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 Module: Sound

```
MODULE DOCUMENTATION
Name:       SOUND
Version:    0.9 Beta.
Date:       December 1997
Author:     Paul Manias
Copyright: DreamWorld Productions, 1996-1997.  All rights reserved.
Notes:      This  document is still being written and will contain errors
            in  a  number  of  places.   The information within cannot be
       treated as official until this autodoc reaches version 1.0.


CHANGES VERSION 0.9B
Deleted: CheckChannel()
         StopChannel()

Added:   StopAudio()
         StopSound()

Edited:  AllocSoundMem()
         FreeSoundMem()
```

1.2 Sound Module

```
 OBJECTS
 Sound

 FUNCTIONS
 AllocSoundMem()
 CheckSound()
 FreeSoundMem()
 SetVolume()
 StopAudio()
 StopSound()
```

1.3   **Sound: AllocSoundMem()**

```
FUNCTION
Name:     AllocSoundMem()
Short:    Allocate memory for sample playback.
Synopsis: APTR AllocSoundMem(LONG Size [d0])

DESCRIPTION
This function allocates a block of memory suitable for playing sound samples.
It  is unlikely that the average developer will need to use this function, as
the Sound object will handle audio memory when loading in sample data.

All audio memory is formatted with 0's when allocated.

INPUTS
Size - The Size of the memory to allocate.

RESULT
Returns a pointer  to  the  allocated memory.  This function returns NULL
if an error occurs.

SEE ALSO
Kernel: FreeMemBlock()
Sound:  FreeSoundMem()
```

1.4   **Sound: CheckSound()**

```
FUNCTION
Name:     CheckSound()
Short:    Check to see if a Sound object is currently playing.
Synopsis: LONG CheckSound(*Sound [a0]);

DESCRIPTION
This function will check a Sound object to see if it is currently playing
through the audio hardware.

RESULT
If  the  Sound is not playing, this function returns FALSE (NULL).  Otherwise
it returns TRUE (1).
```

1.5   **Sound: FreeSoundMem()**

```
FUNCTION
Name:     FreeSoundMem()
Short:    Frees a block of sound memory.
Synopsis: void FreeSoundMem(APTR MemBlock [d0])

DESCRIPTION
This function frees a block of sound memory.  Note that if you forget to free
any  memory  before  your  program exits, you will be presented with a system
error message.
```

For more details on memory handling, see the kernel documentation on this subject.

```
INPUT
MemBlock - Pointer to a Sound memory block allocated from AllocSoundMem() or
    AllocMemBlock()

SEE ALSO
Kernel: FreeMemBlock()
Sound:  AllocSoundMem()
```

1.6 Sound: SetVolume()

```
FUNCTION
Name:     SetVolume()
Short:    Set the volume for a sound.
Synopsis: void SetVolume(*Sound [a0], WORD Volume [d0])
```

```
DESCRIPTION
```
This function will set the volume for a sound object. If the Sound is currently playing, then the volume will be altered as the sound plays.

The Sound->Volume field will be updated to reflect the new volume.

1.7 Sound: StopAudio()

```
FUNCTION
Name:     StopAudio()
Short:    Put an immediate stop to all audio channels that are currently
          playing.
Synopsis: void StopAudio(void)
```

```
DESCRIPTION
```
This function wlil put an immediate stop to all audio channels that are currently playing sound data. Note that if a Music object is playing, then it will be stopped completely (you can restart it with Activate()).

1.8 Sound: StopSound()

```
FUNCTION
Name:     StopSound()
Short:    Stops a Sound that is playing.
Synopsis: void StopSound(*Sound [a0])
```

```
DESCRIPTION
```
This function will immediately stop a Sound from playing. The audio channel will then be made available to other Sound objects.