

SCREAM Tool User's Guide

© 2007 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

"Dolby" is a trademark of Dolby Laboratories.

Table of Contents

1 Introduction and Installation	5
Hardware Requirements	5
Installing the SCREAM Tool	5
Installing the SCREAM Tool Server	8
Repairing or Removing SCREAM Tool or the SCREAM Tool Server	10
Launching the SCREAM Tool and the Server	11
Connecting to the SCREAM Tool Server	13
Using the SCREAM Tool Server	14
Checking the Version and Exiting the SCREAM Tool	17
2 Building a Bank that Plays a Sound / Creating a Bank	18
Key Concepts	18
Interface Overview	18
Creating a New Bank	19
Bank Control – Interface Overview	19
3 Importing a Waveform	21
Importing a Waveform	21
Editing Waveform Parameters in the Waveform Properties Control	22
Editing the Original Source File for a Waveform	24
Refreshing Waveform	24
Looping Sound Using a Looping Waveform	24
4 Creating a Sound	26
Creating a Sound	26
Accessing the Sound Properties Control	27
Specifying Information About a Sound	28
Setting Sound Defaults	29
Determining What Waveforms a Sound References	29
5 Adding Grains to the Sound Script	30
Using the Grain Palette	31
Setting Grain Defaults	32
Using the Grain List	32
More About Grain Validation and Errors	32
Using the Grain Properties Control	32
Playing the Sound	35
Example: Adding Random Pitch Bend Using a Waveform Grain	36
More Complex Examples	37
6 Scripting with Registers	38
Using the Register Control	38
Using Sound Registers	39
Using Global Registers	41
7 Importing and Exporting	44
Importing a Bank (Older Format)	44
Exporting Binaries	44
Exporting a C Header File	44

Exporting File To Be Read By Older Version of SCREAM Tool	44
Exporting a Text File	44
Exporting Multiple File Formats Using the Batch Option	45
8 Modifying Bank Preferences	46
Modifying Bank Preferences (Data Stored and Memory Usage)	46
Checking Current Memory Usage.....	47

1 Introduction and Installation

The Scriptable Engine for Audio Manipulation (SCREAM) offers you the ability to create complex, interactive Sounds with a minimum of programmer support. SCREAM consists of the following components:

- SCREAM Runtime Library – Works in conjunction with the sound banks created using the SCREAM Tool. It offers sound scripting, plug-in support, and priority-based voice management.
- SCREAM Tool Server – Runs alongside the SCREAM Tool to simulate the PLAYSTATION®3's sound capabilities. This allows you to audition and tune the audio while running the SCREAM Tool.
- SCREAM Tool – Provides a GUI for designing *sound banks*; the basic data format used by SCREAM. A sound bank contains waveform data (VAG data) and some number of sound scripts that describe how the waveforms are to be played.

For a detailed system overview that describes how the SCREAM components work together, see the "SCREAM Overview" document

This document focuses on the SCREAM Tool and provides the steps you need to install both the SCREAM Tool and the SCREAM Tool server. It also describes how to use the SCREAM Tool to create sounds with scripts and generate sound banks for use on the PLAYSTATION®3.

Hardware Requirements

The hardware requirements for running the SCREAM Tool and the SCREAM Tool Server include:

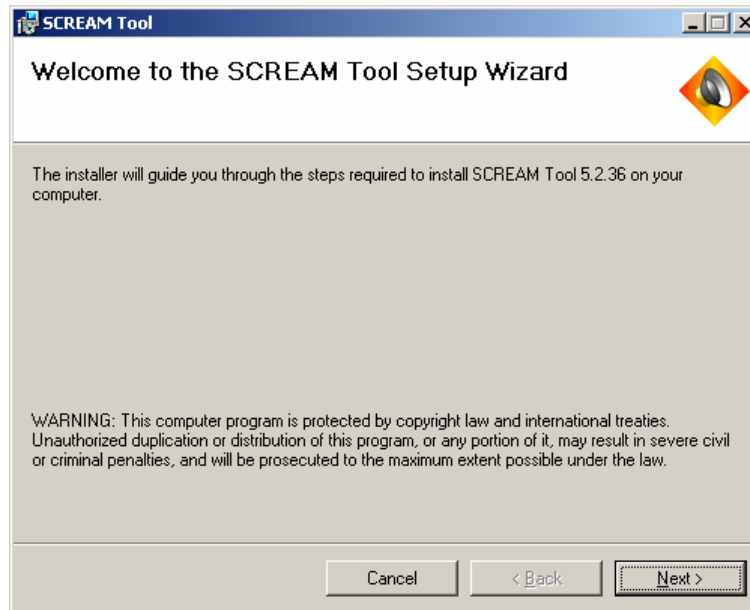
- PC running Windows 2000/XP
- A sound card for audio playback
- Screen resolution of 1280x1024 (recommended minimum)
- 6 Megabytes of hard disk space

Installing the SCREAM Tool

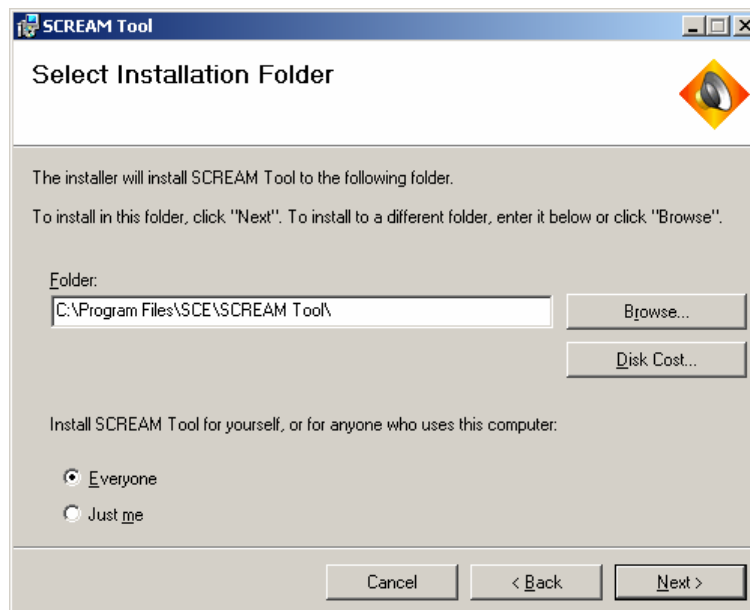
Download the **scream_tool.zip** file then unzip it to a directory (for example, C:\usr\local).

Navigate to the unzipped **SCREAM_Tool** directory (for example, C:\usr\local\cell\tool\SCREAM\SCREAM_Tool), where you will see the SCREAM Tool setup files. Find and run the **SCREAM_Tool_setup.exe** file.

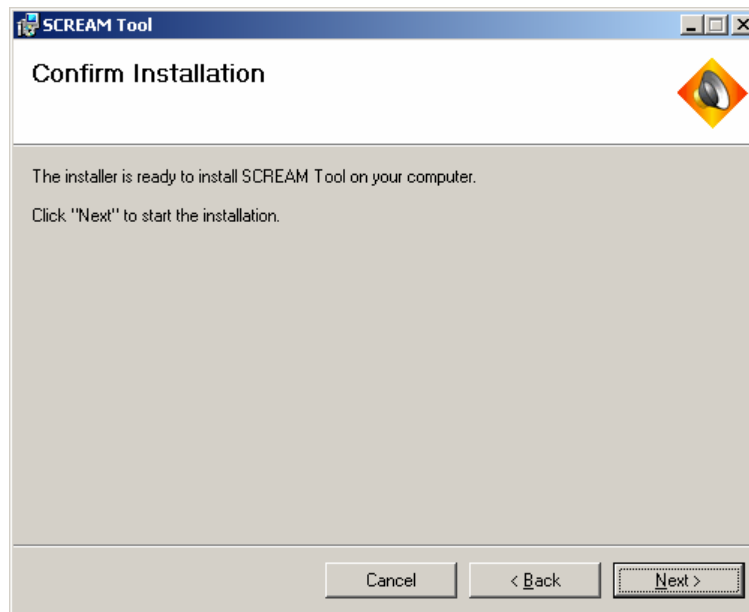
The first window of the SCREAM Tool Setup Wizard displays, as shown in Figure 1. Click **Next** to continue.

Figure 1 SCREAM Tool Setup Wizard

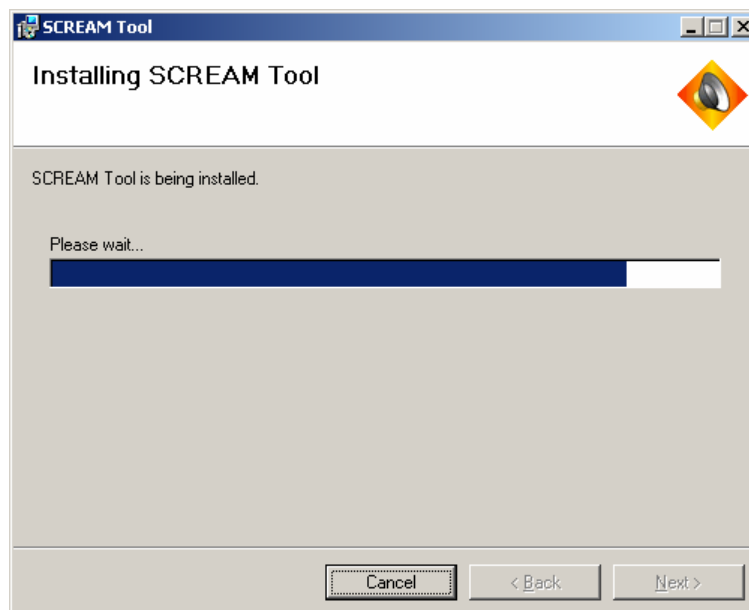
The dialog before the actual installation, shown in Figure 2, prompts you to choose a destination folder to install the SCREAM Tool. Unless you specify otherwise, the tool will be installed in the location shown. Click **Next** to continue.

Figure 2 Tool Installation Location

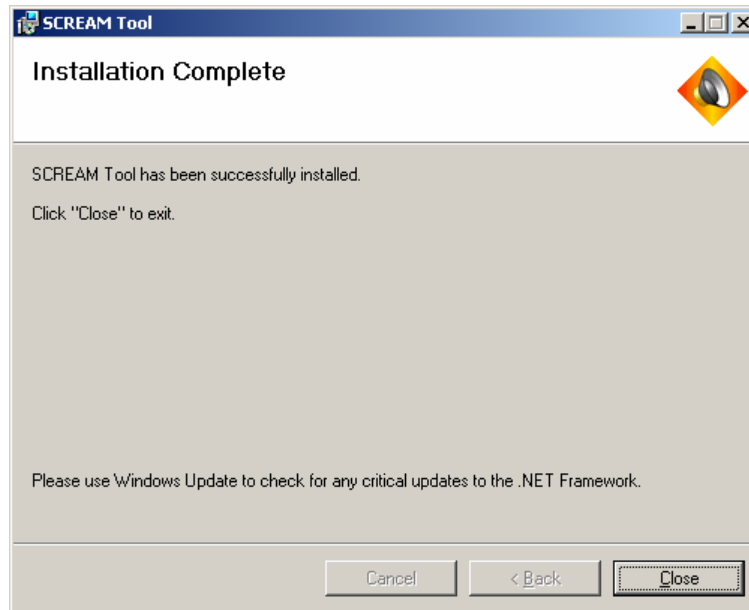
The Setup Wizard displays a confirmation dialog, shown in Figure 3. Click **Next** to continue.

Figure 3 Confirm Installation

The Setup Wizard starts the installation process, as shown in Figure 4.

Figure 4 Installation In Progress

The Setup Wizard finishes installing the tool and displays a final dialog, shown in Figure 5. Click **Close** to close the window.

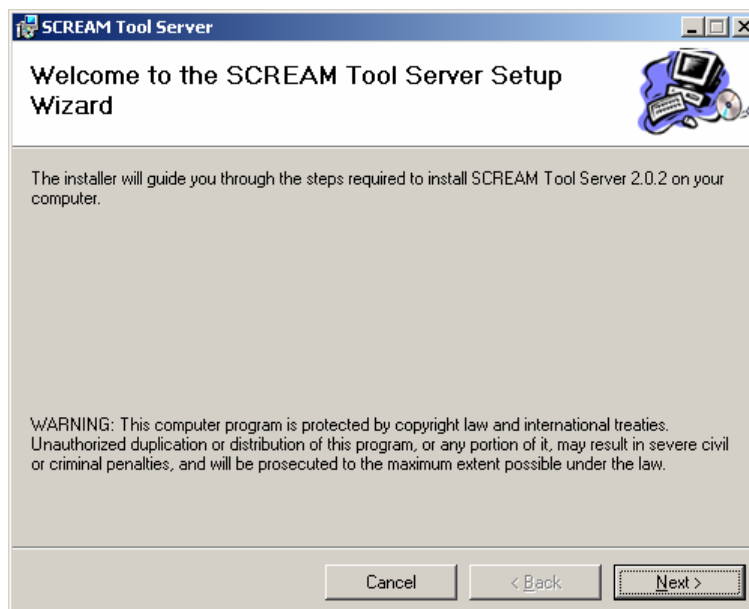
Figure 5 Installation Finished

Installing the SCREAM Tool Server

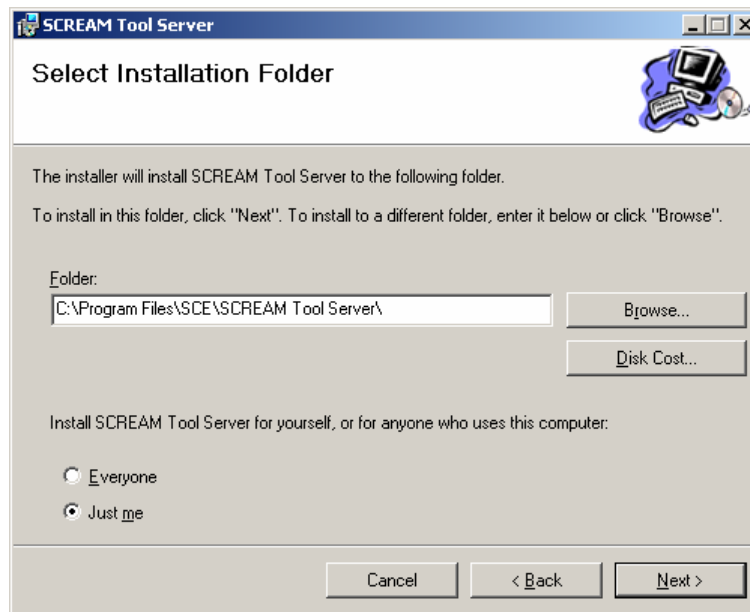
Download the **scream_tool.zip** file then unzip it to a directory (for example, `C:\usr\local`).

Navigate to the unzipped **SCREAM_Server** directory (for example, `C:\usr\local\cell\tool\SCREAM\SCREAM_Server`). Find and run **SCREAM_Server_setup.exe**.

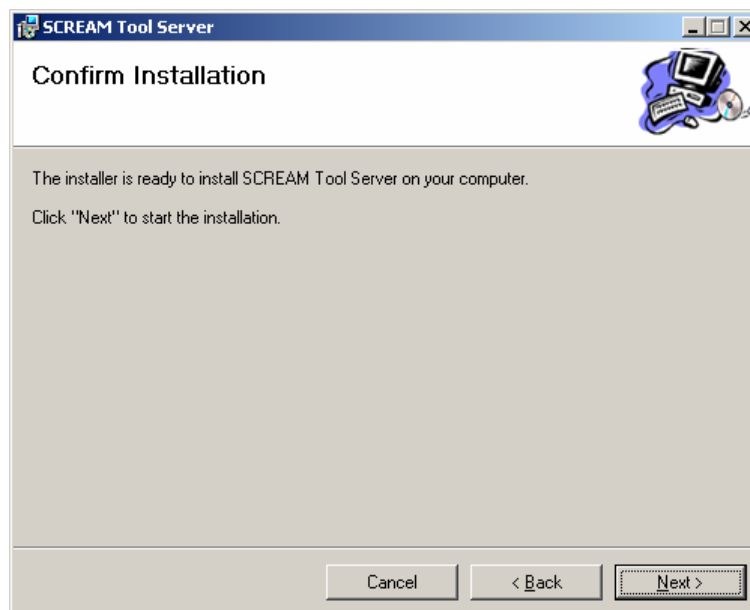
The first window of the SCREAM Tool Server Setup Wizard displays, as shown in Figure 6. Click **Next** to continue.

