Annex E. The FLV File Format

E.1 Overview

Each tag type in an FLV file constitutes a single stream. There shall be no more than one audio and one video stream, synchronized together, in an FLV file. An FLV file shall not define multiple independent streams of a single type.

The simple data types used in FLV are defined in the SWF format specification. FLV files use an additional type that is not defined for SWF files: UI24 representing an unsigned 24-bit integer.

Unlike SWF files, FLV files shall store multi-byte numbers in big-endian byte order. For example, as a UI16 in SWF file format, the byte sequence that represents the number 300 (0x12C) is 0x2C 0x01; as a UI16 in FLV file format, the byte sequence that represents the number 300 is 0x01 0x2C.

See also the SWF File Format Specification at http://www.adobe.com/go/swf_file_format

E.2 The FLV header

An FLV file shall begin with the FLV header:

FLV header

Field	Type	Comment
Signature	UI8	Signature byte always 'F' (0x46)
Signature	UI8	Signature byte always 'L' (0x4C)
Signature	UI8	Signature byte always 'V' (0x56)
Version	UI8	File version (for example, 0x01 for FLV version 1)
TypeFlagsReserved	UB [5]	Shall be 0
TypeFlagsAudio	UB [1]	1 = Audio tags are present
TypeFlagsReserved	UB [1]	Shall be 0
TypeFlagsVideo	UB [1]	1 = Video tags are present
DataOffset	UI32	The length of this header in bytes

The DataOffset field usually has a value of 9 for FLV version 1. This field is present to accommodate larger headers in future versions.

E.3 The FLV File Body

After the FLV header, the remainder of an FLV file shall consist of alternating back-pointers and tags. They interleave as shown in the following table:

FLV File Body

Field	Type	Comment
PreviousTagSize0	UI32	Always 0
Tag1	FLVTAG	First tag
PreviousTagSize1	UI32	Size of previous tag, including its header, in bytes. For FLV version 1, this value is 11 plus the DataSize of the previous tag.
Tag2	FLVTAG	Second tag
...		
PreviousTagSizeN-1	UI32	Size of second-to-last tag, including its header, in bytes.

TagN	FLVTAG	Last tag
PreviousTagSizeN	UI32	Size of last tag, including its header, in bytes.

E.4 FLV Tag Definition

E.4.1 FLV Tag

The FLV tag contains metadata for audio, video, or scripts, optional encryption metadata, and the payload.

FLVTAG

Field	**Type**	**Comment**
Reserved	UB [2]	Reserved for FMS, should be 0
Filter	UB [1]	Indicates if packets are filtered. 0 = No pre-processing required. 1 = Pre-processing (such as decryption) of the packet is required before it can be rendered. Shall be 0 in unencrypted files, and 1 for encrypted tags. See Annex F. FLV Encryption for the use of filters.
TagType	UB [5]	Type of contents in this tag. The following types are defined: 8 = audio 9 = video 18 = script data
DataSize	UI24	Length of the message. Number of bytes after StreamID to end of tag (Equal to length of the tag – 11)
Timestamp	UI24	Time in milliseconds at which the data in this tag applies. This value is relative to the first tag in the FLV file, which always has a timestamp of 0.
TimestampExtended	UI8	Extension of the Timestamp field to form a SI32 value. This field represents the upper 8 bits, while the previous Timestamp field represents the lower 24 bits of the time in milliseconds.
StreamID	UI24	Always 0.
AudioTagHeader	IF TagType == 8 AudioTagHeader	AudioTagHeader element as defined in Section E.4.2.1.
VideoTagHeader	IF TagType == 9 VideoTagHeader	VideoTagHeader element as defined in Section E.4.3.1.
EncryptionHeader	IF Filter == 1 EncryptionTagHeader	Encryption header shall be included for each protected sample, as defined in Section F.3.1.
FilterParams	IF Filter == 1 FilterParams	FilterParams shall be included for each protected sample, as defined in Section F.3.2.

Data	IF TagType == 8
	AUDIODATA
	IF TagType == 9
	VIDEODATA
	IF TagType == 18
	SCRIPTDATA
		Data specific for each media type.

In playback, the time sequencing of FLV tags depends on the FLV timestamps only. Any timing mechanisms built into the payload data format shall be ignored.

E.4.2 Audio Tags

Audio tags are similar to the DefineSound tag in the SWF file format. For formats also supported in SWF, the payload data is identical in FLV and SWF.

E.4.2.1 AUDIODATA

The AudioTagHeader contains audio-specific metadata.

