

# **Sulpha V1.0**

## **User Guide**

SCEE Technology Group

[scee\\_support@ps2-pro.com](mailto:scee_support@ps2-pro.com)

© 2002 Sony Computer Entertainment Europe  
All Rights Reserved

Last updated: 18th July 2003

Copyright of this software is attributed to Sony Computer Entertainment Europe. All rights and restrictions regarding the use and distribution of this software are according to the contract concluded between your company and Sony Computer Entertainment, Inc., Sony Computer Entertainment America and Sony Computer Entertainment Europe.

# Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>VERSION HISTORY</b> .....	<b>4</b>
<b>INTRODUCTION</b> .....	<b>5</b>
WHAT IS SULPHA? .....	5
<b>GETTING STARTED</b> .....	<b>6</b>
PLAYSTATION 2 .....	6
PC .....	7
<b>GETTING CONNECTED</b> .....	<b>8</b>
CONNECTION OPTIONS .....	9
MARKERS .....	10
<b>USING SULPHA</b> .....	<b>11</b>
COLOUR KEY .....	11
OPTIONS .....	11
WINDOWS MANAGER .....	12
<b>REGISTER VIEW</b> .....	<b>13</b>
INTRODUCTION .....	13
VOICE REGISTERS .....	14
CORE REGISTERS .....	17
SWITCHES .....	18
<b>HISTORY DISPLAY</b> .....	<b>19</b>
INTRODUCTION .....	19
HISTORY LIST .....	20
FILTER .....	21
SEARCHING .....	22
RUNNING COMMANDS .....	22
COMMAND VIEWER .....	23
<b>SPU2 RAM VIEW</b> .....	<b>24</b>
INTRODUCTION .....	24
RAM VIEW .....	24
OPTIONS .....	26
SELECTION HIGHLIGHT .....	26
MARKERS .....	26
<b>WARNING AND ERROR VIEW</b> .....	<b>27</b>
INTRODUCTION .....	27
DETECTED ERRORS AND WARNINGS .....	27
<b>CORE + DMA VIEWER</b> .....	<b>28</b>
INTRODUCTION .....	28
VIEWING DMA STATUS .....	28
CHANGING CORE EFFECTS .....	28
VIEWING SPDIF SETTINGS .....	29
<b>ENVX DISPLAYS</b> .....	<b>30</b>
INTRODUCTION .....	30
<b>SOUND EFFECT MANAGER</b> .....	<b>31</b>

INTRODUCTION.....	31
SOUND EFFECT VIEW .....	32
LOADING IN EFFECT LIST .....	32
SAVING OUT EFFECT LIST .....	32
AUDITIONING SOUNDS .....	32
REPLACING SOUNDS.....	32
<b>VOLUME CONTROLS.....</b>	<b>33</b>
INTRODUCTION.....	33
CHANNEL VOLUMES.....	33
CORE VOLUMES .....	33
SSA VOLUMES .....	33
<b>PLAYBACK COMMANDS .....</b>	<b>34</b>
INTRODUCTION.....	34
USING THE PLAYBACK COMMANDS .....	34
<b>EXTRACTING / SENDING DATA TO SPU2 RAM.....</b>	<b>35</b>
INTRODUCTION.....	35
EXTRACTING DATA .....	35
SENDING DATA .....	35

# Version History

## Version 1.00

First release.

# Introduction

---

## ***What is Sulpha?***

Sulpha stands for **S**ound **U**tility for **L**ow-level **P**erformance and **H**ardware **A**nalysis. It is a debug library which requires little or no code change to implement. It is completely transparent and offers the exact functionality of the LibSD libraries. It uses DECI2 to provide 2-way communication between the SPU2 and the PC application, giving full access to all events on the SPU2 and IOP memory.

# Getting Started

---

## *PlayStation 2*

Because Sulpha transparently replaces libsd.irx, the only changes needed is to replace the code used to load libsd.irx and replace it with code to load the two Sulpha irx's (SulphaComm.irx and SulphaSound.irx).

**Note :** Sulpha uses a considerable amount of IOP RAM (around 2.5 MB), and uses the DECI2 protocol for communication. As such it is only possible to run Sulpha on a DevKit. Sulpha will not work on a Test Station.

*Code Sample: Original libsd loading code*

```
while (sceSifLoadModule("host0:/usr/local/sce/iop/modules/libsd.irx", 0, NULL) < 0)
    printf("Can't load module libsd\n");
```

*Code Sample: New Sulpha loading code*

```
#ifdef SULPHAENABLED

while (sceSifLoadModule("host0:/usr/local/Sulpha/PS2/Modules/SulphaSound.irx", 0, NULL) < 0)
    printf("Can't load module SulphaComm\n");

while (sceSifLoadModule("host0:/usr/local/Sulpha/PS2/Modules/SulphaComm.irx", 0, NULL) < 0)
    printf("Can't load module SulphaSd\n");

#else

while (sceSifLoadModule("host0:/usr/local/sce/iop/modules/libsd.irx", 0, NULL) < 0)
    printf("Can't load module libsd\n");

#endif
```

When Sulpha is loaded, it outputs module information via the IOP TTY out:

*Output Sample*

```
/-----\  
| Sulpha Version 1.00  Build 29/04/03 |  
|-----|  
| Written by Oliver Hume @ SCEE |  
|-----|  
Thread priority   : 120  
Thread size      : 8192  
Send Buffer Size : 2500032  
Memory size     : ~2688880 bytes  
Timer Delay     : 2000 ms  
Module Info :  
Id              : 37  
Name            : SULPHA Debug  
Module Version : 101  
Entry Addr     : 684064  
GP Value       : 738048  
Text Addr      : 675888  
Text Size      : 27664  
Data Size      : 1744  
BSS Size       : 2659472  
|-----|
```

## **PC**

Simply execute the Sulpha.exe to start running Sulpha. Sulpha needs to be run on a non-readonly drive as the first time it is executed it will attempt to create a config file in the same location as the executable file. The config file requires ~2 megs of space.

# Getting Connected

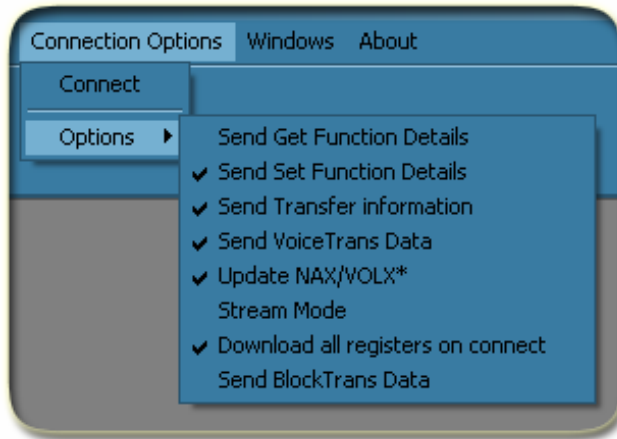
---

To connect the Sulpha PC application to the PlayStation 2, first execute the code on the PlayStation 2. Next enter the IP of the PlayStation 2 Devkit into the IP section of Sulpha. Once this is done, click on the connect button. The status bar will show the message 'PS2 Synced'. This means that the PC application is now connected to the PlayStation 2 and receiving data.





