



Creative Labs, Inc.  
1901 McCarthy Blvd.  
Milpitas, CA 95035

Phone (408) 428 6600  
Fax (408) 428 6611  
[www.creative.com](http://www.creative.com)

---

# ISACT Production Studio

---

## IPS Tutorial

---

# ISACT Production Studio

## IPS Tutorial

© 2004 Creative Technology, Ltd. All rights reserved.

### Revision History

0.3	15 March 2004	Peter Harrison	GDC Beta version
0.4	06 April 2004	Peter Harrison	Beta website revision
0.5	20 July 2004	Peter Harrison	Revised for BETA 0.9
0.51	30 July 2004	Peter Harrison	Revised for RC1 release
0.52	17 August 2004	Peter Harrison	Revised for RC2 release
0.53	21 Dec. 2004	Peter Harrison	Revised demo samples
0.54	4 Feb. 2005	Peter Harrison	Added Interactive Music
0.55	17 Feb. 2005	Peter Harrison	Updated for IPS release 1.5
0.56	14 June 2005	Dan Peacock	Updated for IPS release 1.53
0.57	May 11 2006	Ed Cooper	Updated for IPS release 1.64

### Trademarks and Service marks

Creative, Sound Blaster, and the Creative logo are registered trademarks, and NOMAD, Environmental Audio, EAX, Environmental Audio Extensions, and the SoundBlaster Live! logo are trademarks of Creative Technology Ltd. in the United States and/or other countries.

All other brand and product names listed are trademarks or registered trademarks of their respective holders.

### License Agreement, Limitation, and Disclaimer of Warranties

PLEASE NOTE: BY DOWNLOADING AND/OR USING THE SOFTWARE AND/OR MANUAL ACCOMPANYING THIS LICENSE AGREEMENT, YOU ARE HEREBY AGREEING TO THE FOLLOWING TERMS AND CONDITIONS:

The software and related written materials, including any instructions for use, are provided on an "AS IS" basis, without warranty of any kind, express or implied. This disclaimer of warranty expressly includes, but is not limited to, any implied warranties of merchantability and/or of fitness for a particular purpose. No oral or written information given by Creative Technology Ltd., its suppliers, distributors, dealers, employees, or agents, shall create or otherwise enlarge the scope of any warranty hereunder. Licensee assumes the entire risk as to the quality and the performance of such software and licensee application. Should the software, and/or Licensee application prove defective, you, as licensee (and not Creative Technology Ltd., its suppliers, distributors, dealers or agents), assume the entire cost of all necessary correction, servicing, or repair.

### Restrictions on Use

Creative Technology Ltd. retains title and ownership of the manual and software as well as ownership of the copyright in any subsequent copies of the manual and software, irrespective of the form of media on or in which the manual and software are recorded or fixed. By downloading and/or using this manual and software, Licensee agrees to be bound to the terms of this agreement and further agrees that:

Creative's BBS/FTP/website are the only online sites where users may download electronic files containing the manual and/or software,

Licensee shall use the manual and/or software only for the purpose of developing licensee applications compatible with Creative's Nomad series of products, unless otherwise agreed to by further written agreement from Creative Technology Ltd.; and,

Licensee shall not distribute or copy the manual for any reason or by any means (including in electronic form) or distribute, copy, modify, adapt, reverse engineer, translate or prepare any derivative work based on the manual or software or any element thereof other than for the above said purpose, without the express written consent of Creative Technology Ltd. Creative Technology Ltd. reserves all rights not expressly granted to licensee in this license agreement.

#### Limitation of Liability

In no event will Creative Technology Ltd, or anyone else involved in the creation, production, and/or delivery of this software product be liable to licensee or any other person or entity for any direct or other damages, including, without limitation, any interruption of services, lost profits, lost savings, loss of data, or any other consequential, incidental, special, or punitive damages, arising out of the purchase, use, inability to use, or operation of the software, and/or licensee application, even if Creative Technology Ltd. or any authorized Creative Technology Ltd. dealer has been advised of the possibility of such damages. Licensee accepts said disclaimer as the basis upon which the software is offered at the current price and acknowledges that the price of the software would be higher in lieu of said disclaimer. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitations and exclusions may not apply to you.

Information in this document is subject to change without notice. Creative Technology Ltd. shall have no obligation to update or otherwise correct any errors in the manual and software even if Creative Technology Ltd. is aware of such errors and Creative Technology Ltd. shall be under no obligation to provide to Licensee any updates, corrections or bug-fixes which Creative Technology Ltd. may elect to prepare.

Creative Technology Ltd. does not warrant that the functions contained in the manual and software will be uninterrupted or error free and Licensee is encouraged to test the software for Licensee's intended use prior to placing any reliance thereon.

# Table of Contents



	1
<hr/>	
Table of Contents	4
Introduction	10
About ISACT	10
Run-time Mixing	10
The ISACT Production studio	10
The ISACT Build Utility	10
The ISACT Run-time libraries	10
Document scope	11
Pre-requisites	11
Tutorial Instructions	11
<hr/>	
Creating audio content with IPS	12
ISACT Components	12
Audio Samples	12
3D Buffer Control	12
Global Effects	12
Spatial Paths	12
Sound Events	12
Audio Sequences	13
Timelines	13
Sound Queues	13
Real-time Parameter Controls	13
Transitions	14
Sound Randomizers	14
Sound Entities	14
Getting started with ISACT	15
The ISACT Interface	15
The Project Tree	15
Dividing data into banks	16

Working with audio: Audio Samples	17
Loading audio into IPS	17
The Playback Transport	18
Waveform editing	19
Undoing your changes	19
Exporting samples	20
Audio Samples with 3D Buffer Controls	20
Setting values in IPS	21
Setting distance roll-off using the 3D Buffer Controller	21
Visualising Roll-off settings with the 3D Roll-off Graph	22
Setting Audio Sample directivity	23
A brief explanation of Sound Cones	23
Applying 3D Controls to Audio Samples	24
Auditioning Roll-off settings with the 3D Audition Map	25
Zooming in on the grid	26
Representation of Audio Sample properties	26
Editing 3D Buffer Controller properties while playing content	26
Conclusion	27
Random dynamics: Sound Events	28
Getting started with Sound Events	28
Sound Events Example one – the clock is ticking...!	28
Adding Audio to a Sound Event	28
Auditioning a Sound Event	29
Changing order	30
Changing chance	30
Adding random dynamics to a Sound Event	31
Managing Audio Samples in your Sound Event	31
Looped Sound Events	32
Using silence tracks	33
Other Sound Event examples	34
Counting.sac	34
Grasshopper.sac	34
Conclusion	35
Defining motion: Spatial Paths	36
Applications for Spatial Paths	36
Join the dots.... creating a Spatial Path	36

Editing a Spatial Path	38
Editing in the time domain	38
Sound Events Example two – flying bugs	39
Managing your assets: Indices, Banks and Folders	41
Background to ISACT asset management	41
Referencing data by index	41
Referencing between Banks	41
Sample Bank Indexing	42
Bank Editing	42
Dealing with invalid references	43
Importing Banks	43
Using Folders	43
Adding a new folder	43
Placing content in folders	43
Additional Features of folders	43
Memory management with Banks	44
Bank and Index Example 1: Losing Count!	44
Optimising memory usage via streaming, pre-loading and compression	49
Streaming	49
Pre-loading	50
Compression	50
Run-time de-compression	51
The ISACT Build Utility	52
Run-time implementation of Optimisations	52
Conclusions	53
Using hardware effects processing: Global Effects	54
Effects Manager	55
Conclusions	59
Where the action's at: Sound Entities	60
Sound Entity example one: Too many bugs	60
Players	60
Variables	61
Actions	62
Action Conditions	63
Action Commands	64
Auditioning Entities	65

Conclusion	66
High Command – Real-time Parameter Controls	67
Real-time Parameter Control example one: Get revved up!	67
Mapping properties to inputs	68
Auditioning an RPC	69
Integrating an RPC within a Sound Entity	70
Real-time Parameter Control example two: Cross Fading	70
Simple Sequencing: Sound Queues and Timelines	72
Sound Queues	72
Sound Timelines	74
Making Tracks: Audio Sequences	76
Audio Sequence example one: A moving experience!	76
Audio Sequence example two: The next Level!	80
Sound Randomizer	83
Miscellaneous other stuff	84
Voice Count Tool Window	84
Content Cloning	84
Content Mixer	84
View Properties	85
Sting Control	86
Spatial Editor	86
Boost	87
Setting Global Headroom	88
Setting per-sound object boost	88
Groups	88
Sub-Mix Groups	89
Multi-Properties Dialogue Boxes	91
Global Effect	92
Group Name	92
Sub-Mix Group Name	92
At Start Set Global Variable	92
At End Set Global Variable	92
Gain (mb)	92
Implementing Ducking Using Entities with Groups	92
<hr/>	
Creating interactive music with IPS	96

Interactive music	96
Our demo music	96
Music Loops as Audio Sequences	96
The Master Track	97
Sections	97
Tempos	98
Time Signatures	99
Testing Section playback	99
Interactive music driven by Sound Entities	100
Transition, not Play	101
Stings and motifs	101
The Master Player	101
Starting content on the beat	102
Auditioning stings	103
Stings within the Entity	103
<hr/>	
Run-time Editing with the Network Connection	105
Target Platform Settings	108
Auditioning of Compressed Audio Formats	110
<hr/>	
Conclusions	112
Contact Details	112
USA	112
Europe	112
<hr/>	



---

# ISACT Production Studio

## IPS Tutorial



## Introduction

### About ISACT

The ISACT Production Studio (IPS) is your key to Creative Labs Interactive Spatialised Audio Composition Technology. ISACT represents a set of technologies that are intended to make the design and implementation of advanced 3D audio content for interactive applications quick, simple and convenient.

Using ISACT, you can create compelling audio content that can be played back in 'surround sound' on nearly any delivery system. You can also access flexible randomisation features that are capable of introducing almost infinite variation to your sound effects. And you can design mechanisms for controlling dynamic, layered sound effects using a feature-rich but straightforward state-based programming system.

ISACT also provides facilities for storing your audio assets in a flexible bank structure. Different banks of data can be loaded and unloaded at run-time, enabling efficient dynamic caching of audio data.

For systems with an EAX Advanced HD™ sound device such as the SoundBlaster Audigy and X-Fi series of cards, you can also work with real-time hardware accelerated digital audio effects including environmental reverb, chorus, flanger, EQ etc...

### Run-time Mixing

---

ISACT takes advantage of *run-time mixing*, leveraging a device's native audio mixing and spatialisation capabilities to power a flexible interactive 3D soundtrack. Because ISACT content is never pre-mixed, it is never necessary to down-mix or up-mix for different speaker configurations – you are guaranteed compatibility with any delivery system supported by the rendering device.

### The ISACT Production studio

---

The IPS itself is a powerful Windows-based application for creating and managing all your ISACT content.

### The ISACT Build Utility

---

The build utility can be used to minimize the size of your final content files by removing redundant data and compressing audio data into formats such as ADPCM, Ogg Vorbis, or WMA. Using the IPS, the Sound Designer can flag sound data that will require compression. The ISACT build utility can be used to perform the compression, preparing the data files for distribution.

### The ISACT Run-time libraries

---

The content you create using IPS can be accessed, manipulated and played back from within your application via the ISACT run-time libraries.

The ISACT run-time libraries are built for PC, but they are modular and portable. This means that most parts can be ported to different platforms, and those that can't be ported can instead be replaced with other libraries with similar functionality. This means that, with a little programming work, the content you create using a PC with a high-end sound device can also be played back on other devices such as games consoles.

## Document scope

---

The ISACT tutorial is not intended to be an exhaustive reference of all IPS functionality. Instead, it aims to give you a grounding in the fundamental techniques of content design with ISACT via examples.

Although the tutorial will also touch on aspects of integrating ISACT content within your game or interactive application, it is not intended to be an implementation guide. For full details on how to integrate the ISACT engine into your application's codebase, see the accompanying ISACT API reference.

## Pre-requisites

---

In order to use IPS and follow the tutorial, you will need access to a Windows PC. ISACT content can be created using any PC capable of rendering DirectSound3D / OpenAL spatialised audio, but you can only experience the full functionality of ISACT if your PC has a SoundBlaster Audigy or X-Fi series of soundcard installed. A decent 5.1 or better surround speaker set will also be useful. IPS supports Windows XP and Windows 2000. Of course, you will need to have installed your copy of the IPS.

## Tutorial Instructions

---

Throughout the document, you will see instructions on how to work with the tutorial content. The instructions will be presented in a grey panel, like this:

I am an IPS Tutorial Instruction. Obey me!

All the files necessary to create the tutorial content can be found in the ISACT installation folder, inside the "ISACT Tutorial" sub-folder.

Follow the instructions to get the best out of the tutorial!

# Creating audio content with IPS

## ISACT Components

ISACT provides you with a huge range of different ways to make noise! The basic building block for an ISACT soundtrack is the humble audio sample. But sound samples can be controlled, combined and processed in many different ways. No matter what you want to do, ISACT offers a suitable component type!



### Audio Samples

The Audio Sample is a fundamental building block for more complex ISACT content, as well as being a playable sound in its own right. The Audio Sample contains a sound's sample data.



### 3D Buffer Control

To play a sound within a 3D audio system, the renderer needs to know about the sound's behaviour. For instance, how loud is the sound and how is its intensity affected by distance? Is the sound emitted in a particular direction? Audio Samples don't include any of this information. But the 3D Buffer Control is a set of properties that can be applied to one or more Audio Samples, in order to define their 3D behaviour.



### Global Effects

Creative Labs EAX ADVANCED HD™ soundcards support hardware-accelerated audio processing effects including multi-channel reverberation, chorus, compression, distortion, EQ, flanger, frequency shift, pitch shift and more. ISACT content can be designed to send audio through the effect processing slots. ISACT can control the state of the effect slots using a Global Effect component. This component allows you to load an effect type in each slot, and set each effect's properties.



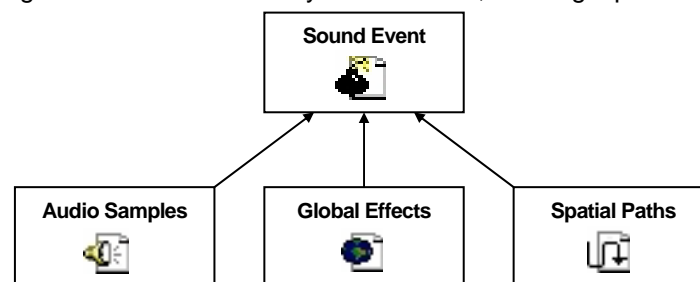
### Spatial Paths

An essential component that helps to bring ISACT content to life in three dimensions is the Spatial Path. A Spatial Path is a flexible way to predefine 3D movement for sound sources over any given period of time.



### Sound Events

For adding what you might call 'random dynamics' to your soundtrack, you can use ISACT's Sound Events. Sound Events are very cool; they allow you to randomise aspects of the audio played back by a particular trigger. Things like sample, pitch, volume, positioning, and on EAX systems filtering and effects sends. So, this allows you to respond to a game event with a variably altered sound, avoiding repetition.

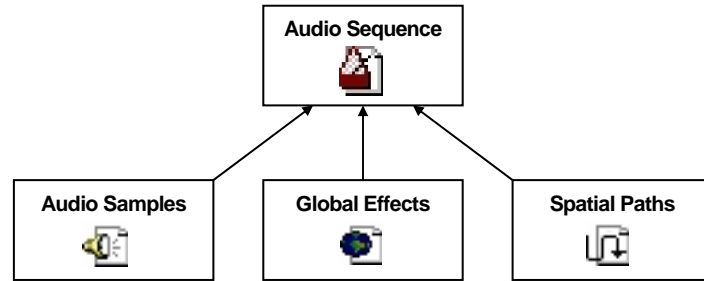


Sound Events can be made from Audio Samples, Global Effects and Spatial Paths.



### Audio Sequences

This is similar to layered audio content built using a traditional tracker or sequencer, and pre-mixed into a stereo or 5.1 stream. However, individual sounds in ISACT's audio sequences can be positioned within a 3D co-ordinate system, based on the concepts you are familiar with from interactive 3D sound. ISACT audio sequences would be useful primarily for 3D music, but could also form immersive 'surround' ambiences, or play alongside visual animations for a movie or cut-scene.



Audio Sequences can be made from Audio Samples, Global Effects and Spatial Paths.



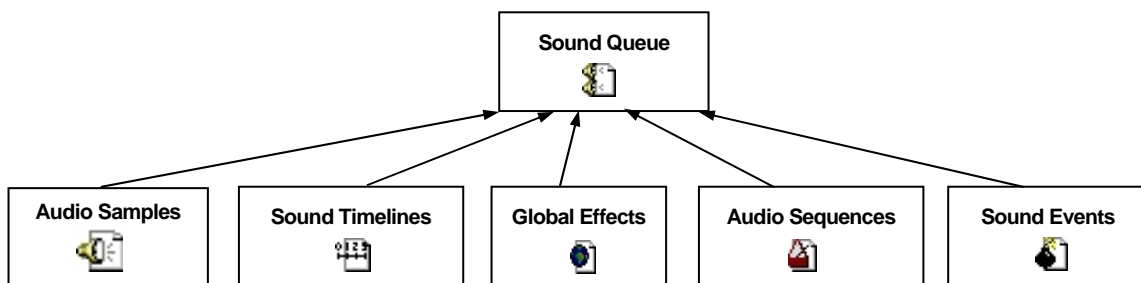
### Timelines

The Timeline allows you to chain together Sound Events. Within a Timeline, Sound Events are positioned absolutely; each Event's start time is relative to the start of the Timeline, and is not effected by the timing of previous Sound Events in the list. This means that in a Timeline several Sound Events can overlap and play simultaneously.



### Sound Queues

You might wish to play several instances of content such as Sound Events, Audio Samples, Sequences and/or Timelines sequentially in response to a trigger in your application. If you simply splice waves together into one long sample you waste resources and also forfeit the random dynamics offered by components such as Sound Events. So ISACT allows you to automate the playing of content sequentially in a Sound Queue. As the name suggests, objects in the queue are played consecutively – no overlap is allowed.



### Real-time Parameter Controls

A Real-time Parameter Control allows the designer to determine how an ISACT audio component can react to a numeric input. The RPC defines a mapping between an input value and one or more properties of a 3D sound source, for example volume, pitch, effects sends etc...

Within IPS, RPCs are typically referenced by Sound Entities to enable complex behaviour. However, using the run-time libraries a programmer can control other content types using an RPC.



### Transitions

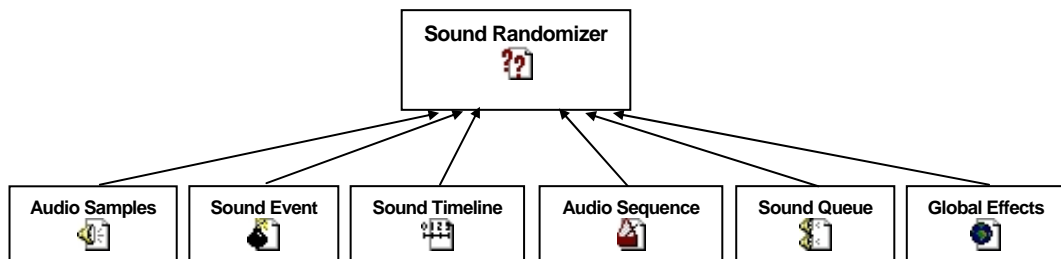
Transitions are used to tell ISACT how to carry out the changeover from playing one piece of content to another. For example, when using ISACT to play back an interactive music track, it is often necessary to switch from one category or section of music to another. In some instances the changes should be abrupt, in other situations a smooth cross-fade might be suitable. The Transition component allows you to shape the shift to your own specification.

Like RPCs, Transitions are only employed within IPS as part of a Sound Entity, but can be programmatically applied to other content types at run-time.



### Sound Randomizers

Sound Randomizers allow you to respond to an event with a randomly selected audio component of any type from the following diagram: -



### Sound Entities

When you want something a bit more complex, ISACT provides 'Sound Entities'. A Sound Entity is a perfect mechanism to use when a sound must be dynamically controllable according to game variables. Typical examples would be engine sounds, which are modulated according to engine speed, load, gear etc... But there are plenty more applications for sound entities, as you'll see later on.

A Sound Entity can control pretty much any ISACT content. Audio Samples, Sound Events, Sequences, Queues, and Timelines can all be started, stopped and adjusted by the Sound Entity's scripting system. Global Effects can be set, and Real-time Parameter Controls and Transitions are both available.

## Getting started with ISACT

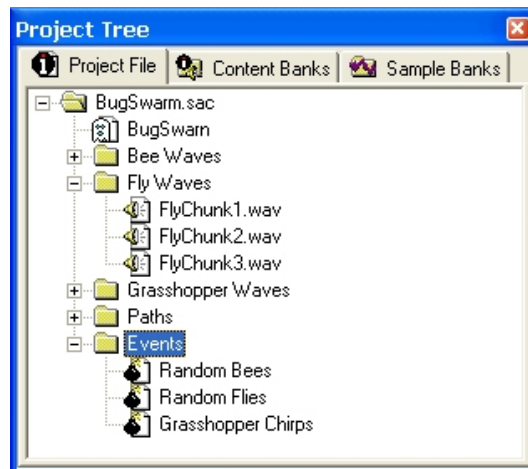
### The ISACT Interface

The IPS interface allows you to have several tool windows open, viewing the different components in your project. As with many similar content creation tools, screen 'real estate' can quickly become used up, so it is advisable to set your system to display at a high graphical resolution. Those with a multiple monitor facilities will be at an advantage – IPS supports multiple screen systems.

If you start the IPS application by running the program without specifying a .SAC file, you will begin with a blank project.

Start IPS by clicking on the ISACT Editor icon on your desktop, or from the IPS group in the Windows start menu.

### The Project Tree



The contents of the current open project are displayed in the Project Tree. When you run IPS for the very first time, you might well find that no tools or viewing windows are visible. Whatever the case, you will always need to have access to the Project Tree.

If you can't see the project tree, then go to the *View* menu, and select *Project Tree*. The project tree should be empty.

Ctrl-T is a keyboard shortcut to toggle the Project Tree on and off

Size the Project Tree, giving you room to view and manipulate components.

Try moving the Project Tree window around the screen. Notice how the window automatically docks with the edges of the application window. (To prevent an IPS window from docking, hold down the *Control* key. If you decide that you want to turn docking off for the window, right click the window's title bar and toggle the 'Dockable' setting to off). Find a handy position for the Project Tree. Note: when you exit IPS, the app remembers the current screen configuration, so that next time you load the application, the windows will re-appear where you left them.

At the top of the Project Tree are three tabs, marked *Project File*, *Content Banks*, and *Sample Banks*. The *Content Banks* and *Sample Banks* tabs give you access to bank management facilities, allowing you to divide your content and sound data into separate banks. The reasons why you should divide your content in this way are detailed below:

#### Dividing data into banks

A project can include practically any number of instances of each ISACT component type such as Audio Samples or Sound Events. For the sake of efficiency, most applications will not want to load all of their audio assets at once.

Organising your ISACT project as multiple banks of indexed objects allows your application to load and unload audio data as it is required thus facilitating memory management, and giving you access to powerful asset management facilities.

ISACT distinguishes between two different bank types: Sample Banks and Content Banks.

Sample Banks can contain Audio Samples, plus 3D Buffer Controls to provide 3D behaviour for the Samples.

Content Banks can contain each of the following object types:

- Metric Time Audio Sequence
- Sound Event
- Sound Timeline
- Sound Queue
- Global Effect
- Spatial Path
- Real-time Parameter Controller
- Transition
- Sound Entity

A project can include references to any number of content banks, and up to 255 sample banks.

For these reasons, all the following tutorial project files have their content divided into content banks and sample banks.

I'm sure you're itching to create some ISACT content now – so let's start with the vital Audio Sample component.