Figure 6 SCREAM Tool Server Setup Wizard

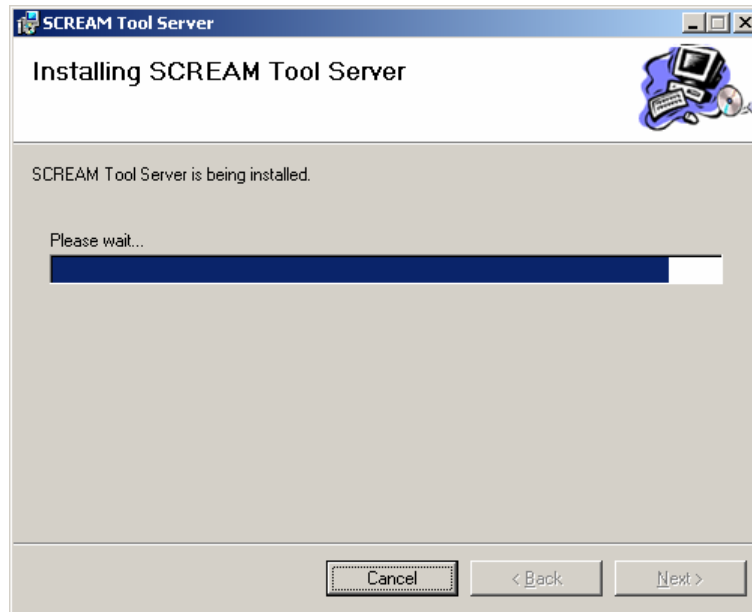
The dialog before the actual installation, shown in Figure 7, prompts you to choose a destination folder to install the server. Unless you specify otherwise, the server will be installed in the location shown. Click **Next**.

Figure 7 Server Installation Location

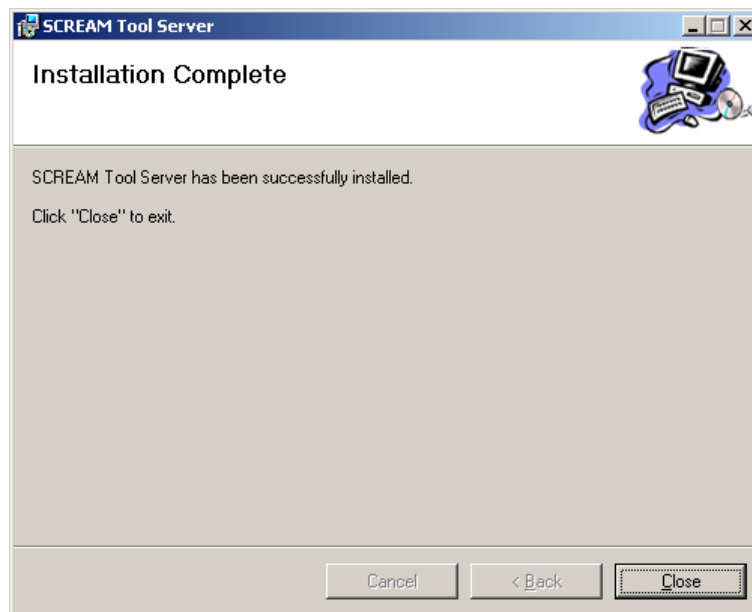
The Setup Wizard displays a confirmation dialog, shown in Figure 8. Click **Next** to continue.

Figure 8 Confirm Installation

The Setup Wizard begins the installation, as shown in Figure 9.

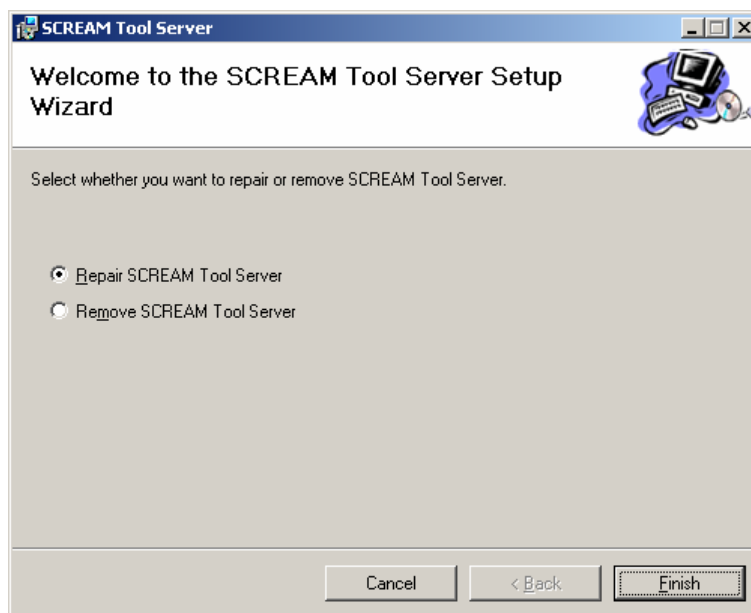
Figure 9 Installation In Progress

The Setup Wizard finishes installing the server and displays a final dialog, shown in Figure 10. Click **Close** to close the window.

Figure 10 Installation Finished

Repairing or Removing SCREAM Tool or the SCREAM Tool Server

If you already have a version of the SCREAM Tool or the SCREAM Tool Server installed, then running the setup file for either of them causes a "Repair or Remove" dialog similar to the following to display:

Figure 11 Repair or Remove

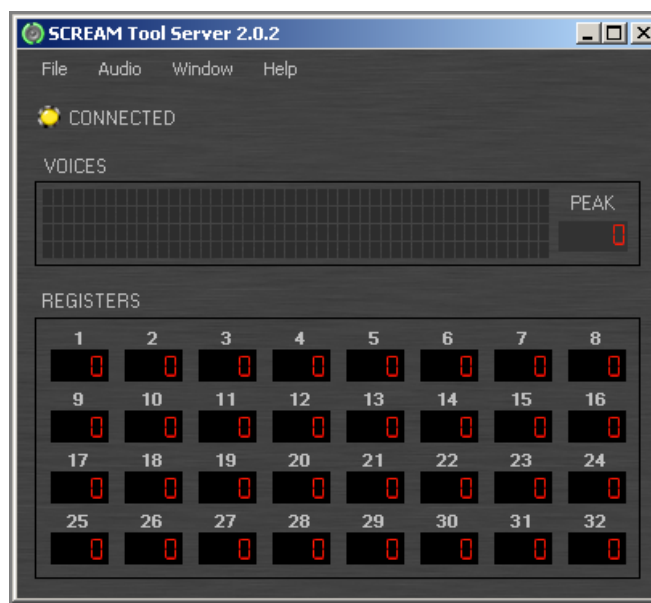
Choose whether you want to repair or remove it, then click **Finish**.

If there is a previous version of SCREAM, prior to version 5.2.36, installed then uninstall that version first by selecting **Start > Settings > Control Panel > Add or Remove Programs** then choosing **SCREAM Tool** and **SCREAM Tool Server**.

Launching the SCREAM Tool and the Server

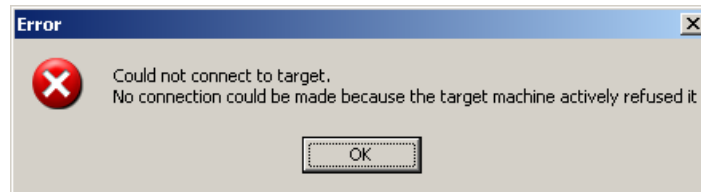
To launch the SCREAM Tool, select **Start > Programs > SCE > SCREAM > SCREAM Tool**.

When you launch the SCREAM Tool, it automatically looks for and launches the SCREAM Tool Server. This is a separate window from the main SCREAM interface and is shown in Figure 12.

Figure 12 Server Window

If you do not see the SCREAM Tool Server window displayed on your screen, or if you receive the following error when trying to connect, then the SCREAM Tool Server is not currently launched and you need to launch it manually.

Figure 13 Server Connection Error



To manually launch the SCREAM Tool Server, go to the Start menu and select:

Start > Programs > SCE > SCREAM > SCREAM Tool Server

SCREAM Tool Server Default Values

If the SCREAM Tool Server does not launch automatically when you start the SCREAM Tool, ensure that:

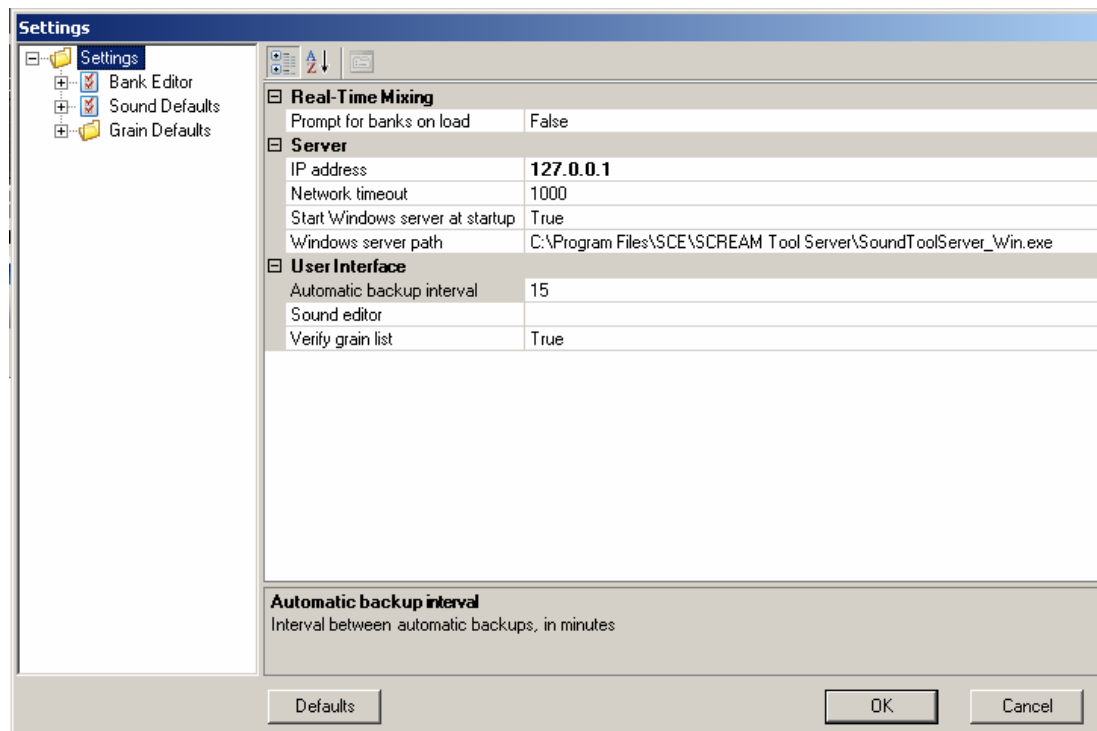
- SCREAM Tool Server was installed properly (as described in the previous section).
- SCREAM Tool is searching for the server in the correct location.

To verify the location of the server, launch the SCREAM Tool and from the main window, go to:

Edit > Preferences

The Settings window displays, as shown in Figure 14.

Figure 14 Preferences Settings



In the Server section on the right side, make sure the following fields contain the proper information:

- Windows Server Path – This field is the location in which the SCREAM Tool looks for and then launches the SCREAM Tool Server. If the server was installed using the default location, the directory path should be:

`C:\Program Files\SCE\SCREAM Tool Server\SoundToolServer_Win.exe`

- IP Address – This field is the default IP address of the Windows server and should be: 127.0.0.1.

Note that it is possible to run the SCREAM Tool Server on a different PC. If this is the case, then specify the IP address of the PC running the server instead.

Connecting to the SCREAM Tool Server

When you try to play a sound, the SCREAM Tool automatically attempts to connect to the server. You can also manually connect to the server from within the SCREAM Tool by choosing **Preview > Connect to Server**.

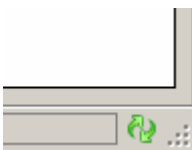
Once connected, you will see a yellow indicator in the top left of the Server Window.

Figure 15 Indicator in Server Window



Another way to check the status of your server connection is using the Connection Status icon, located in the bottom right corner of the main SCREAM Tool window.

Figure 16 Connection Status Icon in SCREAM Tool Window



The icon will display in either of the following ways:

	Not Connected	Indicates no connection to the SCREAM Tool server.
	Connected	Indicates that the SCREAM Tool is connected to the SCREAM Tool server.

The main SCREAM Tool Server window is described in more detail in the following section, "Using the SCREAM Tool Server".

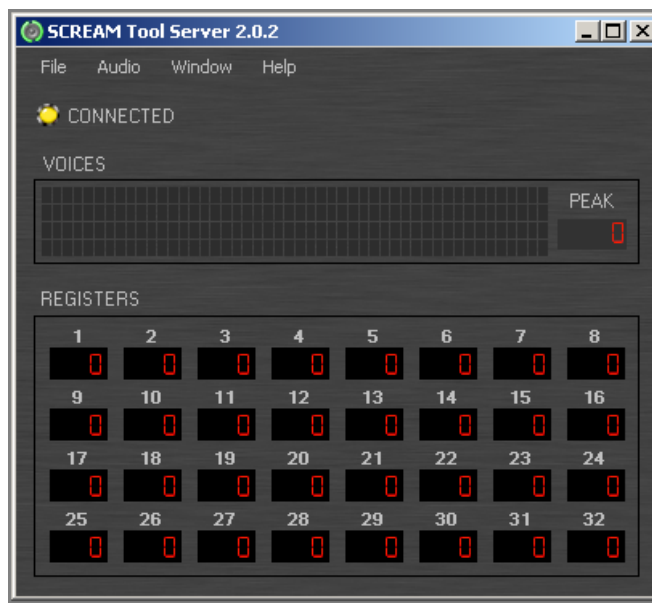
Disconnecting from the Server

You can manually disconnect from the server by going to the SCREAM Tool and choosing **Preview > Disconnect From Server**.

Using the SCREAM Tool Server

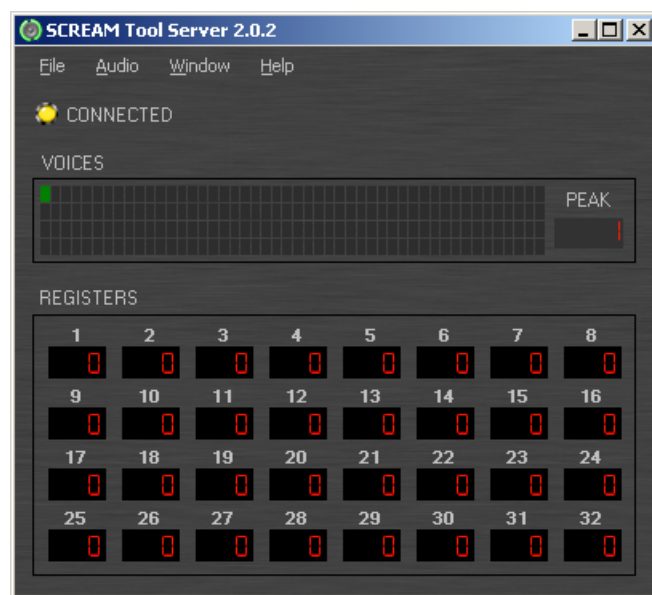
The SCREAM Tool Server is a separate program that runs the SCREAM runtime. When you connect to the server from the SCREAM Tool, it sends a bank (created with the SCREAM Tool) to the server so it can play sounds. This allows you to preview sounds that you create in the SCREAM Tool.