## Connection Options



Option	Meaning
Send Get Functions	Enabling this makes Sulpha send details of *Get Functions (i.e sceSdGetParam), allowing them to be shown in the history list. Some types of Sound Engines (i.e MultiStream) poll the *Get functions excessively to keep their internal variables upto date, causing a lot of data traffic for Sulpha, possibly slowing down other functionality. Default : Off
Send Set Functions	Enabling this makes Sulpha send details of *Send Functions (i.e sceSdSetParam), allowing the correct registers to be updated within Sulpha. Sulpha does not poll 'passive' registers (i.e VOLL) and uses the Set functions to keep its internal details upto date. By turning this off, Sulpha will incorrectly show details of Passive Registers, which can effect Channel Rebuilding, SFX Manager and Report Generation, as well as the Register View. Default : On
Send Transfer Information	Enabling this makes Sulpha send details of transfer parameters. Enabling this will not send the actual VAG/WAV data, but only the details of the transfer. This is overridden if 'Send Transfer Data' is enabled. Default : On
Send Transfer Data	Enabling this makes Sulpha send Voice and Block transfer data from the IOP to the PC. This enables the SPU2 RAM view to be updated, playback of SFX and Channel rebuilding. Turn this off if you are not interested in SPU2 RAM Details, which will save data traffic. Default : On
Update NAX/VOLX*	Enabling this will make Sulpha regularly update any non-static register (a register that is updated by the SPU2 and not only by the User PS2 Code) e.g NAX. Default : On
Stream Mode	Enabling this means Sulpha will not store any of the information received, instead only updating the current views to show the current state. This enables the client to stay connected to the PS2 indefinitely. Turning this on disables the History search, SPU2 playback, Channel Rebuilding and Report Generation. Default : Off
Download all registers on connect	Enabling this updates every register on update making Sulpha up to date from connection. Default : On
Download SPU2 RAM on connect (slow)	Enabling this means Sulpha will download the entire 2 meg of SPU2 RAM when it first connects. Note : downloading this amount of data can take a few seconds.

## **Markers**

It is possible to insert Markers into the data stream, giving the ability to mark and find important sections of capture within the History View. It is possible to insert a Marker via the PC application or within the User PS2 Code.

To insert a Marker via the PC Application, simply click on the Marker button at the appropriate time.

To insert a Marker via the User PS2 code, use the following function at the appropriate point in code :

*Code Sample:*

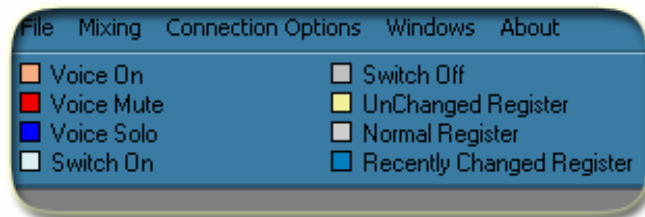
```
Sulpha_AddMarker("My Marker");
```

# Using Sulpha

---

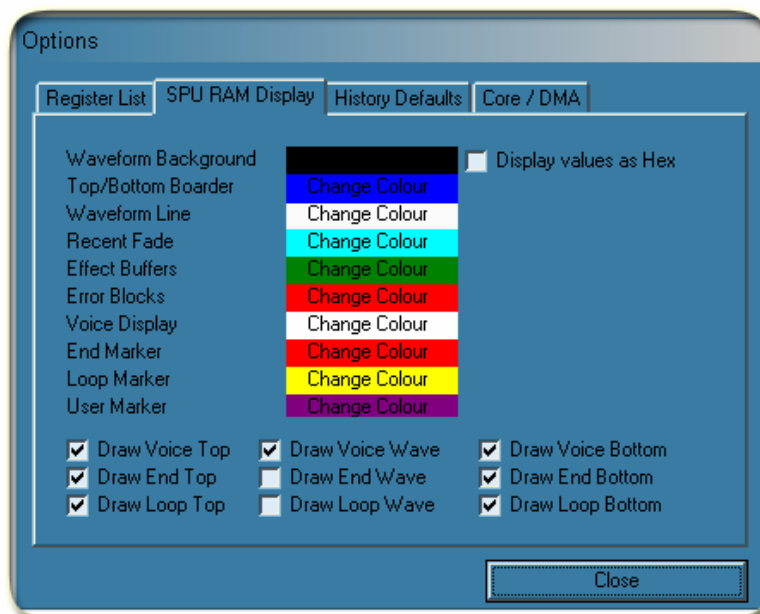
## Colour Key

Whenever you change windows within Sulpha, a colour key is updated to reflect the colours of the current view. This makes knowing what each colour is for each form at a glance easier.



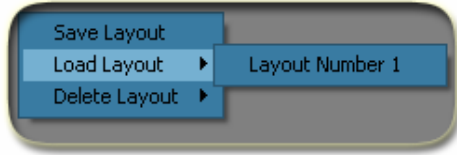
## Options

By selecting File->Options you can open the Option dialog box which enables you to customize any of the colours used by any of the forms. It also allows you to select the soundcard to be used in previewing SFX's.



## ***Windows Manager***

Sulpha contains an inbuilt windows manager, allowing the user to setup any number of layout preferences, and allowing the user to recall any of these layouts at a click of a button.



### **Saving Layouts**

Right click within the parent form of Sulpha, and select the 'Save Layout' option. You will be asked for a layout name.

### **Loading Layouts**

Right click within the parent form of Sulpha, and select the 'Load Layout' option, followed by selecting the layout you wish to load. Once the layout has been loaded, the number of windows and their location will be changed to match the layout when it was saved.

### **Deleting Layouts**

Right click within the parent form of Sulpha, and select the 'Delete Layout' option, followed by selecting the layout you wish to delete. You will be asked to confirm deletion.

# Register View

Registers

	LSAX	NAX	SSA	ADSR1	ADSR2	ENVX	PITCH	VOLL	VOLR	VOLXL	VOLXR
0	20496	54312	20496	255	8143	32759	3763	Constant : 8191	Constant : 0	16382	0
1	78864	112680	78864	255	8143	32759	3763	Constant : 0	Constant : 8191	0	16382
2	20480	20490	20480	0	0	0	1881	Constant : 0	Constant : 0	0	0
3	20480	20492	20480	0	0	0	1881	Constant : 0	Constant : 0	0	0
4	20480	20492	20480	0	0	0	1881	Constant : 0	Constant : 0	0	0
5	20480	20494	20480	0	0	0	1881	Constant : 0	Constant : 0	0	0
6	20480	20482	20480	0	0	0	16383	Constant : 0	Constant : 0	0	0
7	20480	20484	20480	0	0	0	16383	Constant : 0	Constant : 0	0	0
8	20480	20486	20480	0	0	0	16383	Constant : 0	Constant : 0	0	0
9	20480	20488	20480	0	0	0	16383	Constant : 0	Constant : 0	0	0
10	20480	20490	20480	0	0	0	16383	Constant : 0	Constant : 0	0	0
11	20480	20492	20480	0	0	0	16383	Constant : 0	Constant : 0	0	0
12	20480	20494	1191632	255	8128	0	1881	Constant : 1024	Constant : 5818	2048	11636
13	20480	714222	708240	255	8128	32759	1881	Constant : 1024	Constant : 4894	2048	9788
14	20480	748102	743760	255	8128	32759	1366	Constant : 1024	Constant : 4338	2048	8676
15	20480	4807726	4803240	255	8128	32759	4884	Constant : 1024	Constant : 5482	2048	40096

	AVOLL	AVOLR	BVOLL	BVOLR	EVOLL	EVOLR	MVOLL	MVOLR	MVOLXL	MVOLXR
Core 0	0	0	0	0	0	0	C: 16383	C: 16383	32766	32766
Core 1	32767	32767	0	0	0	0	C: 16383	C: 16383	32766	32766

