

asSWFbit Library - a ActionScript based SWF Decompiler / Compiler

ActionScript 3 / Last updating  07.07.2010

asSWFbit_Library is a ActionScript based SWF Decompiler / Compiler for SWF files.

The library should help you to check, test and manipulate SWF files in any way you wanted.











The idea behind asSWFbit is to make it possible to provide a way fix small errors in a SWF file without needing a flash editor or the FLA files.

I hope you understand you should this only do for your own SWF files !

Please keep in Mind that English is not my native language, so I try my best to make a clear documentation in English.

When you find errors or something like that, please send me a short email what I can change or which part is mistakable.

Content

-  [Road map](#)
-  [Manifestations about the asSWFbit Library](#)
-  [Download the asSWFbit Library](#)
-  [Errors, Improvements and Support](#)
-  [Installation of the asSWFbit Library](#)
-  [Read Examples](#)
-  [Overview about asSWFbit Library](#)
-  [Constructor](#)
-  [Method](#)
-  [Objects](#)

Road map



This is only a small Road map for the asSWFbit Library. Because I do this in my free time I will make no promise for any time lines or any other ETA for parts of the project.

Read Support (planned 2010)

- SWF Header Information (Signature, Version, FileLength, Width, Height, FrameRate, FrameCount)
- FileAttributes (UseDirectBlit, UseGPU, HasMetadata, ActionScript3, UseNetwork)
- MetaData (CreatorTool, CreateDate, MetadataDate, ModifyDate, RAW)
- BackgroundColor
- ScriptLimits
- ...
- ActionScript Parser

Modify Support (planned end 2010)

- SWF Header
- FileAttributes
- MetaData
- ...
- ActionScript

Write Support (planned 2011)

- SWF Header
- FileAttributes
- MetaData
- ...

- ActionScript

All items which are line-through are already implemented in the lasted version.
Other items are planned but not released in any version yet.

Manifestations about asSWFbit

The License for asSWFbit is the BSD License, the License for the examples / documentation is the Creative Commons (by-sa) License. Please also check <http://code.google.com/p/asswfbit/> for more Information's.

Download from asSWFbit 0.2

asSWFbit is hosted on Google Code on this page you can download the lasted version of asSWFbit and submit errors and improvements.

 <http://code.google.com/p/asswfbit/>

Errors, Improvements and Support

First of all there is no direct Support for this project, please understand I do this in my free time and it takes a lot of time to write such a documentation.

So I have no time or the resources to answer direct support requests. Support requests to the contact email of the web page will be ignore and not answered.

When you find Errors or have a idea of a improvement, please submit these over the Google Code web page <http://code.google.com/p/asswfbit/>.

Installation from asSWFbit Library

asSWFbit Library 0.2 is available as package, copy the corresponding package folder "de" to the same place as the FLA file.

Read Examples

This examples shows how you can load and parse a flash file with the asSWFbit library.

The FLAs for these examples are also include in the example folder of the asSWFbit library.

Read SWF Header Information

The asSWFbit library has the option only to parse and read the SWF Header of a flash file.

When you only need these Information you should include the switch for a faster parsing of the SWF file.

The SWF Header include the following Information:

- Signature (FWS or CWS)
- Flash Version
- Uncompress file length
- Frame size (Dimension)
- Frame rate (Frames per seconds)
- Frame Count (Number of Frames)

ActionScript Code:

```
1
2  /* Define and Load asSWFbit Library */
3  import de.markusbordihn.flash.as3.asSWFbit_Library;
4  var SWF_data = new asSWFbit_Library("Headers"); // Set Mode to Header only (faster)
5
6  /* Load SWF file into ByteArray */
7  var file:ByteArray = ...; // Could be any ByteArray which has load the SWF file
8
9  /* Check for valid SWF file and parse Information to text fields */
10 if (SWF_data.parse(file)) {
11     txt_signature.text    = SWF_data.infos.Header.Signature;
12     txt_compress.text     = SWF_data.infos.Header.Compress;
13     txt_version.text      = SWF_data.infos.Header.Version;
14     txt_filelength.text   = SWF_data.infos.Header.FileLength;
15     txt_width.text        = SWF_data.infos.Header.Width;
16     txt_height.text       = SWF_data.infos.Header.Height;
17     txt_framerate.text    = SWF_data.infos.Header.FrameRate;
18     txt_framecount.text   = SWF_data.infos.Header.FrameCount;
19 }
```