Figure 17 Server Window



SCREAM runtime has 128 voices; each represented by the small black boxes in the **Voices** section. Each waveform grain uses one voice when it is played; the voice is shown as *active* by one of the black boxes turning green.

Figure 18 Indicator in Server Window



If a box is black then it is not playing a sound. When a box turns green, then that voice is now playing one sound. If multiple boxes are green then each of the voices is playing a sound.

If the entire section of **Voices** is filled with green boxes then you are either using all the voices or using more than the maximum number and are close to dropping voices.

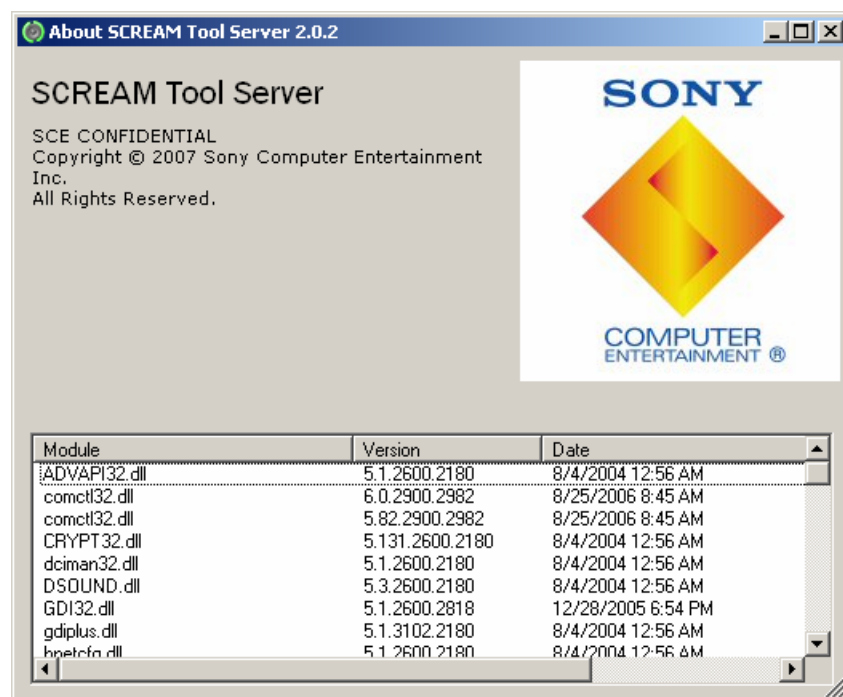
The **Peak** indicator is helpful as a quick snapshot to indicate the maximum number of voices that were recently used. If you see that number display 125, then you know you are close to the limit and you may need to modify your design so you are not playing as many voices or you may need to debug to determine if there is a sound that could potentially exhaust your resources.

The Registers section of the server window contains 32 global registers and is discussed in "Scripting with Registers".

Checking the Version of the Server

You can check the version of the server by choosing **Help > About** from the menus in the server window. A window similar to the following displays.

Figure 19 Server Version



Changing the Audio Output

The Audio menu lets you enable the speaker modes you can set in your game using the function `cellScreamSetPlaybackMode()`. See the "SCREAM Reference" manual for details about this function and these modes.

To enable a speaker mode, choose **Audio** then select one of the following:

- Mono
- Stereo – The default is Stereo.
- Matrixed Surround – Matrixed surround based on Dolby™ Pro Logic™ II technology.
- 5.1 – Discrete 5.1, 6 speaker digital format. This option is grayed out if the sound card does not support it.
- 7.1 – Discrete 7.1, 8 speaker digital format. This option is grayed out if the sound card does not support it.

Note: the settings that are selected within the SCREAM Tool Server are only used to emulate the SCREAM output mode setting on the console. These settings have no effect on the output mode used within the game; the output mode within the game is chosen by the programmer using the SCREAM API at runtime.

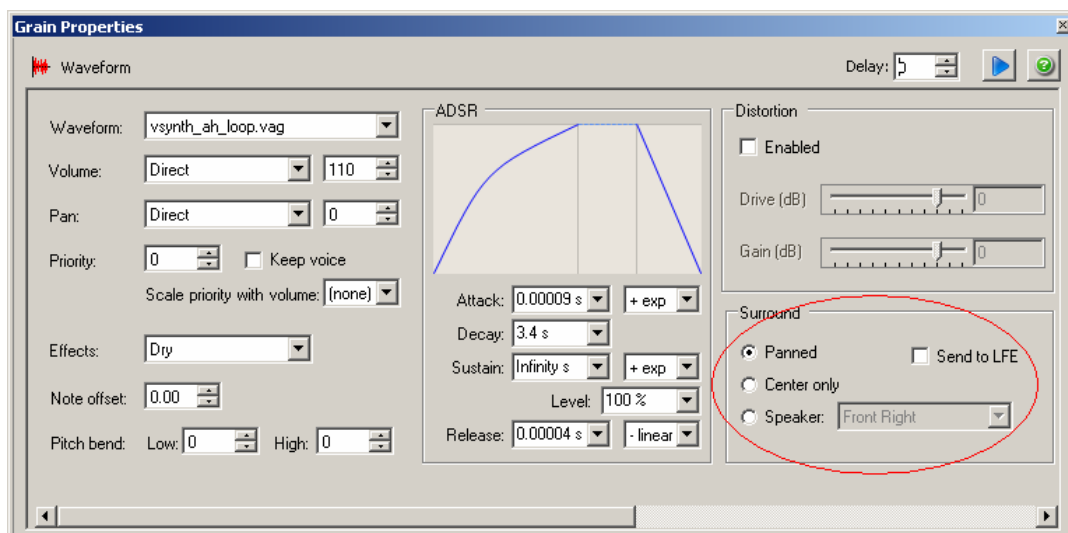
Choosing Pan Table Settings

To choose pan table settings, go to the Audio menu and turn on one of the following:

- **Phantom Center** – Loads a pan table with a phantom center channel. Panned sounds and grains flagged "Center only" will be sent to the front left and right instead of the center channel. 5.1 with phantom center enabled gives you a 5.1 channel layout (3 at the front, 2 at the sides) but will simulate the center channel using the front left and right channels. The "Center only" flag can be enabled or disabled by going to the Surround section of the Grain Properties control of a Waveform, as shown in Figure 20.
- **Pan Excludes Center** – Panned sounds will be rendered with a phantom center channel, but "Center only" grains will still go to the center channel. This is useful, for example, if you want to keep your center channel for dialog or interface sounds only. The "Center only" flag can be enabled or disabled by going to the Surround section of the Grain Properties control of a Waveform, as shown in Figure 20.

Figure 20 shows the "Center only" flag in the Surround section of the Grain Properties control of the SCREAM Tool (see section "Adding Grains to the Sound Script" for more information about the Grain Properties control).

Figure 20 Grain Properties Control



Stopping All Sounds

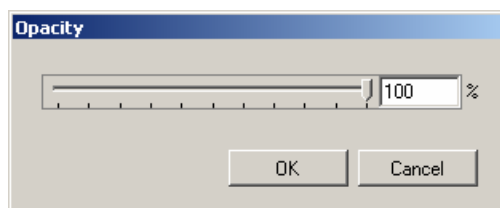
Choose **Audio > Stop All Sounds** to stop all sounds.

Modifying Server Window Position and Opacity

The Window menu offers the following choices for positioning the server window:

- **Always on Top** – Choose this to always display the server window on top of currently open windows.
- **Opacity...** – Use this window, shown in Figure 21, to modify the opacity of the server window. Drag the slider to the setting you want.

Figure 21 shows the Opacity window:

Figure 21 Opacity Window

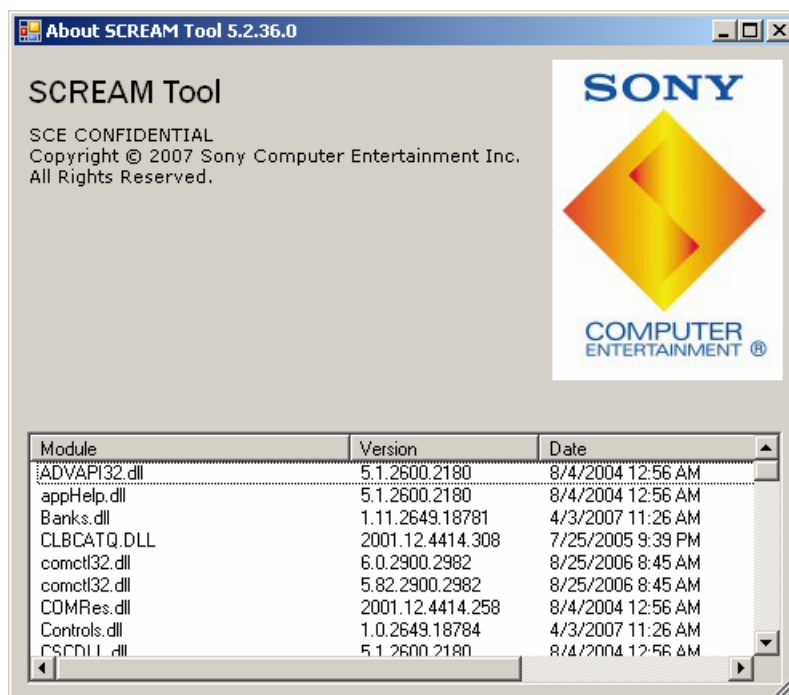
Checking the Version and Exiting the Server

Choose **Help > About** to view the current version of the server.

Choose **File > Exit** or click the X to close the server window. This shuts down the server and stops any sound from being played.

Checking the Version and Exiting the SCREAM Tool

To check the version of the SCREAM Tool, go to the Help menu and select **About**. Figure 22 shows the About Window.

Figure 22 SCREAM Tool About Window

Exit the SCREAM Tool by choosing **File > Exit** or by clicking the X icon in the title bar. The SCREAM Tool server does not automatically close when you shut down the SCREAM Tool; you must shut the SCREAM Tool server down separately.

Close the SCREAM Tool server by clicking the X icon in the title bar.

2 Building a Bank that Plays a Sound / Creating a Bank

Before starting, make sure that the SCREAM Tool and the SCREAM Tool Server are both properly installed and launched.

Key Concepts

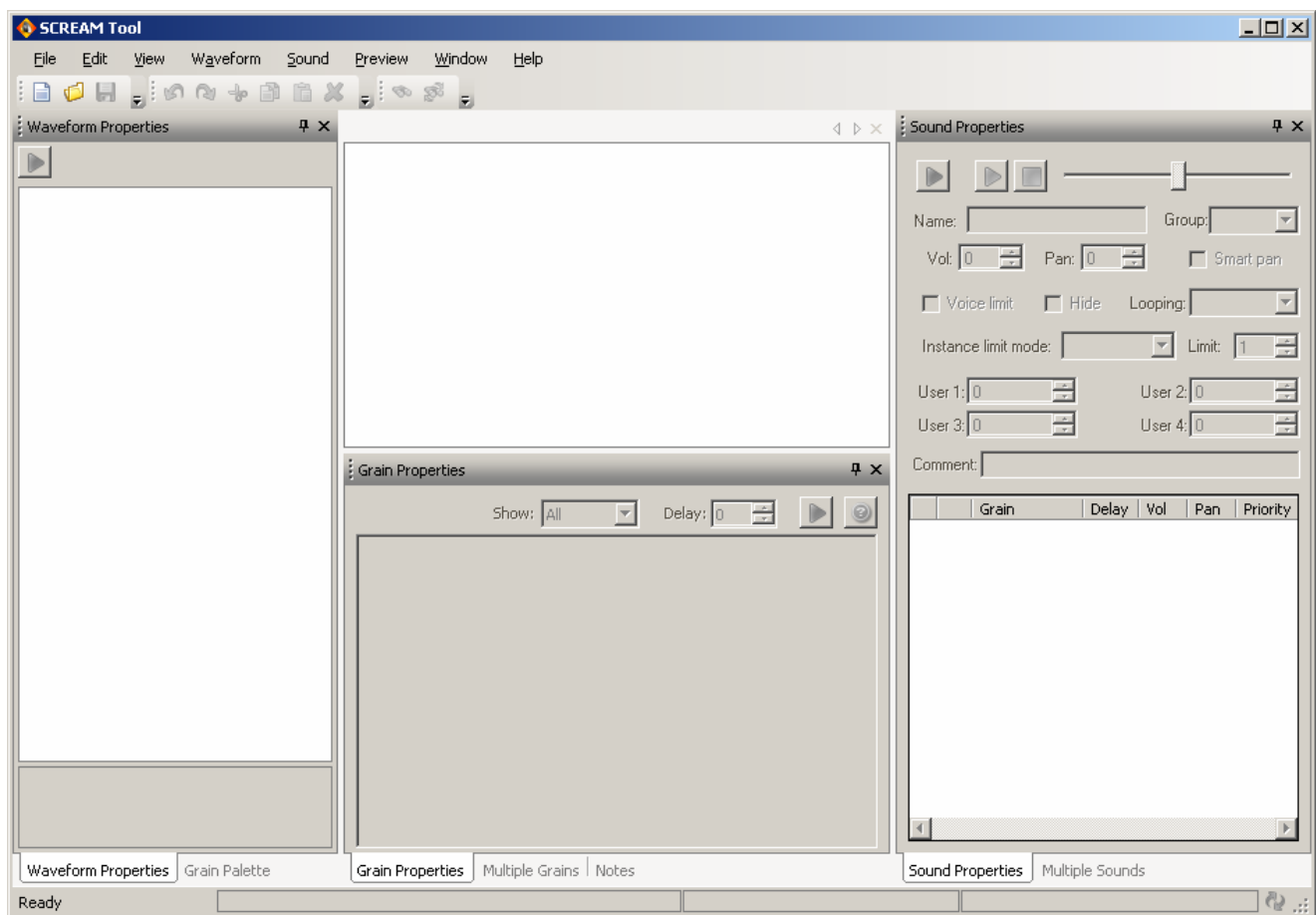
The basic building blocks of SCREAM are:

- **Grain** – A grain is the building block of a complex sound script. Each grain performs one small action. For example, there are grains that can play waveforms, adjust the volume of a sound, or control the execution of other grains.
- **Sound** – A sound is made up of a collection of grains. The game engine typically triggers sounds while the Sound Designer modifies the grains that make those sounds.
- **Waveform** – A waveform is the audio sample data. Waveforms can be played by certain types of grains. The bank file stores the audio data itself, rather than a link to an external audio file. This means that bank files can be copied and moved easily.
- **Bank** – A bank is a collection of sounds, grains, and waveforms.

Interface Overview

Figure 23 shows the main window of the SCREAM Tool:

Figure 23 SCREAM Tool Window



The main window shows (from left to right) the Waveform Properties control, the Grain Properties control, and the Sound Properties control.

Many of the panes in the main window can be adjusted to customize your workspace:

- Click and drag in the title bar of one of these panes to dock it in a new location in the main window, as a tab in another control, or a floating panel.
- Click the pushpin icon to collapse it into a tab on the side of the window.
- Click the **X** icon to remove it from the workspace; it can be shown again by choosing it from the **View > Controls** menu.