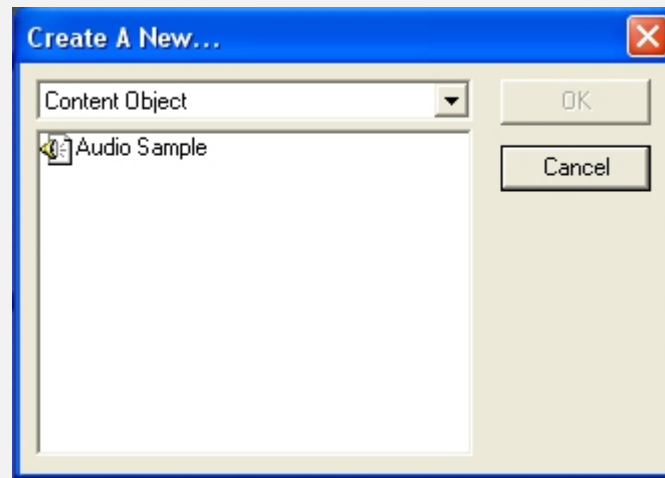


## Working with audio: Audio Samples



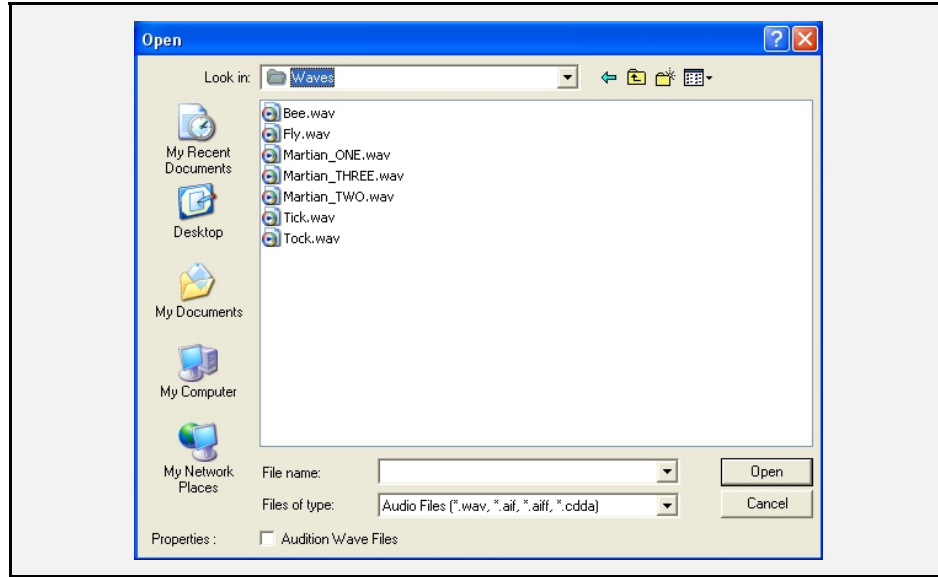
### Loading audio into IPS

In order to load audio into IPS, you first need to create a Sample Bank. Click on the *Sample Bank* tab, then right-click on the *SampleBanks* folder and select *new bank*. Give your new bank a sensible name and then click *Save*. Once you've created your sample bank you can now begin to add some audio samples: Right-click on your newly created sample bank. A context sensitive menu will appear. Select the "New..." option.



The drop-down menu at the top of the dialog allows you to select an object type to add. Audio Samples are Content Objects, so make certain that *Content Object* is selected.

Select *Audio Sample*, then *OK*. The IPS sound-file loading dialog will appear allowing you to browse for audio data.



The IPS sound-file loading dialog supports multiple selections, so you can load several files at once. You can also audition individual audio files before opening them by clicking the 'Audition Wave Files' checkbox and then clicking on the audio file itself. Spatialised audio generally works best with mono audio sources. If you attempt to load stereo or multi-channel wave data, you will be prompted whether to create separate Audio Samples for the individual channels, mix all the channels together into a mono stream, or work with the sample in stereo.

IPS currently works with PCM audio data in .WAV and .AIFF format. ISACT is designed to be able to seamlessly manage compressed data in formats such as ADPCM, Ogg Vorbis, and WMA.

In the file-loading dialog, go to the \ISACT Tutorial\Waves folder.

Locate the files named Tick.wav and Tock.wav.

Load these files, taking advantage of multiple selection if you like.

A new Audio Sample will be added to your project for each file you selected.

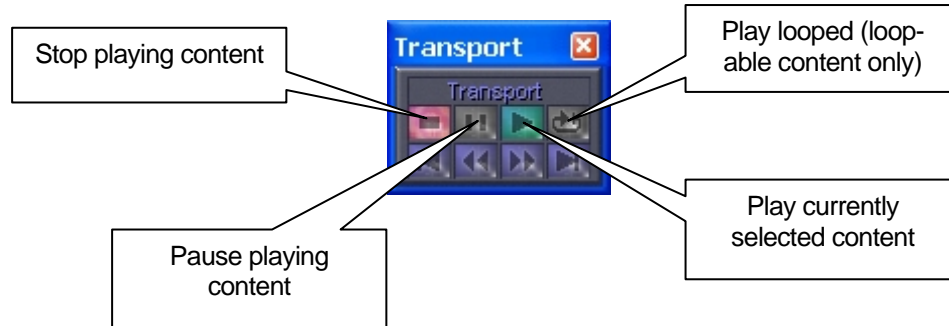
Make sure you have the *Sample Banks* tab selected, then double-click one of the Audio Samples you just created. A waveform display appears, showing the audio sample data.

### The Playback Transport

At this stage, you may be curious to play the sounds you just loaded. So its time to check out the playback transport.

If you can't see the Playback Transport window, go to the *View* menu and select *Tool Windows -> Transport*. The shortcut alternative is *Ctrl-Shift-T*.

Just like the Project Tree, and other IPS editing windows, the transport window can be resized, moved around, and docked. Size the transport to its maximum extent, and find a good location on the screen to position it. The controls on the Playback Transport tool should be familiar to anyone who has used audio software.



While the Audio Sample is the selected edit window, you'll notice that the Playback Transport buttons become coloured, indicating that you have some playable content selected.

Click *Play* to hear the waveform played back. The shortcut alternative is the Space-Bar (when looped playback is enabled, Shift-Space-Bar will start looped playback)

When performing wave playback and editing, you can double-click *Stop* to reset the play cursor to the start of the wave data. The Playback Transport can also be used to operate much of the other playable content in IPS.

### Waveform editing

Whilst IPS is not a full-blown audio editing suite like Wavelab or SoundForge, essential sound editing can be carried out within the application. When an Audio Sample is open and in focus, various options within the *Process* menu are enabled. You can perform manipulations on the entire wave, or click and drag on the waveform display to highlight regions of audio to process. Click on the waveform to release the highlighted region.

You can erase portions of the waveform by pressing the delete key. You can ramp sound up and down using *Process->Fade Volume->Fade In* and *Process->Fade Volume->Fade Out*. You can normalise audio with *Process->Normalise*. Time stretching and Pitch shifting are also available in the *Process* menu.

An alternative way to tag sounds for processing is to highlight one or more Audio Samples in a Sample Bank, and right-click. The context menu that appears exposes a subset of ISACT's processing options.

### Undoing your changes

IPS currently supports 20 levels of undo for all content types. You can hit *Control-Z* to undo your last change.

If you experimented with abandon when you were testing the wave editing, you might find that you have made some undesirable and possibly irreversible changes to your waveform. Instead of using the undo feature, you could simply delete the Audio Sample and load the original back in as a new buffer with the same name.

However, if you were working on a large project, and the Audio Sample you changed was referenced by some higher level content – say for example it was used as part of a Sound Event - how can you revert to the original audio without messing up the project's referencing? Simple! Go to your Sample Bank and right-click on the Audio Sample in question. The context menu that appears gives you the option to '*Replace*' the Audio Sample. Choose this option, and replace the altered wave data by selecting the original file from disk.

In IPS, wave data is actually copied from a wave file when you add it to the project. The copied data resides inside the ISACT project file. So performing destructive edits on waveforms will not change the original wave file. Using *Replace* simply loads new data into the Audio Sample. Anywhere the Audio Sample is referenced in the project, the replacement data will appear. So this would be useful also if you decided to make some alterations to the wave data with another tool, outside of the IPS environment.

#### Exporting samples

If you are particularly pleased with your destructive editing, and want to make your edited waveform available outside of ISACT, you can export the sample. Just right click on the appropriate Audio Sample, and select *Export*. A save dialog will appear allowing you to select a filename and location for your new file.

#### Audio Samples with 3D Buffer Controls

---

None of the manipulations that we have investigated so far have much relevance to 3D audio. Now we'll examine how wave data loaded into an Audio Sample takes on the properties of a spatialised sound source, via the 3D Buffer Controller.

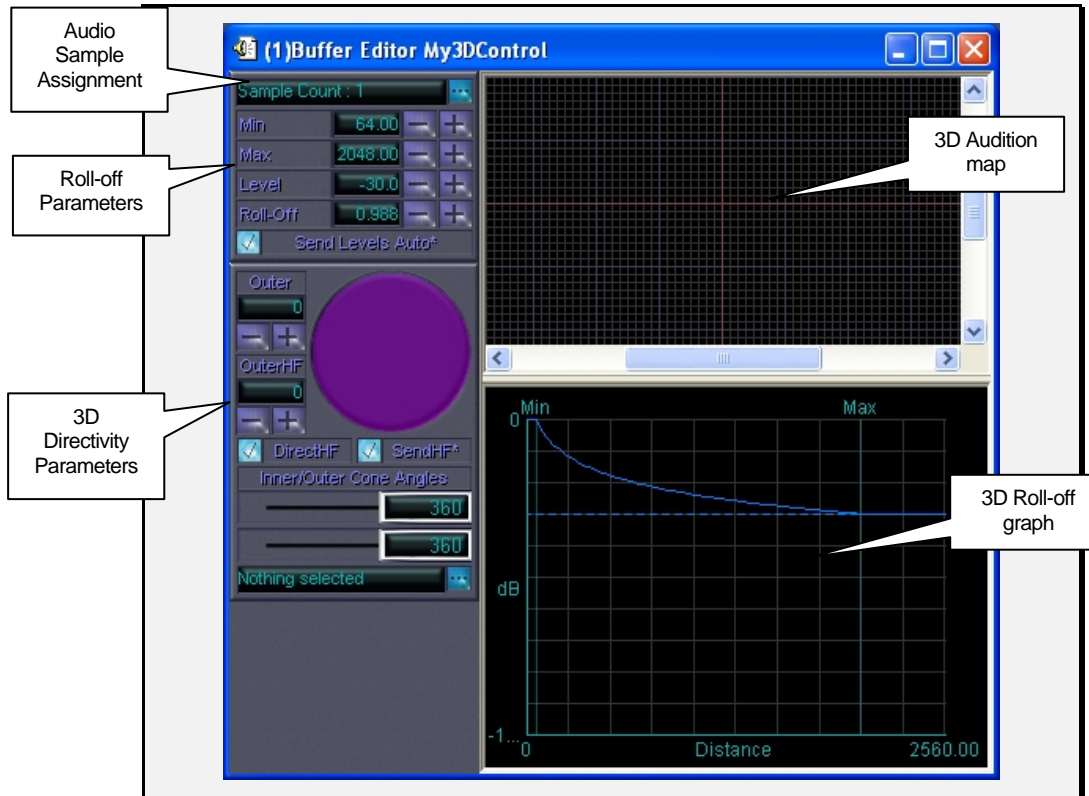
The relationship between Audio Samples and 3D Buffer Controls is many-to-one. Put simply, this means that you can create one 3D Buffer Controller and should you want a group of Audio Samples to exhibit identical behaviour in terms of 3D spatialisation, you can apply it to all of them. Of course that means that when you make a change to the 3D properties of the Control, it changes the behaviour of all Audio Samples associated with it.

Let's look closed into how you create and adjust 3D Buffer Controllers, and how you connect an Audio Sample to a 3D Buffer Controller. First, let's create the new component.

Right-Click your Sample Bank again, and click *New...*

This time, change the object type to *Controller Object* and select *3D Buffer Controller*.

Give your Control a sensible name, click *OK*, and the dialog window appears



### Setting values in IPS

IPS' interface generally offers three ways to set values such as the 3D properties in the 3D Buffer Controller window. Firstly, beside the value display you will notice icons to increment (+) and decrement (-) the setting. Secondly, you can double click the display to obtain a number entry dialog. Alternatively, position the mouse cursor over the value display and use the mouse wheel to scroll the value up or down.

### Setting distance roll-off using the 3D Buffer Controller

A fundamental mechanism for any interactive spatialised audio system is that which controls each sound source's 'distance roll-off'. The aim is to determine the intensity of the sound source as a function of the distance between the sound source and the 3D audio scene's "camera" – the listener.

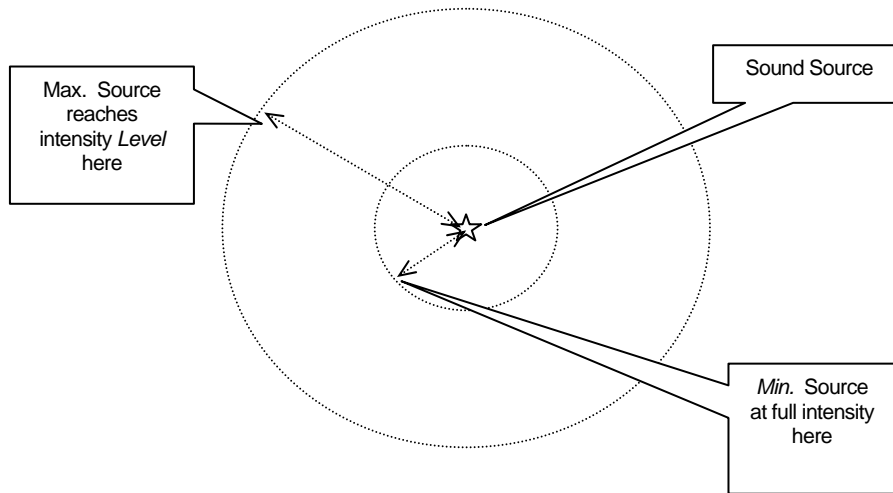
The standard distance roll-off model for 3D audio revolves around the idea of each sound source having a *minimum* or *reference* distance. When the listener is within this radius, the sound will be heard at its full intensity. As the sound moves further away, the intensity is reduced. Typically each doubling in distance over the minimum distance will cause a halving in the audible intensity of the sound.

Each 3D audio programming interface has its own variations on this model, and ISACT IPS is no exception! The ISACT 3D Buffer Controller has a *Min* property, see the top left corner of the Buffer Tools window. *Min* represents the radius of the circle in which the sound source can be perceived at its full intensity. This is similar to the 'minimum' or 'reference' distance concept in 3D audio APIs.

Many audio designers state that audible range is one of the most important concerns when setting up sound source roll-off. In an interactive application where sound is an

essential cue for the user, it is often critical that a given sound is audible, or inaudible, at certain points in the virtual world. This is why ISACT offers the *Max* property for Audio Samples.

Connected to the *Max* property is the *Level* setting. The idea is that when the distance between the sound emitter and the 3D listener is *Max* units, the sound will be heard at an intensity determined by *Level*. So if *Level* is a very high level of attenuation, say  $-60\text{db}$ , then when the distance reaches *Max*, the sound will be close to inaudible.

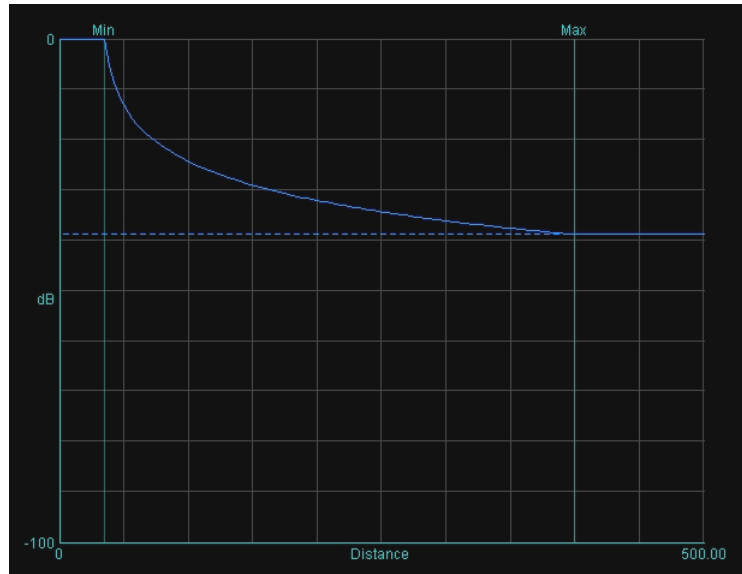


Subsequently, no matter how further than *Max* the sound is, it will never become more attenuated than the intensity described by *Level*.

There is one more value available, which links all the 3D properties together. This is *Roll-off*. The *Roll-off* value determines how steep will be the graph of attenuation between the sound at *min* distance with no attenuation, and the sound at *max* distance with *level* attenuation. With *Roll-off* set to the default value, 1.0, the sound halves in intensity each time the distance doubles. *Roll-off* is a multiplier, so a value of 2.0 makes the effect twice as strong, giving a quarter the intensity each time the distance doubles. *Roll-off* cannot be more than 10.0.

You may by now have noticed that these settings are inter-dependant. If the designer defines a set of values for *Min*, *Max* and *Level*, then a setting for *Roll-off* is implied. Likewise, if you set *Min*, *Level* and *Roll-off* then the *Max* distance must be constant. It might seem a little confusing, but fortunately ISACT has a very clear interface to help you get your distance models just how you want them.

Visualising Roll-off settings with the 3D Roll-off Graph  
Check out the 3D Roll-off graph in the 3D Buffer Controller dialog.



The 3D Roll-off Graph indicates just how the sound will be attenuated according to distance. You can see the lines denoting *Min* and *Max* distances and the dotted line for *Level*, which defines maximum attenuation. You can drag these markers around the graph – watch how the values in the dialog box change according to your settings.

#### Setting Audio Sample directivity

Sound designers also have the capacity to give sound sources directivity. Consider someone holding a megaphone. If you are standing in front of the speaker, in the path of the amplified sound waves being emitted from the megaphone, you will be blasted with loud sound! But the further you are standing from the sound path, the less amplified sound your ears will receive. Stand behind the speaker, and you probably won't hear much more than if the person were talking un-aided.

3D sound APIs provide developers with the facility to define sound cones, allowing them to specify directivity in 3D. ISACT Audio Samples take advantage of this functionality, and IPS makes it easy to define and monitor sound cones on your sound sources.

#### A brief explanation of Sound Cones

The two sliders at the bottom control the inside and outside cone angles. The inside cone angle defines the region where the sound will be at its loudest. Beyond the outside cone, the sound will be at a lower intensity, defined in the *Outer* field. Additional low-pass filtering can occur when the listener is outside the sound's transmission cone – this is defined by *Outer HF*, and is only available on EAX systems.

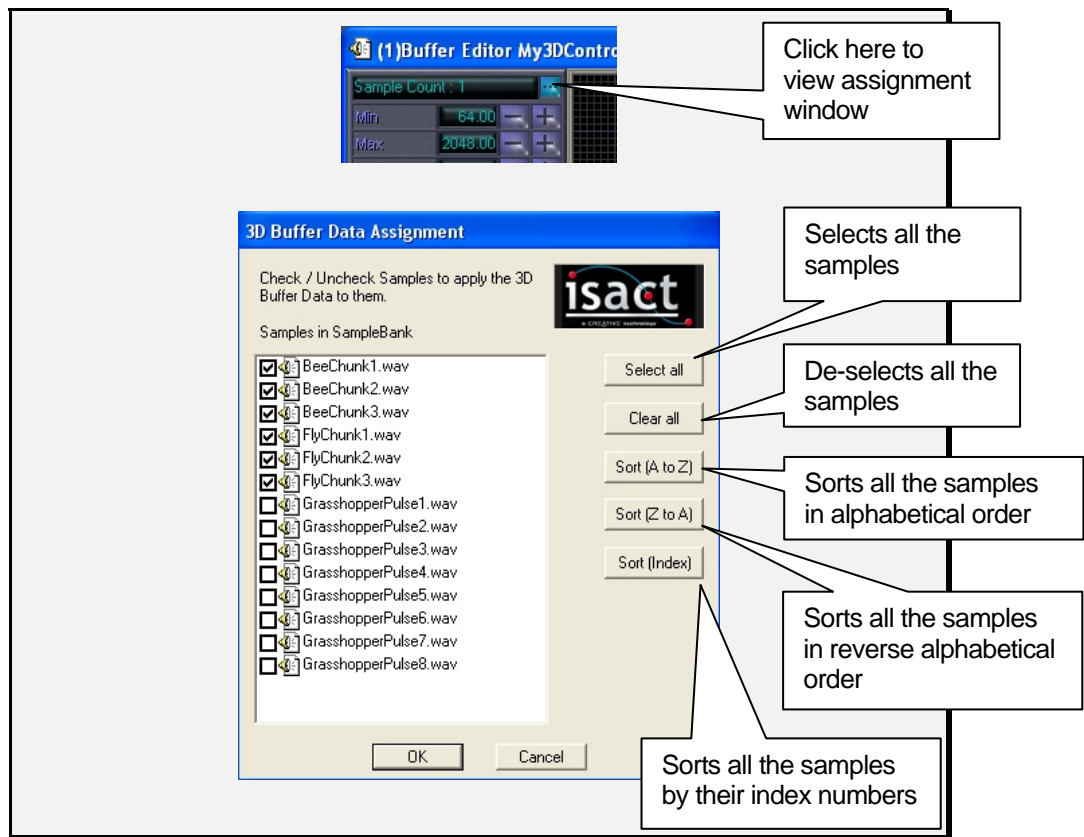


For more information regarding sound directivity, see the OpenAL, DirectSound, OpenAL Effects Extension (EFX) and EAX documentation.

### Applying 3D Controls to Audio Samples

A 3D Buffer Controller can have one or more Audio Samples associated with it. Note that ISACT supports multi-channel sound files, but only mono samples can have 3D controls applied. There are many different ways you can tell IPS to apply a 3D Controller to an Audio Sample. The simplest is to locate one or more Audio Samples within the Project Tree structure, and drag them into an open 3D Buffer Controller dialog window.

However, to fully manage the assignment of Audio Samples, click the blue icon next to the sample count at the top left hand side of the 3D Buffer Control dialog window. This brings up the 3D Buffer Data Assignment dialog window.



This dialog is self-explanatory! It lists all the samples inside the same Sample Bank as the 3D Buffer Controller, and you can flag the samples that you want to be assigned to this 3D Buffer Controller.

Sorting buttons have been included on the right hand side of this dialog to facilitate sample selection: you can sort your samples in alphabetical order, reverse alphabetical order, or by index number. When you have a large number of assets these options can save you lots of time.

Check the Tick and Tock samples.



These samples will now have their 3D Properties set according to this 3D Buffer Controller

Now we have some samples attached to the new 3D Buffer Controller, we can audition them to hear the roll-off in action.

Click the icon to select a Sample to audition



Click here to select Sample to audition

A list of associated samples appears, including Tick.wav and Tock.wav. Choose one of the samples by double-clicking in the list.

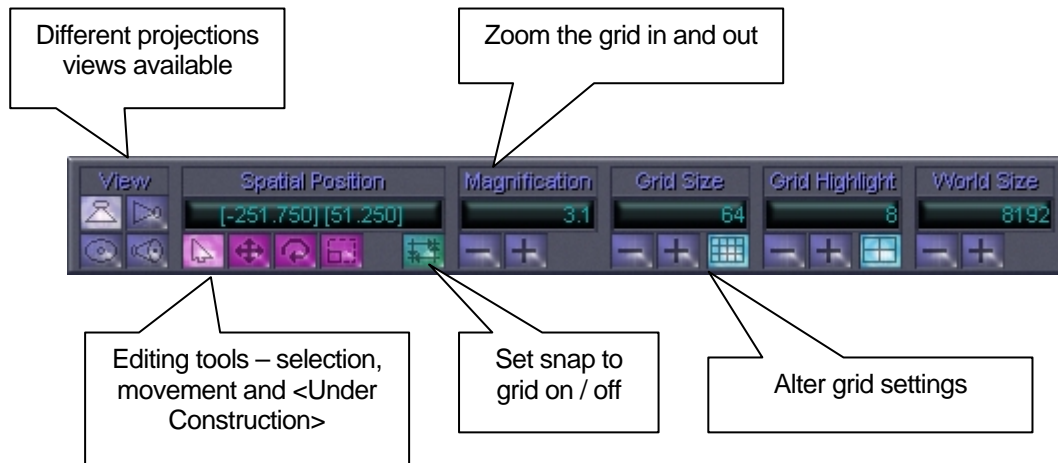
A Node denoting the sound will appear on the 3D Audition map.