In this example you see it is very easy to check for a valid SWF file and read out the SWF Header Information of this SWF file. The Library can be used on any ByteArray which has loaded a SWF file, so you can also use the URLLoader or the FileReference Object to load a SWF file into a ByteArray.

Read FileAttributes

Beside the Header Information there are some more Information which can be read out for more Information.

FileAttributes are available for flash files of version 8 and above.

So when the flash version is 7 and below you will not see any FileAttributes.

The FileAttributes include the following information:

- UseDirectBlit (Hardware Acceleration)
- UseGPU (Graphic Card Acceleration)
- HasMetadata (Metadata)
- ActionScript3 (Define if the file use ActionScript 3)
- UseNetwork (Local Sandbox Setting)

ActionScript Code:

```

1
2  /* Define and Load asSWFbit Library */
3  import de.markusbordihn.flash.as3.asSWFbit_Library;
4  var SWF_data = new asSWFbit_Library();
5
6  /* Load SWF file into ByteArray */
7  var file:ByteArray = ...; // Could be any ByteArray which has load the SWF file
8
9  /* Check for valid SWF file and parse Information to text fields */
10 if (SWF_data.parse(file)) {
11     ...
12     if (SWF_data.infos.Header.Version >= 8) { // Exists only in flash files version 8 or higher
13         txt_usedirectblit.text = SWF_data.infos.FileAttributes.UseDirectBlit;
14         txt_usegpu.text        = SWF_data.infos.FileAttributes.UseGPU;
15         txt_hasmetadata.text   = SWF_data.infos.FileAttributes.HasMetadata;
16         txt_actionscript3.text = SWF_data.infos.FileAttributes.ActionScript3;
17         txt_usenetwork.text    = SWF_data.infos.FileAttributes.UseNetwork;
18     }
19 }
```

Here we use first the SWF Header Information to check if the flash file is version 8 or higher. When yes we can read out and display the FileAttributes Information.

Read Metadata

The Metadata are sometime very helpfully. They can include a lot of Information. You can also define to include your own Information.

The asSWFbit Library parse some of the values automatically, but you have still access to the full raw xml information.

The most common / default information are the following:

- CreatorTool
- ModifyDate
- CreateDate
- Author / Creator
- Description

ActionScript Code:

```

1
2  /* Define and Load asSWFbit Library */
3  import de.markusbordihn.flash.as3.asSWFbit_Library;
4  var SWF_data = new asSWFbit_Library();
5
6  /* Load SWF file into ByteArray */
7  var file:ByteArray = ...; // Could be any ByteArray which has load the SWF file
8
9  /* Check for valid SWF file and parse Information to text fields */
10 if (SWF_data.parse(file)) {
11     ...
12     if (SWF_data.infos.Header.Version >= 8) { // Exists only in flash files version 8 or higher
13         ...
14         if (SWF_data.infos.FileAttributes.HasMetadata) { // FileAttributes tell us if there are Me
15             txt_creatortool.text = SWF_data.infos.Metadata.Description.xmp.CreatorTool;
16             txt_metadatadate.text = SWF_data.infos.Metadata.Description.xmp.MetadataDate;
17             txt_modifydate.text = SWF_data.infos.Metadata.Description.xmp.ModifyDate;
18             txt_createdate.text = SWF_data.infos.Metadata.Description.xmp.CreateDate;
19             txt_creator.text = SWF_data.infos.Metadata.Description.dc.creator;
20             txt_description.text = SWF_data.infos.Metadata.Description.dc.description;
21             txt_title.text = SWF_data.infos.Metadata.Description.dc.title;
22             txt_metadata.text = SWF_data.infos.Metadata.raw;
23         }
24     }
25 }

```

Here you see that most of the xmp entry's and dc entry's are already parsed to a variable for a easy access. The raw xml data are stored in "infos.Metedata.raw". Please keep in mind that this xml is working with namespace, so you need to defined namespace variables to access the different information in the raw data.

