intel Xeon or faster recommended)
- RAM: 1GB minimum, 4GB recommended

Installation

This chapter tells you how to install Sothink Video Encoder Engine for Adobe Flash (Linux Version) on your Linux server.

1. Log on to your server with root account.
2. Download [Sothink Video Encoder Engine for Adobe Flash \(Linux Version\)](#) from our Web site. You will get a file named "fvec.tar.gz".
3. Create a directory "fvec" under /usr/bin:

```
# mkdir /usr/bin/fvec
```

4. Extract all the files from fvec.tar.gz to /usr/bin/fvec.

```
# tar xvfz fvec.tar.gz -C /usr/bin/fvec
```

5. Assign execution permission to users:

```
# chmod 755 /usr/bin/fvec -R
```

Usage

1. Please make sure you have [installed](#) the Sothink Video Encoder Engine for Adobe Flash (Linux Version) correctly.
2. Open the Linux console.
3. Enter the command: `fvec sourcefile [targetfile] [options]`

See the illustration below:

```
# /usr/bin/fvec/fvec
Sothink Video Encoder for Adobe Flash (Linux Version)
Version: 1.1 (Build 90317)

Usage: FVEC sourcefile [targetfile] [options]

options:
-h                show help
-v                show version
-i                show movie info
-h264            use H.264 video codec
-ss hh:mm:ss[.xxx] encoding start position. e.g. 00:00:30
-endpos hh:mm:ss[.xxx] encoding end position. e.g. 00:05:00
-vw num          video width (64 to 1920)
-vh num          video height (64 to 1920)
-vb num          video bit rate (5 to 10000 for H.263, 190 to 50000 for H.264)
-vf num          video frame rate (1 to 30)
-vk num          video key frame interval
-vz letter        video zoom mode (L, P, S)
-vc x1,y1,x2,y2  video crop area. e.g. 20,20,300,300
-vq num          video quality (0 to 6). larger value means better quality
-ad             disable audio
-as num          audio sample rate in Hz (11025, 22050, 44100)
-ab num          audio bit rate in Kbps (8, 16, ..., 320)
-ac num          audio channels (1, 2)
-ag num          audio volume gain in dB (-200 to 60)
-tn pathname     thumbnail file path name
-tw num          thumbnail width (64 to 1920)
-th num          thumbnail height (64 to 1920)
-tf num          thumbnail frame number
-tt hh:mm:ss[.xxx] thumbnail frame time. e.g. 00:00:30
-wn pathname     watermark file path name (only PNG file is supported)
-wp x1,y1        watermark position. e.g. 20,20
-ws duration,interval duration and interval of watermark. e.g. 5,60
```

[Click here](#) to see the detailed description of the command line parameters and options.

[Click here](#) to see a few samples.

Command Line Parameters

sourcefile	The full path of the input video file.
targetfile	The full path of the output .flv file. You can omit this parameter when using the -i option.
-h	Show the usage of the program.
-v	Show the version of the program.
-i	Show information of the video file.
-h264	Use H.264 video codec. If you omit this parameter, Sorenson Spark video codec (based on H.263) will be used. H.264 codec delivers higher quality when to compared to Sorenson Spark video codec. H.264 codec is supported by Adobe Flash Player 9 and later.
-ss	Specify the encoding start position (hh:mm:ss[.xxx]).
-endpos	Specify the encoding end position (hh:mm:ss[.xxx]). This parameter specifies the duration of the encoded video.
-vw	The width of the output video in pixels. Please specify a value between 64 and 1920. If you omit this option: If no cropping area specified (see the -vc option), the width of the input video will be used. If you specified a cropping area (see the -vc option), the width of the cropping area will be used.
-vh	The height of the output video in pixels. Please specify a value between 64 and 1920. If you omit this option: If no cropping area specified (see the -vc option), the height of the input video will be used. If you specified a cropping area (see the -vc option), the height of the cropping area will be used.
-vb	The bit rate of the output video in Kbits/s. The valid value is 5 to 10000 if you use Sorenson Spark H.263 video codec; 190 to 50000 if you use the H.264 video codec. If you omit this option, the program will automatically calculate a proper value for you.
-vf	The fps of the output video. Please specify a value between 1 and 30. If you omit this option, the original fps will be used.
-vk	Specify the video key frame interval of the output video. The smaller key frame interval is, the better picture fluency of Flash Video you can get, but the .flv file size will be larger. If you omit this option, the program will set the key frame interval to 50.
-vz	Video zoom mode of the output video. You can specify L, P or S mode. If you omit this option, the program will use the L option L: Letter Box. If you adopt Letter Box mode, the picture will be shrunk to fit the target dimension while preserving the aspect ratio of source video and black bars will be added to fill the gaps. P: Pan & Scan. Only parts of the source video would be converted with Pan & Scan mode. The program will cut out some detail around the picture, which is not considered that important, to make the width or height fit. S: Stretch. With Stretch mode, the whole video will be recorded at full dimension for preserving all the fine details, but the picture will lost the