#### Auditioning Roll-off settings with the 3D Audition Map

The 3D Buffer Control's audition map is an example of a Spatial Grid view, an interface which IPS uses to display 3D sound positions, and also allows you to move sounds around in 3D space. You can control the appearance and settings of a Spatial Grid view using the Spatial Grid Controls.

To view the spatial grid controls, go to the *View* menu and select *Tool Windows->Spatial Grid Controls*. Keyboard shortcut is *Ctrl-Shift-G*.

Reduce the "World Size" to 1024, and the "Grid Size" to 4.





### Zooming in on the grid

The spatial grid controls allow you complete control over spatial editing. As well as toggling the snap, you can set zoom level with the + and – buttons. Another way to set zoom level is to place the mouse cursor over the graph you are currently editing on, and move the mouse wheel. Moving the wheel up zooms in closer, while down zooms out. You can also work with different axes using the various projection views. If you want to work in the height (y) axis, simply select the appropriate projection, side or front.

### Representation of Audio Sample properties

The Spatial Grid shows a visual representation of the 3D Buffer Control's *Min* and *Max* distance settings, in the form of two concentric circles around the node. The radii of these circles denote the *Min* and *Max* distances.

Set the currently selected Audio Sample to play looping using the Play Looped button on the Transport window. 

Click the Move Node icon on the Spatial Grid Controls. 

The cursor changes to a cross. Pick up and move the node representing the sound on the 3D Audition Map.

Hear how the sound is spatialised as you move it around the listener (represented by the intersection of the two centre lines.) Note also how the sound becomes quieter as the distance between listener and sound increases, according to the graph you defined in the 3D Buffer Control.

A red line marked *Sample Distance* appears on the 3D Rolloff graph, denoting the distance between listener and sound source. The line intersects with the graph you defined, showing you how much attenuation is currently applied.

### Editing 3D Buffer Controller properties while playing content

If you are working on a piece of content and need to adjust the 3D Buffer Controller data for a sample, you can bring up the Buffer Tool Window. This Tool Window can be

launched using *View -> Tool Windows -> Buffer Controls* or by pressing *Ctrl-Shift-B*. This Tool Window is a cut-down version (no auditioning or attenuation graph) of the 3D Buffer Controller edit window but as it is a Tool Window rather than an Edit Window you can use this to adjust parameters *while* still playing content in the active Edit Window. Simply select the appropriate 3D Buffer Controller you wish to edit by pressing the “...” button in the top-right corner, and then adjust the parameters and hear the immediate effect on your playing content.



## Conclusion

---

Audio Samples each contain a buffer of sound data. You can carry out destructive editing of Audio Samples, and undo changes if necessary.

A complete set of 3D Properties is stored in a 3D Buffer Controller. Associate one or more Audio Samples with a 3D Buffer Controller and the properties you set will apply to those sounds. 3D Buffer Controllers provide a visual interface for setting and previewing 3D properties.

## Random dynamics: Sound Events



ISACT has a complete system for playing back dynamic audio in response to a single event or trigger. The Sound Event is the key component that allows you to unlock this mechanism. Use Sound Events and you'll never hear repetitive identical footstep sounds in your game again!

### Getting started with Sound Events

---

Sound Events revolve around providing ISACT with multiple alternative Audio Samples for a given trigger. So this is a good example of higher-level content referencing a basic component.

Several properties can be controlled on each of the alternative Audio Samples. These properties can themselves be restricted, or randomised within a range of settings. Its no problem to include the same Audio Sample multiple times in a Sound Event, each time with different parameter settings - the sample data is not copied multiple times, just a reference to the Audio Sample. So this is a valid method if for example you want alternatives to be composed of the same sample played back at defined pitches.

The Audio Sample can be chosen at random or in order. Silence can be specified as an alternative in both ordered and randomly chosen events. In random events, different probabilities can be set for each alternative Audio Sample. In ordered events, all the alternatives including the 'silence tracks' can be given a specified order.

Once you start to get your head around Sound Events, you quickly realise that they open up a myriad of possibilities! Rather than try to exhaustively list the options open to you, let's look at a few examples of how Sound Events can be employed to create dynamic audio.

### Sound Events Example one – the clock is ticking...!

---

This example shows a trivial application of ordered Sound Events. It also usefully demonstrates how looped and one-shot sound events take advantage of ordered selection.

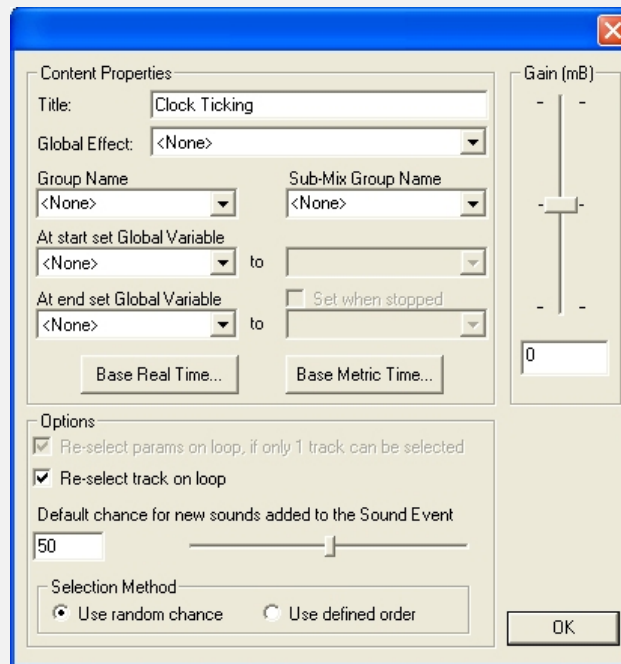
Consider the ticking of a clock. Our clock is of the old-fashioned mechanical type, and the pendulum makes a different sound depending on which way it swings. On the left-to-right swing, it makes a 'tick' and on the return swing, a 'tock'! Really, this example is rather frivolous, as you could easily cater for this scenario with simple sound sample looped, but bear with us – it will show you some important functions of ordered sound events!

#### Adding Audio to a Sound Event

First off, time to add a new Sound Event to your project in progress. Sound events are a high level content object and as such will be stored in a Content Bank. In order to insert a Sound Event to the project, you must first create a new Content Bank:

Click the *Content Banks* tab and right-click the *Content Banks* folder. Select “New Bank...” and give it a sensible name. Now, right-click your newly created content bank, and select “New...”. This time choose *Sound Event* from the menu presented to you

A dialog will appear with some basic characteristics for your event. Set the name to something sensible like ‘Clock Ticking’.



This Sound Event will be based around ordered selection, so select ‘Use defined order’.

(Note there’s a bunch of other settings on that dialog, like “Gain(mB)” and “Base Metric Time...” – we’ll talk about those a little later in the tutorial)

An empty Sound Event dialog window will be displayed.

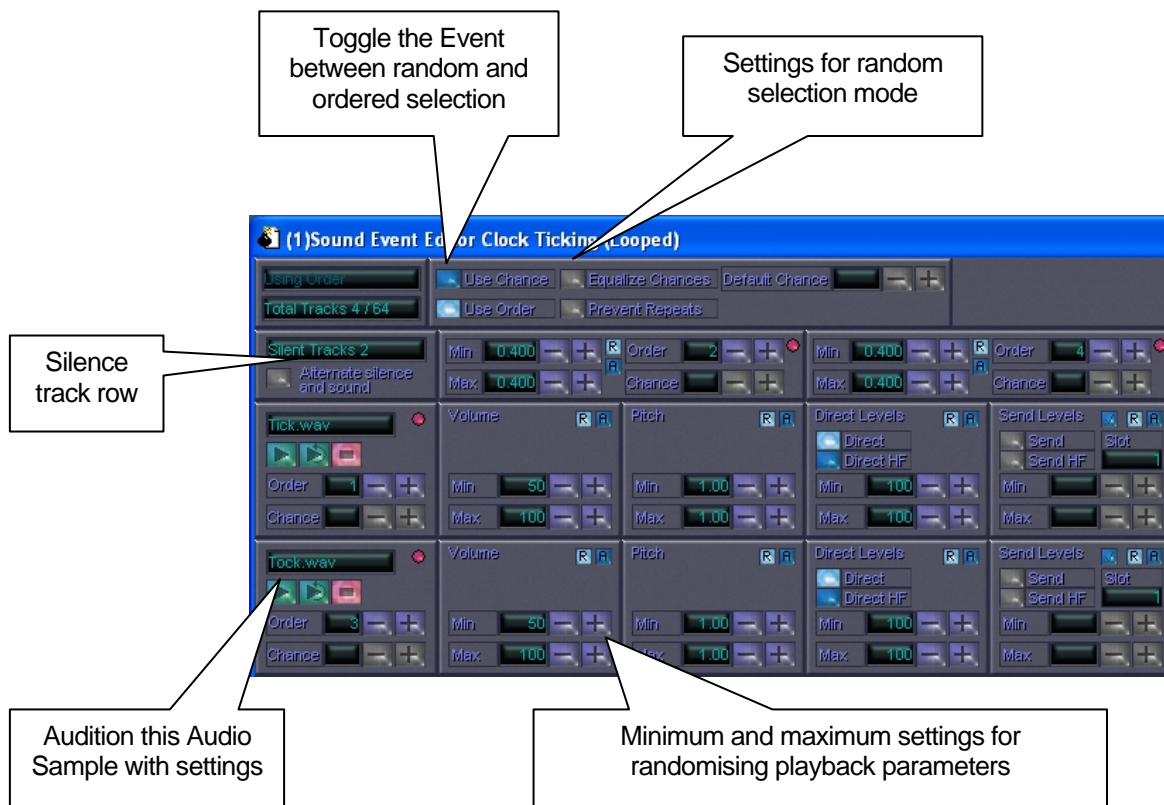
Let’s add some Audio Samples to the Sound Event, so we can start making some noise. The Sound Event dialog window supports drag and drop.

Click the *Sample Banks* tab, grab the Tick and Tock audio samples from your sample bank, and pull them into the Sound Event dialog window.

### Auditioning a Sound Event

Check out how for each audio sample in the Sound Event, a row of information and parameters appears. At the start of the row is a miniature transport, with buttons for Play,

Play looped, and Stop. This allows you to audition this alternative in the Sound Event. The main playback transport window allows you to audition the entire Event.



Test the 'Tick' and 'Tock' alternatives by clicking their Play buttons. Now click the Play button on the main playback transport, a couple of times in succession. Note how when you do this, the event alternates between playing the 'Tick' and the 'Tock' – perfect! As you can see, we marked this Sound Event to use ordered selection. Were you to change your mind, you could toggle between ordered and random selection with 'Use Chance' and 'Use Order', at the top of the dialog.

#### Changing order

Notice how your alternative Audio Samples each have a cell marked 'Order'. At the moment, the Audio Sample you dragged in first will be ordered '1', with the second ordered ... '2'! You can change the order of one alternative and any other alternatives will have their order setting automatically altered to compensate.

### Changing chance

If your Sound Event is in Random mode though, the 'Order' box is disabled, and if you de-select the '*Equalize chances*' option, instead you can set 'Chance'. This allows you to make certain alternatives more likely to be chosen than others. Note that chance is expressed in percentage, so the total chance for all alternatives cannot exceed 100. If the total chance is less than 100, then the remainder represents the percentage chance of no sound playing when the Event is triggered.

### Adding random dynamics to a Sound Event

Now you might like to add some extra dynamics. This being a vintage clock, the 'Ticks' and 'Tocks' probably don't always sound with the same loudness. Examine the row of properties for each Audio Sample. You'll notice that the first settings panel is marked 'Volume'. There are controls for both minimum and maximum. Currently, for each alternative Audio Sample these values are both set to 100 meaning that each is clamped to full volume.

To allow a small variation in the sounds' volume, change the min volume settings on both Tick and Tock Audio Samples to around 50.

If you set these min and max values differently, then ISACT will chose a random setting in between the two values each time that particular alternative is played. You're not just restricted to randomising volume. Look along the row of parameters – you can also apply random dynamics to sample playback pitch, and on systems with EAX support, you can also modulate the amount of dry or wet sound when the Sound Event is feeding effect processing slots.

### Managing Audio Samples in your Sound Event

Note that Sound Tracks in your Sound Event dialog can be highlighted by clicking on some blank space in their row with the mouse. Multiple selections are possible using the shift and control keys. When you highlight one or more Sound Tracks, the right mouse button brings up a context menu, allowing you to delete, copy, cut, paste, etc... the selected alternatives. This allows you to move a track or set of tracks, complete with settings, between different Sound Events in your project.

A very useful feature on the Sound Event context menu is the 'paste custom...' option:

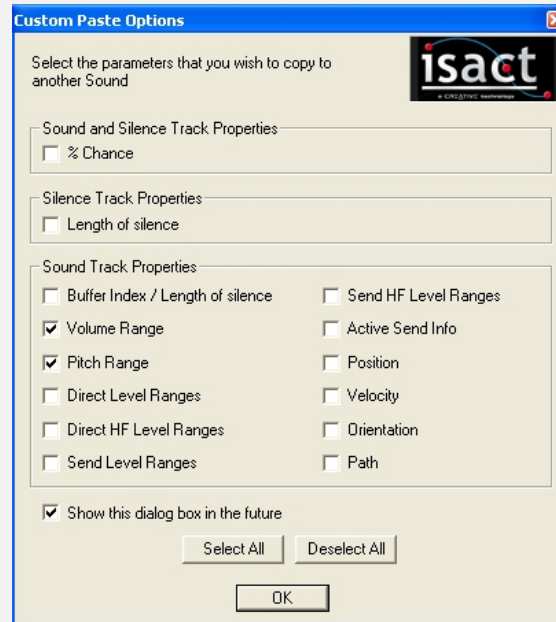
Let's say you have a lot of Sound Tracks in your Sound Event, and you want to assign the same volume and pitch settings to each one. You could edit the settings for every track in the Sound Event Editor window, but this would quickly become laborious. A much quicker way would be to use the 'paste custom...' option as follows:

Create a new Sound Event called 'Custom Paste'. Right click your Sample Bank to add more samples, and add all the samples from the 'Tutorial/Waves' directory.

Drag all these samples into your Sound Event, and alter the first track's volume and pitch settings.

Whilst still in the Sound Event Editor window, select the track by clicking somewhere along the row, and then right click to display the context-sensitive menu, and select 'Copy'.

Now multi-select all the other tracks (using the shift key), right click and select the 'paste custom' option. This will bring up the Custom Paste Options dialogue:



Click the 'deselect all' button and then click the 'Volume Range' and 'Pitch Range' check boxes and then select OK.

The volume and pitch settings will then be copied from the first track to all the tracks you have multi-selected. This method can be used for any of the Sound Track Properties in the Custom Paste Options Dialog and is a great time saver!

The right-click context menu in the Sound Event Editor window also allows you choose to edit the selected sample without having to try and find it amongst all the samples you have loaded into you Sample Banks. This is a great time saver when you have loaded thousands of samples and need to edit a particular sample being used by your Sound Event.

Click on a sound track within your Sound Event. Right-click on it to bring up the context menu, then select "Edit Selected Sample(s)". You will now be able to edit the sample using ISACT's Wave editing facilities.

It's worth noting that all of ISACT's Content Objects have a right-click context sensitive menu, and you can use the same method as above to edit their lower level components too.



Back to the ticks and the tocks...

### Looped Sound Events

The ticking Sound Event we created is all very well if the programmer's code continually triggers clock-ticking noises at regular intervals. However, in some applications a looped event might work better – the programmer might want to just kick off the event, flag it as 'looping', and expect the 'Ticks' and 'Tocks' to keep coming along at the right time and in the right order. Well, this is an equally valid technique in ISACT. So now let's set up an alternative clock Sound Event that works on a looping principle.

Make a new Sound Event, and follow the instructions as before to create a version with identical settings.

Click the 'Play Looped' button on the main Sound Event transport at the top of the dialog.

Everything sounds pretty good, except our clock seems to be running a little fast. We can introduce some quiet time into the event using ISACT's silence tracks.

Before we do that, there are a couple more options that give more control over the looping behaviour of a Sound Event. In the Sound Event's Properties Dialogue Box (accessed by right-clicking on the Sound Event in the Project Tree view and selecting Properties ...) there are two check-box options. "Re-select track on loop" (checked by default) tells the playback engine whether to re-select a new sample when the Sound Event is looped. De-selecting this option will essentially loop the sample (or silence track) that was selected on the first loop iteration. Similarly, the "Re-select track parameters on loop" (also checked by default) tells the playback engine whether to re-select random parameters when the Sound Event loops. For our ticking clock we want to have both options selected, but when modeling other complex sounds, for example a charging weapon sound, it may be preferable to lock the choice of sample and parameter values while the Sound Event is being looped.

### Using silence tracks

The first row in the Sound Event dialog is for the silence tracks. Silence tracks allow Sound Events to sometimes respond with a period of silence. Probably not much use for consecutively triggered events like the first example of the clock tick, but crucial for our looped version.

Right-click the blank area of the Sound Event dialog next to the silence track control panel.

The context menu that appears gives you the option to "Add Silence Track".

Insert two silence tracks.

You can see a new panel appears in the Sound Event dialog for each silence track. On these panels, you can adjust order, just as for each Sound Track. Plus you can define a minimum and maximum duration for each silence track.

Set the order values for the two new silence tracks to two and four

Double-click to set the minimum and maximum duration values on both silence tracks to 335 milliseconds. This will clamp the duration at 335ms.

Now you can audition the whole event with the Play Looped button. As each 'Tick' and 'Tock' event takes approximately 165 milliseconds, and the silence tracks are each 335 milliseconds, we now have an ideal looped sound event, with the 'Tick' and 'Tock' evenly spread over one second.

## Other Sound Event examples

---

Included with the IPS tutorial package is a selection of other sample files containing Sound Events.

Counting.sac

This project contains several Sound Events, each demonstrating a different aspect of event selection. The names are self-explanatory!

### **Example 1 – Random Samples**

Chooses a sample at random from the list of alternatives

### **Example 2 – Random Pitch**

A single sample is used with the pitch randomised between 0.8 and 1.2

### **Example 3 – Random Samples, Pitch, Position**

The sample is picked from a limited list, with randomised pitch. While position cannot itself be randomised, each sample appears three times in the list. Every instance has different positional settings.

### **Example 4 – Ordered Samples**

Three samples played back in a set order

### **Example 5 – Ordered Pitch**

Five instances of the same sample are present, each with pitch clamped to a certain setting. The alternatives are played in order

### **Example 6 – Random Samples with Silence**

This example should be played back looped. Samples are chosen at random, alternated with periods of silence with random duration between 0.1 and 1.5 seconds.

### **Example 7 – Ordered Samples with Clamped Silence**

This example should be played back looped. Samples are selected in order, as with example 4. Samples alternate with silence, with duration clamped at half a second.

### Grasshopper.sac

This project contains a Sound Event called 'Grasshopper'. It makes a random selection between seven different short Audio Samples containing a single grasshopper chirp. Also present is a silence track, with a low probability and a wide range of possible duration. When played back looping, the Sound Event generates continuous grasshopper chirping, interspersed with occasional small periods of silence. Randomisation here makes for almost infinite combinations of sound, with no repetition of phrasing.

### Conclusion

---

Sound Events allow you to introduce a degree of randomness to your soundtrack. You can provide a list of alternative samples, or randomly adjust playback parameters, so that each time a Sound Event is triggered you hear a different effect. Instead of random selection, you can set up samples to play in a set order. You can introduce silence, and play Events with a looping flag. This allows you to produce interesting textural sounds, or granular effects. You can use the right click context sensitive menu to save a lot of time when building up and editing Sound Event content. Sound Events are very flexible, and form the basis of some more sophisticated content as you'll see further along.

## Defining motion: Spatial Paths



ISACT is all about 3D spatialised audio. But up to now, we've not really seen a way to make sound move around in 3D space. That's all about to change!

Spatial Paths allow the designer to create 3D movement maps, which can then be applied to Audio Samples as a part of higher level content like Sound Events. In a traditional 3D application, typically all sound emitters are either tied to actors in the virtual universe, or static either in 3D space or in relation to the listener. The position of a sound source therefore either remains constant or is altered whenever the actor moves.

However, ISACT content using spatial paths allows you to break free from this convention. A sound source can have arbitrary movement applied to it, without the need to create and control a corresponding object within the game universe.

### Applications for Spatial Paths

For instance, your game world may feature a forest full of swooping birds, or a swamp full of buzzing mosquitoes. The sound of the birds or insects circling around the listener's head is essential to convey the atmosphere of this environment. But actually managing say twenty bird actors, or thirty insect entities to realistically buzz around in the game might require more resource than you are prepared to dedicate for this purpose, especially since the visual display of these actors is probably not as critical as their aural presence.

You could choose to create a stream of ambient sound in stereo or 5.1. This would probably be played back as a repeating loop, which could soon get...well...repetitive!

Alternatively, you could use IPS to create a number of Spatial Paths. Gather or record a number of short audio samples for birdcall, or insect buzzing. Sound Events allow you to specify a Spatial Path to be applied to each Sound Track. Of course, combining just three Audio Samples and three Spatial Paths gives you up to nine alternative sounds. Add some variability in terms of pitch, and the possibilities start to multiply further...

Simultaneously play back a few instances of this Sound Event in your game, and immediately you have an ever-changing and extremely immersive ambient soundtrack.

The beauty of this type of ISACT content is that you can make efficient use of a small pool of audio samples with a variety of different parameter manipulations. ISACT's control data such as sound paths, event descriptions etc... are of minimal size in comparison to sample data. So you are likely to make a big saving in terms of memory footprint!

Furthermore, all the 3D panning is taken care of by the sound system's interactive spatialised audio renderer. Therefore your surround sound ambience created using a number of Sound Events will be 'compatible' with all different speaker combinations from simple stereo to 7.1. Plus hardware-accelerated effects, generally restricted to 3D sound source only, can be applied. There's no good reason not to be using techniques like this!

Let's go through the process of making Spatial Paths and applying them to Sound Tracks within Sound Events.

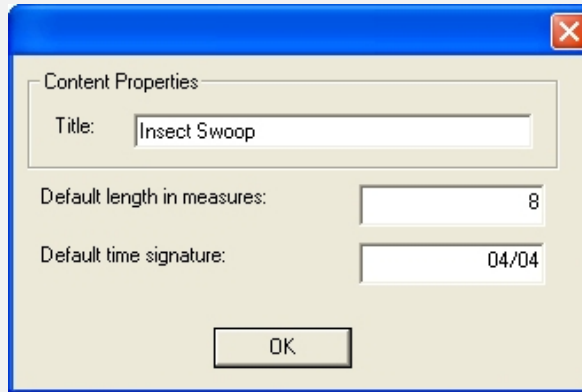
### Join the dots... creating a Spatial Path

---

As with all other ISACT Content Objects, Spatial Paths are stored in Content Banks

Right-click your Content Bank, and select "New...". This time select "Controller Object" from the list box, and choose a "Spatial Path".