Read BackgroundColor and ScriptLimits

The BackgroundColor and ScriptLimits are sometime very intresting information. Please keep in mind that the SWF will only store ScriptLimits when there are different from the default Settings.

ActionScript Code:

```

1
2  /* Define and Load asSWFbit Library */
3  import de.markusbordihn.flash.as3.asSWFbit_Library;
4  var SWF_data = new asSWFbit_Library();
5
6  /* Load SWF file into ByteArray */
7  var file:ByteArray = ...; // Could be any ByteArray which has load the SWF file
8
9  /* Check for valid SWF file and parse Information to text fields */
10 if (SWF_data.parse(file)) {
11     ...
12     if (SWF_data.infos.BackgroundColor) {
13         txt_backgroundcolor.text = SWF_data.infos.BackgroundColor;
14     }
15     if (SWF_data.infos.ScriptLimits) {
16         txt_maxrecursiondepth.text = SWF_data.infos.ScriptLimits.MaxRecursionDepth;
17         txt_scripttimeoutseconds.text = SWF_data.infos.ScriptLimits.ScriptTimeoutSeconds;
18     }
19 }

```

The format of the BackgroundColor is in the RGB (Red Green Blue) Format: **RRGGBB**.

Read all TAGs

You can also read all TAGs of a SWF file to search for a special TAG or to get a general Overview over the SWF itself.
It is possible to access the raw content of each TAG, the parsed content is only available when the asSWFbit Library support the TAG.

ActionScript Code:

```
1
2  /* Define and Load asSWFbit Library */
3  import de.markusbordihn.flash.as3.asSWFbit_Library;
4  var SWF_data = new asSWFbit_Library();
5
6  /* Load SWF file into ByteArray */
7  var file:ByteArray = ...; // Could be any ByteArray which has load the SWF file
8
9  /* Check for valid SWF file and parse Information to text fields */
10 if (SWF_data.parse(file)) {
11     ...
12     if (SWF_data.infos.TAGs) {
13         txt_tags.text = "Found " + SWF_data.infos.TAGs.NumberOfTAGs + " TAGs...\n=====
14         for (var id:uint = 0; id < SWF_data.infos.TAGs.NumberOfTAGs; id++) {
15             if (SWF_data.infos.TAGs["ID" + id]) {
16                 txt_tags.appendText(id+'\t:\t' + SWF_data.infos.TAGs["ID" + id].name + " (size: "
17             }
18         }
19     }
20 }
```

All TAGs from a SWF file are stored at infos.TAGs, so when this is existing then you can be sure that the SWF file has some TAGs inside.
Each TAG has a unique ID over which you can access additional information. For more Information about the structure of this, please look at the "TAGs" section.

Overview over asSWFbit Library



This is a detail overview, which show all functions, events and constructors of the asSWFbit Library.
Below this there are more details information for each function, event and constructor.

Overview over constructors

Syntax	Description
asSWFbit_Library(Mode:String)	Create a asSWFbit_Library-Object. With the mode you can disable or enable additional features. "Headers" will only parse the Header Information and no additional TAGs or Information.

Overview over functions

Syntax	Description
parse(_DATA_:ByteArray) : Boolean	Try to load and parse the SWF file inside the ByteArray. Return true when there is a valid SWF file inside this ByteArray.

Overview over "infos" objects

The "infos" object is only available after parsing over the "parse" function and keep all Information about the SWF file.