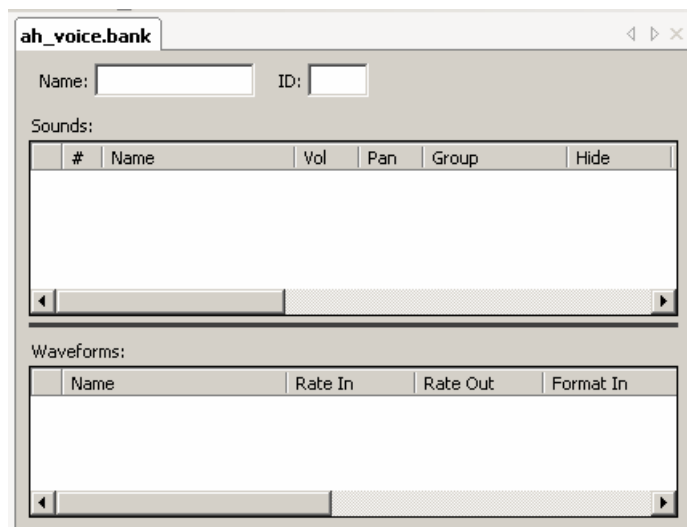
Creating a New Bank

A *bank* contains a collection of Sounds, grains, and waveforms. To create a new bank:

- Click the **Create a New Bank** icon () or select **File > New**.
- Choose **File > Save as**.
- Enter a name then click **Save**.

The name of the bank now appears as the title of the middle tab in the SCREAM window, as shown in Figure 24. In this example, the bank is named **ah_voice.bank**.

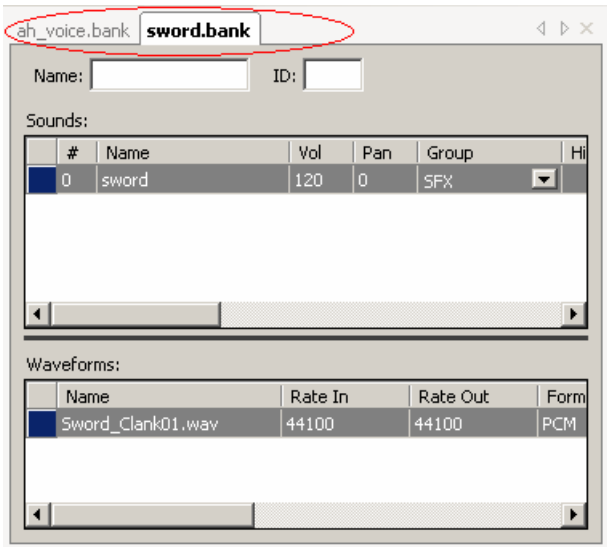
Figure 24 New Bank



Bank Control – Interface Overview

This is the main control that represents your bank. You can have multiple banks open at once; each will show up as a tab on this control.

Figure 25 Multiple Banks



At the top of the control are fields for the bank name and ID. Consult with your game team to develop a naming convention for these values.

The rest of the control is divided between the Sounds list and the waWaveforms list. The dividing line between them can be dragged up or down to change the amount of screen space devoted to each. In both of these lists, some parameters (in black text) can be edited directly in the list itself by double-clicking on the value; other parameters (in gray text) cannot be edited in the list. You can click on a column heading to sort the list by that column; you can also drag a column heading to reorder the columns.

The toolbar offers icons as shortcuts for common tasks, including:

	Action	Corresponding Menu Item
	Cut	Edit > Cut
	Copy	Edit > Copy
	Paste	Edit > Paste
	Delete	Edit > Delete
	Create new bank	File > New
	Open existing bank	File > Open
	Save	File > Save or File > Save as...
	Find	Sound > Find
	Rename	Sound > Rename
	Customize icons	You can customize the toolbars and specify what icons to display by clicking the small down arrow beside each set of tools and turning on/off the icons.

3 Importing a Waveform

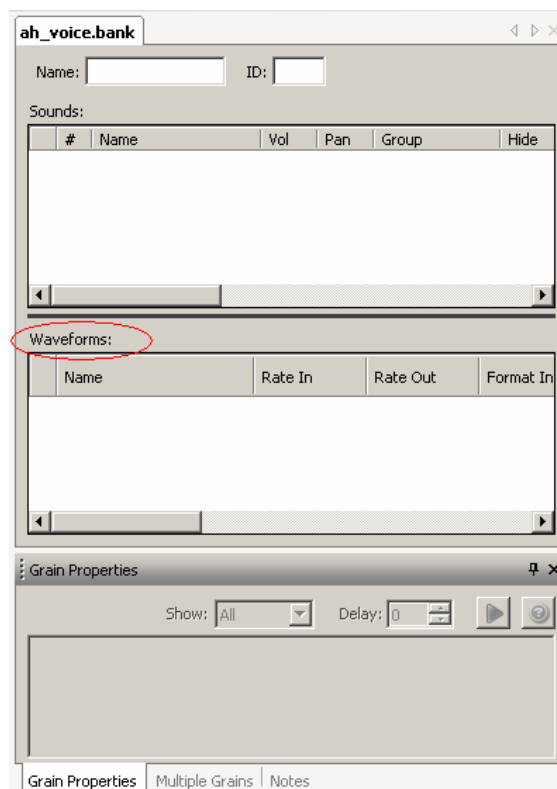
Before importing a waveform, make sure that the SCREAM Tool and the SCREAM Tool Server are both properly installed and launched and you have created or opened a bank.

Importing a Waveform

A Waveform contains audio sample data. The bank file stores the audio data itself, rather than a link to an external audio file. This means that bank files are self-contained and can be copied and moved easily.

The Waveforms Pane, located directly under the Sounds Pane, contains all the waveforms in the bank. It can be expanded by grabbing and dragging the divider line under the Sounds pane.

Figure 26 Waveforms Pane



You can use any of the supported formats which include .wav, .aiff, and .vag. If the file contains multiple channels, each channel will be imported as a separate waveform. For example, a stereo .wav file will be imported as two waveforms; one for the left channel and one for the right channel. To play this back as stereo, you can create two Waveform grains, one panned left that plays the left channel and one panned right that plays the right channel.

To import a waveform, choose any of the following ways:

- Choose **Waveform > Add Waveform...**
- Right-click inside the Waveforms pane then select **Add Waveform...** from the menu that displays.
- Cut and copy then paste a waveform from another bank
- Cut and copy then paste a sound or grain that references a waveform from another bank
- Drag a waveform from another bank's waveform list to this one (hovering the mouse while you drag over the bank's tab to switch banks)
- Drag a file from a Windows Explorer window to the list

Once you get the new waveform into the Waveform Pane, it now displays the information about the file, as shown in Figure 27. Scroll left or right to see a full listing of all the details.

Figure 27 Waveform Pane - Filled

Waveforms:						
	Name	Rate In	Rate Out	Format In	Format Out	Size (P52/P5P)
	vsynth_ah_loop.vag	47985	47985	VAG	VAG	34944

In this example, the newly added waveform is named **vsynth_ah_loop.vag** and is a looping .vag file that sounds like a person saying "ahhhhh."

Editing Waveform Parameters in the Waveform Properties Control

If you select one of the waveforms in the Waveforms pane list, you can edit its parameters in the Waveform Properties Control. The Waveform Properties control, located on the left side of the SCREAM window, is now populated with specific information about the selected waveform, as shown in Figure 28. If the Waveform Properties control is not visible, choose **View > Waveform Properties** or double-click on a waveform in the Waveform Pane.

Figure 28 Waveform Properties Control

Waveform Properties

▶

General

Name

vsynth_ah_loop.vag

References

1

Duration

1.27323127

Input

Format

VAG

Sample rate

47985

Loop

0-61096

Source path

C:\Documents and Settings\WVogt\Desktop

Samples

61096

Output

Resample loops

False

Smart loop

False

Normalization

none

Desired sample rate

47985

Format

VAG

Actual sample rate

47985

Loop

0-61096

Samples

61096

Size

34944

Name

The name of this waveform (need not be the source file name).

Waveform Properties

Grain Palette

The following types of information can be modified; values that are displayed in gray are read-only while those in black can be edited:

General parameters:

- **Name** – Name of this waveform. By default, this is the name of the file from which the waveform was originally read but you can change it to anything you want.
- **References** – Indicates the number of grains that reference this waveform. Waveforms that have zero references will be excluded when exporting a binary .bnk file because they cannot be played by the bank if they are not referenced.
- **Duration** – Duration of the waveform in seconds (ignoring loops).

Input parameters:

- **Format** – Format of the original audio data that was imported (either PCM or VAG).
- **Sample rate** – Sample rate of the original audio data.
- **Loop** – Loop points, if any, in the original audio data, specified in samples.
- **Source path** – The path of the original file from which the audio data was read. This file may not exist on your computer if it has since been deleted or if the bank was created on a different computer.
- **Samples** – Total number of samples in the original audio data. For compressed data, this is the number of uncompressed samples.

Output parameters:

- **Resample Loops** – If set to true, the tool will automatically resample the waveform so that loop points are properly aligned (to 28 samples for VAG data or 16 samples for PCM data). If set to false, improperly aligned loop points will just be moved so they are properly aligned. (this option is disabled for waveforms that do not loop or whose source data is VAG format.)
- **Smart Loop** – If true, the filter coefficients for the first block of the loop are set to 0 when exporting in VAG format. This can prevent clicking at the loops for some waveforms. However, for other waveforms, particularly those whose frequencies are far below the Nyquist frequency, this can actually cause clicking, so you might want to try changing this value if you hear clicks in your loop. (This option is disabled for waveforms that do not loop or whose source data is VAG format.)
- **Normalization** – Normalization to be applied to the waveform, in percent of full scale. For example, if set to 70, the waveform will be scaled so its peak amplitude is 70% of full scale.
 - If set to a negative value or to "none", no normalization will be performed in most cases; however, if a waveform is exported in VAG format, and its maximum amplitude is greater than 95% of full scale, it will be normalized to 95% to prevent clipping due to VAG decompression.
 - If this value is set to a value between 95 and 100, clipping may occur if the waveform is exported in VAG format. This value may be set to a value greater than 100, which will always cause clipping. (This option is disabled for waveforms whose source data is VAG format.)
- **Desired sample rate** – The desired sample rate of the exported audio data. If this is different from the input sample rate, then the waveform will be resampled on export. If the "Resample loop" option is true, the actual sample rate may differ from the desired sample rate so that loop points are properly aligned. (This option is disabled for waveforms whose source data is VAG format.)
- **Format** – The format in which the audio data will be exported in the bank. If the input format is VAG, then VAG will be the only option available; otherwise, you may choose either PCM or VAG.
- **Actual sample rate** – The actual sample rate of the exported audio data. If the "Resample loop" option is true, the actual sample rate may differ from the desired sample rate so that loop points are properly aligned.
- **Loop** – The loop points, if any, in the exported audio data, specified in samples.
- **Samples** – Number of samples in the exported audio data.
- **Size** – The number of bytes that this waveform will use in the exported bank.

Editing the Original Source File for a Waveform

You can easily launch a sample editor to edit the original source file for the waveform by choosing **Waveform > Edit**. If you have not specified a sample editing program, you will be prompted to do so. The sample editor can be changed at any time by choosing **Edit > Preferences**.

Since a Waveform's audio data is stored in the bank itself, if you do edit a sample file, you will need to re-import the file for the changes to have any effect in the bank. See "Refreshing Waveform" for information.

Refreshing Waveform

Since a waveform's audio data is stored in the bank itself, if you edit a sample file you will need to re-import the file, called "refreshing" the file, for the changes to have any effect in the bank. You can refresh the file in any of the following ways:

- **Waveform > Refresh** – Waveform is re-loaded from the path stored as the "Source Path" for the waveform. If that file cannot be read, you are prompted to find a different file.
- **Waveform > Refresh From** – Reloads from a file that is different from the source file.
- **Waveform > Refresh All** – Reloads all files.
- **Refresh From Directory** – Reloads one or more waveforms from files with the same name as the original files but in a different directory when you select the waveform and choose this menu option.

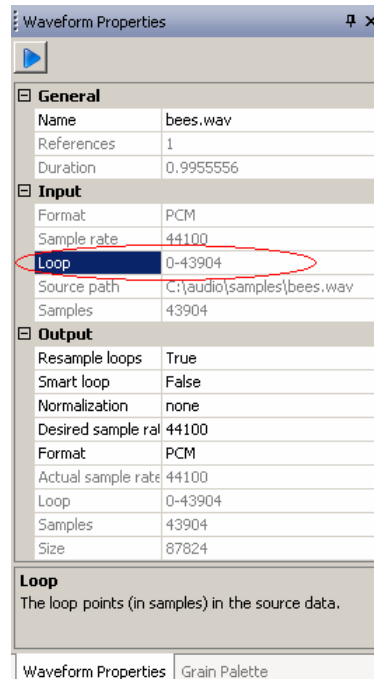
Looping Sound Using a Looping Waveform

A looping waveform contains embedded instructions that send it back to the beginning of the file when it reaches the end of the file, creating a continuous loop.

To import a looping waveform:

- Create a new sound by choosing **Sound > Add Sound** then name it (see "Creating a Sound" for detailed instructions).
- Add a waveform by choosing **Waveform > Add Waveform...** and selecting the looping waveform file.
- Drag the waveform file from the Waveform list over to the Grain list in the Sound Properties control. This adds a Waveform Grain that references that specific looping waveform.

Figure 29 shows the Waveform Properties of a looping waveform; the Loop field in the Input section shows the loop points in the file.

Figure 29 Looping Waveform

The screenshot shows the 'Waveform Properties' dialog box with the 'Input' section expanded. The 'Loop' field is highlighted with a red oval, indicating its importance for looping waveforms. The 'Output' section also shows the 'Loop' field with the same value. The 'Loop' field is currently set to '0-43904'.

General	
Name	bees.wav
References	1
Duration	0.9955556

Input	
Format	PCM
Sample rate	44100
Loop	0-43904
Source path	C:\audio\samples\bees.wav
Samples	43904

Output	
Resample loops	True
Smart loop	False
Normalization	none
Desired sample rate	44100
Format	PCM
Actual sample rate	44100
Loop	0-43904
Samples	43904
Size	87824

Loop
The loop points (in samples) in the source data.

Waveform Properties Grain Palette

If this waveform was not a looping waveform then this field would read "None."

4 Creating a Sound

Before creating a sound, make sure that the SCREAM Tool and the SCREAM Tool Server are both properly installed and launched, and you have created or opened a bank, and imported a waveform.

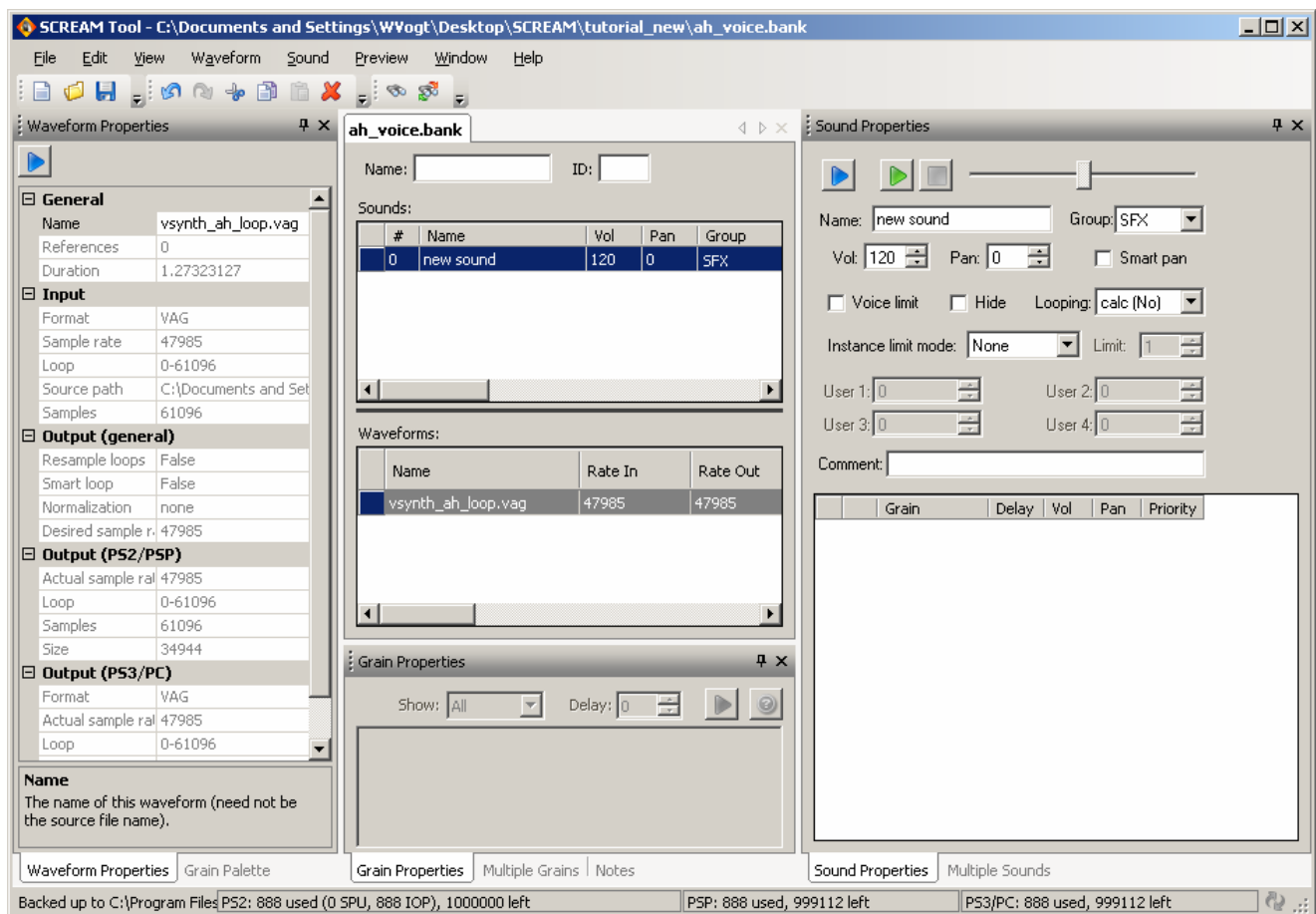
Creating a Sound

A Sound is made up of a collection of grains. During game play, the game engine typically triggers a Sound, which in turn executes a set of grains that have been arranged by the Sound Designer. Grains are the smallest part within SCREAM and represent script functions such as playback of samples, setting of parameters, and logic. The next chapter, "Adding Grains to the Sound Script", describes how to add a grain to a sound script.