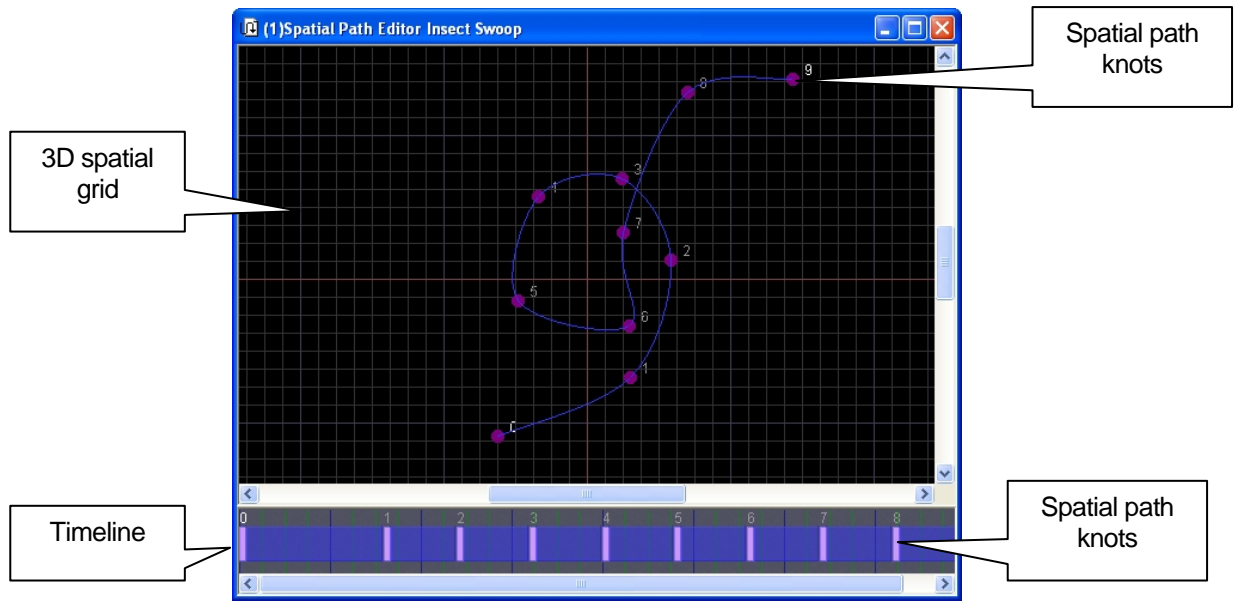
A dialog will appear with some basic characteristics for your Spatial Path. Set the name to something sensible like 'Insect Swoop'.



A spatial path consists of a sequence of 'knots', spread over both time and 3D space - in some senses, Spatial Paths are four-dimensional! Spatial Paths are stored merely as a list of 3D points, with their relative timings.

As you can see from the dialog, you can specify a spatial path's length in measures. When a Spatial Path is used with a Sound Event, its length is not particularly critical as it is automatically scaled to match the length of the sample. When a Spatial Path used in conjunction with a Metric Time Audio Sequence, the length in measures of the Spatial Path and its time-signature is important as these features allow you to move around the path in music time. The use of Spatial Paths in Metric Time Audio Sequences will be discussed later.



When you've created your Spatial Path you will be presented with a new window for path editing.



Double-click on the grid to add a new knot to the path.

You will most likely find that each new knot ‘snaps’ to the grid settings. If you want to work in a finer resolution, you can adjust the grid settings and even turn off the snapping using the Spatial Grid Controls, as described earlier.

### Editing a Spatial Path

Another useful setting is the cursor function. The default cursor  allows you to place new knots. To move knots around you can select a movement cursor . Both cursors allow the selection of one or more existing knots for further editing. Multiple selection can be achieved, either by click-dragging a box over a portion of the graph, or with the control / shift keys. Selected knots can also be deleted.

You’ll notice that each time you add a new knot it is connected to the previous knot by a spline curve, coloured blue. The overall spatial path will follow the route made by the complete set of curves. However, if you don’t want the path to follow curves, there are alternative ways to join up your knots. Right-click a knot or selection of knots. A context menu appears offering different ways to join the knots. Linear is the simplest – a straight line is drawn in purple between the selected point and its subsequent neighbour (single selection) or all selected points (multiple selection). Clockwise and anti-clockwise (counter-clockwise in US terminology) circle segments can be used.

The context menu also shows an option to Loop a path. With this option ticked, a final link will appear in the path, leading from the last node to the first.

### Editing in the time domain

Look along the bottom of the Spatial Path dialog, you’ll see a separate graph showing how the knots are distributed over the duration of the Spatial Path.

You may need to zoom out to see the full duration – as with the positional graph, you can zoom in or out using the mouse wheel. The Time Grid Controls tool window allows you greater control over the view of data in the time domain.

To view the Time Grid Controls, go to the *View* menu and select *Tool Windows->Time Grid Controls*, or *Ctrl-Shift-R*.



To move a knot in time, select it by clicking on it (again, multiple selection is possible). Drag it to the new position, but remember you cannot move a knot forwards or backwards beyond its neighbours.

When new knots are added to the end of a path, IPS tries to keep the tempo of the current knots as consistent as possible, by adding the new know between the last knot and the end of the time graph.

You can impose regular timing on the knots by right-clicking in the background of the time graph and selecting 'Regular time interval'. If you want the speed of movement around the different knots to be regular notwithstanding the different distances between knots, then chose 'Constant speed movement' and their timings will be arranged accordingly.

## Sound Events Example two – flying bugs

---

As described earlier in this section, an ideal use for a Spatial Path is to add motion to an Audio Sample that forms part of a Sound Event.

Load wave files fly.wav and bee.wav into their own Audio Samples.

Create a 3D Buffer Controller, to manage the roll-off on the sounds.

Adjust the Controller's 'Min' parameter, setting it fairly large so that the buffers will not roll-off and become inaudible when they are distant from the listener. Then apply the new 3D Buffer Controller to both Audio Samples.

Create a new random chance Sound Event, and drag each buffer into it twice. Your event should consist of four randomly selected Audio Samples, two flies and two bees.

Create two new Spatial Paths. Name them 'Insect Path 1' and 'Insect Path 2'.

Insects tend to fly around fairly randomly, so place six or seven knots around the centre of each path.

In the Sound Event dialog, look along the information row for the Sound Tracks, until you spot the panel called 'Position'. In this panel, you can either specify a set of 3D co-ordinates, or a spatial path.



Drag Insect Path 1 from your Content Bank to the Path box of the Position panel for the first fly.wav row, and again for the first bee.wav row.

Drag Insect Path 2 from your Content Bank to the Path box of the Position panel for the second fly.wav row, and again for the second bee.wav row.

Add some pitch variation to each alternative Audio Sample in the Sound Event.

Now when you audition the Event, flies and bees at different pitches will fly around your head according to a randomly selected path. Of course, with only two Audio Samples and two Spatial Paths, the combinations are a little limited. So try making some more paths, and working with more samples. If you wish to remove a Spatial Path from a Sound Track, press the blue button in the top-right corner of the 'Position' data. Imagine the possibilities of layering multiple Sound Events.



## Managing your assets: Indices, Banks and Folders

The efficient content creator knows that to avoid confusion, it is essential to properly name, categorise and structure assets. ISACT offers a number of different facilities to facilitate effective asset management.

### Background to ISACT asset management

---

As you know, ISACT content is arranged into projects, banks and folders. An ISACT project file is given the extension “.SAC”. It is entirely possible to hold all your assets, say for an entire game or a whole level, in one project file. However, an interactive application with a large dataset will often require dynamic caching of memory-hungry data such as audio samples. ISACT’s asset management options enable you to store all your audio assets in an orderly way, and also facilitate more efficient data caching at run-time. However, in certain situations, changing the arrangement of your data can ‘break’ your content. Read this section carefully, it will help you in achieving the optimal data configuration whilst avoiding some severe head-scratching sessions when stuff stops working right!

#### Referencing data by index

First of all, it is important to note that ISACT references all content by index. Components that refer to other components rely on those indices remaining constant.

For instance, when an Audio Sample is added to a Sound Event, the Sound Event actually stores the Sample’s index – say, “*Audio Sample index 3*”. If the Sample with index 3 is subsequently removed, then the Sound Event refers to an invalid index – if the ISACT logic requires that this invalid index be played, nothing will be heard!

If a new Audio Sample is imported into the project at Audio Sample index 3, then the new sample will replace the old sample anywhere it is referenced. This is how the IPS ‘Replace Audio Sample’ functionality operates.

This doesn’t just apply to Audio Samples. Each content type has its own index system. The same arrangement happens where a Sound Event references some control data such as a Spatial Path – except here it will refer to ‘*Spatial Path index 2*’ or whatever the index number may be!

ISACT automatically puts new content at the lowest index possible for each object type. But as you’ll see later, it is possible to get into the guts of your ISACT project, and manage different objects’ indices for yourself.

#### Referencing between Banks

Pay attention to this stuff – it could save you lots of trouble! Content can refer to Audio Samples from any sample bank in the current project. But Content can only refer to other non-Audio Sample content such as Sound Events, Spatial Paths, etc... from the same Content Bank in which it resides itself.

For instance, you can’t use a Spatial Path from Content Bank ‘A’ in a Sound Event residing in a separate Content Bank ‘B’, even if they both feature in the same project. This rule even applies to content within the main Project Tree. The IPS interfaces always prevents you from completing invalid operations like this.

## Sample Bank Indexing

As stated above, cross-bank referencing is allowed when content is referring to Audio Samples stored in Sample Banks. How is this achieved?

Under the 'Sample Bank' tab in the Project, the different Sample Banks in the project are listed, along with their Bank Indices. When Audio samples are referenced by content, they are actually identified by two numbers – their bank index *and* their sample index. Audio Samples in the main project file have the bank index 0.

You can alter a Sample Bank's index number – right click on the bank's header in the project tree and select 'Properties'. As well as displaying some details on the bank, the dialog allows you to select a new bank index. If you select an index that is already in use, you will be prompted to confirm when you click Ok. If you do decide to go ahead, then the Sample Bank previously stored at the selected index will be removed from the project and replaced by the current bank. However, the data will still be intact within the bank file, and as you'll see later, bank files can be imported into any project.

## Bank Editing

Here's where you find out how to edit Sample and Content banks at the lowest level!

If you double click on a bank's heading in the Project Tree, or right-click the heading and select "Edit", you will bring up the Bank Editor dialog for that bank. If you are editing a Content Bank, you'll notice a drop-down list at the top of the dialog. This allows you to select the Content type to view and manipulate.

With Content Banks, there are only two columns – the index and the content name. When editing a Sample Bank, you will see several columns, showing you information about the different Audio Samples. This includes format information like frequency, bit resolution and number of channels, as well as size, name and original filename. At the bottom of a Sample Bank Editor dialog, you can see the total number of samples and the total memory used in bytes. The status bar on the Content Bank editor shows the total number of objects present of the currently selected type.

Within the Bank Editor Dialog, it is possible to move an object or a group of objects from one index or index range (Bank Editors support standard multiple selection using the Ctrl and Shift keys) to another by dragging and dropping.

If your proposed operation would result in another object being over-written, then you are prompted to confirm. You should note however that if your operation might cause another component to become invalid because it now references an empty index, you are not warned.

If you right click an object within an editor dialog, you will see a context window offering various operations. These include viewing a property dialog, creating new content of the selected type (or loading wave data in the case of the Sample Bank editor), plus copy, cut and paste functionality.

Due to interface restrictions you cannot actually drag a component or an Audio Sample between edit windows. But using drag and drop in the Project Tree view or the cut-and-paste functionality it is possible to move a component or range of components from one bank to another. Again, where a component references another component and the operation results in the two components residing in separate banks and a reference becoming invalid, you will not be warned.

## Dealing with invalid references

So what happens when a reference becomes invalid, i.e. an Audio Sample or other object, which is referenced from another component, is erased? Although IPS doesn't display a warning dialog, it is typically made clear when viewing the interface slot where the valid reference should be displayed. For instance, when a Sound Event references an Audio Sample that has been moved, the Bank and Sample index for the 'hanging reference' is shown in red.

To remedy the situation, you should either introduce some content at the empty index, or point the broken object towards some other valid content.

## Importing Banks

If you made a Content or Sample Bank in one project, and decide you want to use the content in another, then you can import it. Open the 'File' menu in IPS' main menu bar, select 'Import', and chose your bank type from the list. Or alternatively right-click the header in the Sample Bank or Content Bank tab within the Project Tree, and select Import Bank.

**BEWARE!** Importing a bank into another project actually creates a reference to the bank in the new project, it does not copy all the bank's data to a new file. If a bank file is referenced by multiple projects then you could end up making changes in one project that break another project. It is also possible for a bank to be referenced twice in the same project, which could also lead to heart-ache and potential ruined work. Be very careful when accessing the same bank multiple times

## Using Folders

---

ISACT's project tree and sample / content banks allow nested folders, in which you can place your content. We should re-iterate here that ISACT references components by index, not by folder. So within a given Project Tree or a sample / content bank, it is quite safe to move your components from one folder to another – it won't change the components' indices, and therefore it won't effect the cross-referencing.

### Adding a new folder

Let's add a folder to a Content Bank. To create a new folder within a Content Bank, click the 'Content Bank' tab and right click on a Content Bank. A context sensitive menu will appear offering some different options including the option to add content or create a new folder.

### Placing content in folders

You can now simply drag and drop components around from folder to folder. There's no restriction on what content goes in which folder. You may wish to keep all Sound Events together in one folder. Or you may decide to place related content of different types together. Folders can be nested, for detailed categorisation!

Folders are a mechanism for categorization of components – they help you to achieve an un-cluttered interface to your data. You can expand and collapse folder structures by clicking the "+" or "-" icons next to their names, in the standard Windows manner.

### Additional Features of folders

Some tools such as the Sting Control and the Content Mixer allow you to drag folders of components into them thus simplifying the process of component selection.

Also, right-clicking on a folder brings up a menu with an option called “Folder Content’s Properties”. Selecting this opens a Multi-Properties dialogue box for all the Content in the Folder (and sub-folders). The Multi-Properties dialogue box, Sting Control and Content Mixer will all be discussed later in the document.

## Memory management with Banks

---

If your content is organised into separate banks, it will make your coding team’s memory management tasks much easier. This is particularly relevant with Sample Banks, as sample data can quickly use up great swathes of memory!

As a general rule for efficient sound caching in a location-based application, samples that will be heard in each particular zone or area of an interactive world might be stored in their own bank. That way, the application can be programmed to load in the most appropriate groups of samples according to the position of the player. Sounds which will be heard independent of the current surroundings, such as interface sounds or those associated with the user’s avatar, are likely to be used throughout a gaming session. These might be stored in a separate bank, which may be loaded at the start of the session and never unloaded.

More sophisticated policies will undoubtedly be employed, but a caching system is likely to be based around these principles.

## Bank and Index Example 1: Losing Count!

---

In case you found the bank / indexing topic a little confusing, you can figure it all out with by working through this example

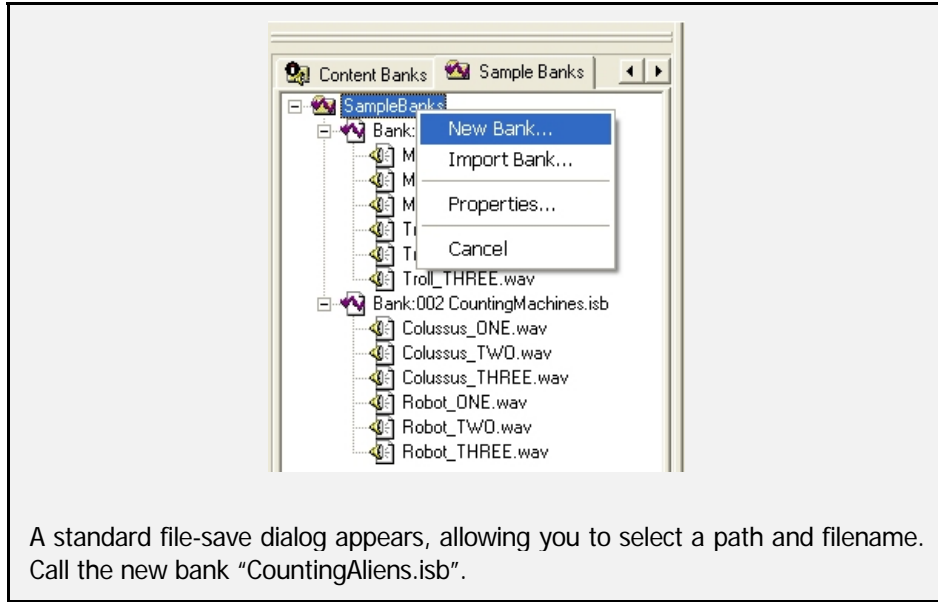
Load in the example Project File called “Counting.sac”, from the Bank Example sub-folder.

Browse through the sample and content banks.

The content might look a little familiar – it’s that set of counting Sound Events again, but this time the samples have been split into two different Sample banks.

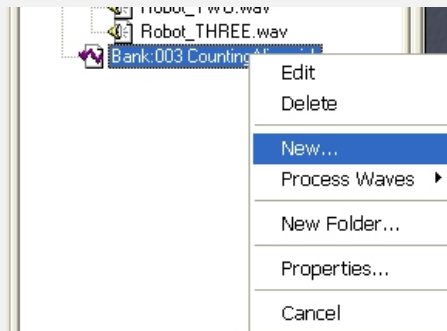
First of all, let’s create a new bank and load some more Audio Samples into it.

Select the Sample Bank tab, right click the head node and select “New Bank...”



The new bank will appear in the Sample Bank list as "Bank:003 Counting Aliens". Now its time to load some alien voice samples into the new bank!

Right-click on the Bank header for your new bank, and select "New..." from the context menu.



Select "Audio Sample" – it's a Content Object.

You will see the familiar interface for loading samples. Go to the Tutorial\Waves folder and locate the files: "Martian\_ONE.wav", "Martian\_TWO.wav" and "Martian\_THREE.wav"

You can take advantage of multiple-selection to load all three in one hit if you like.

Your new Sample Bank now contains three samples. If you double-click the Bank header node or right-click and select Edit, you can see that IPS has put the new three samples to the top of the bank, with indices 0, 1 and 2. Double-clicking within each sample's row in the sheet will play the sample back.

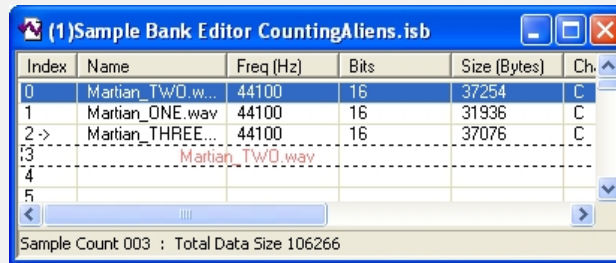
Depending on how the file selection worked, you might well find that the samples are not actually present in the order you might expect – i.e. Martian\_ONE first, followed by Martian\_TWO and then Martian\_THREE.

In any case, this is a good opportunity to test out how to change a component's index.

Make certain that the Bank Editor screen is displayed for the new bank – remember you can bring up this dialog by double clicking a Bank's header node in the Bank List

Notice that Samples are indexed from zero.

Grab an Audio Sample and move it up or down in the list.



Notice how the Audio Sample that you are dragging is displayed in red. The index slot that it is hovering over is highlighted with a dashed box.

When you release the mouse button, the Audio Sample will be moved to the new index. If that index already holds an Audio Sample, then you will be prompted to overwrite the current Audio Sample. If you choose to overwrite, then the current sample will be deleted from the project.

For now, just drag the Audio Samples around so that they are in the correct order – you can use Index Slot 4 as a temporary buffer while you shuffle the Audio Samples around.

If you accidentally erase one of the Audio Samples, then simply right-click the slot where it should have been, and select “Load...”. Find the sample and bring it back into the Bank.

Now let’s see exactly how Content references other content by index, how you can use this to your advantage, and how you might get into a confusing situation if you’re not careful!

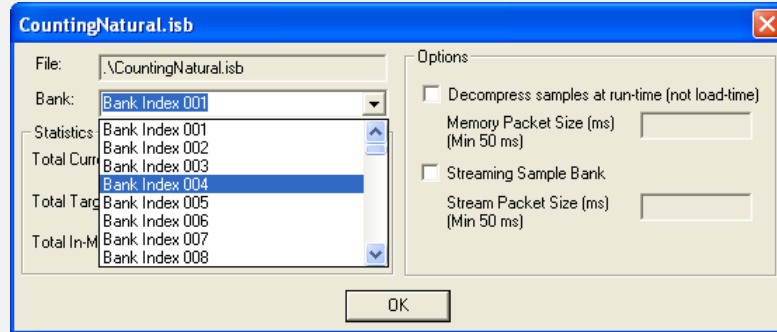
Click the Content Banks tab.

Open the Sound Event called “RandomCounts”. Play the Event a couple of times – you’ll hear human-like voices randomly saying numbers between one and three!

Now go back to the Sample Bank tab, right-click the node marked “Bank: 001 CountingNatural” and select *Properties*.

In the Properties dialog, alter the bank index for the current bank to 004.

Note that banks are indexed from 001, since the main project bank is constant at index 000.



Now go back to your RandomCounting Sound Event – and look at the interface where the Audio Sample names should be displayed.

Instead of a name, each slot shows in red the bank and sample index pair where the previously selected Audio Sample used to reside.



Go back to the Sample Bank tab, and bring up the properties for another bank – Bank: 002 CountingMachines. Change the bank index number to 001.

You should note that if you attempted to move a bank to an index that is already in use, you would overwrite that bank. Although the data will not be lost, the bank will be removed from the project. ISACT will prompt you to confirm you want to do this.

Play with your Sound Event again, and you'll find that instead of human sounding voices counting, you hear the robotic samples stored in the CountingMachines bank.



This functionality can be useful within the context of an application. The programmer has the ability to load Sound Banks at a chosen Bank Index. So you could create a standard set of content, say Sound Events for footsteps, and several alternative sample banks, holding for example samples of footsteps on different surfaces such as snow, gravel, rock etc... When the player moves from one surface to another, simply load in the appropriate sound bank at a pre-defined bank index. The sounds change, but the playback mechanisms remain constant!

With power comes responsibility! Load the wrong sounds into the wrong Sample Bank and your soundtrack's going to sound pretty strange. So, close co-operation between designer and programmer will be required to ensure a consistent bank indexing policy.

Content Banks offer the same functionality – where content references other content, there is actually a link to the correct index. And if you change a component's index, other content which refers to that component will end up with a 'hanging reference'. The big difference is that with content banks, you cannot refer to objects within different banks.

## Optimising memory usage via streaming, pre-loading and compression

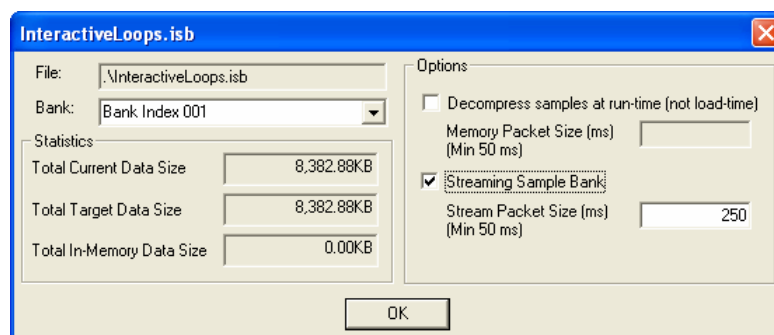
---

You might wish to further optimise run-time memory usage, particularly when authoring for platforms with limited resources. ISACT provides a couple more features to help you to reduce the amount of sound data to be loaded at any one time.

### Streaming

The technique of streaming is a very effective way to reduce memory footprint. When a sound is streamed, the audio data is read from the disk and played back seamlessly in a series of small, sequential chunks. At any one time, only a fraction of the entire file is in main store. This makes the technique particularly suitable for dealing with long sound samples, although there is a pay-off in terms of increased disk access and associated CPU usage.

In ISACT, you can specify audio data to be streamed on a sound bank basis. To declare a sound bank as streamed, right-click on the bank's entry in the Project Tree, and view it's Properties.

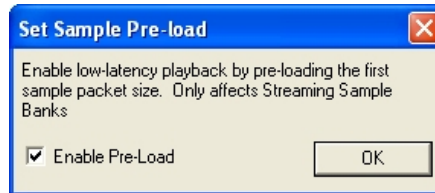


As you can see, it is possible to toggle the bank's streaming status, and also to set the stream packet size. All samples within the bank will now be streamed from disk when they are played. The packet size defines the length of the 'chunks' into which the sound will be split, and therefore indicates how much RAM that sound will take up when it is playing.

## Pre-loading

Accessing the disk and bringing sample data into main RAM is not instantaneous – there will always be a time delay associated with this process. Clearly this means that there will be some latency involved in streaming. Whereas a sound that is in RAM (a 'static' sound) can be played almost instantly, the first 'chunk' of a streaming sound must be retrieved after the call to play has been made.