Path	Description
infos.Header	Include all SWF Header information like Version, Filesize, Compress, Dimension
infos.FileAttributes	Include all the FileAttributes information like ActionScript3, Metadata.

infos.Metadata	Include all Metadata like Creator, CreatorTool, CreationDate.
infos.BackgroundColor	Include only the BackgroundColor in the RGB Format.
infos.ScriptLimits	Include all ScriptLimits information like MaxRecursionDepth and ScriptTimeoutSeconds.
infos.TAGs	Include all TAGs from the SWF file in a special format.

Overview over "infos.TAGs" objects

The "infos.TAGs" object include all TAGs from the SWF file in this special format.

Path	Description
infos.TAGs.index	Small search Index for the most commend SWF Information.
infos.TAGs.NumberOfTAGs	Include the Number of TAGs which was found in the SWF file.
infos.TAGs["ID..."]	For each TAG there is a unique Object with a ID.
infos.TAGs["ID..."].type	Include the Type ID of the TAG.
infos.TAGs["ID..."].name	Include the name of the Type of the TAG.
infos.TAGs["ID..."].start	Start Position of the TAG inside the DataArray
infos.TAGs["ID..."].end	End Position of the TAG inside the DataArray
infos.TAGs["ID..."].size	Size of the raw data, exclude type header
infos.TAGs["ID..."].pos	Overall Position of the TAG in the Object
infos.TAGs["ID..."].raw	Raw data of the TAG, exclude type header
infos.TAGs["ID..."].content	Parsed raw data, depend on the type which information are here.

Constructor

asSWFbit_Library (Mode:String)

This will create the asSWFbit Library object and assign it to a variable, this is needed to access this object later. You can also pass a *Mode*, where you can define which information you need and which information you didn't need.

ActionScript Code:

```

1
2  /*
3   asSWFbit Library Example from Markus Bordihn (http://markusbordihn.de)
4   */
5
6  import de.markusbordihn.flash.as3.asSWFbit_Library;
7
8  /* Example without any Mode (Default) */
9  var SWF_data = new asSWFbit_Library();
10
11 /* Example with Headers only Mode */
12 var SWF_data = new asSWFbit_Library("Headers");

```

In this example the variable "SWF_data" will be the asSWFbit Library Object. You can pass modes such "Headers" to decompile only the header information of a SWF file, in the future other modes will be supported as well.

Method

parse(_DATA_:ByteArray) : Boolean

With the parse method you load and parse the SWF file into the Info Object.
The Method will return "true" when it found a valid SWF file, when the SWF file is invalid it will return "false".
So you can check of this method if you have loaded a valid SWF file and when yes, you can try to access the SWF information.

ActionScript Code:

```
1
2  /* Define and Load asSWFbit Library */
3  import de.markusbordihn.flash.as3.asSWFbit_Library;
4  var SWF_data = new asSWFbit_Library();
5
6  /* Load SWF file into ByteArray */
7  var file:ByteArray = ...; // Could be any ByteArray which has load the SWF file
8
9  /* Check for valid SWF file and parse Information */
10 if (SWF_data.parse(file)) {
11     ...
12 }
```

You should always check first, if you have a valid SWF file before you try to access the information.

Objects



The asSWFbit Library store all of the information into Objects so you can easy check if a object is existing and if it has any information stored.

infos.Header

The "Header" Object store all header information of the SWF file.