	original aspect ratio and be literally squashed out of shape.
-vc	Specify the cropping rectangle of the input video. Please enter four integers here, separated by commas (NO space between comma and integer). The first 2 numbers are the coordinates of the top left of the cropping area, and the second two numbers are the coordinates of the bottom right of the area. For example, if you specify 10,20,330,260, the dimension of the cropping area will be 320x240, locating at (10, 20). If you omit this option, the entire area of the input video will be encoded.
-vq	Specify the video quality (0-6). This parameter is only available when using H.264 video codec. A larger value means better quality but slower encoding speed.
-ad	Specify this option if you want to disable the audio. The output .flv will contain video stream only.
-as	Specify the audio sample rate in Hz. The available values are 11025, 22050 and 44100. If you omit this option, 44100 Hz will be used.
-ab	Specify the audio bit rate in Kbits/S. The available values are related to the sample rate and audio channels. If you use Sorenson Spark video codec, please refer to the MP3 Audio Settings table for more information. If you use H.264 video codec, please refer to the AAC Audio Settings table for more information. If you omit this option, 128 Kbps will be used.
-ac	Specify the audio channels. The available values are 1 (mono) and 2 (stereo). If you omit this option, 2 channels (stereo) will be used.
-ag	Sets the desired gain in dB for the audio stream from -200 to 60. Please note that if the gain is too high, the audio may get distortion.
-tn	Specify the full path name for the thumbnail image file. The created file is JPEG format.
-tw	Specify the width of the thumbnail image. The program will use the width of the output video if this parameter is omitted.
-th	Specify the height of the thumbnail image. The program will use the height of the output video if this parameter is omitted.
-tf	Specify the frame to create the thumbnail file. For example, "-tf 5" means the program will use the 5th frame to create the thumbnail image. Please note if you specify a very large value, the program may take a long time to create the thumbnail image. If you omit this parameter, the program will use the first non-blank frame to create the thumbnail image.
-tt	Specify the time (hh:mm:ss[.xxx]) to create the thumbnail file. For example, "-tt 00:00:05" means the program will use the frame at the 5th second to create the thumbnail image. Please note if you specify a very large value, the encoder may take a long time to create the thumbnail image. If you omit this parameter, the program will use the first non-blank frame to create the thumbnail image.
-wn	Specify the full path name of the image file used to create the watermark. The format of the image file must be PNG.
-wp	The desired position of the watermark, for example, 10,10. The coordinates are related to the top left corner of the video.

-ws	Duration and interval of watermark in seconds, for example, 5,60.
-----	---

[Click here](#) to see a few samples.

Samples

Important note: The file paths in the parameters must be surrounded by quote marks if they contain spaces.