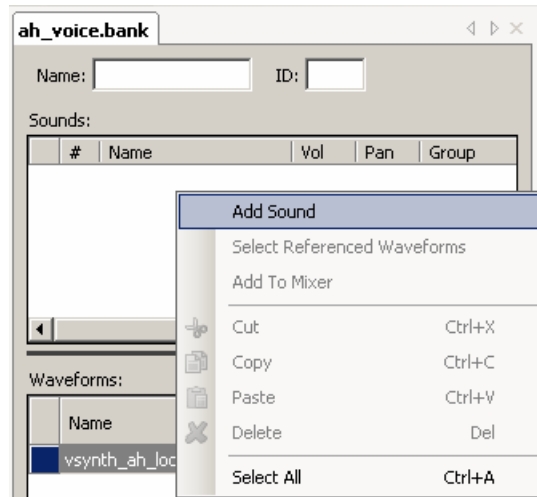
Create a new Sound using any of the following methods:

- Choose **Sound > Add Sound**. By default, this new Sound is automatically named "new sound" as shown in Figure 30.

Figure 30 Sounds Pane – New Sound



- Inside the Sound Properties pane, located directly below the Bank tab, right-click and select **Add Sound** from the menu. Figure 31 shows this menu. By default, this new Sound is automatically named "new sound".

Figure 31 Add Sound Menu

- Cut, copy, and paste a sound from another bank. By default, the new Sound's name is the name of the pasted sound.
- Drag a sound from another bank's sound list to this one (hovering the mouse while you drag over the bank's tab to switch banks). By default, the new Sound's name is the name of the sound you just dragged in from the other bank.
- Drag a waveform from the waveform list to the sound list to create a sound with one waveform grain. By default, the new Sound's name is the name of the waveform dragged from the waveform list.

Rename the Sound by double-clicking the Name field in the Sounds pane or by going to the Sound Properties pane, located on the right side of the SCREAM window, then entering the desired name. In this example, the Sound is named **ah_voice**.

To view all the information about a Sound, scroll left or right in the Sounds pane, located in the middle of the SCREAM window. The Sounds pane contains all the sounds in the bank. If you select one sound in the list, you can edit its parameters in the Sound Properties Control. If the sounds are sorted by index, you can also reorder the list by selecting one or more lines and dragging them to a new position. You can preview a sound on a connected test server by holding down the Control key and clicking on the sound with the right mouse button.

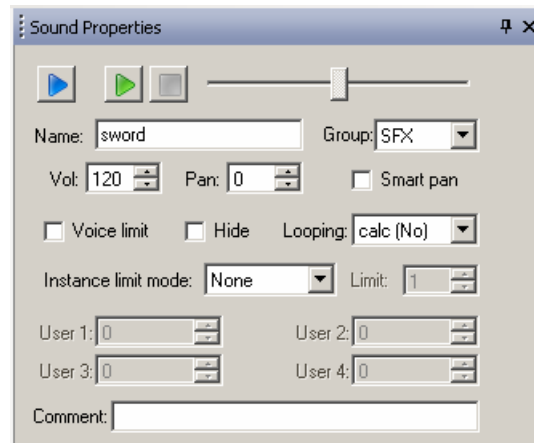
Figure 32 Sounds Pane

Sounds:													
#	Name	Vol	Pan	Group	Hide	Voice Limit	Smart Pan	Looping	User 1	User 2	User 3	User 4	Comment
0	ah_voice	120	0	SFX	☐	☐	☐	Yes	0	0	0	0	

Accessing the Sound Properties Control

The Sound Properties Control, as shown in Figure 33, is located in the right side of the SCREAM window and lets you edit the sound that is currently selected in the Bank Control. If Sound Properties control is not visible, you can display it by:

- Choosing **View > Sound Properties**
- Double-clicking on a sound in the Sounds pane.

Figure 33 Sound Properties Control

Specifying Information About a Sound

Both the Sounds pane and the Sound Properties control allow you to modify the following types of information (if you change it in one, it automatically updates to the other):

- **Name:** The name of the sound. If you are including names in the bank when you export it (see "Importing and Exporting" for information about exporting), the game engine can use this name to call this sound. Even if you are not exporting names, you can name sounds to help you keep track of them. See "Document Preferences" in the SCREAM help files for an explanation of the effect of exporting names.
- **Group:** You can assign a sound to a specific "volume group". This allows the volumes of all the sounds in the group can be scaled easily by the game engine. For information about a related SCREAM function, see `cellScreamSetMasterVolume()` in the "SCREAM Reference" manual.
- **Volume:** Master volume of the sound, from 0 to 127 inclusive. Any volumes of individual grains in this sound will be scaled by this volume value.
- **Pan:** The pan value, in degrees. A sound at a pan of 0 degrees plays at center; a sound at 90 degrees plays at hard right, and so on. The pan value may be negative. Note that individual grains can have their own pan values and the game engine may supply a pan value; all of these pans are added together to create the final pan value of the sound.
- **Smart Pan:** If you enable this option, then pans for all grains in this sound are scaled toward 0 as the volume of the sound is reduced. This is useful if you have a stereo sound that is attached to some localized area of the game.

For example, let's say you have a waterfall in your game. You create a pair of grains, one panned left and one panned right, to make a wide stereo sound that surrounds the listener. This works fine when the listener is standing next to the waterfall, but as the listener moves away from the waterfall, the sounds remain panned left and right, so the sound is not directional. In this case, if you enable **Smart Pan**, then as the listener moves away from the waterfall and the game engine decreases the volume of the sound, the stereo waterfall sound collapses to mono so it can provide a more directional audio cue.

- **Voice Limit:** If enabled, any playing instances of the sound are keyed off when a new instance is started. This is useful for sounds that are quickly repeated many times in succession. Note that the playing instances are keyed off, not immediately silenced, so the total number of voices that will play at once depends on the release time of the sound's waveform grains. Also note that if the sound is triggered as a child of another sound, the voice limiting will not take effect. This option cannot be enabled at the same time as Instance Limit (see below).

- **Hide:** A "hidden" sound is not exported in C header files or text files so if your game's programmers are using these files to find out which sounds are in a bank, they will not know about the sound if this is checked. This is useful for sounds which you only want to be triggered as child sounds (see the "Start Child Sound" grain and the "Branch To Child Sound" grain descriptions in the SCREAM help files for details).
- **Looping:** The sound system has a function that the game engine can call to find out if a sound loops. The menu provides three options: Yes, No, and calc (no).

If you choose **calc (No)** from the drop-down list, the SCREAM Tool will try to figure out whether the sound loops or not. If the "calc" item reads "**calc (Yes)**" or "**calc (No)**", then the tool has determined that the sound loops or does not loop.

There are some cases where it cannot determine for certain; in these cases, the "calc" item will read "calc (?No?)". If your game engine needs to be able to determine whether this sound loops, you should override the calculation by choosing "Yes" or "No" from the drop-down list.

Note that if your game engine never queries the sound system about the looping state of a sound, then you need not worry about this setting. Also, keep in mind that this setting has no effect on whether your sound actually loops or not; it only affects the information available to the game engine.

- **Instance limit mode:** Choose from None, reject new, reject oldest, or reject softest. This is similar to the **Voice Limit** option (see above), but more flexible. If enabled, the sound system will only allow a certain number of instances of the sound to play at once.

When the instance limit is reached, and the game engine tries to start another instance, the resulting behavior depends on the setting in the drop-down list:

- **Reject new:** the sound system will prevent the new instance from playing.
- **Stop oldest:** the sound system will stop the oldest of the currently playing instances.
- **Stop softest:** the sound system will stop the quietest of the currently playing instances, if that instance is quieter than the new instance; otherwise it will prevent the new instance from playing.

In both of the last two cases, the sounds are stopped by being "keyed off", so the actual voice usage will depend on the ADSR envelope.

Note that if the sound is triggered as a child of another sound, the voice limiting will not take effect.

This option cannot be enabled at the same time as Voice Limit.

- **User Data:** If you have enabled user data in the Document Preferences dialog, you can enter four (4) integer values that will be associated with this sound User 1, User 2, User 3, User 4. For more information, see "Modifying Bank Preferences (Data Stored and Memory Usage)".

Setting Sound Defaults

You can set the default values for new sounds by choosing **Edit > Preferences** and selecting "Sound Defaults."

Determining What Waveforms a Sound References

To find out which sounds reference which waveforms:

- Select one or more sounds in the Sounds Pane and the waveforms that are referenced by those sounds are displayed in bold in the Waveform Pane, OR
- Choose **Sound > Select Referenced Waveforms** to display the referenced waveforms in the list.

You can also select one or more waveforms and choose **Waveform > Select Referencing Sounds**. This causes all sounds that reference the selected waveforms to be selected in the Sounds Pane.

5 Adding Grains to the Sound Script

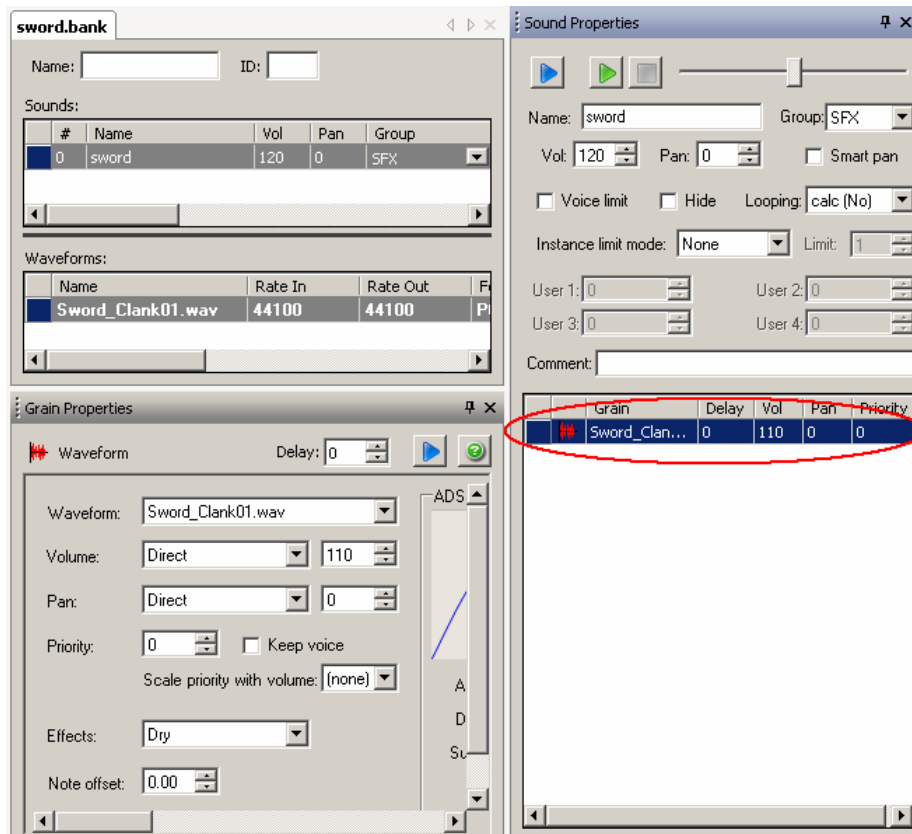
A Grain is the building block of a complex Sound script. Each grain performs one small action. For example, there are grains that can play waveforms, adjust the volume of a Sound, or control the execution of other grains.

After you create a Sound, you can then create a grain in the Sound script. There are many ways to add a grain to the Grain List, including:

- Choose **Sound > Add Grain** then select the grain type from the sub-menu.
- Cut/copy and paste a grain from another sound, in this bank or another bank
- Drag a waveform from the waveform list in the Bank Control to the grain list to add a Waveform Grain that references that waveform
- Drag one or more grains from the grain list to a sound in the Sound list
- Drag one or more grains from the grain list, drag over a sound in the Sound list to select the sound and drop the grains in the grain list
- Drag one or more grains from the Grain Palette to the grain list. See "Using the Grain Palette" for information.

For example, if you wanted to add a Waveform Grain that referenced a waveform called "Sword_Clank01", right-click in the empty Grain List (located at the bottom of the Sound Properties pane) then choose **Add Grain > Waveform > Sword_Clank01.wav** from the pop-up menu. Figure 34 shows the result. Notice that the file remains automatically selected after it is added and the new grain's properties are now displayed in the Grain Properties pane, located in the bottom middle of the window, as shown in Figure 34.

Figure 34 Bank with a Sound and Waveform Grain



Using the Grain Palette

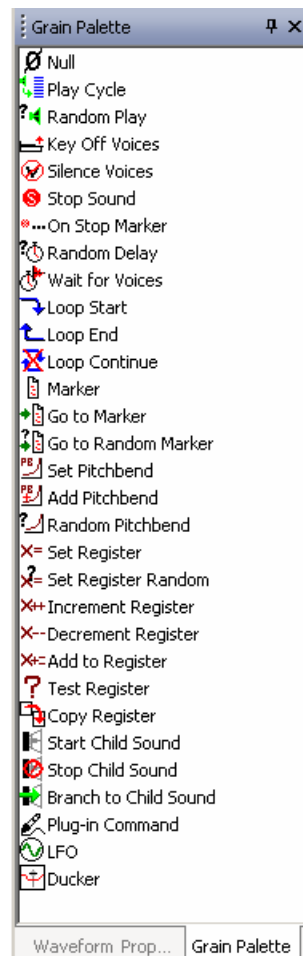
The Grain Palette provides a list of all available grain types. Make sure the Grain Palette is visible by doing one of the following:

- Choose **View > Grain Palette**.
- Click the Grain Palette tab along the left side at the bottom of the main SCREAM window.

Each of the following grains is discussed in detail in the SCREAM help. See the Grain Reference in the SCREAM help system for detailed information.

The Grain Palette looks like this:

Figure 35 Grain Palette



Adding Grain(s) from the Grain Palette

To add a grain to a sound:

- Select the sound
- Drag one or more grains from the Grain Palette to the sound's grain list OR
- Copy and paste a grain from the Grain Palette to the sound's grain list.