Structure:

```
1
2  infos.Header = {
3      Signature:    ... // Signature of the SWF file (CWS = compress / FWS = uncompress)
4      Compress:     ... // Indicate if this is a compress or uncompress SWF file
5      Version:      ... // Version of the flash file
6      FileLength:   ... // FileLength of the SWF (uncompress) file
7      FrameSize:    ... // Frame size in twips format
8      {
9          Xmin: ... // x minimum position
10         Xmax: ... // x maximum position
11         Ymin: ... // y minimum position
12         Ymax: ... // y maximum position
13     },
14     Width:        ... // Calculated width of the Frame size in pixel
15     Height:       ... // Calculated height of the Frame size in pixel
16     FrameRate:    ... // Frames per second
17     FrameCount:   ... // Total number of frames
18 }
```

The "Width" and the "Height" information are only calculated values of the "FrameSize" Object, there are not stored in the SWF file directly.

infos.FileAttributes

The "FileAttributes" Object store all file attributes information of the SWF file.
This Object is only available for SWF files equal or higher then version 8.

Structure:

```
1
2  infos.FileAttributes = {
3      UseDirectBlit: ... // DirectBlit (Hardware Acceleration) will be used
4      UseGPU:        ... // GPU Acceleration will be used
5      HasMetadata:   ... // SWF file has Metadata
6      ActionScript3: ... // SWF file use ActionScript 3
```

```
1 UseNetwork:    ... // Local Sandbox can use Network
2 }
```

The values of these field are "1" for "true" and "0" for false. Please keep in mind that Hardware Acceleration is only available for flash player 10 and above.

infos.Metadata

The "Metadata" are only available when the `infos.FileAttributes.HasMetadata` is true, which should be the case.

Structure:

```
1
2 infos.Metadata = {
3   Description : {
4     xmp : {
5       CreatorTool:  ... // The Tools which create this SWF file
6       CreateDate:  ... // Create date of this SWF file
7       MetadataDate: ... // Create date of the Metadata
8       ModifyDate:  ... // Date of the last modification
9     },
10    dc: {
11      format:        ... // Fileformat, should be only "application/x-shockwave-flash"
12      description:   ... // Description of the SWF file
13      creator:       ... // Name of the Creator of the SWF file
14      title:         ... // Title of the SWF file
15    }
16  },
17  raw:              ... // Raw Metadata in XML Format
18 }
```

The raw Metadata use namespaces to access the different variables, so you need to know how you can use such namespaces.

infos.BackgroundColor

Store the BackgroundColor of the SWF file in a RRGGBB format.

Structure:

```
1
2 infos.BackgroundColor = ... // Background Color in RRGGBB format
```

It could be that future versions are store the BackgroundColor in RRGGBB and split in R,G and B.

infos.ScriptLimits

Store the ScriptLimits Information of the SWF file.

Structure:

```
1
2 infos.ScriptLimits = {
3   MaxRecursionDepth:  ... // Maximum of recursion of the ActionScript
4   ScriptTimeoutSeconds: ... // Display a warning message when the ActionScript is above this limit
5 }
```

These Information are only available when the SWF file don't use the default values.

infos.TAGs

The **infos.TAGs** is a very important object, it store all TAGs from the SWF file in parse and raw format. For each TAG there will be a unique identifier, so you can easy access any of these TAGs individual.

Structure:


```

1
2  infos.TAGs = {
3     index:           ... // Small Search Index for important information
4     NumberOfTAGs:    ... // Save Number of TAGs for a faster access
5     "ID[Position]": { ... // Example Entry for for a TAG each TAG has a unique ID
6         type:         ... // Tag Type (Number)
7         name:         ... // Parsed Tag Type Name (String)
8         start:        ... // Start Position of the DATA
9         end:          ... // End Position of the DATA
10        size:         ... // Size of the information (not include RECORDHEADER)
11        pos:          ... // General Position of all TAGs
12        raw:          ... // Raw Content of the DATA
13        content:      ... // Parsed Information of the TAG when possible / available
14    }
15 }

```

You can always access the raw content of any TAG even when there is no parser implemented for this type of TAG.

The parsed Information depend on the Tag Type, so for e.g: in the "BackgroundColor" TAG you will only found the BackgroundColor information. When you need to check all TAGs, you can create a loop which use the "NumberOfTAGs" information for the numbers of TAGs which needs to be checked.

In the Example section there is also a example of such a loop, to display all TAGs.