01234567891011121314151617181920212223

242526272829303132333435363738394041424344454647

ENDXALL:

☐ Switch On☐ Switch Off

## Introduction

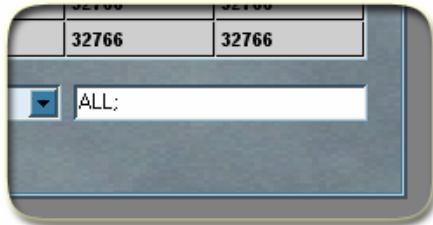
Sulpha allows real time representation, and changing , of all the SPU2 Parameters, Address and Switch registers The Register View window is split into three sections, Voice Registers (LSAX, NAX, SSA, ADSR1, ADSR2, PITCH, VOLL, VOLR, VOLXL, VOLXR) at the top, Core Registers (AVOLL, AVOLR, BVOLL, BVOLR, EVOLL, EVOLR, MVOLL, MVOLR, MVOLXL, MVOLXR) in the middle and Switches (ENDX, KON, KOFF, NON, PMON, VMIXL, VMIXR, VMIXEL, VMIXER) at the bottom.

The register view uses fading visual cues to show register changes. For the NAX registers, all voices which are above the 0x5010 value (playing) are highlighted to make seeing activate voices easier. For all other registers, any change will set the cell colour to the fade colour.

## Voice Registers

### Voice Filter

Included on the form is the filter text box.



This filter can be used to only show the Voices based on a set of rules. The possible rules that can be used are :

#### **All Voice Rule**

By simply specifying the rule “All”, it is possible to enable all channels.

#### **Individual Voice Rule**

It is possible to specify an individual channel to be viewed by entering the channel number. e.g. “12”.

#### **Voice Range Rule**

It is also possible to specify an ranger of channels to be viewed by entering the lower and upper voice numbers separated by a ‘-’ e.g. “4-15”.

#### **Register Rules**

Filters can also be register dependent. By entering the name of the register, followed by any valid symbol, followed by a value, it is possible to create complex filters.

Symbol	Meaning
=	Is equal to
!	Is not equal to
>	Is greater than
>=	Is greater or equal to
<	Is less than
<=	Is less or equal to

**e.g. ‘NAX>60000’ would only display voices that have a NAX larger than 60000.**

Note : it is possible to enter values as hex by prefixing the number by ‘0x’ e.g 0x5010

Any number of rules can be entered, separated by a comma, and ended with a semi-colon. For a voice to be displayed, it must be true on ALL the filter rules. If the rules entered are a invalid combination, the text box will be coloured red.

For example :

If the filter were set to “**2, 6-10, LSAX>20496, VOLL!0, VOLR!0;**” only channels 2,6,7,8,9,10 would be displayed IF the LSAX for those voices were greater than 20496 and both the VOLL and VOLR were not equal to 0.

## Locking Registers

Sulpha is able to lock individual or groups of Registers from the User PS2 code. By doing so, any attempt by the User PS2 code to change that register via a LibSD function call will be ignored. This is completely transparent from the User PS2 code, and in such, any values returned from any LibSD Get functions will return the User settings. For example, if the VOLL on channel 1 was locked while the value was set to 1000, and the User PS2 code called `sceSdSetParam(VOICE_1 | SD_VP_VOLL, 4000)`, the register will still maintain a value of 1000. But if the function `sceSdGetParam(VOICE_1 | SD_VP_VOLL)` was called, it would return the changed value of 4000.

To lock a register only on a specified voice number, right click on the cell and select 'Freeze Value'. To lock all voices of a register type, right click on any of the cells for that register and select 'Freeze Column'. It is also possible to lock all voices of a register type by left-clicking on the register name along the top of the table.

When a register is locked, it displays a red key in the cell :



VOLL	
Constant : 8120	! 🔑 Cor
Constant : 1024	🔑 Cor
Constant : 1024	🔑 Cor
Constant : 1024	! 🔑 Cor
Constant : 1024	🔑 Co

## Unlocking Registers

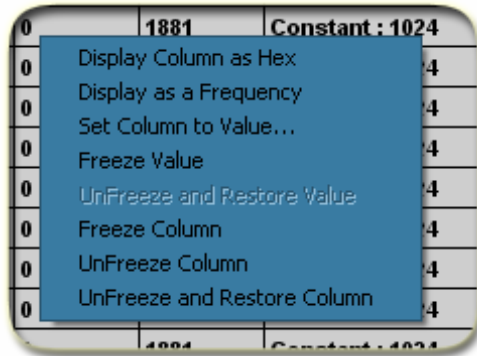
To unlock a register only on a specified voice number, right click on the cell and select 'UnFreeze Value'. This will unlock that register on that voice, and allow the User PS2 code to access this register. It is also possible to unlock the register and change it's value to the latest set by the User PS2 code while locked by selecting 'UnFreeze and RestoreValue'. To unlock all voices of a register type, right click on any of the cells for that register and select 'UnFreeze Column'. In the same way it is possible to restore the last set User PS2 code value per voice, it is possible to select 'UnFreeze and Restore Column'. It is also possible to unlock all voices of a register type by right-clicking on the register name along the top of the table.

## Muting Channels

Sulpha is able to mute and solo channels whilst running. By left-clicking on the voice number, the mute status for that channel will be toggled. This muting is completely transparent to the User PS2 code, and as such all readings of the registers from LibSD will return that the channel is still valid and playing. To solo a channel, right click on the channel number. This will mute all other channels, leaving the chosen channel as the only one playing.

## Viewing Options

By using the popup menu, it is possible to change any of the columns display type to either Decimal or Hexadecimal. It is also possible, for the SD\_VP\_PITCH column, to set the display type to Frequency, translating the SPU2 pitch to a standard Hz representation.



## Changing Values

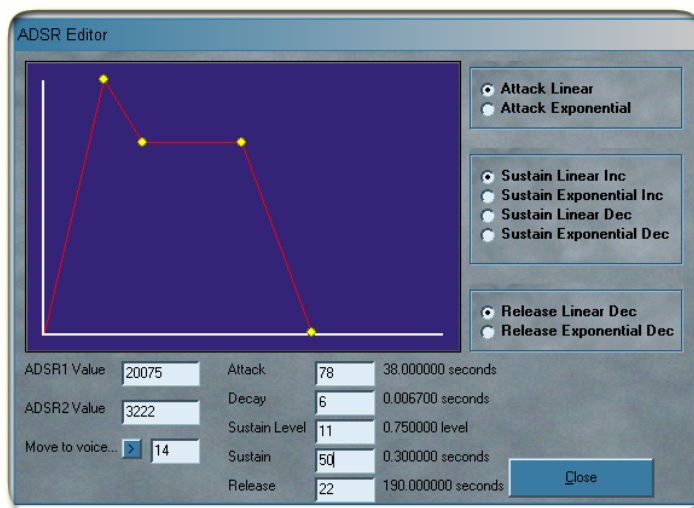
Sulpha allows two way communication between the PC and PlayStation 2. This allows the user to change values within the PC application and have the changes sent to the PlayStation 2. To change a value, simply double click within the cell of the desired Voice/Register and enter the new value.

Note : it is not possible to change LSAX, NAX, VOLXL or VOLXR registers as these are specified as read-only.