Once you add a grain to the sound's grain list, you can double-click on it to display the Grain Properties control. Figure 40 shows the Grain Properties control for a waveform.

Setting Grain Defaults

You can set the default values for new grains by choosing **Edit > Preferences** and selecting the grain type in the "Grain Defaults" group.

Using the Grain List

If you select one grain from the list, you can edit the grain's properties in the Grain Properties control (see "Using the Grain Properties Control" for details). The delay, volume, pan, and priority values can also be edited directly in the grain list by double-clicking on the value. To re-order the list, select one or more grains and drag them to their new position in the list. Columns in the list can be re-ordered by dragging the column header to a new position. You can preview a grain on a connected test server by holding down the Control key and clicking on the grain with the right mouse button.

Grains in the grain list are checked for simple errors (for example, having a Loop Start grain without a corresponding Loop End grain). If an error is detected, all grains beginning with the first incorrect grain are highlighted in red in the list; hovering the mouse over one of the red grains will display some text below the list in red, that explains the reason for the error.

More About Grain Validation and Errors

Grain errors are used to highlight unexpected grain configurations. Some conditions may be flagged as errors are even though they harmless, in that they will not cause the game to crash. Such configurations are assumed not to be what the user intended and although their behavior is undefined, it should be consistent. It is recommended that users try to correct grain errors to ensure the desired behavior.

Advanced users may want to build sounds using unusual grain constructs that are flagged as errors in order to achieve specific behavior. Such constructs are legal and can be used within game so in such cases, you may want to turn off grain validation:.

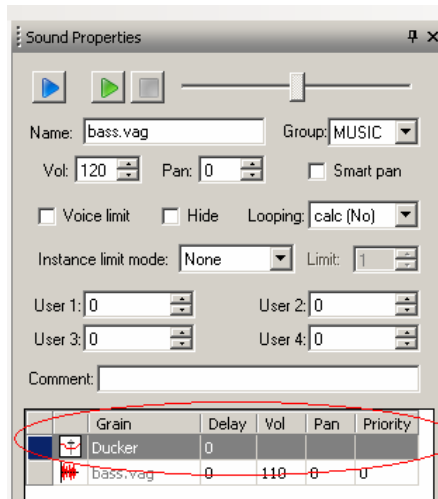
- Choose **Edit > Preferences** and go to "Bank Editor".
- In the "User Interface" section, set "Verify grain list" to **False**.

However, as this is a global setting, it is recommend that grain errors remain highlighted in case of situations where unusual constructs were *not* intended. The "Verify grain list" option only controls whether unusual constructs are displayed, it does not allow or prevent their use within game.

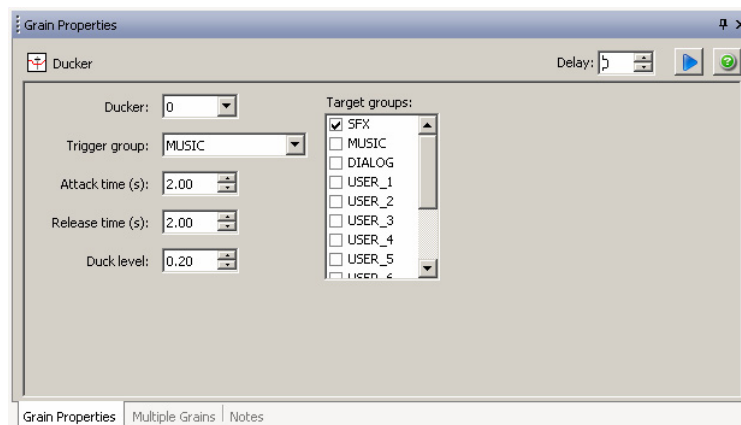
Using the Grain Properties Control

The Grain Properties Control allows you to edit parameters on the currently selected grain in the Sound Properties Control. Each grain will have different parameters displayed in the Grain Properties control.

If, for example, you chose a Ducker grain from the Grain Palette and dragged it to the grain list, then it would show up similar to this:

Figure 36 Ducker Grain Example

To display the Grain Properties Control for the Ducker grain, double-click on the grain or choose **View > Grain Properties** and the following Grain Properties dialog for the Ducker grain displays:

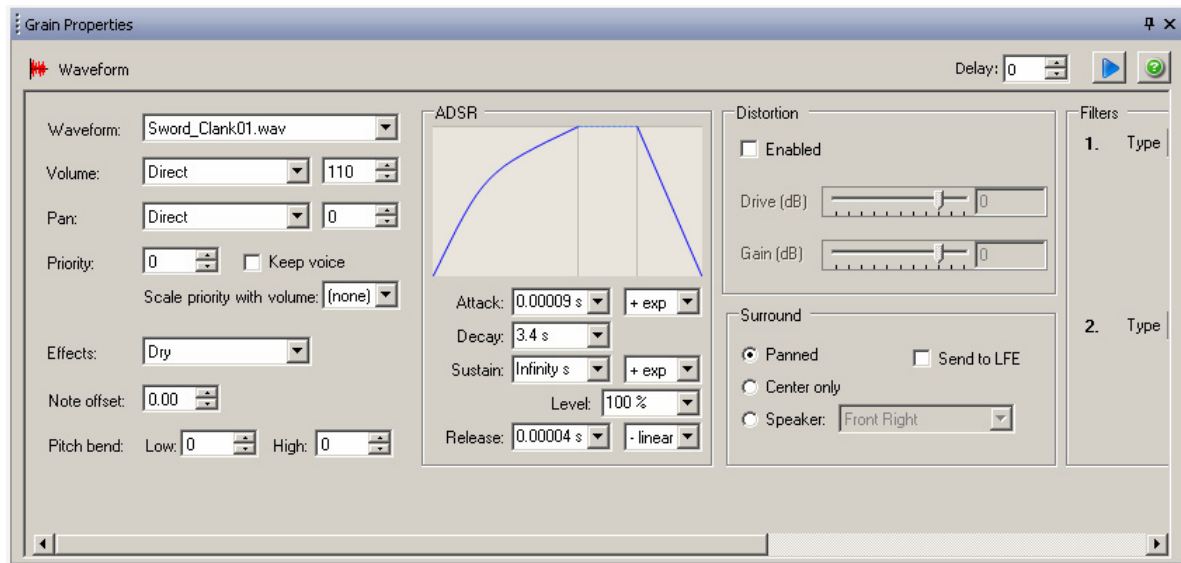
Figure 37 Grain Properties for a Ducker Grain

The information on this dialog, below the Delay field, is specific to the Ducker grain; each grain will have its own unique set of parameters or fields. The delay value indicates the offset of the selected grain from the previous grain, in sound system ticks (there are 240 ticks per second). For example, if there are three (3) grains, and the second and third grains have delays of 120 then the first grain will be played when the sound is triggered, the second will be played 120 ticks after that and the third will be played 120 ticks after that.

For details regarding the grain properties for the Ducker Grain, see the "Grain Reference" section of the SCREAM Help system.

Grain Properties Control for the Waveform Grain

One of the more commonly used grain types, the Waveform Grain, is also one of the more complicated. The following example shows the Grain Properties for a Waveform Grain and discusses the dialog in detail.

Figure 38 Grain Properties for a Waveform Grain

A waveform grain plays a single waveform from the bank. Note that each waveform is mono; to play a stereo sound you must add each channel as a separate waveform and play them panned left and right. The following information is available in the Grain Properties control for a waveform grain:

Waveform: Select the waveform to be played by this grain.

Volume: Set the volume (0-127). Possible volume modes are:

- **Direct:** the volume is set to the specified value.
- **Random:** the volume is set to a random value from 0-127.
- **Register 0-3:** the volume is set to the value in the specified sound register. See "Scripting with Registers" for information about sound registers.
- **Global Register 1-32:** the volume is set to the value in the specified global register. See "Scripting with Registers" for information about global registers.

Pan: Set the pan (in degrees). A grain at a pan of 0 degrees plays at center; a grain at 90 degrees is played hard right, etc. The pan value may be negative. Note that the final pan at which a sound is played is the sum of the grain's pan, the sound's pan, and any pan value specified by the programmer when the sound is triggered. The pan modes are analogous to the volume modes.

Priority: Sets the priority of this waveform in allocating voices. If the value is between 0 and 125 inclusive, the allocator can steal this grain's voice if there are no free voices and the priority value is less than or equal to the requested voice's priority. In other words, voices for grain with low priorities can be stolen and given to grains with higher priorities. If the value is 126, the voice will never be stolen, even by grains with priority 126. In other words, voices with priority 126, once playing, will never be stolen.

Keep voice: If this is checked, the voice will not be released, even when the sound stops playing. This is useful to ensure that not only will this sound's voice not be stolen while it is playing, but it will also always have a voice available when it is played again.

Scale priority with volume: The priority of the grain can be reduced by the chosen percentage from the value set in the "Priority" field as the grain's volume is decreased to 0.

Effects: Sets the routing for hardware effects. Possible effects settings are:

- Dry (no effects)
- Wet (effects)
- Wet & Dry (mix of sound with and without effects)

Note Offset: Adjust the pitch of the grain, in units of semitones. For example, 12 indicates an octave up; -1.5 indicates one and a half semitones down.

Pitch Bend: Sets the range for pitch bend adjustment. Note that both values should be positive.

ADSR: Set the volume envelope for this grain. As you adjust the parameters (attack, decay, sustain, and release), the volume envelope is represented in the graph. Note that the exact shape of the volume envelope will depend upon how long the note is sustained, so the graph will not exactly represent the volume envelope in all circumstances. Also, the horizontal dimension of the graph is scaled to fit the display space, so it is possible for ADSR values with very different time scales to look approximately the same in the graph; pay attention to the actual values in the drop-down lists to avoid confusion. (The rate values in the lists are the amount of time required for the volume to change between zero and full volume.)

Filters: Adjust parameters for each of the four filters that are applied to the sound. You can also view a more detailed graphical view of a filter's response curve by clicking the "Plot" button.

Distortion Adjust parameters for the distortion effect that can be applied to the sound.

Surround: Adjust parameters for surround sound.

- If "Panned" is selected, the sound is positioned among the surround sound speakers according to the sound's pan setting.
- If "Center only" is selected, the sound is played through the center speaker only.
- If "Speaker" is selected, the sound is played through the selected speaker, regardless of the pan value. (Setting "Center only" differs from setting the sound to the center speaker in that the latter will respond to pan changes through the SCREAM API.)
- If "Send to LFE" is checked, the sound is sent to the LFE (low-frequency element) and whatever fullrange channels it might be playing through.

Playing the Sound

You can now preview the Sound you've created.

In the Sound Properties pane, use the following buttons to play a Sound:

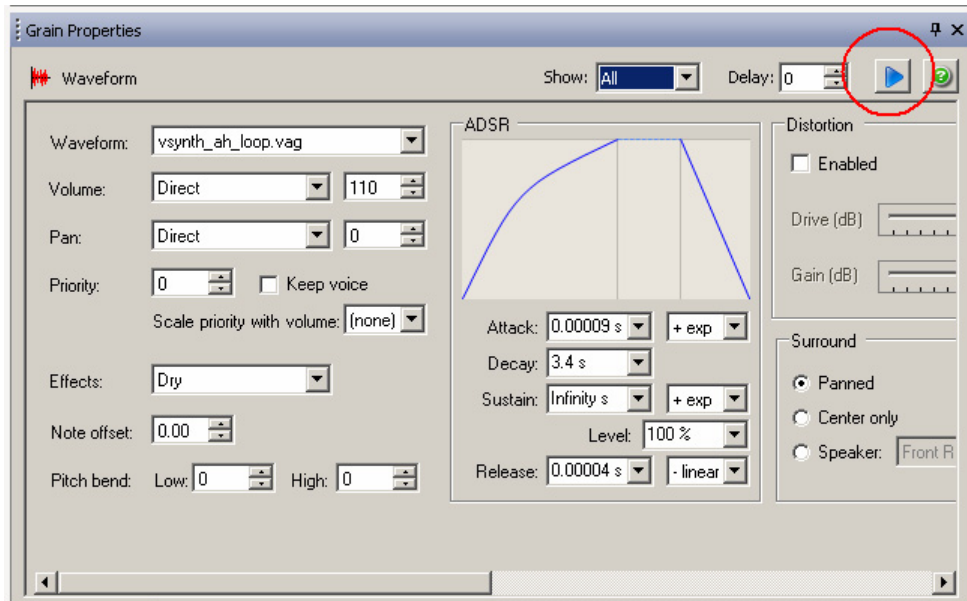
	Play / Stop When Released	Starts playing the Sound and stops as soon as you let go of the button. Use this button for testing release settings and envelope settings.
	Play Continuously	Plays the Sound all the way through. If the selected sound is a loop, you can start it and leave it running by clicking this button; the sound will continue to play until you stop it.
	Stop	Stops the Sound that is playing as a result of pressing the green Play button. You can also choose Preview->Stop All Sounds .

As soon as you try to play a Sound, the SCREAM Tool automatically tries to connect with the SCREAM Tool Server and load the sound bank. The server connection icon in the bottom right corner of the window may display gray (not connected) at first but as soon as you try to play a Sound, it will automatically connect to the server and turn green when successful. If it remains gray and no Sound plays then the connection is broken and you may need to re-launch the SCREAM Tool server.

If you were to play the **ah_voice** Sound used in this tutorial, the **Green Play** button would play the **vsynth_ah_loop.vag** waveform in a loop until you stopped the Sound using the **Stop** button.

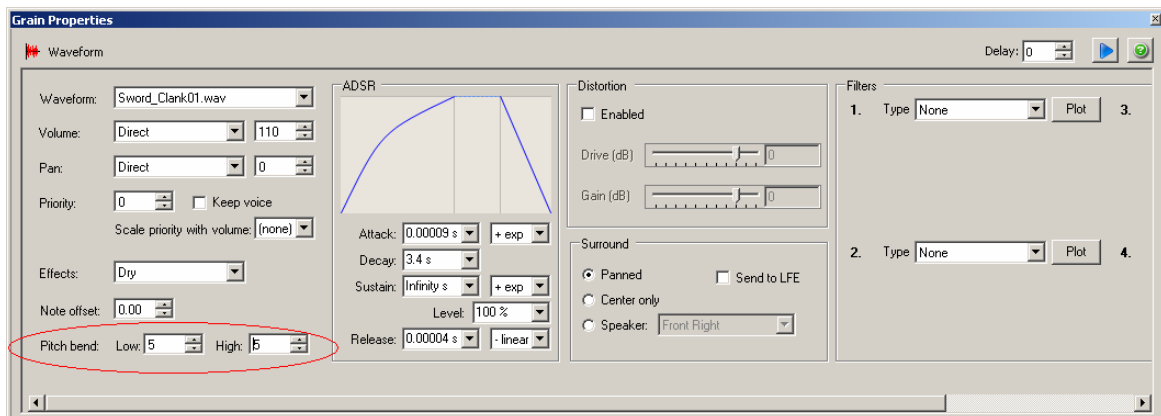
Previewing a Grain

To preview one of the grains within a Sound, highlight the desired grain then preview it by pressing and holding the  button in the Grain Properties pane:

Figure 39 Preview Sounds Using the Grain Properties**Example: Adding Random Pitch Bend Using a Waveform Grain**

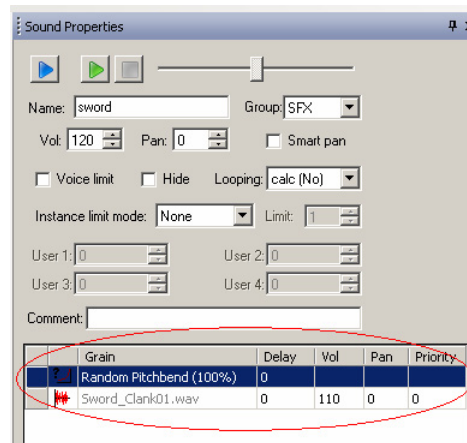
This example shows how to modify a simple sound effect by adding some random pitch bend to the sound, so it will sound slightly different every time it is played.