However, ISACT has a mechanism to alleviate this latency without impacting too heavily on the memory footprint. If a streaming sound is tagged to 'pre-load', then the first chunk of data for that sample will be read into memory as soon as its sample bank is loaded. So this essentially provides a compromise between the potentially large memory cost of static sounds and the potential latency penalty of streamed sounds. Streaming samples can be tagged to pre-load on a per-sample basis. Right-click the sample in the Project Tree or sample bank editor, and select 'Pre-Load...'. Multiple-selection can be used in order to apply this property to many samples.



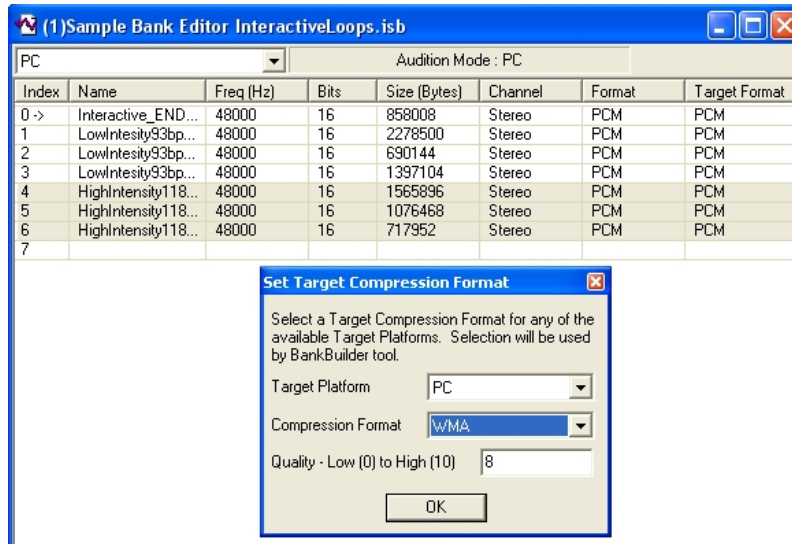
For every sound in bank that is tagged to pre-load, one chunk's worth of memory will be taken up when that bank loads.

Note that it is possible for a programmer to over-ride your streaming and pre-loading settings when the sample bank is loaded at run-time. Also, if the run-time libraries detect that a sound tagged to stream actually uses less memory than would be utilised by the streaming system (taking into account the packet size), then the streaming behaviour will be turned off and the sound loaded as normal.

## Compression

To further drive down the amount of RAM used by audio samples, ISACT sound data can be squashed down in size using a variety of audio compression formats.

The process of applying compression is quite simple. Compression properties are set on a per-sample basis. So right-click the sample or samples which you want to apply compression format to (again, multiple-selection is possible) and select 'Target Format...'. The Set Compression Format dialog appears allowing you to set the target compression formats for each destination platform.



Available compression formats include ADPCM, Ogg Vorbis, and WMA.

You can actually set up your own Target Platforms and choose which Compression formats will be available to these platforms, but that will be described later in the document.

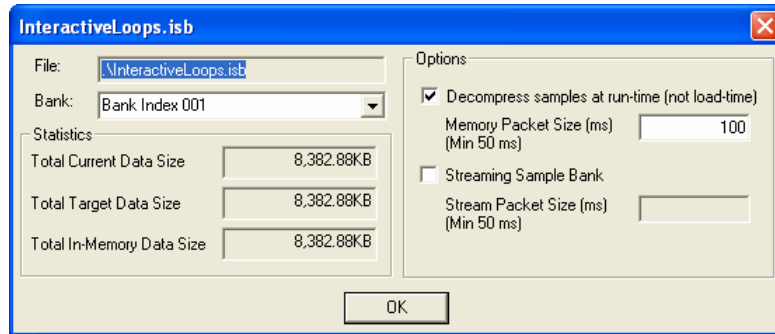
The ADPCM variant supported by ISACT is IMA 4bit ADPCM with 64 + 1 samples per block.

Note that with Ogg Vorbis and WMA you can set a quality factor enabling you to choose on a scale between fidelity and size. You can also return a sample to the uncompressed setting by selecting PCM. It is possible for a sample to be both streamed and compressed, although pre-loading is currently disabled for Ogg Vorbis and WMA streams.

If you have made some adjustments to the compression settings on Audio Samples within a sample bank or within the main project, then you will subsequently need to run the data through the ISACT Build Utility, which will actuate the settings by applying the compression. Once this is done, any affected Sample Banks (.isb) and Project files (.sac) will be ready for distribution.

#### Run-time de-compression

Another way to reduce the amount of main RAM required is to keep compressed audio samples in their compressed format until they need to be played. The trade-off is that there will be an increased CPU load for performing the decompression at play-time instead of Sample Bank loading time.



As you can see, it is possible to toggle whether the bank uses run-time decompression or load-time decompression, and also to set the stream packet size. If run-time decompression is enabled, all compressed samples within the bank will be streamed (and decoded) from memory when they are played. The packet size defines the length of the 'chunks' into which the sound will be split for streaming.

### The ISACT Build Utility

The ISACT Build Utility is a command line application that takes a standard uncompressed sample bank and applies the specified compression to the Audio Samples inside the bank. It also rids the file of various bits of meta-data which are useful for the ISACT Production Studio but of no value for the run-time.

For this reason it is critical not to over-write your original Projects and Sample Banks with the output from the ISACT Build Utility. Always back up your files in a safe place before processing them with the ISACT Build Utility.

For more information on the ISACT Build Utility, see the accompanying readme file.

### Run-time implementation of Optimisations

At the bottom of the sample bank edit dialog, various statistics are displayed. This includes a target data size for the current bank, which gives an estimate for the memory usage that the bank will require when loaded at run-time. Note that this value is a 'best-guess', which tries to take into account the implications of streaming, pre-loading and compression. This estimate is particularly vague when the Ogg Vorbis or WMA formats are used, since they use variable bit-rate compression and hence the compression rate applied will vary depending on the sample content.

As with all factors which effect run-time resource usage, it's well worth discussing the use of streaming, pre-loading and run-time / load-time compression with the programming team, to ensure that the policies adopted are compatible with the rest of the application's code-base. For instance, although de-compression and streaming save memory, they cost CPU cycles and disk access. All these factors must be taken into account when you are attempting to maximise an application's use of resources.

At run-time, the ISACT libraries will decode any ADPCM data automatically. But the libraries depend on the presence of third party decode DLLs to de-compress Ogg Vorbis data (for more information, see [www.xiph.org](http://www.xiph.org)) and WMA data (see [www.microsoft.com](http://www.microsoft.com)).

## Conclusions

---

ISACT's referencing is carried out using indices for all objects. Audio Samples are stored in sound banks, along with 3D Buffer Controllers. Other content is stored in Content Banks. Since these banks can be dynamically loaded and unloaded at runtime, the bank system offers scope for memory optimisation – you don't have to keep all your sound data in memory all of the time. But be cautious – since you can change the indices of your objects, you have the capacity to 'break' content by altering the index of an object that is referred to by some other content.

You can save memory at run-time by streaming, pre-loading and compressing your samples. You can combine many of these techniques to suit your content. But be aware of the limitations and pay-offs. The ISACT Build Utility performs compression on Audio Samples, and strips out meta-data from Projects and Banks. But be aware that processed files will not load back into ISACT – they are for run-time use only.

## Using hardware effects processing: Global Effects



Audio effects, such as those hardware accelerated by EAX ADVANCED HD™ compatible soundcards are commonly used in many interactive applications which feature 3D audio. EAX 3.0 compatible cards support one effect-processing slot (FX Slot), which can render multi-channel reverberation. EAX 4.0 compatible cards support four FX Slots, each slot can render one effect type from a list including multi-channel reverberation and also stereo auto-wah, chorus, compression, distortion, EQ, flanger, frequency shift, pitch shift, vocal morpher, and more. Future EAX soundcards may offer more processing slots and extra effect types.

ISACT content can be designed to send audio through the effect processing slots. ISACT can control the state of the effect slots using a Global Effect component. This component allows you to load an effect type in each slot, and set each effect's properties.

IPS does not provide sophisticated effects editing and auditioning features, although you can adjust parameters one-by-one if you drill into the effect settings for the slot. However, since IPS is compatible with the .EFX extendable audio effects file format you can access effects presets created using another .EFX compatible application, for instance the EAXShow tool, which is included in the EAX 4.0 SDK.

Open the project "Effects.sac" – this contains a basic ordered Sound Event. We're going to add some effects!

Select the Content Bank tab and right-click on 'Effects.icb', and create a new 'Content Object' of type Global Effect. Give it a sensible name...like *MyEffects!*

Right-click the Global Effects root and select *Add...* to open a new effect slots. Remember, EAX 4.0 will render effects in slots 1 to 4. Slot 1 must contain Reverb settings, Slot 2 must contain Chorus.

We want to work with all four effect-slots, so add slots 1 thru 4 by the same process.

For future compatibility, up to 32 FX Slots can be controlled, although on current hardware only four slots are actually available. When new effect types are defined in the future, you will be able to add them to IPS simply by providing the application with an effect library file (.EFX) containing information about the new effects.

## Effects Manager

---

The key interface for importing EAX effect data is the Effects Manager. To show the Effects Manager dialog, open the *View* menu and select *EFX Effects Manager* – the hotkey combination to toggle the Effects Manager dialog is *Ctrl-E*.

The effects manager allows you to view effects stored in the standard “.EFX” effect file format. For more information on the EFX format, see Creative Labs Developer Relations. The Effect Manager can access multiple EFX files simultaneously, and allows you to view all the effect settings in each and bring the presets into your ISACT projects.

Open the Effects Manager (Ctrl-E)

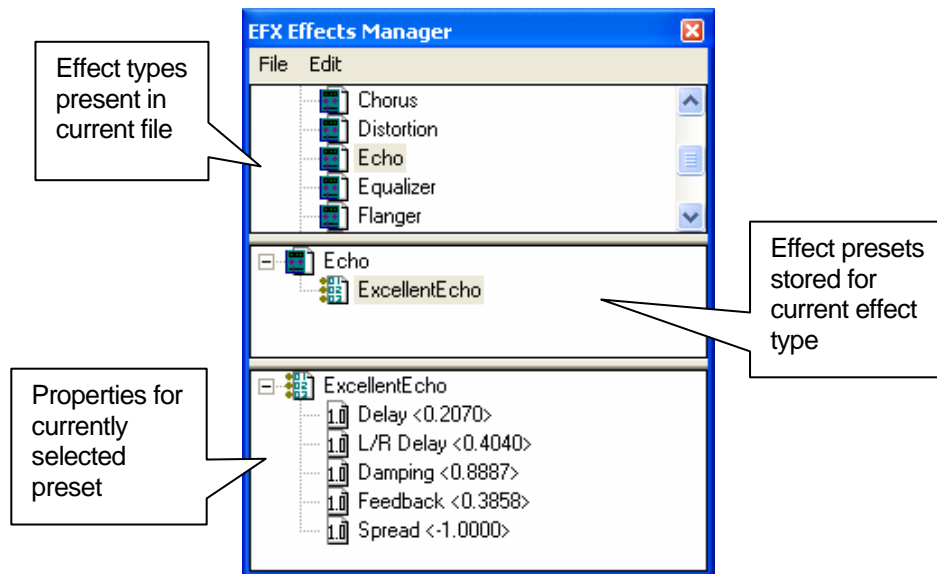
On the Effects Manager dialog menu, select *File -> Add...*

You'll see a standard file browser which you can use to locate one or more “.efx” files containing EAX preset data.

Go to the Tutorial\EFX folder, and select the file “TutorialFX.efx”.

Use the “File” -> “Paths” menu to select between the different .efx files currently being accessed by the Effects Manager. Make sure you are looking at “TutorialFX.efx”

If you expand one of the Effect Slot folders in the Global Effect's Tree-View, you can right-click on the Effect Type to bring up a context menu. This allows you to select directly from all the effect types and effect presets available in the .EFX files.



The effect manager dialog has three panes. The top pane shows the different effect types present in the currently selected file. Choose an effect type from the top pane and the middle pane will show the effects presets available for that effect type.

An .efx file can be arranged into a tree structure with folders and subfolders, so the middle pane allows you to navigate such a structure. If nothing appears in the middle pane, then no data is present for the chosen effect type.

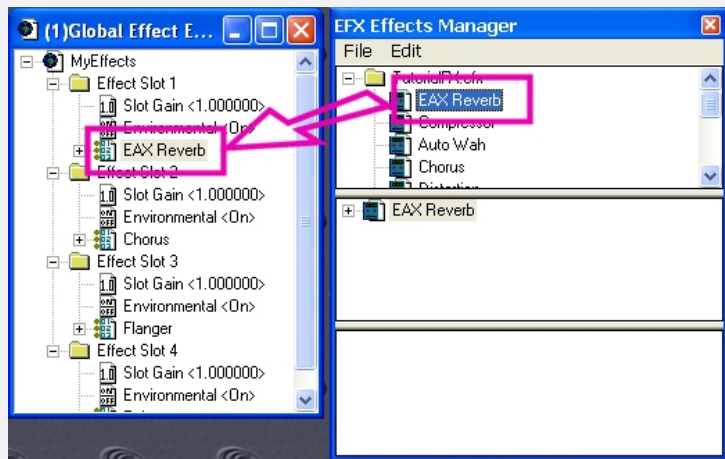
Select a preset in the middle pane and the lower pane will show the settings for that preset.

You can drag and drop effect types and effect preset data from the Effect Manager dialog into the slots contained in your Global Effects object. But you can only drag effect preset data into a slot which is configured to render the same type of effect.

So first of all, open all the effect folders in the Global Effects dialog.

From the Effect Manager, drag one of the "EAX Reverb" icons into the Effect Slot 1 effect type setting. (If Slot 1 already contains EAX Reverb, then you don't need to do this)



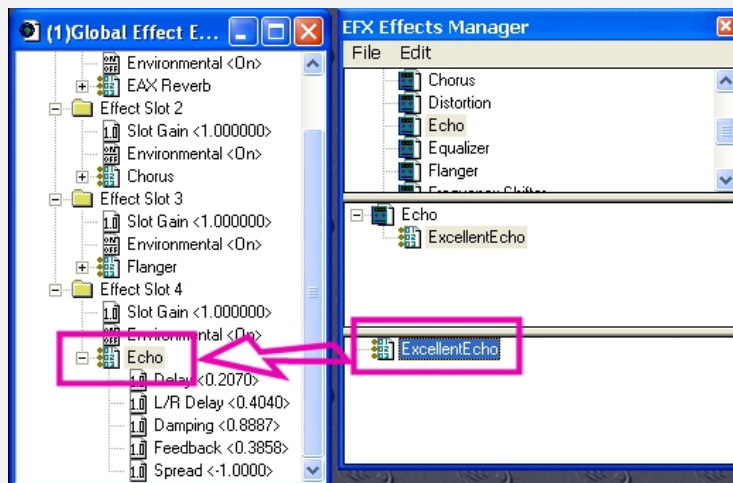


Make certain that Slot 2 contains Chorus, Slot 3 Flanger and Slot 4 Echo.

Now you can drag the preset data across as well.

Go back to EAX Reverb on the Effects Manager, and expand the tree in the middle pane – you should see a preset called “RockinReverb”. Drag the reverb across to the Global Effects. If you have the slot parameters expanded on the Global Effect dialog, you will see the new figures being applied.

Do the same for the Chorus (“CoolChorus”), the Flanger (“FunkyFlanger”) and the Echo (“ExcellentEcho”)



Once you have finished creating your Global Effect, be sure to play it using the Transport – you won’t hear anything, but the FX Slots on the soundcard have now been configured with your desired effect presets.

Note that you can adjust effect parameters directly for yourself within the Global Effects dialog. Just drill down to the individual properties for an effect slot, and double click a property to adjust it.

Open the SoundEvent “EffectNames”. Test the SoundEvent, you’ll hear the names of different effect types played in order...but they are dry.

Bring up the SoundEvent’s properties by right-clicking it’s entry in the Content Bank. Set the ‘Global Effect’ property to point to your newly created Global Effect object.

Now it’s just a question of setting up the Sound Event entries to each send audio to the appropriate effect slot. To the right hand side of the Sound Event rows you’ll find a panel with effect send properties.



For the first entry in the Sound Event, which plays “Reverb.wav”, set the slot to 1 and click on the tick icon to turn on the send into effect slot 1.

Now set both Min and Max values to 100 to ensure a good signal in the send.

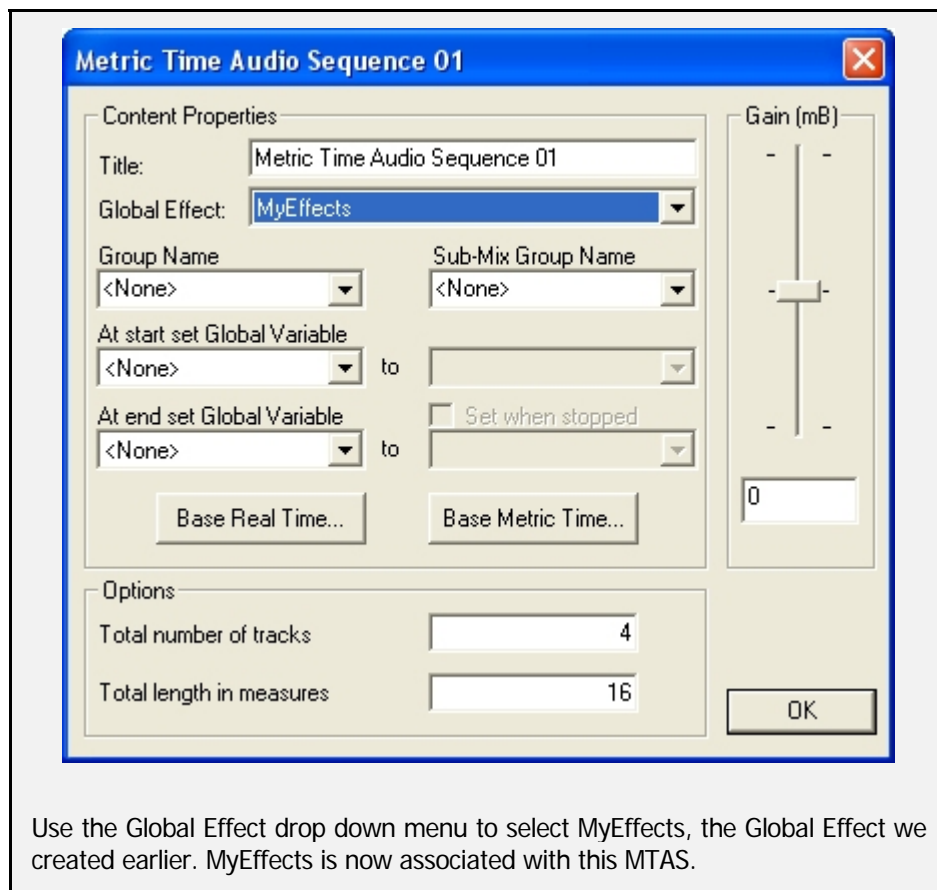
Go through the SoundEvent, setting each alternative to send to consecutive slots. So “Chorus.wav” should go through slot 2, “Flanger.wav” through slot 3, and “Echo.wav” through slot 4.

Now when you audition the Sound Event, the samples should play through the appropriate effect!

A Global Effect can be associated with a piece of playable content such as a Sound Event, a Sound Entity or a Sequence. When the content is played, the referenced Global Effect settings will be passed to the hardware if supported.

Create a new Metric Time Audio Sequence by right-clicking on “EffectsFinished.icb”, selecting *New > Content Object > Metric Time Audio Sequence* and then click *OK*.

Now right click the MTAS you have just created and select *Properties...* to bring up the properties dialogue. You will see that the second drop down menu in the Content Properties section is labelled Global Effect:



You can load a finished version to compare against – “EffectsFinished.sac”. Remember this functionality will only work on EAX 4.0 compatible systems – a PC containing an Audigy or X-Fi series of soundcard with up-to-date drivers installed.

## Conclusions

---

DSP audio effects are available in ISACT through Creative's EAX SDK. IPS itself doesn't have the sophisticated effect editing and auditioning features of a dedicated editor application. The best way to bring your own effects into ISACT is using the Effect Manager, which understands the standard .efx extendable effect file format.

Create Global Effect settings to control the state of the EAX 4.0 DSP system. Associate the Global Effect with a content object, and those effect settings will be activated when the content is played. Relevant object types expose effect send parameters, so you can manipulate how each sound feeds the different slots.

## Where the action's at: Sound Entities



Up to this point, we haven't seen how to dynamically adjust ISACT content in real time from your application's code. Sure, you can trigger Audio Samples directly, or with random dynamics via Sound Events. But you can't make changes to sound playback to reflect in some way the state of the application...not without Sound Entities anyway!

### Sound Entity example one: Too many bugs

---

One of the tutorial projects is called 'BugSwarmUNFINISHED.sac'. This project contains several complete Sound Events that produce insect sounds, including the grasshopper and flying bugs events mentioned earlier in the document.

We're going to add to the project a Sound Entity that plays several of these events together to produce a dynamic 3D ambience of varied insect sounds. The Entity will be controlled by a variable, which will determine the density of insects present.

Load the tutorial .SAC file name 'BugSwarmUNFINISHED.sac'.

Audition the different insect Sound Events for yourself, and imagine playing them all layered together.

Add a new 'Controller' object to the Content Bank 'bugswarmunfinished.icb' – a Sound Entity. Call it 'BugSwarm'.

We want to play a maximum of ten insects at once, so set Players to 10.

### Players

A Player represents a playback channel for a single piece of content being played by an entity. When dealing directly with components like Sound Events or Audio Samples, you don't have to worry about managing playback channels – ISACT will automatically take care of Player allocation.

However, Entities supported layered sounds, each of which is a piece of content in its own right – for instance a Sound Event. Furthermore, within an Entity you can implement your own commands to play and stop sounds, or even adjust their properties, all according to the Entity's state. Players provide the reference to each bit of content that you are controlling within your entity.

It makes sense to decide beforehand what content you will be controlling within your entity. In this case we're going to work with three instances of 'Bee' Sound Events, three instances of 'Fly' Sound Events, and four instances of 'Grasshopper' Sound Events. For the sake of clarity, it would be sensible to name each Player within the Entity according to the content you plan to play on it – Fly1, Fly2, Fly3, Bee1, Bee2, etc.... To rename a Player right click on it and select "Player Properties ..." and give it a unique name.