Sample 1: Create a Flash Video with default settings. Because we did not specify the dimension of the output video, the encoder will use the dimension of the source video (320x240) by default.

```
# /usr/bin/fvec/fvec /home/test/1.mpg /home/test/1.flv
Percent: 2%
Percent: 43%
Percent: 84%
Percent: 100%
```

Sample 2: Create a Flash Video, set the dimension of the output video to 160x120, video bit rate to 300 Kbps, and with stereo audio.

```
# /usr/bin/fvec/fvec /home/test/1.mpg /home/test/1.flv -vw 160 -vh 120 -vb 300 -ac 2
Percent: 3%
Percent: 50%
Percent: 100%
```

Sample 3: Create a Flash Video with mono audio, and crop the picture of source video with a cropping rectangle (100,100,260,220). Please note we did not specify the dimension of output video, thus the program took the cropping area as the dimension setting.

```
# /usr/bin/fvec/fvec /home/test/1.mpg /home/test/1.flv -vc 100,100,260,220 -ac 1
Percent: 5%
Percent: 91%
Percent: 100%
```

Sample 4: Create an H.264 encoded Flash Video.

```
# /usr/bin/fvec/fvec /home/test/1.mpg /home/test/1.mp4 -h264
Percent: 5%
Percent: 91%
Percent: 100%
```

Sample 5: Create a 160x120 thumbnail image using the first non-blank frame of the input video.

```
# /usr/bin/fvec/fvec /home/test/1.mpg -tn /home/test/1.jpg -tw 160 -th 120
```

Sample 6: Encode only a part (2 minutes) of the input video.

```
# /usr/bin/fvec/fvec /home/test/1.mpg /home/test/1.flv -ss 00:00:30 -endpos 00:02:00
Percent: 5%
Percent: 91%
Percent: 100%
```


The Output and Exit Code of the Program

When you run the program from the Linux console or execute it in another application, the program will print certain information to the STDOUT. You can catch the STDOUT in your application to get desired information from the program at run-time.

Output

If the input video file is a valid video file and can be encoded, the program will print the conversion progress to the STDOUT. Below is an example:

```
# /usr/bin/fvec/fvec /home/test/1.mpg /home/test/1.flv
Percent: 2%
Percent: 43%
Percent: 84%
Percent: 100%
Duration: 00:00:59
```

Percent: Print the Conversion Progress in percent periodically. The percent of 100% means the conversion is accomplished and the program has terminated automatically.

Duration: Print the total length of the encoded video.

If the input video file is not a valid video file, or the file can not be encoded correctly, the program will print one line of error message to the STDOUT and then terminate. The error message starts with the text "Error:" and is followed by a description of the error reason. For example, when you type the following command in the console (assume "nonexistent.mpg" does not exist):

```
# /usr/bin/fvec/fvec /home/test/nonexistent.mpg /home/test/1.flv
Error: Can not find the source file "/home/test/nonexistent.mpg".
```

Exit Code

The program may exit with the following exit codes. You can check the exit code to know if the conversion is finished successfully or not.

Exit Code	Description
0	Conversion is finished successfully.
1	Conversion failed.
2	Reserved for future use.
3	Program initialization error.
4	Insufficient parameter supplied. For example, you passed the -vb parameter, but did not specify the video bit rate.
5	Unrecognizable parameter found. Please refer to the usage section for more information.

6	Invalid parameter passed to the program. For example, the bit rate value exceeded the permitted range.
7	Can not parse the input file.
8	Can not encode the input file.
9	Can not find the encoder. Please make sure the software has been installed correctly.

Appendix I - MP3 Audio Settings

The following table describes the dependency relationship between the audio bit rate and the audio sample rate.

Sample rate	11025 KHz	22050 KHz	44100 KHz
Bit rate	8 Kbps	8 Kbps	32 Kbps
	16 Kbps	16 Kbps	40 Kbps
	24 Kbps	24 Kbps	48 Kbps
	32 Kbps	32 Kbps	56 Kbps
	40 Kbps	40 Kbps	64 Kbps
	48 Kbps	48 Kbps	80 Kbps
	56 Kbps	56 Kbps	96 Kbps
	64 Kbps	64 Kbps	112 Kbps
	80 Kbps	80 Kbps	128 Kbps
	96 Kbps	96 Kbps	160 Kbps
	112 Kbps	112 Kbps	192 Kbps
	128 Kbps	128 Kbps	224 Kbps
	144 Kbps	144 Kbps	256 Kbps
	160 Kbps	160 Kbps	320 Kbps

Appendix II - AAC Audio Settings

The following table describes the dependency relationship between the audio bit rate and the audio sample rate.

