AIFF

Olaf `Olsen' Barthel

COLLABORATORS

	TITLE : AIFF	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Olaf `Olsen' Barthel	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AIFF

1.1 AIFF sound datatype

AIFF Sound DataType for Workbench 3.0
Written by Olaf ‘Olsen’ Barthel
Public Domain

1.2 introduction

"AIFF" is an acronym for "Audio Interchange File Format". This file format was devised by Apple Computer, Inc. and a number of third party developers. It is mostly used on Apple Computers, such as the notorious Apple II, Apple IIGS and the Apple Macintosh. With the introduction of System 7 the Sound Manager was enhanced to support direct playback of "AIFF" and "AIFC" format files, thus making this file format an integral part of the Macintosh operating system. Most digital sound recording software on the Apple Macintosh supports "AIFF" as the native sound file format. Rather recently, "AIFF" was adopted for use with Kodak Portfolio CDs since the corresponding development software is largely Macintosh-based.

"AIFF" and its ‘sister format’ "AIFC" are in many ways similar to the "8SVX" IFF format which is commonly used on the Amiga. Both are stored in IFF format, can contain sampled sound data and offer means to store musical instrument data. Unlike "8SVX" it provides additional services, such as MIDI support, data markers and audio recording information. Sampled sound data

is stored interleaved for multiple channels, making it easy to replay stereo sound. Samples do not need to be eight bits wide, such as required by the "8SVX" format. In fact, they can be anything from 1-32 bits wide.

While both "AIFF" and "AIFC" share the IFF file structure and thus are designed as interchangeable file formats to be used on many computer platforms Apple Computer, Inc. added data compression support to "AIFC". As always, all the standard data compression methods are based upon proprietary algorithms to which Apple Computer, Inc. probably holds patents and is unwilling to share. In essence, this restricts the portability of compressed "AIFC" files which violates the essential idea behind the IFF concept. To do it right, a new chunk would have had to be added to the original "AIFF" format description.

1.3 installation

The "AIFF" datatype distribution should consist of the following files:

- aiff.datatype

- AIFF
- AIFF.info

- AIFF.guide
- AIFF.guide.info

- Source code (in the "source" drawer)

Copy the file "aiff.datatype" into the "SYS:Classes/DataTypes" drawer. The "AIFF" and "AIFF.info" files should be placed in the "DEVS:DataTypes" drawer. In order to use the datatype you will need to reboot the machine.

1.4 features

The "AIFF" datatype can read and process both "AIFF" and "AIFC" format files which include sampled sound data. In the case of "AIFC" files the sound data needs to be stored in uncompressed format.

Sound samples will be cropped to eight bits per sample, stereo and multichannel data will be added up to yield monophonic sound.

In order to compensate for memory shortages or sound replay rates the Amiga sound hardware would be unable to support, the quality of the sound may get scaled down.

1.5 bugs

When running under Workbench 3.0, sounds larger than approximately 102,400 bytes will not play correctly. This is actually not a fault of the aiff.datatype, but rather of the sound.datatype superclass which does no

double-buffering. Application software can, however, easily compensate for this limitation.

1.6 author

The "AIFF" datatype was written by Olaf 'Olsen' Barthel using SAS/C 6.56. In order to recompile it you will need the Includes Release 40.15 or later. The datatype, the documentation and the accompanying source code are placed in the public domain.

The author can be reached at:

Olaf Barthel
Brabeckstraße 35
D-30559 Hannover
Federal Republic of Germany

or via Internet:

olsen@sourcery.han.de

1.7 history

v1.19
Rewrote part of the library close routine which could clash with the open routine if both were run at the same time.

v1.18
The dispatcher now supports saving sound data in AIFF file format. The library open code properly maintains its open counter even if the library initialization failed. If the initialization of libraries failed, no two successive calls to the library open routine will end in desaster.

v1.17
Internal development version only.

v1.16
The dispatcher now uses the DTA_Handle rather than opening the input file all on its own. I also rewrote large parts of the dispatcher and library core code for better readability.

v1.15
Internal development version only.

v1.14
Removed the asyncio and stack swap routines, polished and shortened the code.

v1.12
Removed the normalization code.

v1.11

The sound reader could trash 1-3 bytes following the allocated sample data, this has been fixed. The reader now uses Martin Taillefer's asyncio routines which I modified slightly for reentrancy. I also added a new routine which this datatype does not yet need.

v1.10
Small changes to the sound normalization code.

v1.9
This release no longer requires v40, but you may still experience problems when running under v39. See the section titled
Bugs
for more information. There is no longer so much stack space required to use this datatype. If necessary it will allocate new stack memory for itself and use it. Older releases could slightly distort the sound while normalizing it, this has been fixed.