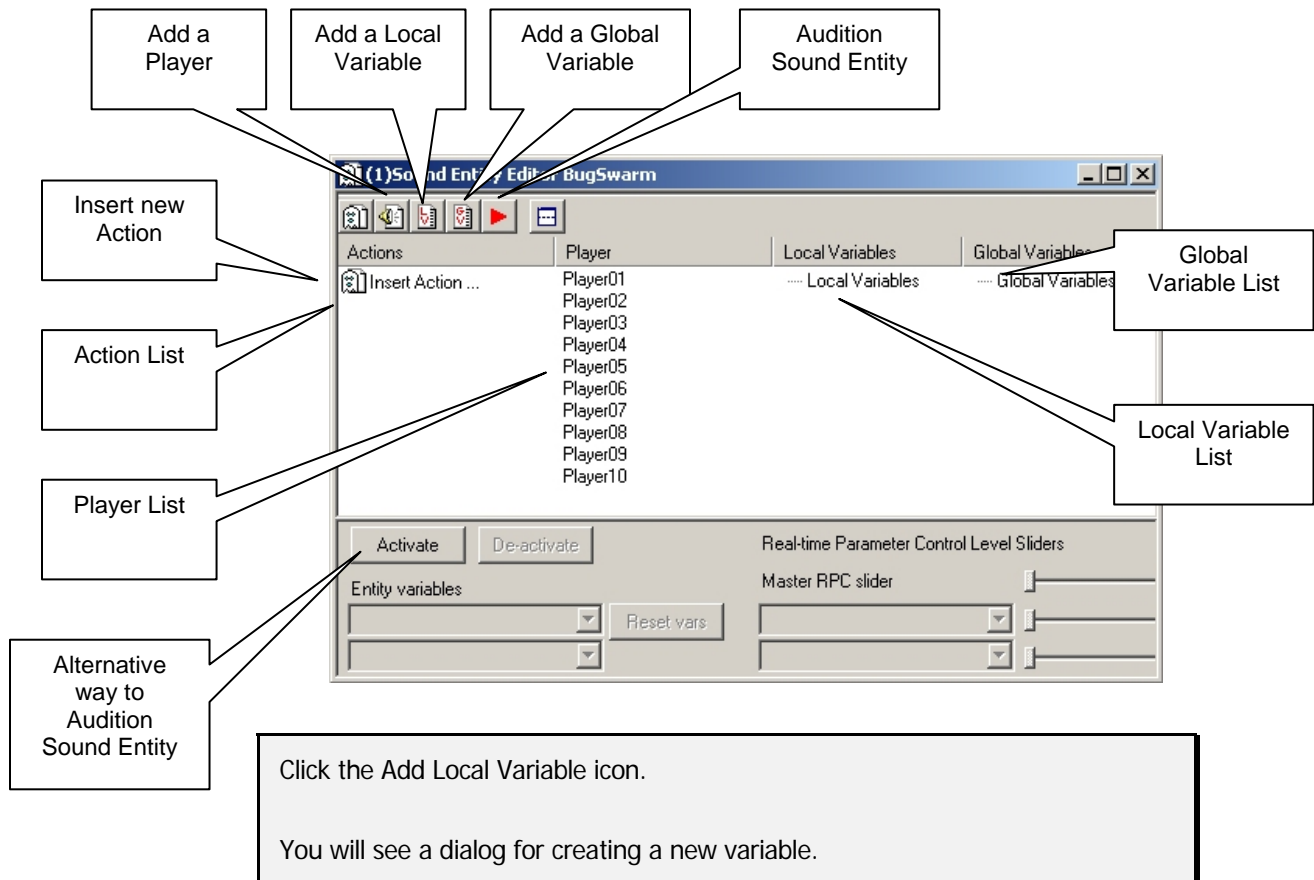
## Variables

Sound Entities expose variables, and respond to changes in their state. Entity variables are not actually values like integers, floating point numbers or strings. Instead, the designer defines the Entity variable as a finite set of states.

Variables can be Local or Global. Local variables can only be seen by the Entity in which they are defined. Global variables are common to all Entities within a single Content Bank.

At run-time, the application can construct instances of an Entity defined in an ISACT project, and alter the variables connected to each instance. Let's create a variable to control the density of our bug swarm Entity.

When you edit a Sound Entity, you see the Sound Entity Editor dialog, which will look something like this:



Name your variable 'Level'. Add four states, naming them 'Low', 'Medium', 'High' and 'Off'. The state marked with an asterisk is the default – i.e. the state the variable will be when the Entity is created. Highlight the "Off" state and press the "Set to Default" button to set 'Off' to the default state.

The variable you just created will control our Entity. We need the Entity to behave so that the number of Events playing will relate to the state of 'Level'. At a high level, many insects should be audible. When level is 'Low', things will be relatively quiet.

Once you have added the variable to your Entity, you can start creating some Actions.

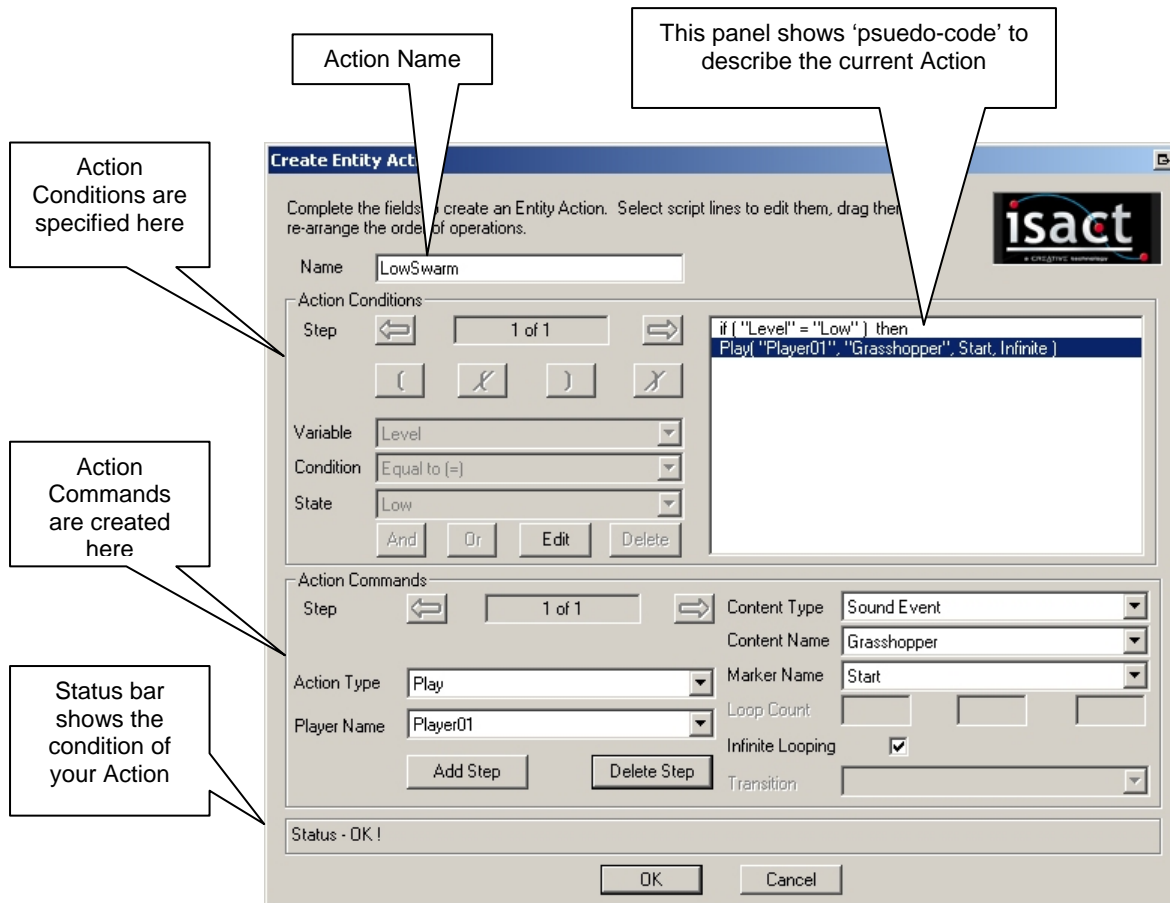
### Actions

The purpose of an Action is to check for a particular combination of variable states. If the conditions are met, the Action carries out one or more commands.

So an action consists of two parts. First of all, there is a set of criteria that must be met if the action is to be carried out – the 'Action Conditions'. Then for when the criteria are met and the test passed, the command or commands to be performed – the 'Action Commands'.

Click the 'Insert Action' icon or right-click in the area beneath the Actions tab, and select "Add Action ..." from the menu.

The 'Create Entity Action' dialog looks complex, but so long as you go through the process of creating an action systematically, it's actually quite simple!



### Action Conditions

The Action Conditions are a set of logical checks on the Entity's variables. Any number of tests can be carried out, and the outcomes combined, but the test is a big 'IF' statement, so the eventual outcome must be binary – true or false.

Firstly we want to create an Action for when the Level is set Low. Initially we must test for this situation.

For our first test step, we will test the 'Level' variable, so select this variable from the first drop-down box available to you. The next drop-down box becomes active.

Here you can select from a set of different comparisons. We need to identify when 'Level' equals 'Low', so select 'Equal to' from the list.

And when the next list becomes available, select the state 'Low'.

Note in the large text box to the right, some logical 'pseudo-code' is displayed, showing our progress.

A status box at the bottom of the dialog shows whether our Action is valid or not. Of course right now it's not finished, so it says 'Result Step 1 : Blank line found'

## Action Commands

Similarly, the designer can create a list of different Action Commands to be carried out in the event of a positive result to the test. Action Commands can be operations on other content, for instance playing or stopping a Sound Event. Alternatively, they can set parameters on playing back content, such as volume, position and pitch. Or they can adjust variables.

Be careful playing with Action Commands – by changing variables from within multiple actions, it is actually possible to set up circular references, which could cause ISACT to go into an infinite loop!

For now, let's set up commands to play two Grasshopper SoundEvents, one Bee SoundEvent, and also a Fly!

Press the 'End' button to complete the 'conditions' and move on to the 'commands'. Again you can move from one drop-down box to the next. Our first action will be to Play a Grasshopper Event. Set the Action Type to 'Play'

We'll always play the first grasshopper Event on Player01 – select this player in the Player Name box.

Grasshopper is a Sound Event, select this component type in the 'Content Type' box.

All the Sound Events in the project are offered in the next box, Content Name. Select 'Grasshopper Chirps'.

Ignore Section Name for now as the Grasshopper content does not contain any section markers.

It makes sense to keep looping the Grasshopper infinitely. We can halt the Event again when the Entity reaches a state where it should stop – for instance if the application stops or destroys the Entity. So tick the Infinite Looping box.



You'll also need Action Commands to play the Fly and Bee Events infinitely looped. Hit 'Add Step' to continuing adding commands to the list. Remember to use a different Player for each piece of content so that you hear all the bugs simultaneously.

Plus there's plenty more to remember with Entities. For instance, in this example your state might be changing down from High, or Medium, to Low, in which case it will be necessary to stop some of the currently looping Events from states when more Events are looped. It's safest to assume the worst, and in this action add commands to stop any sounds that shouldn't be playing. Plus you might want to spatialise the grasshopper sounds by setting a position. And apply some change in pitch to some of the Fly and Bee Events to provide some variation.

As a tip, its best to map out beforehand which component you want to play on which Player, and allocate a player for each. You can name Players, making it easier to identify which Player is supposed to be playing which content.

Try to set up this Entity yourself. You'll need Actions for when the Entity is playing at Low, Medium and High levels, and one for when the Entity is stopped, i.e. when 'Level' is 'Off'.

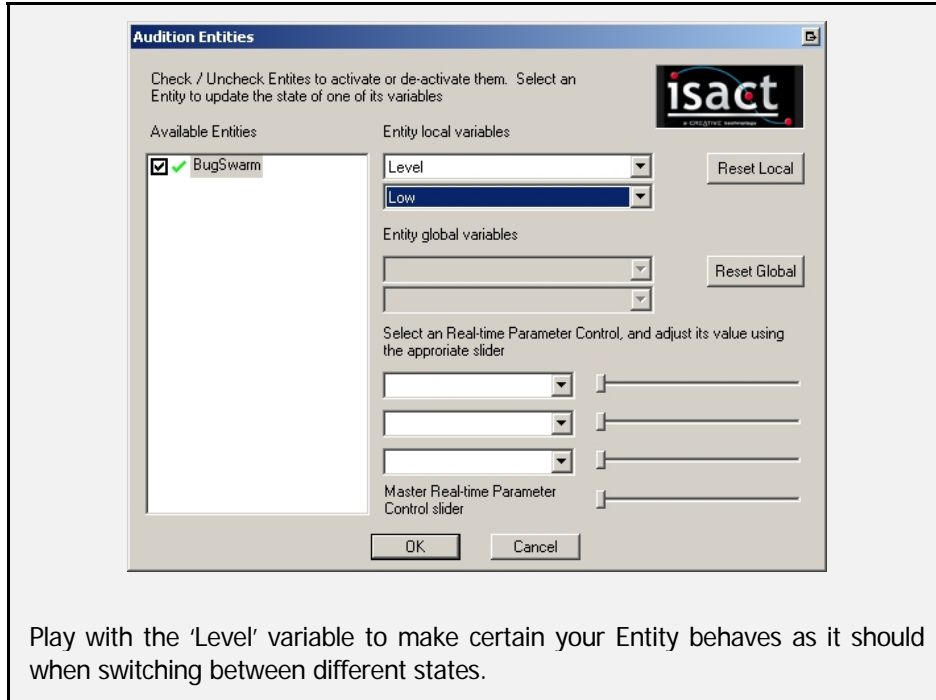
Remember that each time an entity variable's value is altered by the application, ISACT will run through all the actions in the entity in the order in which they appear. If the action refers to the altered variable, then it will be processed. In case the action order is critical, you can change the order of the actions by dragging them around the action list in the entity editor dialog.

You can load 'BugSwarmFINISHED.sac' to analyzse and audition a complete Sound Entity that implements the behaviour discussed. To see how Transitions (discussed later) work within Sound Entities, load 'BugSwarmFINISHED\_TRANSITIONS.sac'.

#### Auditioning Entities

You can audition Sound Entities within the IPS environment, meaning you don't need to write code to test your audio designs.

Click the Audition icon, you'll see the Audition Entity dialog



## Conclusion

Entities enable you to design a complex sound effect, which can react to discrete state variables. Within an entity, you control the playback of one or more audio components, using a simple scripting system.

## High Command – Real-time Parameter Controls



We saw in the previous section how the sound designer can create a simple state variable. You can script how a Sound Entity reacts to different states using actions. But you might want more in-depth control of your sounds – for this you'll need an RPC.

The RPC allows you to define mappings between numeric inputs and 3D sound properties like volume and pitch. The input for an RPC is a floating-point value between 0 and 1. It is trivial for the program code to scale a game variable into this range before it is passed in to the ISACT engine at run-time.

### Real-time Parameter Control example one: Get revved up!

---

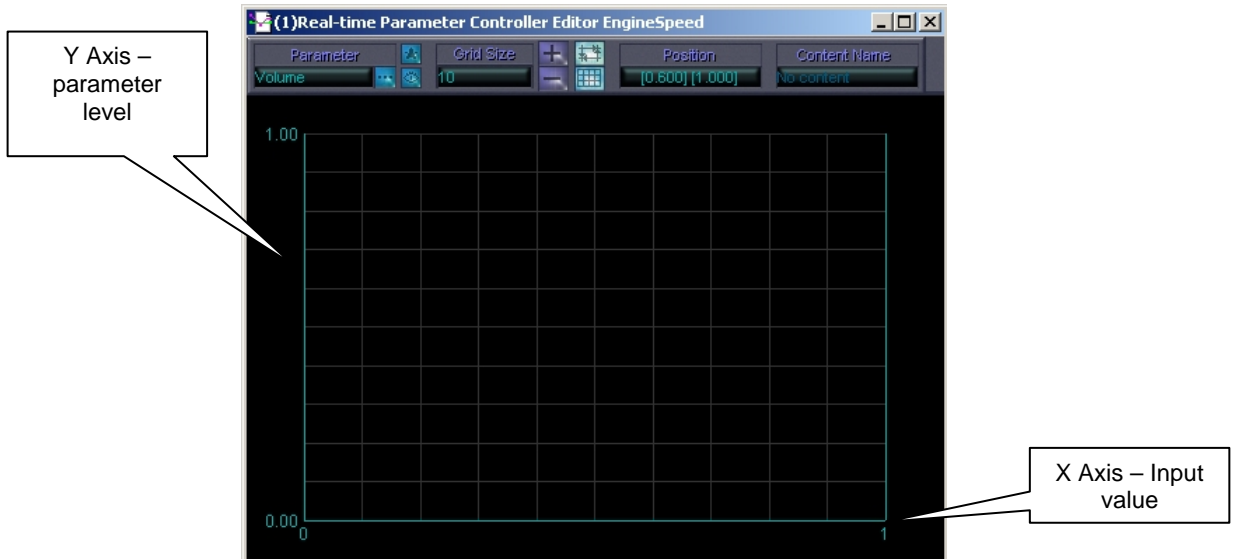
We can base a simple example around vehicle sounds. Vehicle sounds in games tend to work on a simple principle of varying sample playback pitch on a small loop to simulate the engine speed. Is it possible to achieve this with ISACT? Damn right it is!

Load up the project called 'CarSimpleUNFINISHED.sac'. This file features a very simple Sound Event, which combines an Audio Sample containing a short engine sample with a Global Effect setting containing a distortion effect in FX Slot 3. Play the Sound Event looped, and the engine sound is acceptable. It gets old quickly though. Something is missing...a little *je ne sais quoi!*

Let's liven things up by adding some dynamics.

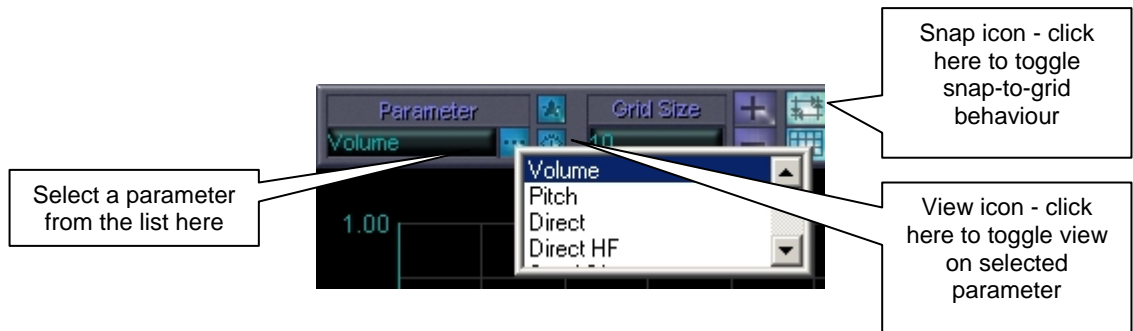
Click the Content Banks tab and right-click the Content Bank 'CarSimpleUNFINISHED.icb'. Select "New..." and add a controller object - a Real-time Parameter Controller.

Give it a sensible name like 'EngineSpeed'. A dialog containing a blank graph appears.



### Mapping properties to inputs

The graph allows you to draw mappings between the input on the X-axis, and the 3D sound property on the Y-axis. It is blank because no sound property is currently being viewed. To view a property, select the property you want from the Parameter list, and click on the icon that looks like an eye to show and hide the property's mapping.



The crucial dynamic in a vehicle sound is for the vehicle's motor speed to modulate the engine sample's playback pitch.

- Click the Parameter panel
- Select the Pitch parameter from the parameter list and double-click
- Click the 'view' icon.

The Y-axis scale changes, to show that pitch can be varied from 0 to 2.0 – pitch is a parameter that is applied multiplicatively. Right now, you can see that pitch is at 1.0 for the entire input range.

Parameter mappings are controlled with knots. To create a new knot, simply double-click on the graph. You can select knots by dragging a box over them in the graph view. When a knot is 'in-focus' for adjustment, the cursor changes to a pointing finger. Try moving the knots at the left and right of the graph. As these knots are the first and last in the mapping, they are fixed to input values 0 and 1. To remove a knot, highlight it and press 'delete'. To turn the snap-to-grid feature off, click the snap icon.

Create a mapping that changes pitch from low to high as the input variable moves from 0 to 1.

It should be shown as a diagonal line from bottom left to top right.

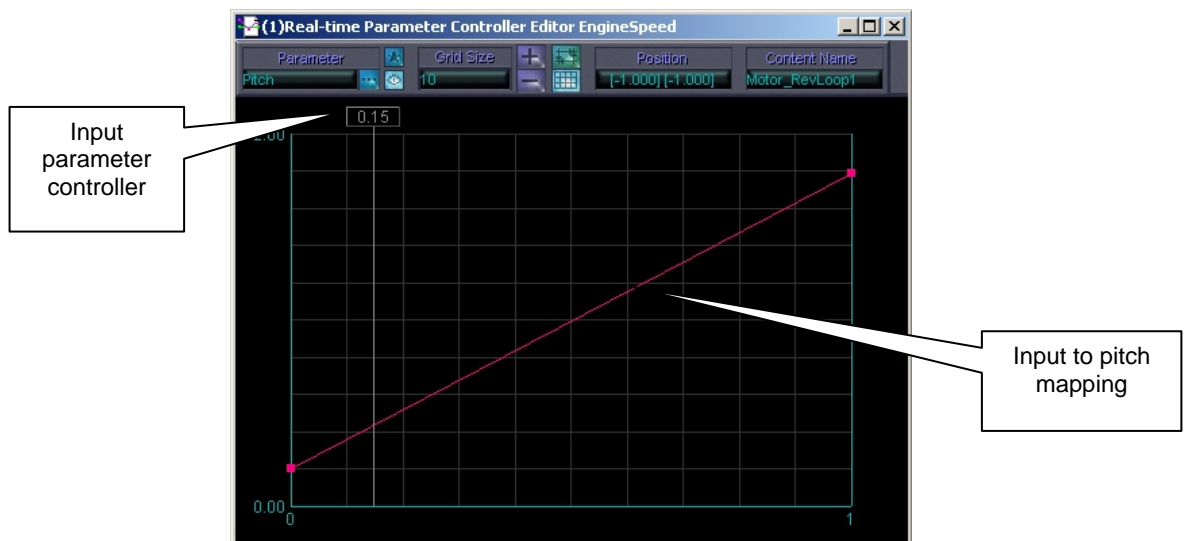
#### Auditioning an RPC

Now its time to audition the RPC, to check it works as desired.

Locate the RevLoop Sound Event in the project tree, and simply drag it into the RPC dialog window.

Click on Play Loooped in the main playback transport window (remember, its *Ctrl-Shift-T* to toggle the playback transport window).

The sound will play, and a grey box will appear on the RPC dialog. This allows you to alter the input setting on the RPC. As you move the input to the right, the pitch of the engine sound should rise.



You can control more than just one property with an RPC. Load up the file 'CarSimpleFINISHED.sac'. This project uses the same components, but the

EngineSpeed RPC is more complex. It modulates the DirectHF property of the sound source, to open and close a low pass filter on the engine sound. It also adjusts the effect send to the distortion effect, so there is more distortion at higher revs.

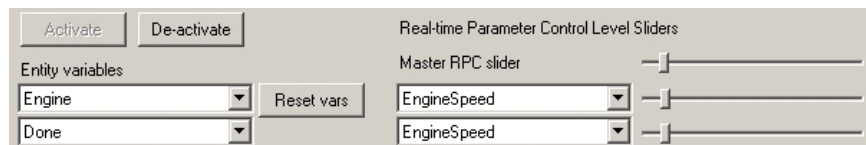
### Integrating an RPC within a Sound Entity

This project also shows how an RPC can be integrated into a Sound Entity. From the previous section, you will remember that an Entity's state variable can control whether a sound is played or stopped. Each sound source in an Entity is associated with a Player.

Open the Car Engine Sound Entity, you'll see it includes a single Player, named Engine. Right-click this Player and select 'Player Properties...' You'll see how the project's RPCs can be selected from a drop-down list, and that the EngineSpeed RPC is selected on this player.



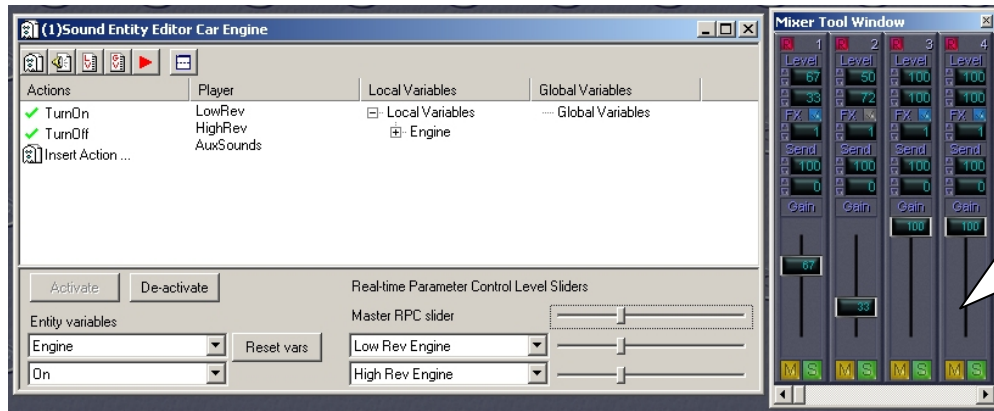
When you audition the Sound Entity, you will see that sliders are available for each RPC. Set the Engine variable to On, and adjust the master RPC slider. This way you are able to associate RPCs with sound sources inside of Entities, and test how the Entity as a whole reacts to the changing RPC.



### Real-time Parameter Control example two: Cross Fading

The second example takes the engine model a little further. Have a look at 'CarCrossfade.sac'. Here we have two engine sound loops, one for 'idling' and one for 'revving'. There are also two separate RPCs. One is louder for low input values, and ramps down for higher values, the other is the opposite, ramping up the volume at higher inputs. This enables us to cross-fade between the 'idling' loop at low revs and the 'revving' loop at high revs.