Sample rate	11025 KHz		22050 KHz		44100 KHz	
Channels	Mono	Stereo	Mono	Stereo	Mono	Stereo
Bit rate	8 Kbps					
	16 Kbps	16 Kbps	16 Kbps			
	20 Kbps	20 Kbps	20 Kbps			
	24 Kbps	24 Kbps	24 Kbps	24 Kbps		
	28 Kbps	28 Kbps	28 Kbps	28 Kbps		
	32 Kbps	32 Kbps	32 Kbps	32 Kbps	32 Kbps	
			40 Kbps	40 Kbps	40 Kbps	
			48 Kbps	48 Kbps	48 Kbps	
			56 Kbps	56 Kbps	56 Kbps	56 Kbps
					64 Kbps	64 Kbps
					80 Kbps	80 Kbps
					96 Kbps	96 Kbps
					112 Kbps	112 Kbps
					128 Kbps	128 Kbps
					160 Kbps	160 Kbps
					192 Kbps	192 Kbps
					224 Kbps	224 Kbps
					256 Kbps	256 Kbps
				320 Kbps	320 Kbps	

Appendix III - Glossaries

AVI: That is Audio Video Interleave. This is a container video format that specifies certain structure how the audio and video streams should be stored within the file. AVI itself doesn't specify how it should be encoded, so the audio/video can be stored in very various ways.

ASF: That is Advanced Streaming Format. Microsoft's new audio/video format, meant specifically for streaming purposes. It doesn't specify how the video or audio should be encoded, but instead just specifies the structure of the video/audio stream. This means that ASF files can be encoded with basically any audio/video code and would still be in ASF format.

MPEG: That is Moving Pictures Expert Group. A working group of ISO/IEC in charge of the development of standards for coded representation of digital audio and video. MPEG is not an acronym for any standard, it is the acronym for the group who develops these standards!

MOV: A file extension used by the QuickTime-wrapped files.

SWF: SWF is a proprietary vector graphics file format produced by the Flash software from Adobe (formerly Macromedia). Intended to be small enough for publication on the web, SWF files can contain animations or applets of varying degrees of interactivity and function. SWF is also sometimes used for creating animated display graphics and menus for DVD movies, and television commercials. The Flash program produces SWF files as a compressed and uneditable final product, whereas it uses the .fla format for its editable working files.

FLV: FLV (Flash Video) is a proprietary file format used to deliver video over the Internet using Adobe Flash Player (also called Macromedia Flash Player). FLV content may also be embedded within SWF files. Notable users of the FLV format include Google Video, Reuters.com, YouTube and MySpace. Flash Video is viewable on most operating systems, via the widely available Macromedia Flash Player and web browser plugin, or one of several third-party programs such as Media Player Classic (with the ffdshow codecs installed), MPlayer, or VLC media player.

JPG/JPEG: An image format optimized for "natural" images developed by the **J**oint **P**hotographic **E**xperts **G**roup, JPEGs are probably second only to GIFs in level of acceptance. JPEGs manage to capture wonderfully detailed images in millions of colors in minimal space by taking advantage of limitations with human vision, and performing little lossy compressions. This means that each time a JPEG is saved, it will lose a little more quality, although each individual loss will be nearly invisible to the human eye. If care is not taken, however, a JPEG image can become worse looking than a color-limited GIF. JPEGs are also not good at storing cartoon-like images or line drawings; for these cases either GIFs or PNGs or X-bitmaps are better choices. JPEGs also cannot contain a transparent color. If a transparent color is needed, GIFs, PNGs, and X-bitmaps are better choices. JPEG has nothing to do with either JBIG or MPEG in spite of the similarity of names.

MP3: MP3 stands for MPEG-1 Audio Layer III. It is not a separate format, but a part of MPEG-1 video encoding format, developed by MPEG group in early 1990's. MPEG-1 Audio Layer III (*MP3*) is a method to store good quality audio into small files by using psychoacoustics in order to get rid of the data from the audio that most of the humans can't hear. MP3's bit rates vary from 8kbps (*that is 8 kilobits per second, not kilobytes*) to 320kbps. When MP3 phenomenon began in 1996, most of the audio files were encoded using 128kbps bit rate, which is still the most popular bit rate in the world -- although most of the people agree that by using slightly higher bit rates, like 192kbps or 256kbps, the audio quality can be compared with the CD quality.