Note : it is possible to enter values as hex by prefixing the number by '0x' e.g 0x5010, and to enter frequencies for the PITCH register by prefixing the number by 'hz' e.g 44100hz.

## ADSR Editor

When either the ADSR1 or ADSR2 column is double-clicked to change the value, instead of manually entering a value, a ADSR Editor is displayed.



To use the ADSR editor, you can either enter the individual ADSR parameters into the corresponding fields, or instead, by clicking and holding the mouse on any of the boxes in the ADSR graph display, physically change the graph to match the desired specifications.

To set the ADSR settings for a voice, select the voice number and click on the '>' button.



## Core Registers


### Locking Volumes

Sulpha is able to lock individual or groups of Volumes from the User PS2 code. By doing so, any attempt by the User PS2 code to change that register via a LibSD function call will be ignored. This is completely transparent from the User PS2 code, and in such, any values returned from any LibSD Get functions will return the User settings. For example, if the MVOLL on channel 1 was locked while the value was set to 1000, and the User PS2 code called `sceSdSetParam(CORE_0 | SD_P_MVOLL, 4000)`, the register will still maintain a value of 1000. But if the function `sceSdGetParam(CORE_0 | SD_P_MVOLL)` was called, it would return the changed value of 4000.

To lock a volume only on a specified core, right click on the cell and select 'Lock Value'. It is also possible to lock both cores of a register type by left-clicking on the register name along the top of the table.

When a core volume is locked, it displays a red key in the cell :



MVOLL		
C: 16383		C
C: 16383		C

### Unlocking Volumes

To unlock a register only on a specified core number, right click on the cell and select 'UnLock Value'. This will unlock that register on that voice, and allow the User PS2 code to access this register.. It is also possible both cores of a register type by right-clicking on the register name along the top of the table.

### Changing Values

Sulpha allows two way communication between the PC and PlayStation 2. This allows the user to change values within the PC application and have the changes sent to the PlayStation 2. To change a value, simply double click within the cell of the desired Cell/Register and enter the new value.

Note : it is not possible to change MVOLXL or MVOLXR registers as these are specified as read-only.  
Note : it is possible to enter values as hex by prefixing the number by '0x' e.g 0x5010.

## Switches

### Selecting Switches

To change the current Switch in view, simply select the appropriate switch from the pull down menu.

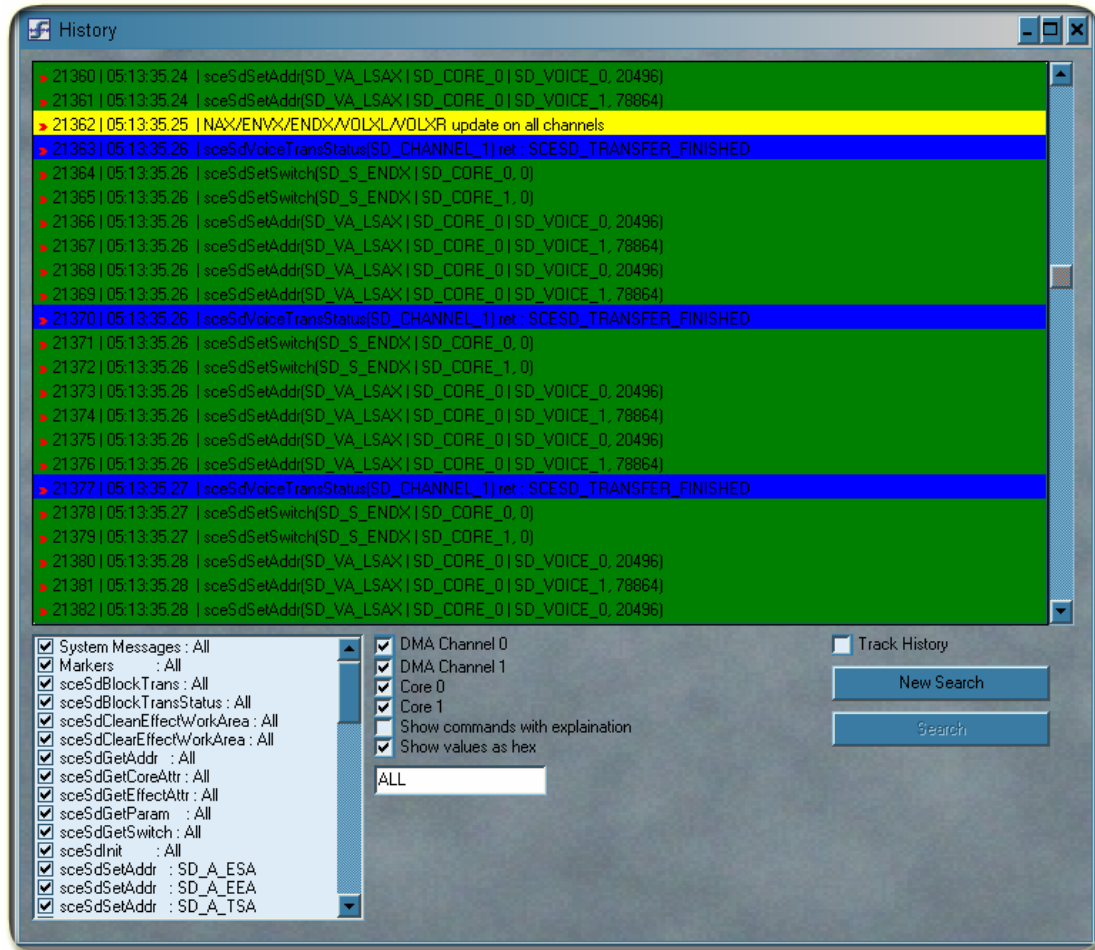


The switches can be divided into two types. Those which are triggers (KON, KOFF, ENDX) and those which are toggles (NON, PMON, VMIXL, VMIXEL, VMIXR, VMIXER). For switches that are triggers, the latest switch sent to the SPU2 is displayed. .e.g KON displays the last channels to be keyed on, and not which channels are keyed on. Toggle switches show all channels which are enabled e.g VMIXL shows all the channels which are currently actively being mixed.

### Setting Switches

To change the status of a switch, simply click on the voice number you wish to change. For trigger switches, this will trigger that individual channel. For toggle switches, this will change the status of that channel.

# History Display

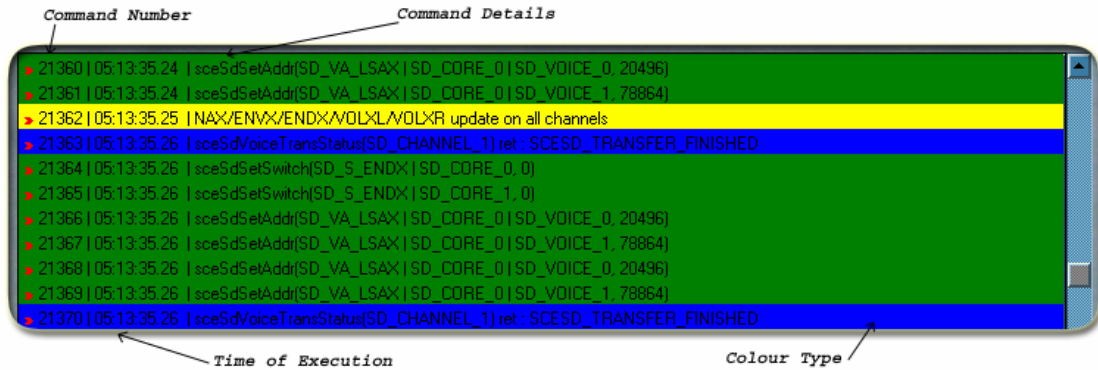


## Introduction

Every command Sulpha receives while connected is stored, and it is possible to view all the events captured, the time of capture and the parameters used via the History Display.