AudioTagHeader

Field	**Type**	**Comment**
SoundFormat (See notes following table, for special encodings)	UB [4]	Format of SoundData. The following values are defined: 0 = Linear PCM, platform endian 1 = ADPCM 2 = MP3 3 = Linear PCM, little endian 4 = Nellymoser 16 kHz mono 5 = Nellymoser 8 kHz mono 6 = Nellymoser 7 = G.711 A-law logarithmic PCM 8 = G.711 mu-law logarithmic PCM 9 = reserved 10 = AAC 11 = Speex 14 = MP3 8 kHz 15 = Device-specific sound Formats 7, 8, 14, and 15 are reserved. AAC is supported in Flash Player 9,0,115,0 and higher. Speex is supported in Flash Player 10 and higher.
SoundRate	UB [2]	Sampling rate. The following values are defined: 0 = 5.5 kHz 1 = 11 kHz 2 = 22 kHz 3 = 44 kHz
SoundSize	UB [1]	Size of each audio sample. This parameter only pertains to uncompressed formats. Compressed formats always decode to 16 bits internally. 0 = 8-bit samples 1 = 16-bit samples

SoundType	UB [1]	Mono or stereo sound 0 = Mono sound 1 = Stereo sound
AACPacketType	IF SoundFormat == 10 UI8	The following values are defined: 0 = AAC sequence header 1 = AAC raw

Format 3, linear PCM, stores raw PCM samples. If the data is 8-bit, the samples are unsigned bytes. If the data is 16-bit, the samples are stored as little endian, signed numbers. If the data is stereo, left and right samples are stored interleaved: left - right - left - right - and so on.

Format 0 PCM is the same as format 3 PCM, except that format 0 stores 16-bit PCM samples in the endian order of the platform on which the file was created. For this reason, format 0 should not be used.

Nellymoser 8 kHz and 16 kHz are special cases, as the SoundRate field cannot represent 8 or 16 kHz sampling rates. When Nellymoser 8 kHz or Nellymoser 16 kHz is specified in SoundFormat, the Flash Player ignores the SoundRate and SoundType fields. For other Nellymoser sampling rates, specify the normal Nellymoser SoundFormat and use the SoundRate and SoundType fields as usual.

If the SoundFormat indicates AAC, the SoundType should be 1 (stereo) and the SoundRate should be 3 (44 kHz). However, this does not mean that AAC audio in FLV is always stereo, 44 kHz data. Instead, the Flash Player ignores these values and extracts the channel and sample rate data is encoded in the AAC bit stream.

If the SoundFormat indicates Speex, the audio is compressed mono sampled at 16 kHz, the SoundRate shall be 0, the SoundSize shall be 1, and the SoundType shall be 0. For information regarding Speex capabilities and limitations when stored in a SWF file, see the SWF File Format Specification at http://www.adobe.com/go/swf_file_format.

The AUDIODATA segment contains the audio payload.

AUDIODATA

Field	Type	Comment
IF Encrypted		See Annex F. FLV Encryption for details.
Body	EncryptedBody	AudioTagBody encrypted as specified in Section F.3.3.
ELSE		
Body	AudioTagBody	

The AudioTagBody holds the audio payload.

AudioTagBody

Field	Type	Comment

SoundData	IF SoundFormat == 10
	AACAUDIODATA
	ELSE
	Varies by format	

E.4.2.2 AACAUDIODATA

The AAC format is supported in Flash Player 9,0,115,0 and higher.

AACAUDIODATA

Field	Type	Comment
Data	IF AACPacketType == 0	The AudioSpecificConfig is defined in ISO

	AudioSpecificConfig ELSE IF AACPacketType == 1 Raw AAC frame data in UI8 [ ]	14496-3. Note that this is not the same as the contents of the esds box from an MP4/F4V file.

E.4.3 Video Tags

Video tags are similar to the VideoFrame tag in the SWF file format, and their payload data is identical. See also the SWF File Format Specification at http://www.adobe.com/go/swf_file_format

E.4.3.1 VIDEODATA

The VideoTagHeader contains video-specific metadata.

VideoTagHeader

Field	**Type**	**Comment**
Frame Type	UB [4]	Type of video frame. The following values are defined: 1 = key frame (for AVC, a seekable frame) 2 = inter frame (for AVC, a non-seekable frame) 3 = disposable inter frame (H.263 only) 4 = generated key frame (reserved for server use only) 5 = video info/command frame
CodecID	UB [4]	Codec Identifier. The following values are defined: 2 = Sorenson H.263 3 = Screen video 4 = On2 VP6 5 = On2 VP6 with alpha channel 6 = Screen video version 2 7 = AVC
AVCPacketType	IF CodecID == 7 UI8	The following values are defined: 0 = AVC sequence header 1 = AVC NALU 2 = AVC end of sequence (lower level NALU sequence ender is not required or supported)
CompositionTime	IF CodecID == 7 SI24	IF AVCPacketType == 1 Composition time offset ELSE 0 See ISO 14496-12, 8.15.3 for an explanation of composition times. The offset in an FLV file is always in milliseconds.