WAV: WAV is the standard Windows audio file format and it was originally jointly developed by Microsoft and IBM. Format itself doesn't specify any particular audio compression scheme, but actually supports several compression types. However, the most commonly used -- and the one people typically refer as ".wav file" -- is the one that uses *IMA/ADPCM* compression at 4:1

compression level for 16-bit sounds. Windows operating system uses .wav files extensively for its audio operations, such as notify sounds, etc. Also, virtually all CD rippers use "raw" (*the above-mentioned IMA/ADPCM compression*) WAV format as their default method for outputting extracted CD audio tracks to file format. Due its de facto standard status, virtually all Windows-based audio and video applications can use and play WAV files.

HTML: HTML is a Hypertext Marked Language, it is a simple marked language that is used to make hypertext document. Hypertext document is called as HTML document because it is written by HTML.

LINK: In this program, we need to fill the URL in the link field. URL is Uniform Resource Locator, it is the address that defines the route to a file on the web or any other Internet facility.

FRAME RATE: This is the number of frames per second. The default frame rate is 12.

HERTZ: Hertz is the standard unit of frequency (of change in state or cycle in sound wave, alternating current, or other cyclical waveform) of one cycle per second.

BIT RATE: The transmission speed of binary coded data regarding online media delivery. This is the amount of data that needs to be delivered to the viewer per second. This is most usually measured in kilobits per second (or kbps), just like modem (eg.56kbps or 28kbps).

STREAM: A stream of data is a continuous flow of data, when information online is set to stream, it means that the user does not need to download all the information (say, an entire song) before listening to it. Stream information can be processed as it is received and does not need to remain on the user's hard-drive.

Overview

Before you setup the server, please ensure that you deploy the product on computers that meet the minimum system requirements. For information, see the [System Requirements](#) section.

This tutorial guides you to use the Sothink Video Encoder Engine for Adobe Flash (Linux Version) to setup a video hosting site for instant video uploading, encoding and exhibition. The technologies used in this development kit include:

- Sothink Video Encoder Engine for Adobe Flash (Linux Version)
- Apache Server
- PHP
- MySQL
- JavaScript

To setup your video hosting site now, please follow these steps:

- Step 1: [Install the apache server and PHP](#)
- Step 2: [Setup directories and files](#)
- Step 3: [Configure PHP](#)
- Step 4: [Setup MySQL Database](#)
- Step 5: [Test your Web site](#)

Install Apache Server

Please download the Apache server from the [Apache Software Foundation](#) and follow [this document](#) to know how to install, configure and run Apache under Linux.

Install PHP

PHP (recursive acronym for "PHP: Hypertext Preprocessor") is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Please download PHP from <http://www.php.net> and follow the [installation and configuration guide](#) to know how to install and configure PHP.

Setup Directories and Files

1. Log on to your server with root account.
2. Download [Sothink Video Encoder Engine for Adobe Flash \(Linux Version\)](#) from our Web site. You will get a file named "fvec.tar.gz".
3. Create a directory "fvec" under /usr/bin:

```
# mkdir /usr/bin/fvec
```

4. Extract all the files from fvec.tar.gz to /usr/bin/fvec.

```
# tar xvfz fvec.tar.gz -C /usr/bin/fvec
```

5. Assign execution permission to users:

```
# chmod 755 /usr/bin/fvec -R
```

6. Download the [PHP source of the demo site](#) from our Web site and you will get a file named "php-demo.tar.gz"; Extract all the files to a sub-folder under your www root, for example, /var/www/html/php-demo.

```
; Create the directory
```

```
# mkdir /var/www/html/php-demo
```

```
; Extract the files
```

```
# tar xvfz php-demo.tar.gz -C /var/www/html/php-demo
```

```
; Assign permissions
```

```
# chmod 777 -R /var/www/html/php-demo
```

7. To prevent users from uploading executable files to the directory "/var/www/html/php-demo/files" and executing them, please open the file httpd.conf in the conf folder of Apache and insert the following lines to deny the execution of files in this folder.

```
# vi /etc/httpd/conf/httpd.conf
```

```
<Directory /var/www/html/php-demo/files>  
    php_flag engine off  
</Directory>
```

Important Note:

If [SELinux](#) is installed on your system, the video uploading and encoding functions may not work correctly due the secure settings of [SELinux](#). If you met this problem, please disable [SELinux](#) or modify the configuration of [SELinux](#) to allow PHP to create and modify files in directory "/var/www/html/php-demo" and its sub-directories.