Double-click a waveform grain in the Grain list to display the Grain Properties control. The example in Figure 40 shows the Grain Properties for the waveform grain named **Sword_Clank01.wav**.

Figure 40 Grain Properties of a Waveform

To add some variation to the sound, set the Pitch Bend Low and the Pitch Bend High values to 5; this sets the pitch bend range of this grain to 5 half steps up or down.

Right-click in the Grain list and choose **Add Grain > Random Pitchbend**. Drag the new grain to reorder the list so it comes before the Waveform grain in the grain list, as shown:

Figure 41 Grain List Example

Click on  in the Sound Properties control to preview your sound several times. You will notice that each time, it plays at a random pitch within the range specified. In this example, the randomization is anywhere within the ± 5 semitone range, selected values are not restricted to semitone intervals.

More Complex Examples

You can see a more complex example using other grains from the Grain Palette in "Scripting with Registers".

6 Scripting with Registers

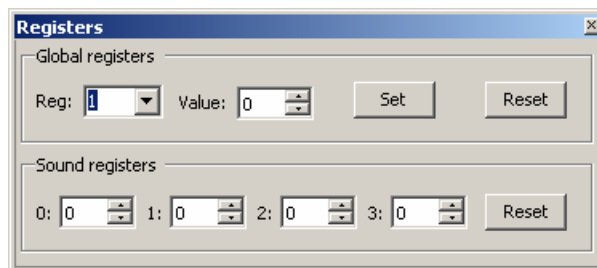
Registers are variables that can be used for implementing complex logic in sounds and allowing communication between the game engine and SCREAM runtime. For example, grains can alter or test register values, and some parameters such as volume and pan can be set from register values. There are two kinds of registers:

- Global Registers – Use global registers when you want to share register values among sound instances.
- Sound Registers – Use sound registers when the register value is relevant only to the internal logic of the sound's grains, or there will be multiple sound instances running and you don't want the register operations of one sound instance to affect those of another.

Using the Register Control

The Register control is used to set register values on the test server. It is used only when previewing on the test server and it is disabled when not connected to a server. It is initially hidden; to display it choose **View > Registers**.

Figure 42 Registers

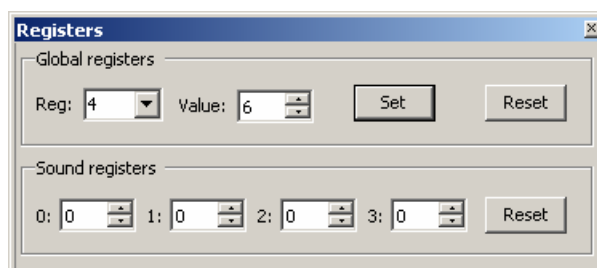


When connected to a server, you can set the value of a global register by selecting a register, entering a value, and clicking the **Set** button. You can also set the initial values of the sound registers that are used when you start a new sound. Remember that this control is only used for controlling the test server; changing its settings does not change any data in the bank itself.

Note: This control sets the register values on the server but the server does not notify it when register values change. Therefore you should not assume that the values displayed on the control represent the current values on the server. For example, if you set Global Register 1 to a value of 10 and then you play a sound that sets Global Register 1 to a value of 11, the value will still be displayed as 10 on the Register Control.

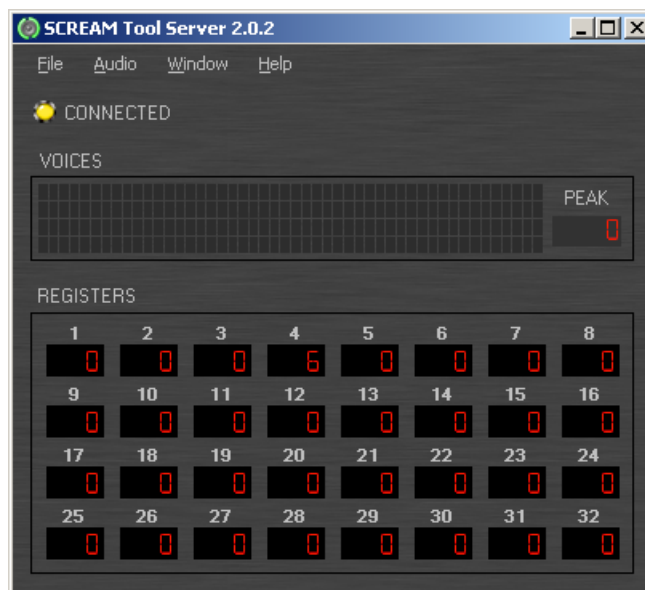
To set the value of a register, you can access the Register dialog by choosing **View > Registers**. Select the Global Register you want to set, from Register 1 to 32 then enter the desired Value and click **Set**. In the example below, Register 4 is set to 6.

Figure 43 Registers



You can also set the value of a Global Register by going to the SCREAM Tool Server window. Just click on the value and type in a new one. The SCREAM Tool Server window shows the current values of the 32 global registers. Figure 44 shows how the server window would correspond to Register 4 having a value of 6.

Figure 44 Global Registers in Server Window



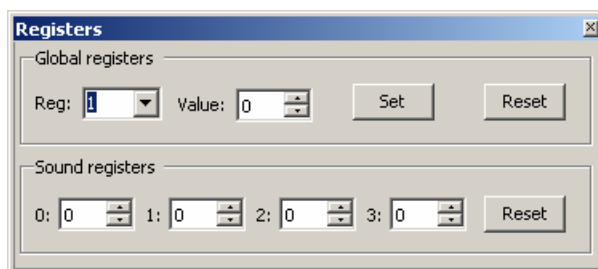
Using Sound Registers

Sound registers are useful when the register value is relevant only to the internal logic of the sound's grains, or there will be multiple sound instances running and you don't want the register operations of one sound instance to affect those of another.

Every sound instance has 4 "registers", each of which can hold a value from -128 to 127. These registers are private to that sound; the game code can view and change these register values but no other sound can look at another sound's registers.

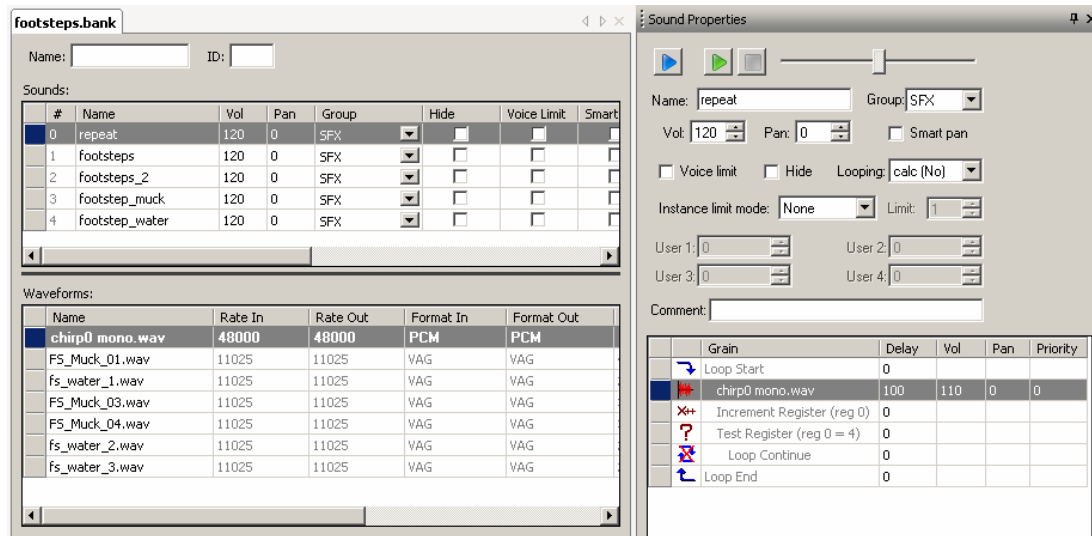
To access the Register dialog, choose **View > Registers**.

Figure 45 Registers



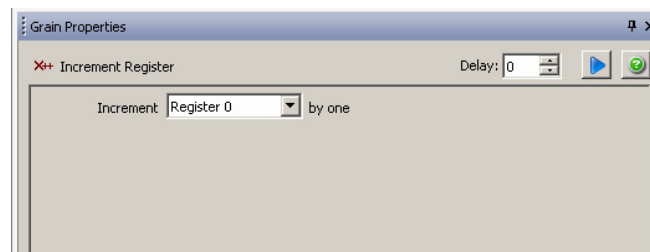
Simple Example: Setting Up an Incrementing Register

An example of that sets up an incrementing register is shown in Figure 46. In this example, a sound called "repeat" uses Loop Start and Loop End grains and a Sound Register to play a chirp sound three times. Each time through the loop, the Increment Register grain increments the Sound Register by 1. When the register value reaches 4, the Test Register grain causes the Loop Continue grain to execute.

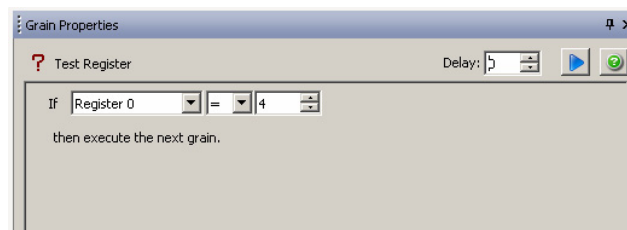
Figure 46 Looping Sound Using a Looping Script

To create an example like this one, you would choose grains from the Grain Palette then set them up in the Sound Properties dialog to specify the logic in the following way:

- Use the Loop Start grain starts the loop.
- Choose a waveform file (chirp0 mono.wav, in this example) that will play after a delay of 100 ticks. The delay value is located to the right of the waveform grain and it is there that you set it to 100; if it is set to 0 then the loop would not work properly since all three iterations of the loop would happen in very rapid succession.
- Use the Increment Register grain to increment the Sound Register by 1. Double-click the Increment Register grain to display and edit the Grain Properties control; specify which Sound Register you want to increment by 1 (in this example, Sound Register 0 is specified). That control looks like this:

Figure 47

- Use the Test Register grain to check the value of the Sound Register and compare it to some value. Choose this value by double-clicking the Test Register grain and then editing the Grain Properties control; for this example, that value is set to 4 for Sound Register 0.

Figure 48

When the test is satisfied, processing proceeds to the Loop Continue grain.

- The Loop Continue grain causes loop processing to jump to the first grain following the next Loop End grain.
- Loop End grain ends the loop.

If you set up an example like this yourself, then you can test the logic by pressing the  button in the Sound Properties control. The waveform grain should play four times. Press the  button to stop.

Using Global Registers

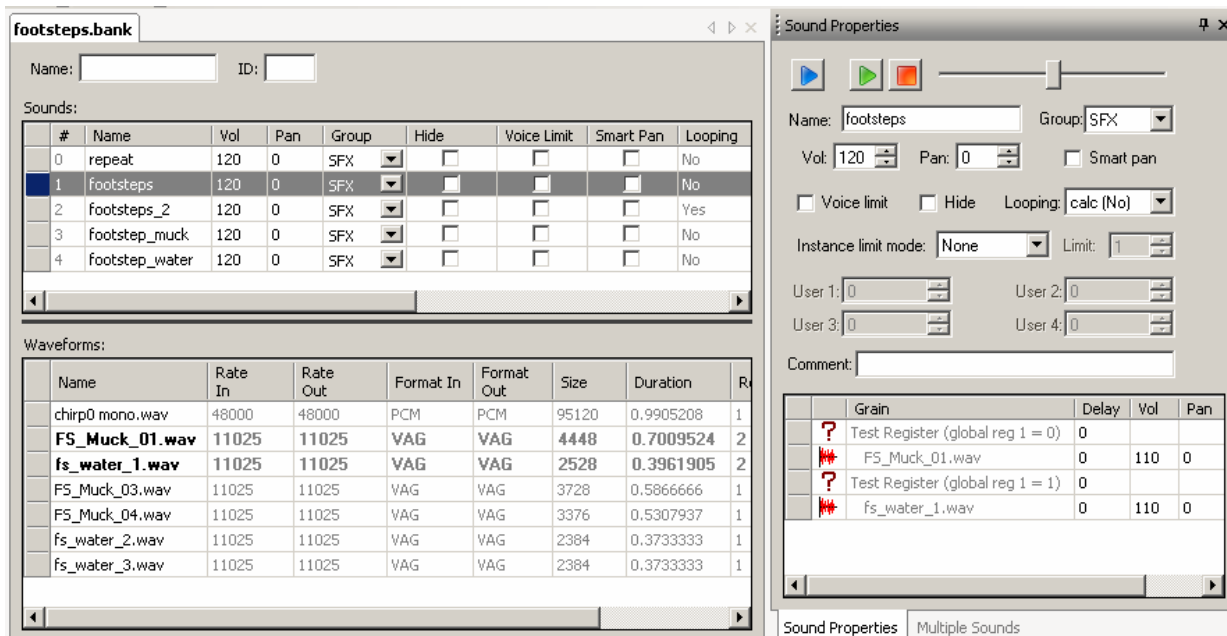
Global registers are variables that can be set by the game or by sounds to allow communication between the game engine and the SCREAM runtime. There are 32 global registers that can be seen by all sound instances. Global registers are numbered from 1 to 32.

A global register would be useful, for example, if you wanted the game to play different types of footsteps sounds depending on the type of environment the character was walking in; a muddy environment would require a "mucky" footstep sound while a watery, rainy environment would require a "watery" footstep sound. SCREAM enables the programmer to communicate this to the audio engine by setting a global register using the SCREAM API. The value of the global register can then be used by a sound designer within the sound as follows:

- If Global Register 1 is set to a value of 0, then the "muck" waveform should be played.
- If Global Register 1 is set a value of 1, then the "water" waveform should be played.

This method requires minimal intervention from a programmer when selecting any sound to be played; as long as the register usage is agreed between sound designer and programmer, the sound designer retains complete flexibility of how sounds are played back under different game circumstances. Figure 49 shows an example of the logic that could be set up using grains in a sound (for the footsteps scenario described above):

Figure 49 Global Registers Example



The screenshot displays the SCREAM software interface. On the left, the 'footsteps.bank' window shows a list of sounds and waveforms. The 'Sounds' table lists: repeat, footsteps, footsteps_2, footstep_muck, and footstep_water. The 'Waveforms' table lists: chirp0 mono.wav, FS_Muck_01.wav, fs_water_1.wav, FS_Muck_03.wav, FS_Muck_04.wav, fs_water_2.wav, and fs_water_3.wav. On the right, the 'Sound Properties' dialog is open for the 'footsteps' sound. It shows various settings like Volume, Pan, and Looping. The 'Grain' section at the bottom of the dialog shows two grains: 'Test Register (global reg 1 = 0)' and 'Test Register (global reg 1 = 1)'. The 'Waveforms' pane in the 'footsteps.bank' window shows the waveforms associated with these grains: 'FS_Muck_01.wav' and 'fs_water_1.wav'.

In the Sound Properties dialog, you can see the grains that specify the logic for the different footsteps:

- Test Register grain checks the value of the Global Register 1 and if it equals 0, then it plays the waveform named **FS_Muck_01.wav** (listed in bold in the Waveforms pane).
- Test Register grain checks the value of the Global Register 1 and if it equals 1, then it plays the waveform named **fs_water_1.wav**.

