Sound

Paul Manias

COLLABORATORS			
	TITLE : Sound		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Paul Manias	July 26, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Sound

1.1 Sound Object Documentation

```
OBJECT DOCUMENTATION
Name:      SOUND
Version:   0.9 Beta
Date:      October 1997
Author:    Paul Manias
Copyright: DreamWorld Productions, 1996-1997.  All rights reserved.

CHANGES VERSION 0.9B
Added:   Sound->LastChannel
         Sound->Attrib:SSTOPLAST

Deleted: Sound->Attrib:SBIT8
         Sound->Attrib:SBIT16
         Sound->Channel
```

1.2 Object: Sound

```
OBJECT
Name:    Sound
Module:  Sound
Version: 1
Type:    Simple

DESCRIPTION
The  purpose of the Sound object is to provide the means for any program to
load  and  play  a  sound file.  By default all loading and saving of sound
data  is  in  IFF  8SVX.   Formats  such  as WAVE can be supported in child
objects of the Sound class.

ACTIONS
The Sound object supports the following actions:

   * Activate()      Plays a Sound.
     CopyStructure() Copy sound fields.
     Deactivate()    Stop a sound from playing.
```

```
     Free()          Free a Sound object.
     Get()           Get a new Sound structure.
   * Init()          Initialise a Sound.

STRUCTURE
The Sound structure consists of the following public fields:

     Attrib      Sound attributes.
     Data        Address of sample data.
     File        Where the sound comes from.
     Frequency   Sampled frequency.
     Header      Sample info header, if any.
     LastChannel The last channel that this sound played through.
     Length      Length of sample data in bytes.
     Octave      Octave/Note setting.
     Priority    Priority.
     Volume      Volume of sample (1 - 100).
```

1.3 Object: Sound

```
FIELD
Name:      Attrib
Type:      LONG
On Change: Cannot change after initialisation.
Status:    Read/IWrite

DESCRIPTION
Specifies the attributes for the sound.

  SMODVOL
  Modulates the volume with the next channel.

  SMODPER
  Modulate the sound's period with the next channel.

  SREPEAT
  Repeats the sample forever.

  SLEFT
  Preferably play the sound through the left speaker.  Note that setting in
  combination with SRIGHT will produce unreliable results.

  SRIGHT
  Preferably  play  the sound through the right speaker.  Note that setting
  in combination with SLEFT will produce unreliable results.

  SFORCE
  Force  playing  through the selected speaker (eg if SFORCE|SLEFT then the
  sound will only play through left speaker channels).

  SSTOPLAST
  If  the  sound  is still playing through its last channel, stop it before
  playing through a new channel.
```

1.4 Object: Sound

```
FIELD
Name:        Data
Type:        APTR
Inheritance: Header
On Change:   Cannot change after initialisation.
Status:      Read/IWrite

DESCRIPTION
This field points to the actual sample data that is going to be played.
```

1.5 Object: Sound

```
FIELD
Name:        Header
Type:        APTR
Inheritance: Source
On Change:   Cannot change after initialisation.
Status:      Read/IWrite

DESCRIPTION
Points  to  the  very  start of the sample, which in most cases will be the
something  like an IFF 8SVX header.  If there is no header, then this value
will be NULL.

You  should only set this field if you want to intiailise your sound from a
memory location.
```

1.6 Object: Sound

```
FIELD
Name:   LastChannel
Type:   WORD
Status: Read Only.

DESCRIPTION
Specifies  the channel that this sound was last played through.  Acceptable
channel  numbers  range  from 1 to whatever the hardware limit is (absolute
maximum of 32,767).

This  field  is  managed  by the Activate() action.  If the sound could not
play  from the last Activate() (usually because there are no channels left)
then this field will be set to NULL on return.

If  the audio hardware does not use a channel based system, then this field
will be set to a random value each time you play a sound.
```

1.7 Object: Sound

```
FIELD
Name:        Length
Type:        LONG
Inheritance: Header
On Change:   Cannot change after initialisation.
Status:      Read/IWrite

DESCRIPTION
The length of the sample data in bytes.
```

1.8 Object: Sound

```
FIELD
Name:        Octave
Type:        WORD
Inheritance: Header
On Change:   Dynamic
Status:      Read/Write

DESCRIPTION
The  octave  at  which  to  play this sample.  The highest pitched value is
OCT_G0S, the lowest is OCT_A7S.  There are about 94 available settings, see
games/sound.i to look at the complete list.

SEE ALSO
Include: games/sound.i
```

1.9 Object: Sound

```
FIELD
Name:      Priority
Type:      WORD
On Change: Dynamic
Status:    Read/Write

DESCRIPTION
The  priority  of your sound goes here.  This field is used to determine if
your  sound  should be played if the destination Channel is currently busy.
Naturally, higher values are played over samples with lower values.
```

1.10 Object: Sound

```
FIELD
Name:      Source
Type:      APTR
On Change: Cannot change after initialisation.
Status:    Read/IWrite

DESCRIPTION
```

If your sound is located on disk, place a pointer to the source here. This will allow Init() to load the data in for you and fill in the Sound fields.

1.11 Object: Sound

FIELD
Name: Volume
Type: WORD
Inheritance: Header
On Change: Dynamic
Status: Read/Write

DESCRIPTION
The volume of the sound, which lies in the range 0 - 100. A volume of zero will not be heard, a volume of 100 is the loudest.

1.12 Object: Sound

ACTION
Name: Activate()
Object: Sound
Short: Play a sound through an audio channel.

DESCRIPTION
Plays a sound according to the settings in the sound object. Prioritised sounds will not be played if they cannot equal or better the channel's current priority setting.

Prioritisation of sounds allows you to play sound effects according to their importance. Make sure that you take care in ordering your sounds so that they play effectively!

It is recommended that you use ALLCHANNELS in the Sound->Channel field so that your program makes maximum use of all the available sound channels.

1.13 Sound: Init()

ACTION
Name: Init()
Object: Sound
Short: Initialise a sound structure for the play routines.

DESCRIPTION
This function will initialise a sound for use in the play routines. Its main job is to load and assess the sound header, and fill in any missing fields. It can also unpack sounds in some cases.

NOTE
If the sound is in RAW format you will need to set most of the fields yourself. (Keep in mind that storing sounds as raw data is definitely not

recommended).