Configure PHP

You should change some php.ini directives to make the demo site work properly. Generally, php.ini is in the directory /etc. The demo site uses some new functions available since PHP 4.3. If you use older versions, please upgrade to version 4.3 or later. Please check the [PHP configuration document](#) for details.

; Enable file uploading:
file_uploads = On

; Setup the temporary directory used for storing files when doing file upload. The directory must be writable by whatever user PHP is running as.
upload_tmp_dir = /var/tmp

; Maximum allowed size for uploaded files.
upload_max_filesize = 20M

; Maximum size of POST data that PHP will accept. This value should be larger than upload_max_filesize because extra data will be posted while uploading.
post_max_size = 30M

Save php.ini and restart Apache to make the changes take effect.

```
# /sbin/service httpd restart
```

Setup MySQL Database

1. Open the file in your browser: <http://yourdomain/php-demo/installdb.php>
2. Input the server name, user name and password of the of MySQL server.
3. Click install to start the setup process.

If succeeded, a database named "fvec" and a table named "VIDEOS" will be created in MySQL. A file 'db.php' will also be created in the current directory (/var/www/html/php-demo).

Test Your Web Site

To test the site, please open your browser and navigate to <http://yourdomain/php-demo/upload.htm>. If everything goes well, you can upload video files to your server and encode them now. To view our demo site, please visit <http://flv-encoder-linux.sothinkmedia.com>.

Overview

After purchasing Sothink Video Encoder Engine for Adobe Flash (Linux Version) , you can get [Sothink Video Encoder for Adobe Flash](#) Windows GUI version for free. With the Windows GUI version, you can make customized Flash Video Players for general use on your video hosting and sharing server.

Table of Content

- [Installation](#)
- [Create a Flash Video Player](#)
- [Deploy the Flash Video Player](#)

System Requirements

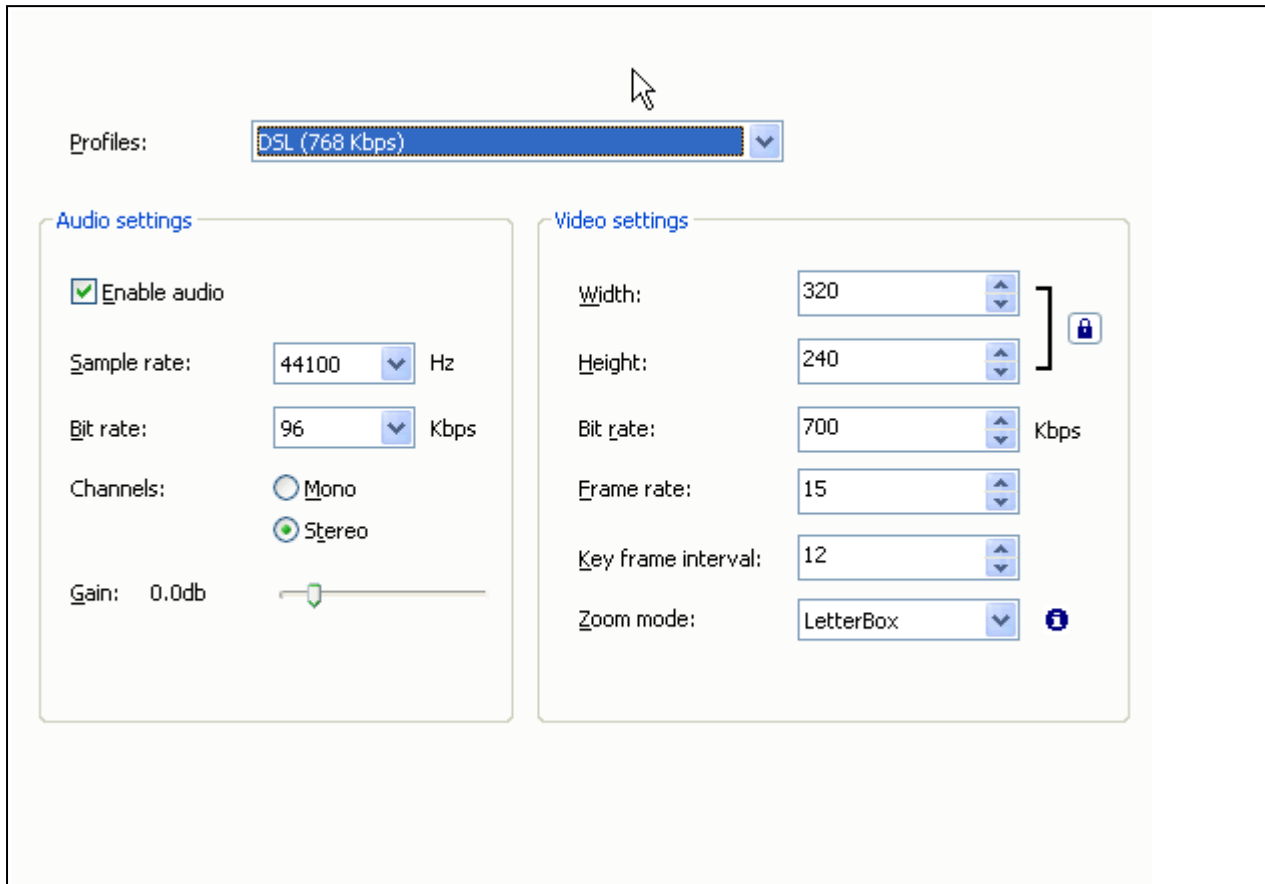
- Windows 2000
- Windows XP
- Windows 2003 Server
- Windows Vista