When you audition the Entity, view IPS' mixer panel (*View->Tool Windows->Mixer* or *Ctrl-Shift-M*). ISACT's voice management system will automatically select two 3D channels to play the different loops, most likely the first two channels, on the left of the mixer. You can see how the volume of one channel is reduced as the other is raised when you move the master RPC control slider. You can also see how the Effect Send amounts change on the channels in real-time.



## Simple Sequencing: Sound Queues and Timelines

There are a couple of different ways to string together multiple simple sound like Audio Samples and Sound Events to create a short sequence.

### Sound Queues

---



The Sound Queue is a 'monophonic' sequence – only one sound can be playing at any one time in this sequence. But looping sounds are supported. To demonstrate this component, we're going to create a sound queue for a car's starter motor.

Open the tutorial project "StarterMotor.sac".

Have a listen to the samples, and check out the "StarterMotorLoop" Sound Event.

We aim to set up a queue, which starts with the initial starter motor click, followed by the looped sound of the starter turning, before the sound of the engine catching.

Add a new Sound Queue to the Content Bank – it's a Content Object. Call it something sensible, such as "StarterQueue". For now, you can ignore the other properties.

Drag the Audio Sample called "StarterBegin" from the Sample Bank into the Sound Queue dialog. It will appear at the top of the list.

Now drag the Sound Event "StarterLoopEvent" into the queue. We want this sound to be looped while the starter is working, so double-click on the box to the left of the Sound Event's entry in the queue.



Finally, drag the EngineCatch Audio Sample to the end of the queue.

If you got the sounds in the wrong order somehow, you can re-order them by dragging them up and down the queue. Only one group of consecutive entries can be looped, but in this instance we only want to loop the StarterLoopEvent.



Now you can test your queue. Play the queue using the transport. You'll hear the click of the starter, followed by the looped random event simulating the starter motor turning over without catching.

When you click Stop in the transport, the loop ends, and the "EngineCatch" sound is played. All this activity is shown in the playing status column to the right of the Queue dialog.

A Sound Queue can be told to immediately break-out of a looped section without finishing the current iteration of the loop by checking the "Immediately break out of looped section on stop" check-box in the Sound Queue's Properties Dialogue Box.

Note how different component types can be queued –

- Audio Sample
- Sound Event
- Sound Timeline
- Metric Time audio sequence
- Global Effect

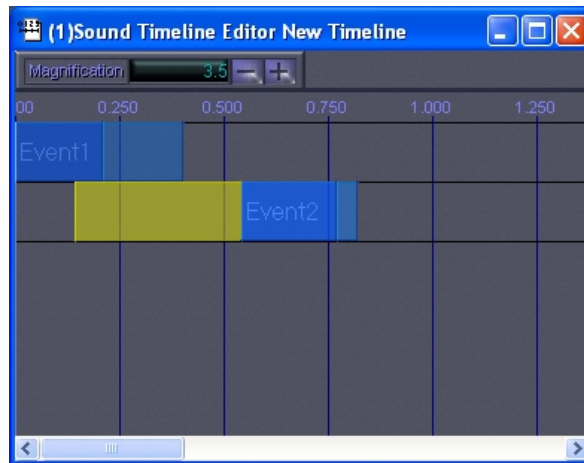
## Sound Timelines



The Timeline has slightly different functionality. For starters, Sound Timelines are constructed solely from Sound Events. In the Sound queue, sounds follow each other one by one – one sound can only start once the preceding one finishes. But you might wish to give your sounds an absolute start time. Since Sound Events feature elements of randomisation, you can't tell in advance the duration of the sound.

For this reason, the visual interface for the Timeline displays Sound Events' minimum and maximum durations, making it easier to position several together with confidence. In the Sound Timeline, Sound Events can overlap.

To place Sound Events on a Timeline, simply drag them from the project tree into the Timeline dialog. Notice how the minimum and maximum duration of the event are indicated. This allows for the fact that a sample might be pitched down, causing it to play for longer.



You can add a random period of silence to the start of each Event in a Timeline. Simply right-click an Event in the Timeline and select the properties option. The maximum potential delay before an Event in the Timeline is displayed as a yellow rectangle.



To demonstrate this component, we're going to use a timeline to make a sound effect for a rocket launcher weapon.

Open the tutorial project 'WeaponsUNIFINISHED.sac' from the 'Weapons Demo' directory.

Have a listen to the Rocket Launcher samples, and check out the 'RocketTimeLine' time line.

At the moment, the time line is pretty simple and thus our rocket launcher sound effect is a little boring, so let's make it more interesting by adding some more content onto the timeline:

Drag the Sound Events 'Bang01' and 'Bang02' onto the timeline. Right-click 'Bang01' on the time line and select "Track Properties...". Set its Minimum offset to 0.010 and its Maximum offset to 0.050. Do the same for 'Bang02'. Now playback the timeline.

A little better!

Now drag the 'Whoosh01' and 'Whoosh02' Sound Events onto the time line. Right-click 'Whoosh01' on the time line and select "Track Properties..." Set its timestamp to 0.202 seconds. Do the same for 'Whoosh02'.

Play back 'RockTimeLine'. As you can hear, our rocket launcher sound is now much more interesting. Try experimenting with different values for the Minimum offset and Maximum offset of the different Sound Events, and also moving their position on the timeline.

## Making Tracks: Audio Sequences



Audio sequencing is a familiar concept to most audio technology enthusiasts. Gather a set of audio recordings; arrange them together on a multi-track timeline. Place control data to modulate properties of the audio when it is played – things like volume ramps, pan changes etc... And when the whole track is ready, mix it together as a stereo or multi-channel stream!

Well, an ISACT audio sequence follows many of these conventions. Except that ISACT's sequences are built around 3D positioned mono sound buffers. Plus at the end of the process, no 'mix-down' takes place. Instead, the audio and control data are stored in the project file. Audio Sequences reference lower level ISACT components – Spatial Paths can be used to provide panning, Global Effects to set up effect processing, and Audio Samples contain the sound data.

ISACT audio sequences are intended for playback within interactive applications. ISACT's programming interface ensures that, just like other ISACT content, the sound and control data contained in Audio Sequences can be rendered by most popular audio spatialisation interfaces.

This means that ISACT Audio Sequences take full advantage of whatever 3D audio renderers might be present, to give the best 3D audio experience possible on a user's system.

A metric sequence is laid out on a system of measures and bars, controlled by a tempo. There are no automatic tempo adjustment features in ISACT, so if you are working on music and want all your tracks to be in time, you'll need to adjust the source samples for yourself.

### Audio Sequence example one: A moving experience!

---

We'll start to examine Audio Sequences with a simple single-track example. This example will show how Audio Samples and Spatial Paths can be combined in Sequences. The tutorial includes a project file with an Audio Sample and a Spatial Path already prepared.

Open the project file 'SimpleSequenceUNFINISHED.sac'

The project includes an Audio Sample called 'ElectroBeat' and a Spatial Path 'Circle'

Add a new 'Metric time audio sequence'

Give your new sequence a sensible name.

We won't be using any effects in this sequence. However, you may need to alter the tempo – 'ElectroBeat' has a tempo of 120 beats per minute, so select the 'Base Metric Time' option and set this value.

We'll only play one sound in this sequence, so you can reduce the number of tracks. The "Total number of tracks setting" is important because it determines how many 3D voices are reserved when the sequence is played at run-time. If you made a sequence of 8 tracks but only placed audio on 4 of them, then when the sequence is played back by an application, four voices will be wasted.

You can change all these settings later if necessary.

The Audio Sequence interface is quite extensive.

Each track has its own row in the dialog, with a control panel at the left hand side



You can drag and drop Audio Samples into a track.

Drag the 'Electro' Audio Sample into the track in the Audio Sequence.

Ensure your new audio data is flush up against the start of the track. Content can be moved around the sequence by click-dragging.

If your tempo is set correctly, purple rectangle representing the sound will take up exactly one bar.



Content can also be copied. Select the Audio Sample rectangle and hold down the control key. Drag the audio to a new location.

Just as with Spatial Paths, you can use the Time Grid Controls dialog window to set the zoom level on the sequence. You can also turn on and off the measure markings, and determine whether dragged content will 'snap' to measures.

Sample data is not stored in the sequence. When you work with audio in a sequence, all you are really doing is manipulating references to Audio Samples. So if you made a change to the Audio Sample, for instance shortening the waveform, the change is reflected wherever the sequence references that Buffer.

Notice how audio rectangles are displayed in yellow when they are overlapped by other audio. Only one Audio Sample can play at any one time on each of a sequence's tracks.

To delete some audio from the sequence, highlight the offending sound markers and press 'delete', or right-click and select delete.

Multiple selection is supported for processes such as deleting, moving etc...

Fill the track with eight instances of the electro sound.

Double click the name on the control panel, to change the track name from '<Untitled>' to something more sensible – like 'Electro Beat'



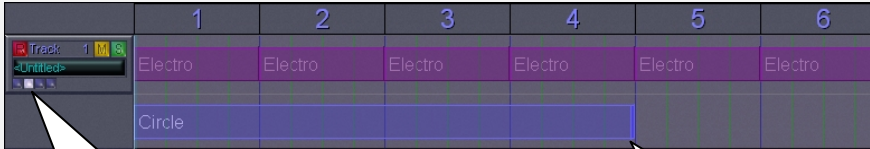
The Transport window provides control over sequence playback. Click the Play icon to start the sequence playing. Stop halts playback, while double-clicking stop resets the play cursor.

Playing back the sequence so far isn't particularly amazing – the beat just plays over. Let's add some movement with a Spatial Path.

The square boxes at the bottom of the control panel give access to extra control data for each track. The second button allows you to manage Spatial Paths for the track.

Click the second box under the track name. An extra row appears.

You can drag Spatial Paths onto this area – so grab the 'Circle' Path from the project tree and drag it into the sequence



	1	2	3	4	5	6
Track 1	Electro	Electro	Electro	Electro	Electro	Electro
Circle						

Toggle Spatial Paths Data Track

Use the handle to resize

You'll notice that we made the Circle Spatial Path four bars long. You can stretch a Path to fit content by dragging the blue handle at the right hand side.

Drag the Path duration handle to make the path eight bars long.

Play the Sequence!

Now you have an audio sequence where the looping beat sound revolves around the listener. If you don't believe it, then just open the Spatial Editor (View > Spatial Editor) and watch the sound emitter circle around as the sequence plays! You can see a finished version of this sample by loading 'SimpleSequenceFINISHED.sac'

For more information on the Spatial Editor see the 'Miscellaneous other stuff' section.

## Audio Sequence example two: The next Level!

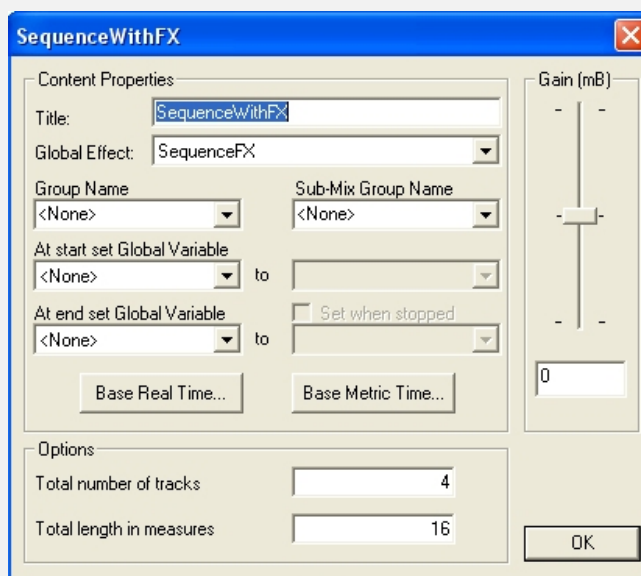
So far with Audio Sequences, you saw how to apply control data in the form of a Spatial Path to a sequenced Audio Sample. ISACT offers more control data, enabling you to adjust nearly any aspect of a sequence at playback time.

This example will add some effects processing to the sequence, and control data to adjust the effects send levels.

Open the project file 'LevelsSequenceUNFINISHED.sac'

Create a new Metric Time Audio Sequence, this time specify the Global Effect component "SequenceFX" in the properties page. This ensures that "SequenceFX" is 'played' before the Sequence begins and therefore configures the Effect Slots appropriately.

The tempo should be 120 again.



Add 'ElectroBeat' to the sequence and copy it multiple times to fill up 8 measures.

To use Effects Processing in Audio Sequences requires two main steps. Because ISACT effects processing is based around EAX 4.0 hardware-accelerated effects, you have to explicitly tell IPS which effect slots a track will feed. So first of all, we will enable the correct effects sends on the track. The effects defined in 'SequenceFX' Global Effect are placed in slots number three and four – don't forget that Slots 1 and 2 are reserved for Reverb and Chorus.



Open the sequence, and take a look at the buttons under the track controls for the track containing audio data.

Ensure that the play cursor is set to the start of the sequence, by double-clicking the stop button on the transport.

Click the third button, to show the Effect Send Enable controls.

Put the track into 'record' mode, by clicking the red button marked "R". This means that changes in control data will be stored.

Scroll through the different sends by clicking +, until you reach send number three.

Click the Tick button, to activate this send. Do the same for send number four.

A red panel will appear in the track, at the start of the sequence. The third and fourth markers will be coloured green, to signify that the sends are active.

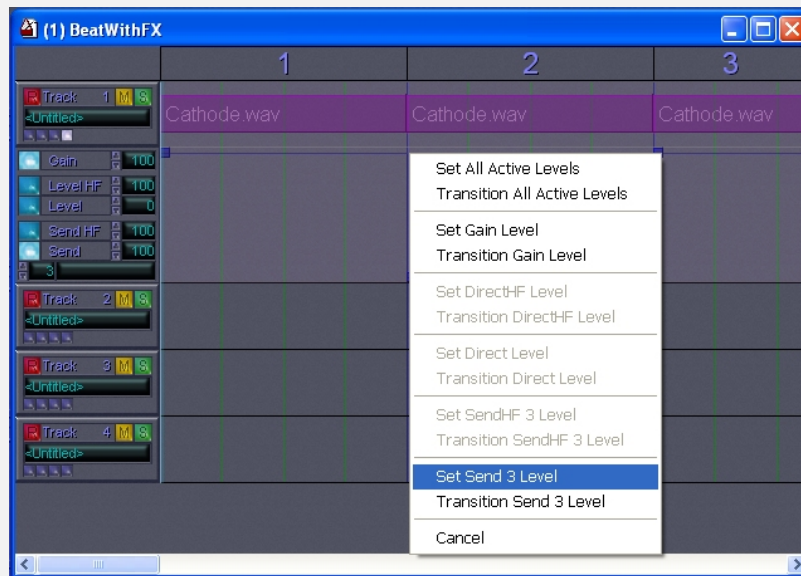


Turn record mode off again.

The FX Sends have been activated, but you'll need to turn the send levels up to hear the effects – by default they are set at zero. By the way, 'SequenceFX' Global Effect set weird, distorting chorusing and flanging effects in the slots...

You can hide the Effect Send Enable control by clicking the third button again.

The fourth button shows the level controls. Here you can set Volume and Send levels.



We want to work with Send levels. So click the Send button.

Our effects are in slots three and four. Set the Send value to three using the up and down arrows. This means we are working with the effect send levels into slot 3.

Right-click the sequence area, to the right of the level / send controls, to bring up a context menu.

The menu allows you to set or transition the different levels.

In this instance, select "Set Send 3 Level". A square 'knot' will appear, attached to a blue line.

The Send 3 Level is marked with this line. The higher the line, the more sound will be sent to the Effect in Slot 3.

Adding a transition enables you to create diagonal lines for 'fades'.

Experiment with Send Levels on Slots 3 and 4.

You can also try playing with the main track volume ("Gain"), the dry path ("Direct") and the EAX low pass filter ("Direct HF")

You can audition the results by simply playing the sequence.

Load the file 'LevelSequenceFINISHED.sac' to hear the sequence with effects applied and levels adjusted.

## Sound Randomizer

The Sound Randomizer is similar to the Sound Event. It chooses from a list of alternative responses to the application's trigger. The concepts of random and ordered selection are carried over from Sound Events. However, in a Sound Randomizer, the alternatives can consist of any object from the following list:

- Audio Sample
- Sound Event
- Sound Timeline
- Sound Queue
- Metric Time Audio Sequence
- Global Effect

Consequently, many of the playback parameters offered in the Sound Event are not available.



As with Sound Events, you can simply drag components from the project tree into the Sound Randomizer dialog box, where you can set chance and order properties.

To see an example of a Sound Randomizer being used in a musical context, open the file *MusicWithRandomizers.sac* and listen to each of the Metric Time Audio Sequences in the Content Bank.

Now open the Sound Randomizer 'Music Randomizer' and examine how it has been put together. Finally, play 'Music Randomizer' looped to hear it in action.

## Miscellaneous other stuff

There's a few other IPS and ISACT features which are very useful

### Voice Count Tool Window

---

The Voice Count Tool is accessed by selecting "View" > "Tool Windows" > "Voice Count" from ISACT's main menu.



This window is useful for keeping track of voice usage. The number of voices in use is displayed next to the maximum number of available voices. The 'peak' number of voices represents the highest number of voices that were in use at one time. You can reset the peak count by pressing the reset button.

### Content Cloning

---

If the nature your project requires you to create a lot of similar content, you can save yourself a lot of time using ISACT's Content Cloning facility: Simply right click on the content you need to replicate and select "Clone" from the menu that appears. An identical copy of the content and all its settings will be created with "<Clone>" appended to its name. If you wish to rename this content you can do so from its properties dialog.

### Content Mixer


---

The Content Mixer window allows you to drag in individual ISACT Content objects and samples and then control their volumes. Take our example of the rocket launcher:

Open WeaponsFinished.sac from the 'Weapons Demo' directory. Let's say we want to alter the volume of the Rocket Launcher sound. Rather than alter the volume of each Sound Event that is in the 'RocketTimeLine' or create a submix group consisting of all those Sound Events and then alter that, it is much quicker do the following:

Select View > Tool Windows > Content Mixer from the menu bar at the top of the screen to open the Content Mixer.

Drag the 'RocketTimeLine' time line into the Content Mixer



Clicking the content icon launches the appropriate edit window for that content

Select 'RocketTimeLine' and play it looped. You now control its volume by using the gain fader in the content mixer.

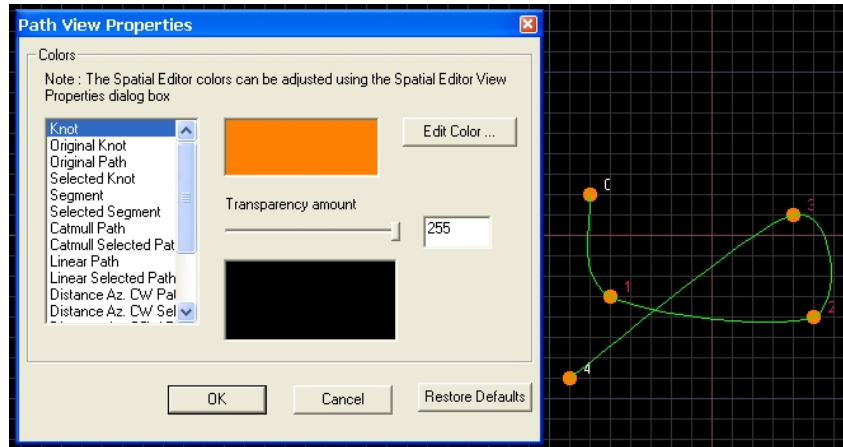
Note that the relative volumes of the Sound Events within the timeline remain the same.

You can drag as many Content objects into the Content Mixer as you want. You can also drag whole folders of ISACT content into the content mixer.

## View Properties

---

You can alter the visual settings for most IPS edit windows. When a dialog window is in focus, go to the "View" menu and select "Properties". The dialog allows you to alter colour and transparency settings for many visual components of the view. Some view properties also allow you to toggle associated warning and prompt dialogs.



## Sting Control

---

The Sting Control allows preview of multiple components simultaneously.



Simply open the Sting Tool Dialog using *View -> Tool Windows -> Sting Control* or the hotkey combination *Ctrl-Shift-N*. Then, drag any content that you want to test into the dialog box. You can start and stop components independently using the transport at the top of the dialog. Multiple-selection allows you to start several objects playing at once. You can drag whole folders of content into the Sting Control to save time.

## Spatial Editor


---

The spatial editor provides an easy way to place hi-level ISACT content in the perceived 3D space surrounding the listener:

Click the R button so the editor saves the changes you make to the position of the selected ISACT content

This square represents the relative position of the selected ISACT content to the listener

To use the spatial editor, you double click on the ISACT content that you want to position such as a Sound Event or a MTAS, then select *View > Spatial Editor* from the main menu.

Then click the red R button in the Spatial Editor dialogue, and click the  icon in the Spatial Grid Controls. Finally, select the blue square that represents the selected ISACT content and move it wherever you wish (the crosshair at the center of the dialogue represents the listener's position).

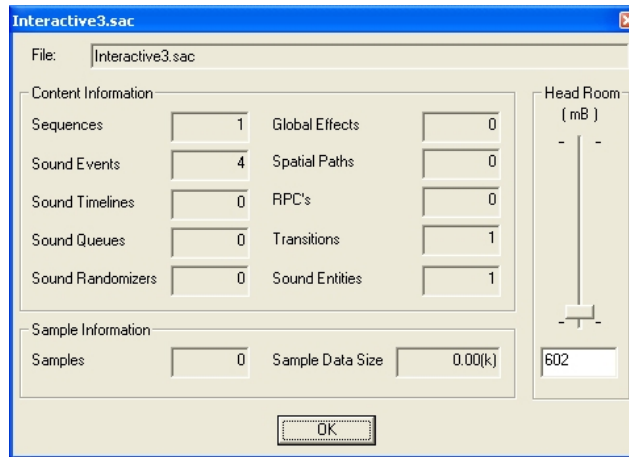
## Boost

---

Traditionally in interactive 3D audio, you have been able to specify the amount of attenuation applied to your sounds, but never to boost the sounds above 0dB – there is no facility for adding gain. Adding gain could enable you to increase the dynamic range of your sound scene, meaning that certain sounds could be emphasised. Now, to actually change this it would be necessary to re-write APIs and software and hardware drivers, the whole 3D audio system – not a realistic proposition. But in order to give you the facility to emphasise certain sounds with boost, ISACT simulates a gain control. It does this by allowing you to specify a global headroom setting for your project. All sounds in the project will be attenuated by a certain amount to provide headroom, allowing boost on particular sounds that require emphasis.

## Setting Global Headroom

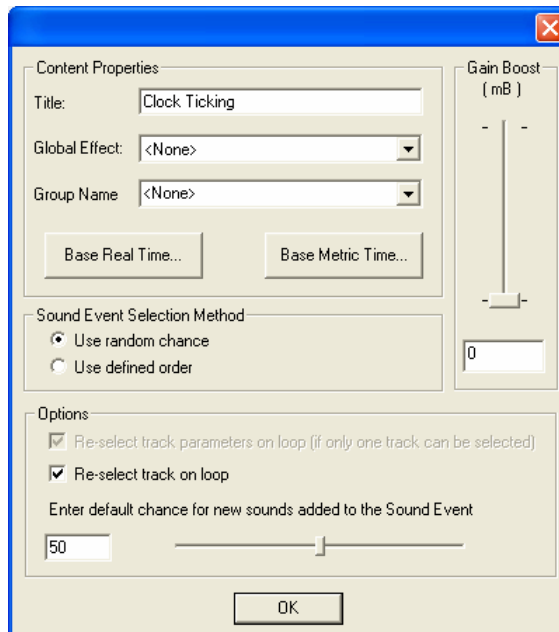
To set the global headroom, bring up the project properties (right-click on the project's root node in the Project Tree and select 'Properties...').



The Head Room slider allows you to set the amount of attenuation applied to non-boosted sounds in millibels.

## Setting per-sound object boost

Now for any sounds that you want to be emphasised, simply bring up the sound component's properties and adjust the Gain Boost setting.