The VIDEODATA segment contains video metadata, optional encryption metadata, and the video payload.

VIDEODATA

Field	Type	Comment
IF Encrypted		See Annex F. FLV Encryption for details.
Body	EncryptedBody	VideoTagBody encrypted as specified in Section F.3.3.
ELSE		
Body	VideoTagBody	

The VideoTagBody contains the video frame payload.

VideoTagBody

Field	Type	Comment

VideoTagBody	IF FrameType == 5
	UI8
	ELSE (
	IF CodecID == 2
	H263VIDEOPACKET
	IF CodecID == 3
	SCREENVIDEOPACKET
	IF CodecID == 4
	VP6FLVVIDEOPACKET
	IF CodecID == 5
	VP6FLVALPHAVIDEOPACKET
	IF CodecID == 6
	SCREENV2VIDEOPACKET
	IF CodecID == 7
	AVCVIDEOPACKET
	)
		Video frame payload or frame info
		If FrameType == 5, instead of a video payload, the Video Data Body contains a UI8 with the following meaning:
		0 = Start of client-side seeking video frame sequence
		1 = End of client-side seeking video frame sequence
		For all but AVCVIDEOPACKET, see the SWF File Format Specification for details

E.4.3.2 AVCVIDEOPACKET

An AVCVIDEOPACKET carries a payload of AVC video data.

AVCVIDEOPACKET

Field	Type	Comment

Data	IF AVCPacketType == 0
	AVCDecoderConfigurationRecord
	IF AVCPacketType == 1
	One or more NALUs (Full frames are required)	

See ISO 14496-15, 5.2.4.1 for the description of AVCDecoderConfigurationRecord. This contains the same information that would be stored in an avcC box in an MP4/FLV file.

E.4.4 Data Tags

Data tags encapsulate single-method invocations, which usually are called on a NetStream object in Flash Player. Data tags comprise of a method name and a set of arguments.

E.4.4.1 SCRIPTDATA

The SCRIPTDATA segment contains optional encryption metadata, and the script payload.

SCRIPTDATA

Field	Type	Comment
IF Encrypted		See Annex F. FLV Encryption for details.
Body	EncryptedBody	ScriptTagBody encrypted as specified in Section F.3.3.
ELSE		
Body	ScriptTagBody	

The ScriptTagBody contains SCRIPTDATA encoded in the Action Message Format (AMF), which is a compact binary format used to serialize ActionScript object graphs. The specification for AMF0 is available at: http://opensource.adobe.com/wiki/display/blazeds/Developer+Documentation

ScriptTagBody

Field	Type	Comment
Name	SCRIPTDATAVALUE	Method or object name. SCRIPTDATAVALUE.Type = 2 (String)
Value	SCRIPTDATAVALUE	AMF arguments or object properties. SCRIPTDATAVALUE.Type = 8 (ECMA array)

E.4.4.2 SCRIPTDATAVALUE

A SCRIPTDATAVALUE record contains a typed ActionScript value.

SCRIPTDATAVALUE

Field	Type	Comment
Type	UI8	Type of the ScriptDataValue. The following types are defined: 0 = Number 1 = Boolean 2 = String 3 = Object 4 = MovieClip (reserved, not supported) 5 = Null 6 = Undefined 7 = Reference 8 = ECMA array 9 = Object end marker 10 = Strict array 11 = Date 12 = Long string

ScriptDataValue	IF Type == 0
	DOUBLE
	IF Type == 1
	UI8
	IF Type == 2
	SCRIPTDATASTRING
	IF Type == 3
	SCRIPTDATAOBJECT
	IF Type == 7
	UI16
	IF Type == 8
	SCRIPTDATAECMAARRAY
	IF Type == 10
	SCRIPTDATASTRICTARRAY
	IF Type == 11
	SCRIPTDATADATE
	IF Type == 12
	SCRIPTDATALONGSTRING
		Script data value.
		The Boolean value is (ScriptDataValue ≠ 0).

E.4.4.3 SCRIPTDATADATE

A SCRIPTDATADATE record stores date and time.

SCRIPTDATADATE

Field	Type	Comment
DateTime	DOUBLE	Number of milliseconds since Jan 1, 1970 UTC.
LocalDateTimeOffset	SI16	Local time offset in minutes from UTC. For time zones located west of Greenwich, UK, this value is a negative number. Time zones east of Greenwich, UK, are positive.

E.4.4.4 SCRIPTDATAECMAARRAY

A SCRIPTDATAECMAARRAY record stores an ECMA array. An ECMA Array is an associative array, and shall be used when an ActionScript Array contains non-ordinal indices. All indices, ordinal or otherwise, are strings instead of integers. For the purposes of serialization, this type is very similar to an anonymous ActionScript Object. The list contains approximately ECMAArrayLength number of items. A SCRIPTDATAOBJECTEND record follows the list of items.