## History List

The list of commands received consist of the following information:



### Command Number

This is the commands item index in the history list.

### Time of execution

This shows the execution time of the command. The time represented is actually the PlayStation 2 execution time, and not the clock time.

Note : Certain system messages are generated by the PC application itself, and as such are displayed having a time of 00:00:00.

### Command Details

This shows the details of the command received. If 'Show commands with explanation' is selected the command is described. If not, the command is displayed in its original code format.

It is possible to show function parameters in hexadecimal by selecting 'Show values as hex'.

### Colour Type

Each command is colour coded as being one of the following :

*A System Command – a message generated by Sulpha.*

*A Write Command – a write message i.e sceSdSetParam.*

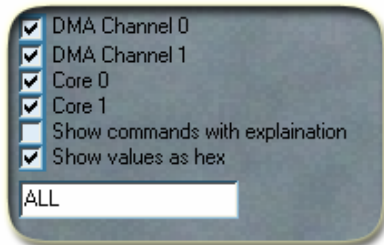
*A Read Command – a read message i.e sceSdGetParam.*

*A Marker – a marker message, as executed by the user.*

## Filter

It is possible to filter the history list to only show certain commands or specific core/channels.

To only view specify individual DMA Channels (for transfers) or individual Cores (for Core commands) toggle the appropriate check boxes:



To specify which channels (for channel commands) are used, use the channel filter. This filter can be used to only show the Voices based on a set of rules. The possible rules that can be used are:

### All Voice Rule

By simply specifying the rule “All”, it is possible to enable all channels.

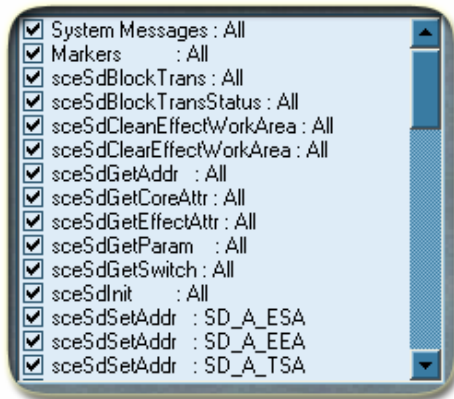
### Individual Voice Rule

It is possible to specify an individual channel to be viewed by entering the channel number. e.g. “12”.

### Voice Range Rule

It is also possible to specify an ranger of channels to be viewed by entering the lower and upper voice numbers separated by a ‘-’ e.g. “4-15”.

To specify which commands are shown in the history list, you can use the function list :



Simply left clicking on the command name will toggle wether it is displayed or not. Right clicking on the command name will toggle that command, and toggle every other command to the opposite.

Note : If the filter specified only finds a very few matching commands, it can make browsing the commands slower than normal.

## Searching

To search for a specific type of command, click on the 'New Search' button. This will bring up the search form :

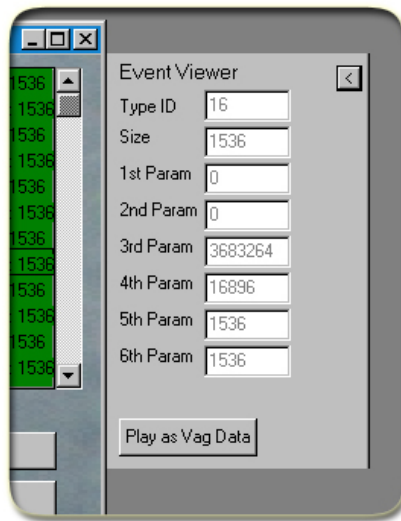
In the search form select the type of the command you wish to search for. It is also possible to specify parameter type, channel number, core number and function parameter values (where applicable). Once this is entered click on the 'Search' button to find the first entry which matches the specified parameters. To continue searching for more matches, click on the 'Search' button on the main History form.

Note : It is only possible to search for function types which are currently selected in the filter section.

## Running Commands

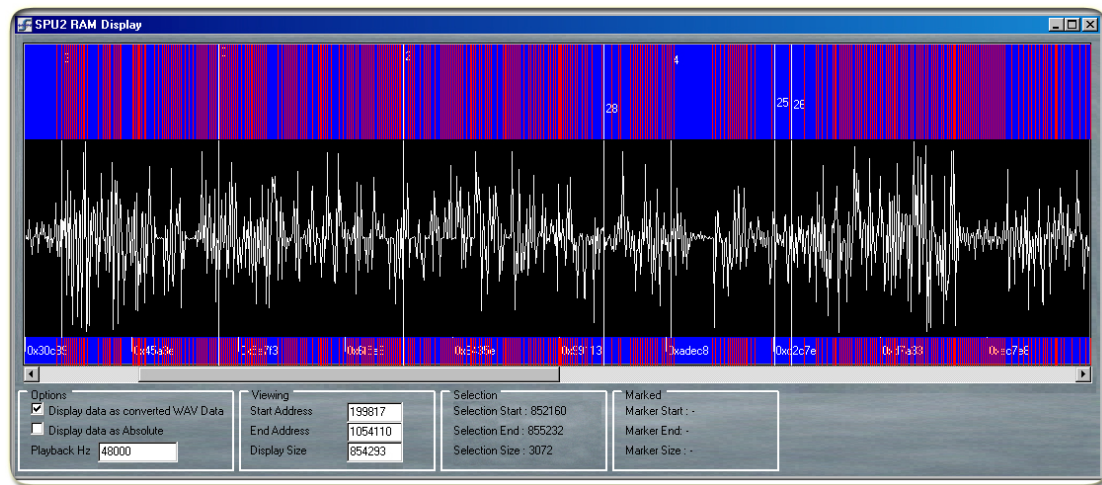
When you have disconnected from the PlayStation 2, it is possible to view the state of the SPU2 at any point of the capture. It is possible to do this from the history form, by selecting the point in the history list you wish to see, and right clicking, selecting 'Run to this command'. This will set all the displays to show the details of the SPU2 at that particular point of capture.

## ***Command viewer***



It is possible to view the packet information of each entry in the history list. This is useful for debugging any problems with Sulpha, but also serves another purpose. If the current selected item is a transfer instruction, it is possible to audition the transfer data by pressing 'Play as Vag Data'.

# SPU2 RAM View



## Introduction

Sulpha keeps an internal copy of all data transferred to SPU2 RAM, and can represent this data as a single wave. On this display it is possible to view the current viewing position all the channels and see loop/end markers within the VAG data received.

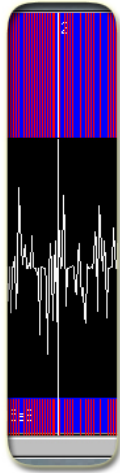
## RAM View

The SPU2 RAM view can be zoomed in by clicking the left mouse button on the place you wish to zoom into, and can be zoomed out by right clicking on any part of the wave form. You can specify what area of the SPU2 RAM you wish to view by entering the details into the 'Viewing' section of the form :

Viewing	
Start Address	199817
End Address	1054110
Display Size	854293



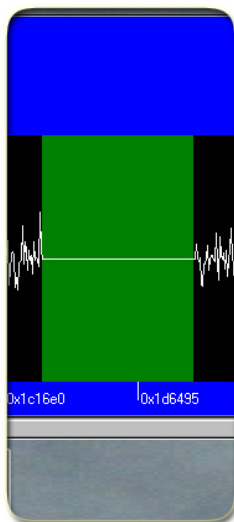
The SPU2 RAM View also shows the current location of any channels that aren't looping in the 0x5000-0x5010 range, making it easy to see which channels are currently playing and where.