Installation

1. Download [Sothink Video Encoder for Adobe Flash](#) from our web site.
2. Extract the ZIP file to your hard disk
3. Run Setup.exe and follow the wizard to install the software.
4. Launch Sothink Video Encoder for Adobe Flash by double-clicking the program icon on your desktop.
5. Enter the registration key to register the software. If you have not get the registration key yet, please send your registration information of Sothink Video Encoder Engine for Adobe Flash (Linux Version) to support@sothinkmedia.com and we will send you the registration key then.

Create a Customized Flash Video Player

1. Launch Sothink Video Encoder for Adobe Flash by double-clicking the program icon on your desktop.
2. Select any video file and click "Next".
3. In the "Encoding preferences" page, change the "Width" and "Height" to desired values, then click Next.



4. In the "Select output folder and files" page, check "Generate SWF" and uncheck "Generate FLV", "Generate HTML" and "Generate thumbnail". Enter "player.swf" as the name of the output SWF file. Select "Get FLV path from a URL variable and play progressively" and enter "flv" as the name of the URL variable. You can also change the playback settings as you like. Click "Next" after done.

Output directory: C:\Documents and Settings\ch.SOURCESOFT\Desktop

Generate FLV: test.flv

Generate SWF: player.swf

Embed the video in SWF and play in timeline

Play the created external FLV file progressively

Get the FLV path from a URL variable and play progressively: flv

Generate HTML: test.htm

Create thumbnail: test.JPG

Playback

Auto play Auto rewind Repeat

URL to launch when clicking: Target:

URL to launch at the end: Target:


Load a SWF at the end:

Buffer time (seconds): 5 Initial sound volume: 75

- In the "Select play control skin" page, check "Use a skin" and select a desired skin. You can also specify a color scheme for the skin and adjust the saturation of the color.

Use a skin: Metal (Full) [Get more skins](#)

Skin color: Saturation: 8



Minimum width: 280 Minimum height: 2

- Click "Next" and a file name "player.swf" will be created in the specified folder.

Deploy the Flash Video Player

1. Find the file "player.swf" created in the previous steps in Windows Explorer.
2. Copy the file "player.swf" in to the folder of the PHP demo site and overwrite the old one.
3. Now you can upload a video and test the created Flash Video player on your web site.