SCRIPTDATAECMAARRAY

Field	Type	Comment
ECMAArrayLength	UI32	Approximate number of items in ECMA array
Variables	SCRIPTDATAOBJECTPROPERTY [ ]	List of variable names and values
List Terminator	SCRIPTDATAOBJECTEND	List terminator

E.4.4.5 SCRIPTDATALONGSTRING

SCRIPTDATASTRING and SCRIPTDATALONGSTRING records store strings.

SCRIPTDATALONGSTRING

Field	Type	Comment
StringLength	UI32	StringData length in bytes
StringData	STRING	String data, with no terminating NUL

E.4.4.6 SCRIPTDATAOBJECT

A SCRIPTDATAOBJECT record encodes the properties of an anonymous ActionScript object. A SCRIPTDATAOBJECTEND record follows the list of properties.

SCRIPTDATAOBJECT

Field	Type	Comment
ObjectProperties	SCRIPTDATAOBJECTPROPERTY [ ]	List of object properties
List Terminator	SCRIPTDATAOBJECTEND	List terminator

E.4.4.7 SCRIPTDATAOBJECTEND

The SCRIPTDATAOBJECTEND record terminates a list of SCRIPTDATAOBJECTPROPERTY records. The SCRIPTDATAOBJECTEND record is a SCRIPTDATAOBJECTPROPERTY record with a zero-length string and an Object end marker.

SCRIPTDATAOBJECTEND

Field	Type	Comment
ObjectEndMarker	UI8 [3]	Shall be 0, 0, 9

E.4.4.8 SCRIPTDATAOBJECTPROPERTY

A SCRIPTDATAOBJECTPROPERTY record defines an object property of an ActionScript object or a variable of associated array.

SCRIPTDATAOBJECTPROPERTY

Field	Type	Comment
PropertyName	SCRIPTDATASTRING	Name of the object property or variable
PropertyData	SCRIPTDATAVALUE	Value and type of the object property or variable

E.4.4.9 SCRIPTDATASTRICTARRAY

A SCRIPTDATASTRICTARRAY record stores a strict array. A strict array contains only ordinal indices, which are implied, not stored in the record. The indices can be dense or sparse. Undefined entries in the sparse regions between indices shall be serialized as undefined. The list shall contain StrictArrayLength number of values. No terminating record follows the list.

SCRIPTDATASTRICTARRAY

Field	Type	Comment
StrictArrayLength	UI32	Number of items in the array
StrictArrayValue	SCRIPTDATAVALUE [StrictArrayLength]	List of typed values

E.4.4.10 SCRIPTDATASTRING

SCRIPTDATASTRING and SCRIPTDATALONGSTRING records store strings.

The SCRIPTDATASTRING record may be used for strings no longer than 65535 characters.

SCRIPTDATASTRING

Field	Type	Comment
StringLength	UI16	StringData length in bytes.
StringData	STRING	String data, up to 65535 bytes, with no terminating NUL

E.5 onMetaData

The FLV metadata object shall be carried in a SCRIPTDATA tag named `onMetadata`. Various properties are available to a running ActionScript program via the `NetStream.onMetaData` property. The available properties differ depending on the software creating the FLV file. Typical properties include:

onMetadata properties

Property Name	Type	Comment
audiocodecid	Number	Audio codec ID used in the file (see E.4.2.1 for available SoundFormat values)
audiodatarate	Number	Audio bit rate in kilobits per second
audiodelay	Number	Delay introduced by the audio codec in seconds
audiosamplerate	Number	Frequency at which the audio stream is replayed
audiosamplesize	Number	Resolution of a single audio sample
canSeekToEnd	Boolean	Indicating the last video frame is a key frame
creationdate	String	Creation date and time
duration	Number	Total duration of the file in seconds
filesize	Number	Total size of the file in bytes
framerate	Number	Number of frames per second
height	Number	Height of the video in pixels
stereo	Boolean	Indicating stereo audio
videocodecid	Number	Video codec ID used in the file (see E.4.3.1 for available CodecID values)
videodatarate	Number	Video bit rate in kilobits per second
width	Number	Width of the video in pixels

E.6 XMP Metadata in FLV

The XMP metadata object shall be carried in a SCRIPTDATA tag named `onXMPData`. The tag shall be placed at time 0. The tag should be after all time 0 `onMetaData` tags, and before all time 0 audio or video tags, but readers should not require this ordering.

XMPMetadata object

Property Name	Type	Comment
liveXML	String or Long string	XMP metadata, formatted according to the XMP metadata specification

For further details, see www.adobe.com/devnet/xmp/pdfs/XMPSpecificationPart3.pdf