It also displays any detected End/Loop markers in the received VAG data, and displays these as lines through the data (it is possible to specify which sections these are shown in via the Options screen)

<input checked="" type="checkbox"/> Draw Voice Top	<input checked="" type="checkbox"/> Draw Voice Wave	<input checked="" type="checkbox"/> Draw Voice Bottom
<input checked="" type="checkbox"/> Draw End Top	<input type="checkbox"/> Draw End Wave	<input checked="" type="checkbox"/> Draw End Bottom
<input checked="" type="checkbox"/> Draw Loop Top	<input type="checkbox"/> Draw Loop Wave	<input checked="" type="checkbox"/> Draw Loop Bottom

The current effect work area's and size's are represented on the wave form as coloured line (colour specified in the Options screen)



## Options

The options available are :

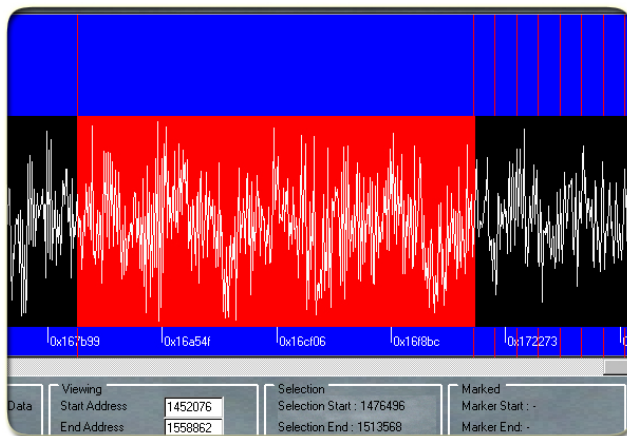
'Display data as converted WAV data' – Displays the VAG data received as converted WAV data on the wave form. It is useful to turn this off if you store WAV data in the SPU2 RAM, and wish to view it in its native form.

'Display data as Absolute' – Changes the display to either a standard wave display, or an absolute wave display, where the sample value is taken as an absolute of itself, and shown both above and below the centre line.

'Playback Hz' – When a SFX is auditioning by holding down Ctrl and clicking the left mouse button, the SFX is auditioned at the sample rate specified here.

## Selection Highlight

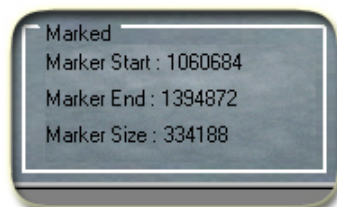
When you move the mouse over the SPU2 RAM display, Sulpha attempts to detect the start and end of the sound effect under the mouse cursor. This SFX is highlighted, and it is possible to play the SFX on the PC (hold the Ctrl key and press the left mouse button). Details of the SFX are shown under the Selection section :



Note : It is possible to drag a VAG file from outside of Sulpha, and drop it into the SPU Display. When this is done, the file is transferred to the starting address of the selection highlight it was dropped onto.

## Markers

It is possible to insert your own markers in the SPU2 Display by holding down the Alt key and left-clicking on the SPU2 RAM display. The first click will set the beginning marker. The second click will set the end marker. A third click will reset the markers. Once these markers have been set, their details are shown under the Markers section :

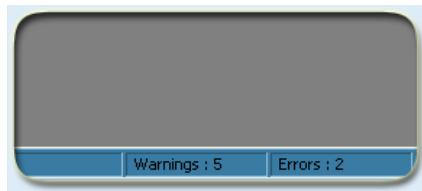


# Warning and Error View

---

## *Introduction*

Sulpha monitors all commands sent to the SPU2, and runs an internal sanity check on each command to make sure each is valid. Any command sent to the SPU2 which is invalid or possibly invalid is highlighted via the Warning and Error view. If any warnings or errors have been detected, the status panel will flash and display how many have been detected :



## *Detected Errors and Warnings*

Only SD\_VP\_\* and SD\_P\_\* registers are used with sceSdSetParam and sceSdGetParam.

Only SD\_VA\_\* and SD\_A\_\* registers are used with sceSdSetAddr and sceSdGetAddr.

Only SD\_S\_\* registers are used with sceSdSetSwitch and sceSdGetSwitch.

Only CORE\_0 or CORE\_1 are used for Core specific functions.

Only VOICE\_0 to VOICE\_23 are used for Channel specific functions.

Any IOP or SPU2 RAM address specified for transfer is between 0 and 2 Meg.

Any Effect Mode specified is a valid Effect Mode.

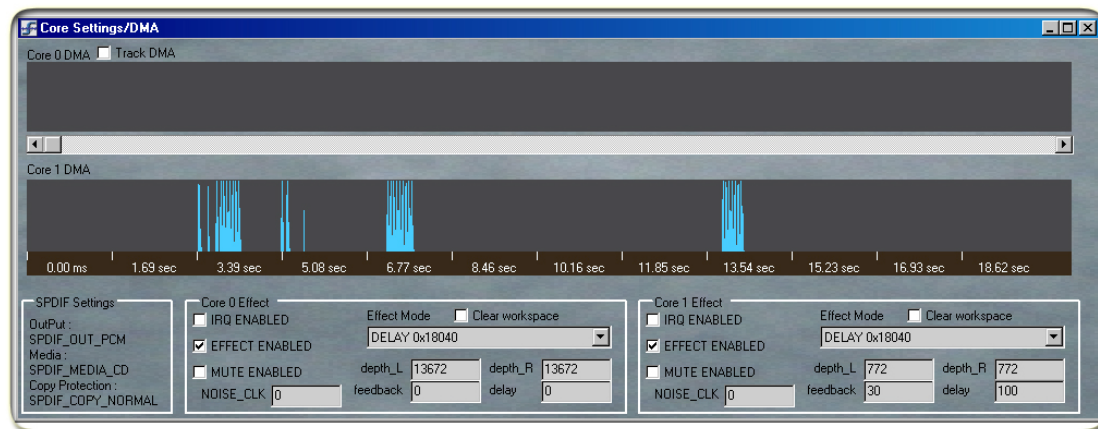
All Param/Addr values are valid.

Any Voice or Block transfer doesn't transfer into the 0x5000 – 0x5010 range.

Any Voice or Block transfer doesn't transfer into the effect work areas.

All Effect Params are valid.

# Core + DMA viewer



## Introduction

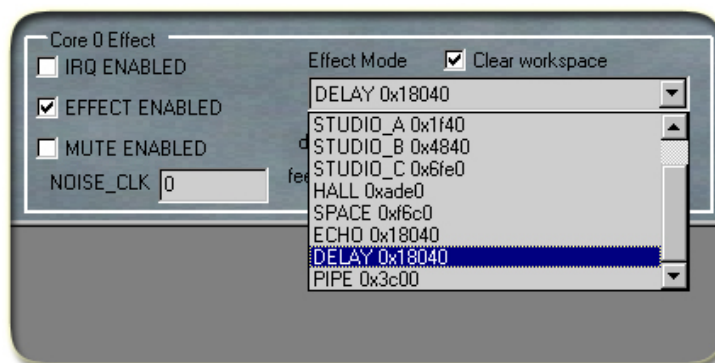
The Core+DMA viewer screen can be used to view/change the Core Effect modes/params. It also shows the SPDIF settings, and an activity monitor of both the DMA channels showing DMA activity on a time graph.