You can test the logic yourself by setting the value for the Global Register using either of the following methods (as discussed previously):

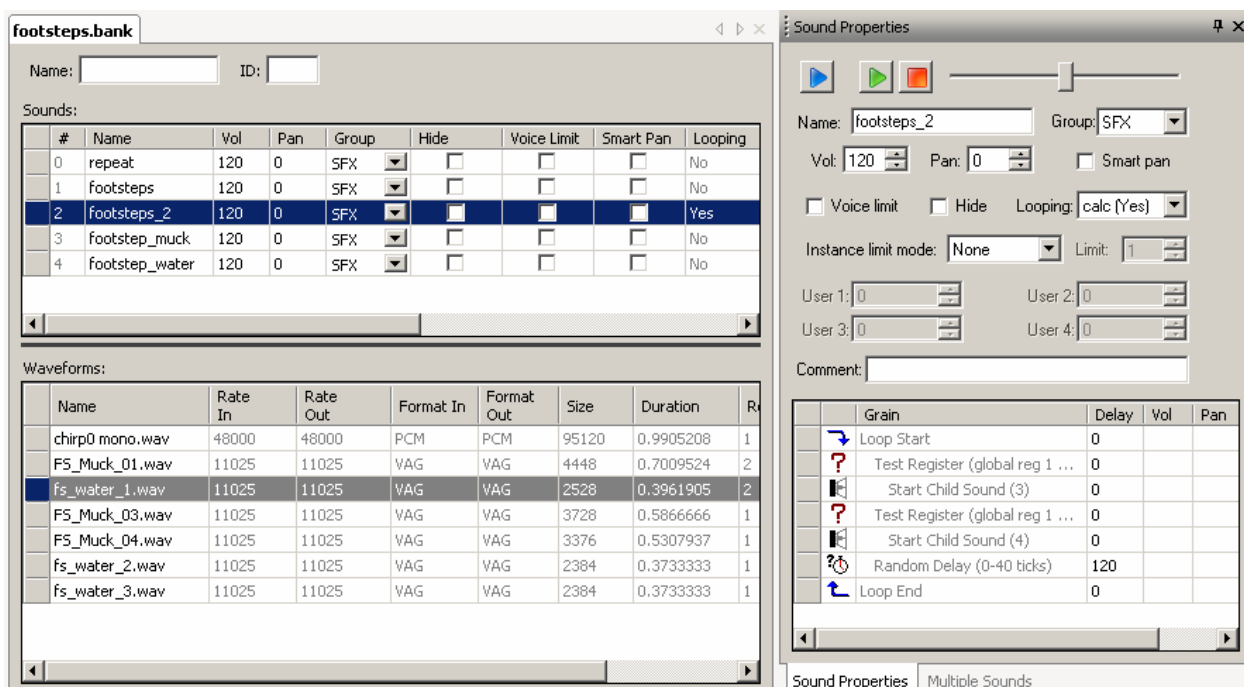
- Go to the SCREAM Server window and entering a 0 or 1 in Global Register 1.
- Choose **View >Registers**, select Global Register 1, enter a 0 or 1 in Value, then click **Set**.

After setting the value, press the  button in the Sound Properties control. Depending on the value, a different waveform will play. Press the  button to stop.

Complex Example

A more complicated example of the footsteps scenario would enable the script to play a few similar, randomly chosen waveforms then apply a random pitchbend to them to make the footsteps sound less regular and more realistic. You can accomplish this by using a child sound, which allows you to trigger other sounds to play. Figure 50 shows an example of logic which does this using a sound called "footsteps_2".

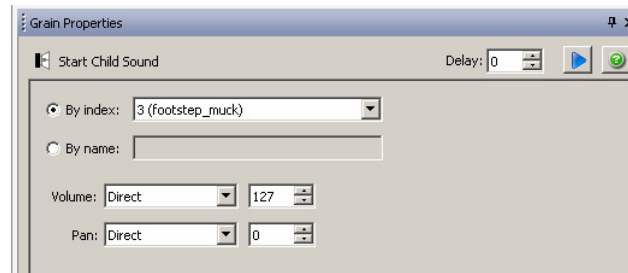
Figure 50 Global Registers Complex Example



In the Sound Properties dialog, you can see the grains that specify the logic for this:

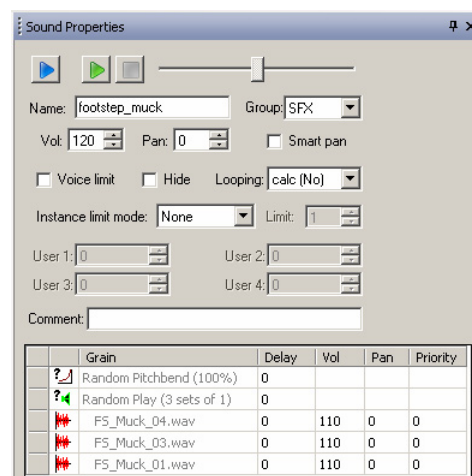
- Loop Start grain starts the loop.
- Test Register grain sets up a conditional expression using Global Register 1. If the test result equals 0, the grain immediately following this one is executed, in this case it executes the Start Child Sound grain. If the result is not 0, the grain immediately following this one is skipped.
- Use the Start Child Sound grain to specify a sound to be played. Double-click the Start Child Sound grain to display and edit the Grain Properties control; specify which sound you want to play (in this example, "footstep_muck" is specified). That control looks like this:

Figure 51



When the Start Child Sound grain is triggered, it plays the "footstep_muck" sound. The "footstep_muck" sound contains grains that randomly choose among three similar waveforms (FS_Muck_01.wave, FS_Muck_03.wav, and FS_Muck_04.wav), then apply a random pitchbend to them, as shown here:

Figure 52



- Test Register grain checks the value of the Global Register 1 and if it equals 1, then it plays the child sound. If the result is not 1, the grain immediately following this one is skipped.
- Use the Start Child Sound grain to specify the sound to be played. When the Start Child Sound grain is triggered, it plays the "footstep_water" sound. The "footstep_water" sound contains logic that randomly chooses among three similar waveforms (fs_water_1.wav, fs_water_2.wav, fs_water3.wav), then applies a random pitchbend to them. (Similar logic as for the previous Start Child Sound grain in this example.)
- Random Delay grain applies a random delay so the timing of footsteps is not too regular.
- When grain processing reaches a Loop End grain, it returns to the previous Loop Start grain.

You can test the logic yourself by setting the value for the Global Register using either of the following methods (as discussed previously):

- Go to the SCREAM Server window and entering a 0 or 1 in Global Register 1.
- Choose **View >Registers**, select Global Register 1, enter a 0 or 1 in Value, then click **Set**.

After setting the value, press the  button in the Sound Properties control. Depending on the value, a different waveform will play. Press the  button to stop.

Note: You can also change the global register value while the sound is playing; the footstep sound will change accordingly.

7 Importing and Exporting

The SCREAM Tool provides a GUI for designing sound banks; the basic data format used by SCREAM. A sound bank contains waveform data (VAG data) and some number of sound scripts that describe how the waveforms are to be played. The sound bank file is referred to as a **.bank** file and is only used by the SCREAM Tool. It cannot be played in-game at runtime. You can use the **File > Open** and **File > Save** or **File > Save As...** menu commands to save your bank as a .bank file.

For in-game use, SCREAM exports an optimized representation of the .bank file, referred to as a **.bnk** file. The .bnk file only retains the data from the tool that is necessary for in-game use. All wave data is represented in the correct export format and is stored using endianness that is suitable for the PLAYSTATION®3.

To read or write other file formats, you must use the **File > Import** or **File > Export** menus, as described in the next sections.

Importing a Bank (Older Format)

You can import a bank file that was saved from an older version of the SCREAM Tool. To import this file, called a .sfx file, choose **File > Import > SFX...** then select the file in the dialog that displays.

Exporting Binaries

For in-game use, SCREAM exports an optimized representation of the .bank file, referred to as a .bnk file. The .bnk file only retains the data from the tool that is necessary for in-game use. All wave data is represented in the correct export format and is stored using endianness that is suitable for the PLAYSTATION®3.

To export a binary .bnk file that can be read by the sound system on the corresponding target platform, use either of the following:

- File > Export > Binary (PC)
- File > Export > Binary (PS3)

Generally, the only .bnk file you will need to export is the PS3 Binary. The PC Binary is a PC optimized format which is not currently supported by any SCREAM tools.

Exporting a C Header File

To export a C Header file with preprocessor definitions for sound names and indices and other information about the bank, choose **File > Export > C Header**. This header can be used by the programmer to index sounds without performing unnecessary string lookups.

Exporting File To Be Read By Older Version of SCREAM Tool

To export a .sfx file readable by an older version of the SCREAM Tool, choose **File > Export > SFX (SCREAM 3.x)**.

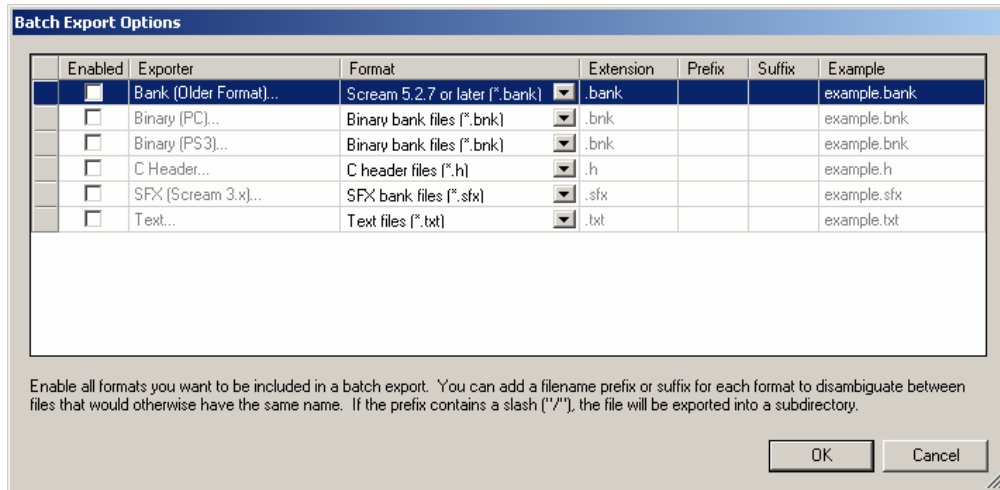
Exporting a Text File

To export a simple text file containing a list of the sounds in the bank, choose **File > Export > Text**.

Exporting Multiple File Formats Using the Batch Option

You can export several file formats at one time. Choose **File > Export > Batch Setup** to display the Batch Export Options dialog, as shown in Figure 53, which lets you configure which file formats are exported during a batch export and how the exported files are named.

Figure 53 Batch Export Options

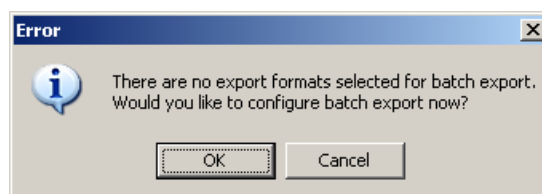


By default, all available export formats are disabled. To enable a format, :

- Click the box in the "Enabled" column beside the appropriate format.
- Choose the appropriate format by clicking the down arrow in the "Format" column and choosing from the list. (Some exporters can export in more than one format and others only have one.)
- Each format has a configurable prefix and suffix. To determine the filename of the exported file, the prefix is prepended to the basename of the file chosen by the user, then the suffix is appended, followed by the file extension. The prefix can include slashes, which is useful for putting files in separate directories. To edit the prefix or suffix for a particular format, double-click on the value in the list.

If you try to perform a batch export without specifying the export formats, you see this message window:

Figure 54 Batch Export Error



Click **OK** to display the Batch Export Options dialog and continue.

Once you configure the Batch Export Options, you can perform a batch export using the **File > Export > Batch** menu command.

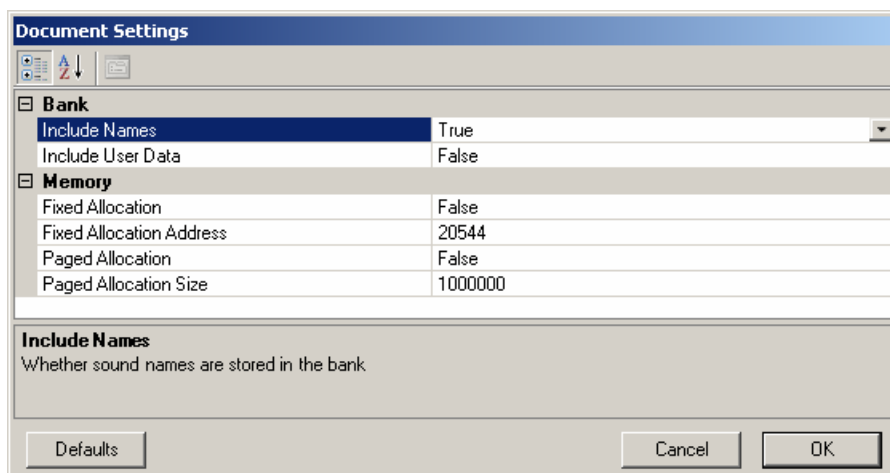
8 Modifying Bank Preferences

You can specify some preferences for the current bank using the Document Preferences Settings dialog. It is worth discussing all of these settings with your audio programmer to ensure optimal use of the SCREAM API within a game. The settings can then be customized to the needs of each project.

Modifying Bank Preferences (Data Stored and Memory Usage)

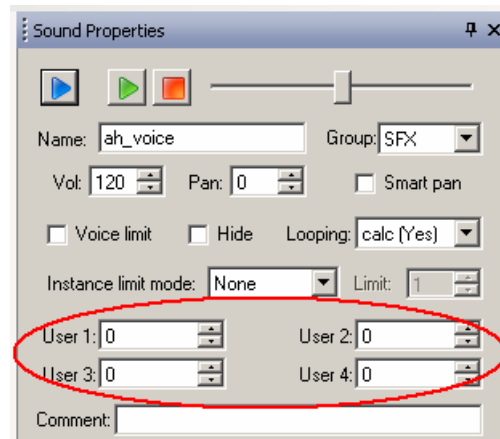
Choose **Edit > Document Preferences** to display the Settings dialog.

Figure 55 Document Preferences



You can modify the following settings:

- **Include Names** – Specifies whether sound names are stored in the bank. If true, the bank and sound names are included in the exported binary .bnk file and are accessible to the game engine. The game engine can trigger sounds by index or by name. Triggering them by name is generally less error-prone, because the index of a sound can change if another sound is added or removed from the bank. If you expect sounds to be triggered by name, make sure that names are included in the bank. Not including names will make the exported binary file slightly smaller, so it will consume slightly less memory at runtime. If your game team plans on triggering sounds by index, it is probably a good idea to export a C header file for the programming team, so they can refer to sounds with preprocessor macros rather than raw index values. See "Exporting a C Header File" for information. Triggering sounds by index should also be slightly faster than performing a text lookup so it is worth considering this option if it is necessary to optimize SCREAM usage.
- **Include User Data** – Specifies whether user data is stored in the bank. If true, four integer values are associated with each sound. the values correspond to fields which are called User 1 through User 4. If true, the User 1 through User 4 fields are active and can be edited in the Sound Properties Control, as shown in the following dialog.

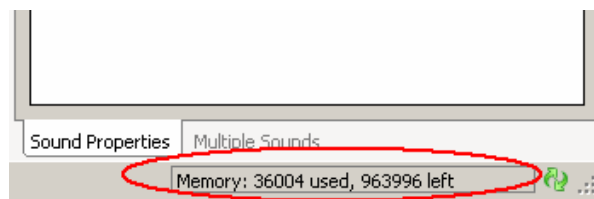
Figure 56 User Fields

If **false**, the User 1 through User 4 fields are grayed out and inactive. The game engine can interpret these values in any way the game programmers wish.

Note: The rest of the settings, Fixed Allocation, Fixed Allocation Address, Paged Allocation, and Paged Allocation Size, are not supported in this version.

Checking Current Memory Usage

The memory usage of the current bank is indicated in the status bar at the bottom of the main SCREAM window.

Figure 57 Memory Usage

The maximum memory usage desired for a bank can be set by editing the "Paged Allocation Size" value in the Document Preferences dialog. When the usage exceeds the maximum, the indicator turns red.