## Viewing DMA Status

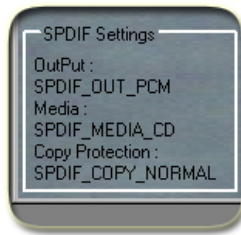
The status of both DMA channels can be observed from the Core + DMA Viewer window. As the capture progresses, an activity graph is generated. It is possible to zoom in on these graphs by clicking the left mouse button on either of the DMA channels. It is possible to zoom out by right clicking on either DMA channels. All the different types of DMA activity are displayed as individual colours.

## Changing Core Effects

To change either of the Core Effect types, firstly toggle the 'Clear Workspace' option. Next display the drop down list of Effect types and select the appropriate one. To change the effect parameters, simply type the new value into the box and press enter.



## Viewing SPDIF Settings

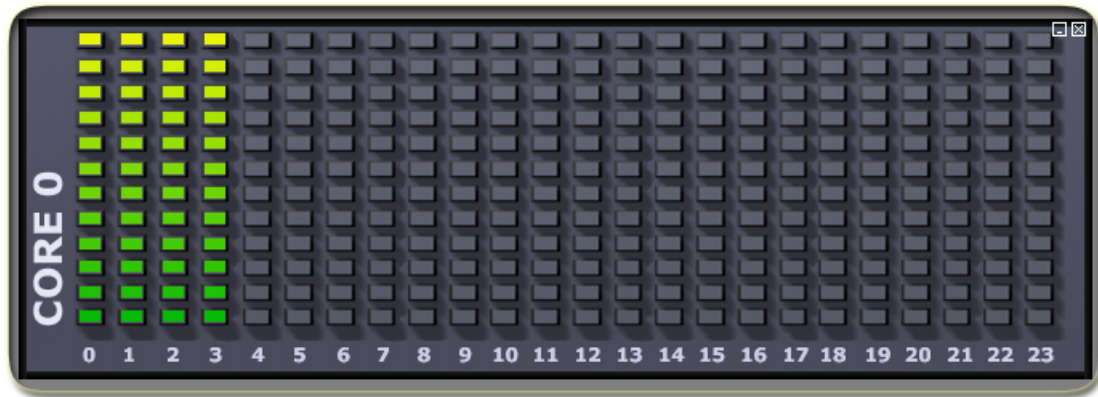


The SPU2 SPDIF settings are shown within the CORE+DMA viewer screen. These cannot be edited.

Note : If the SPDIF\_COPY\_PROHIBITED mode is turned on, Sulpha will not transfer any digital data from the PlayStation 2 (i.e transfer data).

# ENVX Displays

---

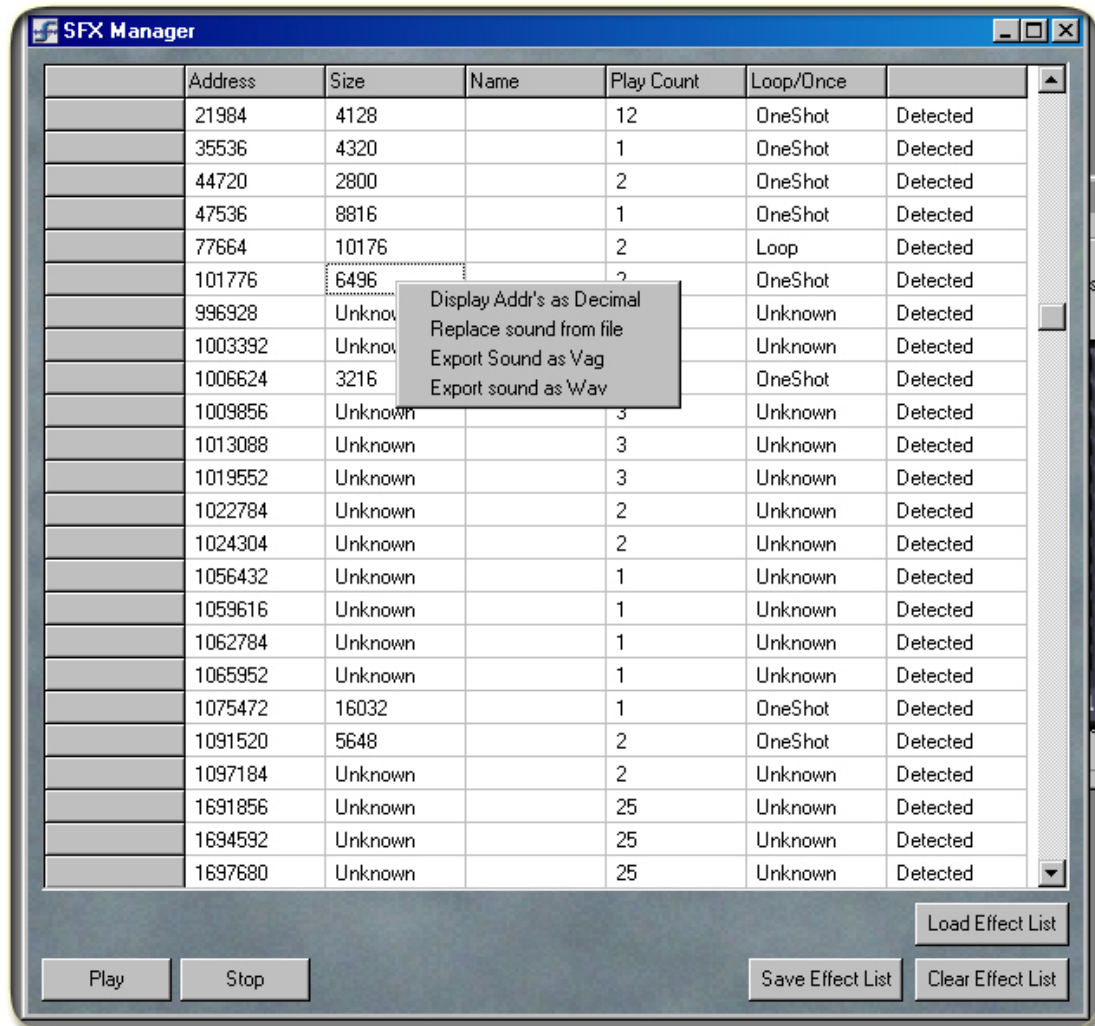


## *Introduction*

Sulpha contains an ENVX view for both cores, which can be used for quick visual reference to the state of all 48 channels. To create a new instance of an ENVX core viewer, goto Mixing->Create ENVX Display and then choose the core you wish to view.

# Sound Effect Manager

---



## Introduction

While Sulpha is capturing SPU2 data, it monitors all Keyon events, and deciphers this information to extract possible individual SFX locations. Using this information it is possible to replace the SFX from the PC, view how many times the effect is used, name the effect for ease of use, and highlight used effects on the SPU2 RAM Display.

## ***Sound Effect View***

The main part of this panel displays a number of entries per SFX. Each time a new SFX is played by the PS2, the information is updated accordingly. If the SFX has now been played before, a new entry is amended to this list. Otherwise, relevant information for this SFX is updated.

The information for each SFX contains the following entries:

SPU2 RAM Address:	Start address of SFX
Size:	Size of SFX
Name:	Name of SFX (user can give each SFX an individual name if required)
Play Count:	Amount of times the SFX has been triggered
Loop/Once	Displays if the SFX is either looping or once-off

The Address parameter is calculated by Sulpha checking the address which was set when a KeyOn was issued by the users' code.

The Size parameter is calculated by Sulpha looking ahead from the KeyOn SPU address until it reaches either an end of sample marker or a start marker for the next sample in SPU RAM.

## ***Loading in Effect List***

It is possible to load in SFX names from previously saved Effect lists, or from lists created by the user or users programs.

The effect list is a list of simple entries in the following manner :

0x5010 , NameOfEffect

50000, NameOfEffect2

When this list is loaded, any SFX with a SSA of 0x5010 will be named of NameOfEffect, and any with an SSA of 50000 will be names NameOfEffect2. This only works with statically loaded SFX.

## ***Saving out Effect List***

It is possible to extract all SFX name information (including information entered by the user concatenated with any lists previously loaded) and save it as a list file. Simply click on the 'Save Effect List' button and select a output location.

## ***Auditioning Sounds***

To audition any sound within the SFX manager, simply select the entry, and either double click on any column apart from name, or click on the button labeled 'Play'. Note : if you have not captured the transfer information for this SFX, it will play silence.

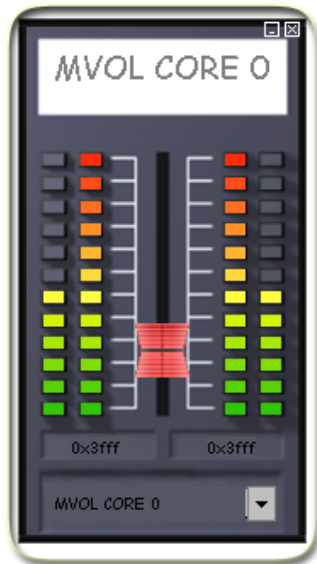
## ***Replacing Sounds***

It is possible to replace the data stored at the SFX location with data from the PC. It is possible to do this either by dragging the file from Windows Explorer onto the correct SFX Row. It is also possible to right click on the row, and select the popup menu item 'Replace sound from File'.



# Volume Controls

---



## ***Introduction***

Sulpha has the ability to transparently fade volumes being used by the SPU2, enabling the user to mix the volumes of the Cores, Channels and SFX's in game to get the perfect mix.

## ***Channel Volumes***

Select Mixing->Create Volume Control->Voice to create a new instance of a individual channel fader. You can select the channel you wish to fade, but dropping down the Selection box at the bottom of the control.

## ***Core Volumes***

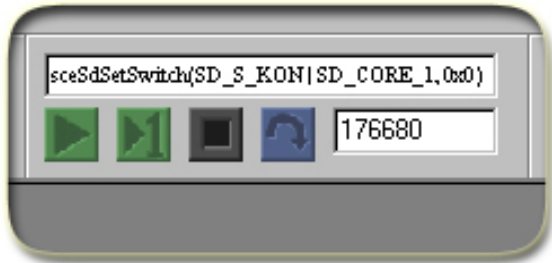
Select Mixing->Create Volume Control->Master Volumes to create a new instance of a core volume fader. You can select the core volume you wish to fade, but dropping down the Selection box at the bottom of the control.

## ***SSA Volumes***

Select Mixing->Create Volume Control->SSA Volumes to create a new instance of a core volume fader. You can select the SFX volume you wish to fade, but dropping down the Selection box at the bottom of the control. The SFX you wish to fade must have been captured by the SFX Manager. Whenever that SFX is played on any channel, Sulpha will detect this and fade that voice, allowing for fading of SFX's on dynamically allocated channel systems.

# Playback Commands

---



## ***Introduction***

Once a capture is complete, Sulpha is able to play through the commands received, replaying the actions of the SPU2. This allows methodological step-by-step debugging of the SPU2.

## ***Using the Playback Commands***

There are several ways to playback the commands stored by Sulpha. The simplest method is to enter the command number you wish to view up to into the debug number text box, next to the 4 playback buttons. Once this has been entered Sulpha will restart from the beginning of the captured list and run through to the number you have specified.

The second method is to use the History List, right click on the entry you wish to run too and select the option 'Run to this Command'.

The third method is to use the debug command buttons. The buttons offer the following functionality:



: Attempts to play through received command list in real time from the current entry selected.



: Plays through the received command list at a reduced speed.



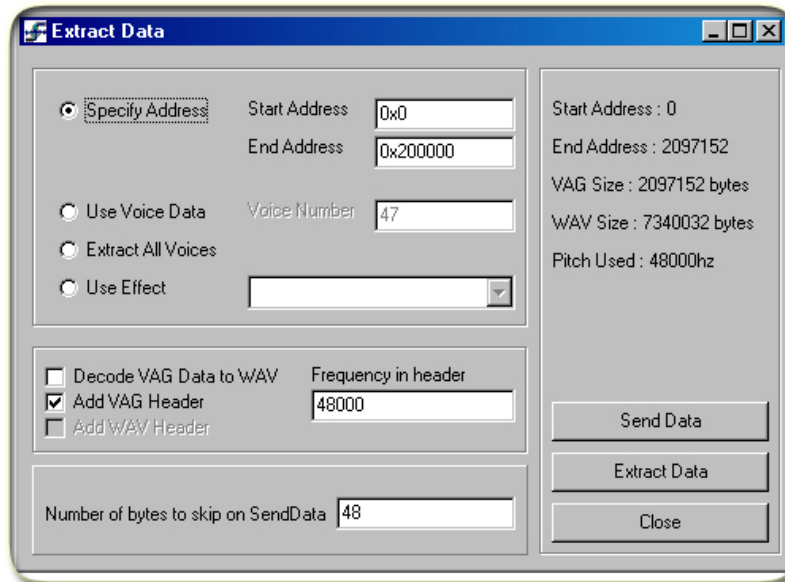
: Stops any currently running debug replay.



: Steps through the received command list a single command at a time.

# Extracting / Sending Data to SPU2 RAM

---



## Introduction

Sulpha has 2 way communication with the SPU2 RAM, and as such can extract data from the RAM, and send data to the RAM.

## Extracting Data

It is possible to extract data from the RAM using different types of parameters. It is possible to extract a range of data, by selecting 'Specify Address' and entering the start and end address and clicking 'Extract data'. It is also possible to extract data on a per voice basis, or from all voices. Using this option allows Sulpha to get the start address from the Voice's SSA, and search through memory for the first End/Loop Marker it finds, to form the end address for the extraction. It is also possible to use this method to get the start and end extraction address's from the SFX listed in the SFX Manager.

Note : When extracting data from the RAM, it is important to note this data isn't sent over from the SPU2 on request, but rather from the reconstruction of the RAM on the PC. This means that any transfer commands must have been captured or the 'Dump SPU2 RAM on connect' option is enabled.

## Sending Data

To send data to the SPU2 RAM, select the address you wish to send too, and enter this in the Start Address section (please note, End Address is ignored for sending data, the size of the file is used). If the file you are sending contains a header you do not want to be sent, use the 'Number of bytes to skip on SendData' section to specify how many bytes to skip. Next click on Send Data and select the file you wish to send. Note : It is important to remember that sending data via the PC to the SPU2 is very similar to sending data via the PS2 to SPU2. Do not write in area's below the 0x5010 range. Also remember all transfers sizes will be rounded upto 64 byte multiples.