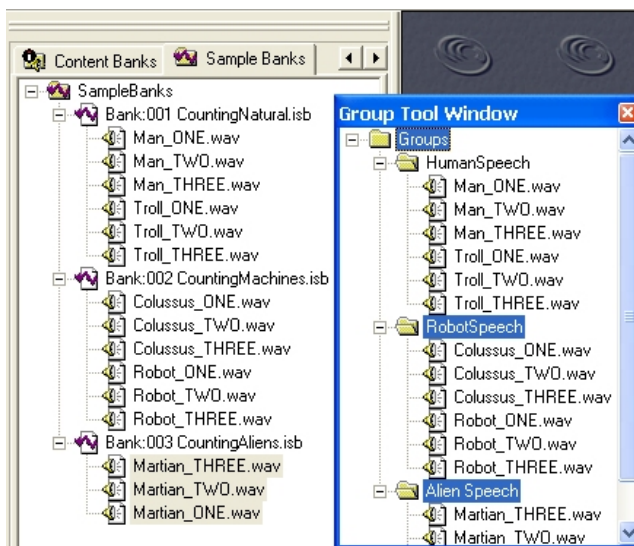
## Groups

ISACT allows you to create sound groups, enabling advanced mixing techniques. Groups have a volume level associated with them, which the programmer can manipulate at run-time. For example this makes it easy to set up interface sliders in your applications audio menu giving the user separate control over the level of different parts of the soundtrack



such as music, effects, speech etc... You can also use groups to implement ducking techniques, where background audio is faded out to emphasise important sounds like speech.

Groups are defined using the Group Tool window. You can bring up this dialog from the "View" menu by clicking "Tool Windows" -> "Group Controls" or using the hotkey "Ctrl-Shift-P".



You can add groups by right-clicking in the Group Tool dialog and selecting "Add Group" from the context menu. Groups are represented by simple folder icons. Groups cannot be nested. To add an ISACT component to a group, simply drag it from the Project Tree or bank where it resides into the appropriate group folder in the Group Tool dialog. Right-clicking in the Group Tool Window displays the context-sensitive menu which allows you to delete and rename a selected group, and remove content objects. Alternatively, every Content Object's Properties Dialogue box allows you to assign it to a Group and / or a Sub-Mix Group (see next section).

For an example of the use of Global Variables and Groups within entities to implement ducking techniques, examine the project "BugSwarmFINISHED\_DUCKING.sac". For a full explanation of how this example works and how to recreate it, refer to the "Implementing Ducking Using Entities with Groups" section at the end of this section.

## Sub-Mix Groups

---

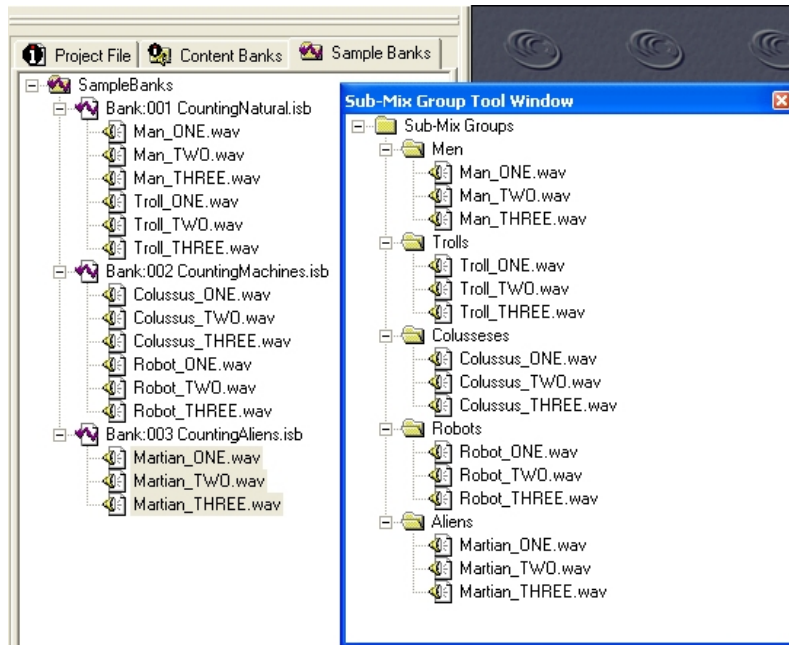
Like Groups, Sub-Mix groups allow you to group sounds together for ease of manipulation. Unlike groups, the volume level associated with them cannot be controlled at run time. The purpose of Sub-mix groups is to allow you as a sound designer to set up mixes that you wouldn't want or need a game user or programmer to be able to alter.

For example, let's say you created a piece of music for a game using ISACT: you could use sub-mix groups to easily control the relative volumes of your stringed instruments, wind instruments and percussion until you have a mix that you are happy with. You wouldn't really want this mix to be able to be altered at run time.

However, you could then add all these instruments to a group called music, thus giving the programmer the option of allowing the gamer to be able to control the overall level of the music at run time. Using this method, the mix of instrument types you created using the sub-mix groups would remain intact.

You can use any ISACT Content Object in a sub-mix group.

Sub-Mix Groups are defined using the Sub-Mix Group Tool window. You can bring up this dialog from the “View” menu by clicking “Tool Windows” -> “Sub-Mix Group Controls”

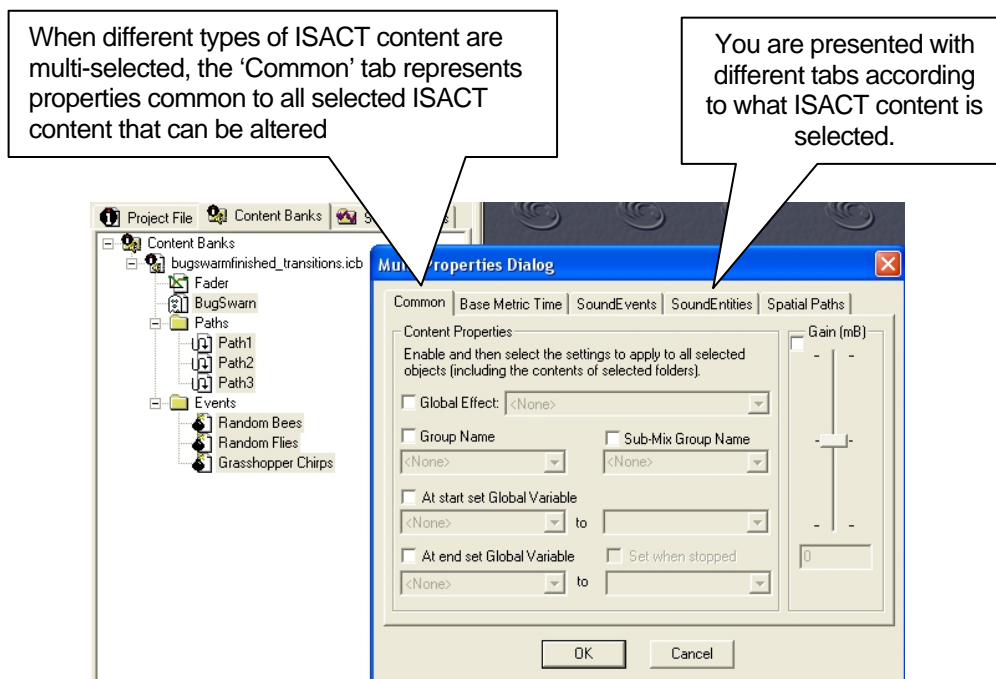


Controlling the volumes of the sub-mix groups you create is very easy. Once your sub-mix groups are set up as you wish, select View > Tool Windows > Sub-Mixer to bring up the Sub-Mixer Tool dialogue. Each Sub-Mix group you created will have an associated fader in this dialogue.



## Multi-Properties Dialogue Boxes

Multi-Properties Dialogue Boxes allow you to select multiple ISACT objects and alter their properties in very convenient manner. You will be presented with different tabs according to what types of ISACT content you have selected, but you will always be presented with a *Common* tab which allows you to alter properties that are common to all the content you selected at the same time. These properties are known as common object properties.



To bring up the Multi Properties Dialog, you select multiple types of ISACT content using the Shift key or CTRL key and then right-click and select the "Properties..." menu option. If you have multiple different types of ISACT content selected, a common tab will be displayed allowing you to alter the common properties of the selected content.

In the above example there are also tabs for 'SoundEvents', 'SoundEntities', and 'Spatial Paths' representing the three different types of ISACT content originally selected. These tabs allow you to alter properties specific to the type of content they represent. The more content types you have multi-selected, the more tabs that will be available.

The Multi Properties Dialog is a great time saver as it allows you to change the properties of many items at once.

Let's examine some of the options on the Common properties tab in more detail:

#### Global Effect

Clicking the Global Effect checkbox allows you to select any Global Effect you have already set up to apply to all the ISACT components that you have multi selected.

#### Group Name

Clicking the Group Name checkbox allows you to assign all your multi-selected content to a group of your choice by selecting it from the drop down menu.

#### Sub-Mix Group Name

As with groups, clicking the Sub-Mix Group Name checkbox allows you to assign all your multi-selected content to a sub-mix group of your choice by selecting it from the drop down menu

#### At Start Set Global Variable

Clicking this checkbox allows you to pick a global variable you have created and automatically set it to the value of your choosing when one of the components that you have multi selected starts playing.

#### At End Set Global Variable

Clicking this checkbox allows you to pick a global variable you have created and automatically set it to the value of your choosing when one of the components that you multi selected stops playing.

#### Gain (mb)

Clicking this checkbox allows you to use the fader to alter the volume settings for all the components that you multi selected

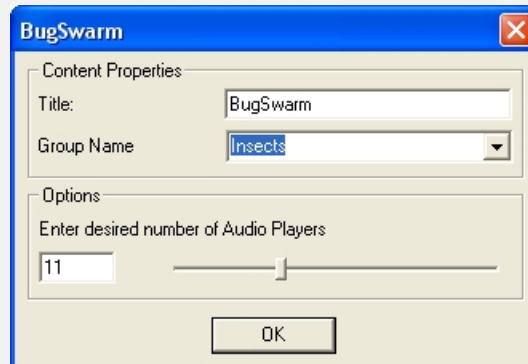
### Implementing Ducking Using Entities with Groups

This example will demonstrate the use of Global Variables and Groups in conjunction with Entities in order to perform a volume ducking effect.

Open the project "BugSwarmUNFINISHED\_DUCKING.sac" and double click the entity 'Bugswarm'.

Select "View" > "Tool Windows" > "Group Controls" from the menu at the top of the screen, and right click the Group folder in the Group Tool Window. Select "Add Group..." and call it 'Insects'.

Now right click the BugSwarm entity, select "Properties..." and select 'Insects' from the drop down Group Name menu thus adding it to the 'Insects' group



One of the action commands available in an entity is called "Adjust Group Gain". This command allows you to set the volume level of the specified group. Now that 'BugSwarm' is part of a group, we can set up some action commands that will alter the volume of anything played in the 'BugSwarm' entity.

Open the BugSwarm Entity and create a new global variable called 'Ducking' and give it two states: "On" and "Off" with "Off" being the default state. It will become clear why 'Ducking' needs to be a global variable soon.

Now create an action called 'Duck Volume' and set "Ducking", "Equal to (=) and "On" as your action conditions. Click 'End'

Set your action type to "Adjust Group Gain", the Group Name to "Insects", type 10 in the Gain box, and 250 in the Fade-time box. Click OK

This action will fade the volume of everything in the insects group down to 10% over ¼ of a second, when global variable "Ducking" is equal to 'On'.

Now create an action called 'Raise Volume' and set "Ducking", "Equal to (=) and "Off" as your action conditions. Click 'End'

Set your action type to "Adjust Group Gain", the Group Name to "Insects", type 100 in the Gain box, and 250 in the Fade-time box. Click OK

This action will fade the volume of everything in the insects group up to 100% over ¼ of a second when global variable "Ducking" is equal to 'Off'.

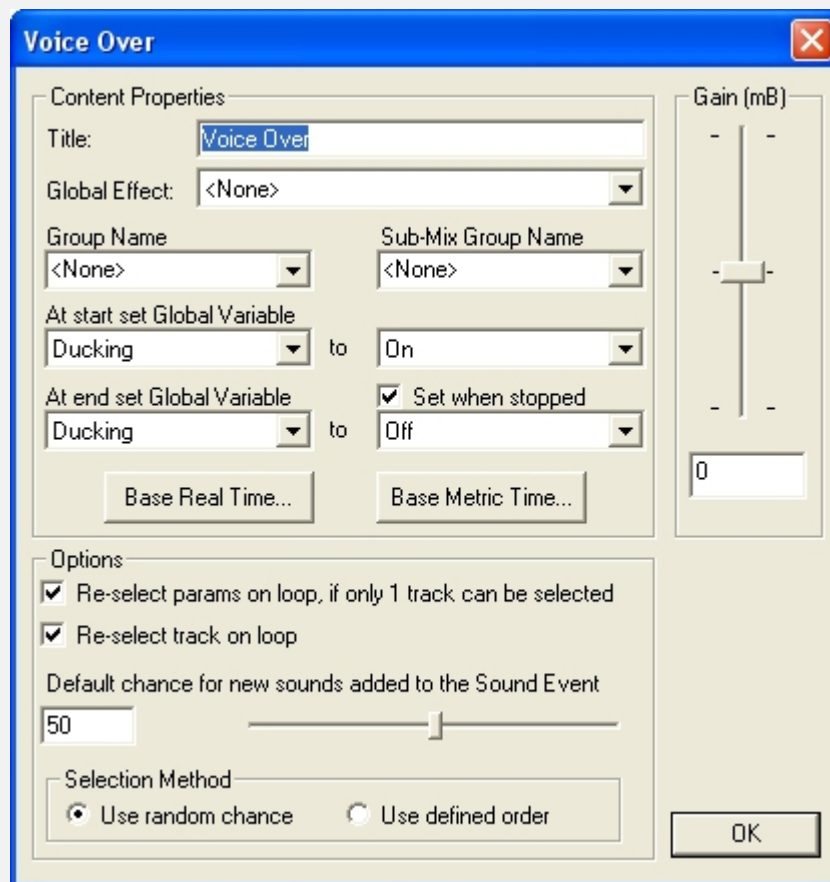
Audition the entity and set 'Swarm' Variable to large. Now set the 'Ducking' global variable to 'On' and you will hear the volume of your insects fade to 10%.

If we want to trigger this ducking effect when another is sound playing, we can do so as follows.

Create a new Sound Event and call it Voice Over. Use the 'At start set Global Variable' drop down menu to select your 'Ducking' global variable and then select 'On' from the drop down menu next to this. This means that when this Sound Event starts playing, 'Ducking' will be set to on.

Now use the 'At end set Global Variable' drop down menu to select your 'Ducking' global variable again, and this time select 'Off' from the drop down menu next to this. Also, click the 'Set when stopped' check box. This means that when this Sound Event stops playing, 'Ducking' will be set to off, and the 'Set when stopped' check box option tells ISACT to run the 'end' action if the content is stopped prematurely.

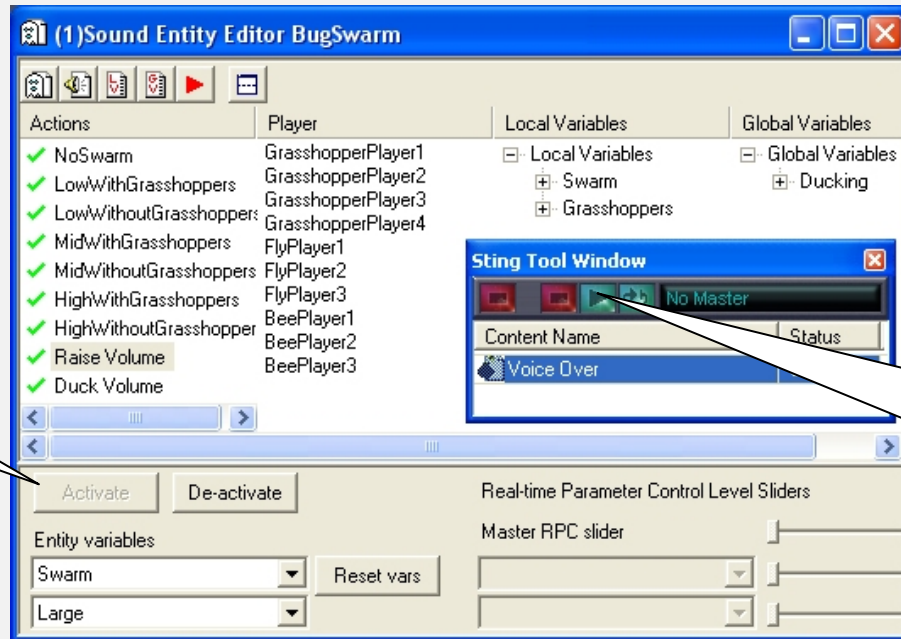
The Voice Over Sound Event properties dialogue should now look like this:



Now add the sample 'Martian\_ONE4.wav' to this Sound Event.

Everything is now set up so lets audition the entity and see ducking in action.

Firstly select View > Tool Windows > Sting Control, and drag the Voice Over Sound Event into the Sting Control dialogue. Now double click the BugSwarm entity and click the 'Activate' button at the bottom left hand corner of the Entity Editor dialogue. Set the 'Swarm' variable to 'large' using the drop down menus to start playing the insect noises. Now select the 'Voice Over' Sound Event in the Sting Control dialogue and press play.



Clicking the Activate button enables the auditioning of the BugSwarm entity

Clicking the play button plays the Voice Over Sound Entity thus ducking the volume of the Insects group.

As you can hear, when 'Voice Over' is playing, the volume of all the insects is ducked, and when it stops playing, the volume of all the insects is returned to normal.

## Creating interactive music with IPS

### Interactive music

ISACT's wave playback synchronisation capabilities and scripting features make it an ideal tool for creating and manipulating loop-based interactive music.

This section shows how you can build music tracks by importing loops into Metric Time Audio Sequences. Whilst playing back this content, you can then move around within or between sequences. Markers help you in setting entry and exit points in order to define distinct passages of music and control the jumps between sections. With a little intervention, ISACT can make certain that jumps only occur on musically sensible boundaries, so the rhythm is never interrupted.

You can control sequence playback from within a Sound Entity. This means you can define how an interactive music track will react to variable state changes. The idea is that at run-time, all the programmer needs to do is start and stop the music track, and pass in information on the program state via state variables. The audio designer can script exactly how the tune will change.

Finally, you can set up other content such as Sound Events etc... to start 'on the beat', synchronised in time with a given Audio Sequence. This way you could introduce musical flourishes and motifs as responses to game events, and have them integrated with your project's music.

Let's jump straight in and look at some loop-based music created using an Audio Sequence. This part of the tutorial assumes some knowledge of the components that make up an ISACT project, such as Metric Time Audio Sequences, Sound Entities, Audio Samples and Transitions. If you haven't done already, you should read and understand the material presented in the first section, Creating Audio Content in ISACT.

#### Our demo music

---

The interactive music that we're using to demonstrate the technique has two intensity levels. Some of the music is fast paced and busy, some of the music is slow and sparse. The idea is to emulate a situation where the pace and intensity of the music will reflect the action in the game.

### Music Loops as Audio Sequences

As you may remember from the previous section, Metric Time Audio Sequences allow you to place Audio Samples on a grid that is arranged by musical time (bars, beats, clocks). Of course, the tempo of your samples must match the tempo defined for the sequence, although using markers it is possible to vary the tempo as the sequence progresses. ISACT has no facilities for changing samples' tempos in real time, but it can analyse a wave file and attempt to determine the sample's tempo.

Since you should already be familiar with techniques such as creating sample banks, sequences and entities, we'll use a set of pre-prepared components to make our interactive music. Go to the tutorial folder and open the music sub-folder. You'll notice a

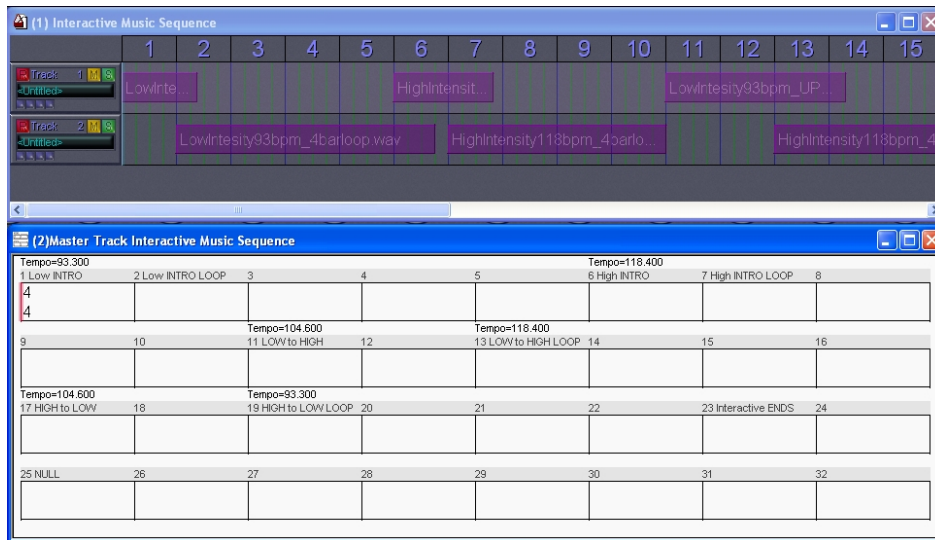


project (.sac) file and a couple of soundbanks. Load the project file, and open the Metric Time Audio Sequence – it’s called “Interactive Music Sequence”.

Note that the sequence we are using for the tutorial comprises a set of stereo loops, but you can just as easily use positioned mono sounds, and all the control settings such as effect sends, spatial paths etc...that were introduced earlier.

## The Master Track

You’ll notice that the loops are spread around the track, seemingly in a rather meaningless order. Don’t dismay, there is method behind the madness. When you’re working with interactive music, the audio sequence just acts as a container for the sequenced loops. The map that really defines how the loops fit together is called the Master Track. To bring up the master track, select the menu option *View->Master Track* or use the hotkey combination *Shift-M*.



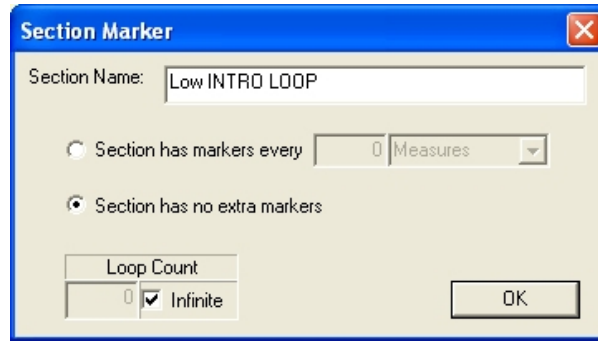
The master track enables you to divide your sequence up into distinct sections. You can give each section a set of properties, and also place tempo and time signature changes over the sequence.

### Sections

A Section defines a set of consecutive bars (known in ISACT as measures), which form a distinct passage of music. To define a new Section within a sequence, right-click on the grey bar in the Master Track, at the position you want the new section to start. Select “Set Section...” from the context menu, and a dialog box will appear enabling your to define your new Section’s settings.

Note that you can erase Section definitions by right-clicking and selecting “Delete Section”. Also, if you select “Set Section...” where a Section already exists, you can alter the Section’s settings.

You can see that the demo music is already split into sections. The Section properties dialog is shown below:-



Each section has a name, which is its unique identifier within the sequence. The loop count property allows you to define how many times a Section should be looped when it is played.

Markers define regular 'exit points'. These are points where the Section can end and another Section can take over playing. If you tick "Section has no extra markers" then a Section can only exit at its end. Alternatively you can allow a section to exit on measure (bar) boundaries, or at an even higher resolution such as beats or clock ticks.

Don't forget that these Sections will be referenced by the programmer or by other ISACT components (Sound Entities) when an audio sequence is being played back interactively. The logic in the Sound Entity or the programmer's code will determine the order in which Sections should be played back, using a Transition component to describe how the changeover should sound. But the Section settings that you define in the Sequence component will describe the behaviour of each region in the Sequence.

When defining your own sections, don't forget that ISACT can't start playing a sequence halfway through an Audio Sample.

### Tempos

As stated earlier, the demo music varies not only in intensity but also in pitch. This allows us to demonstrate how tempo changes can be inserted in the Master Track.

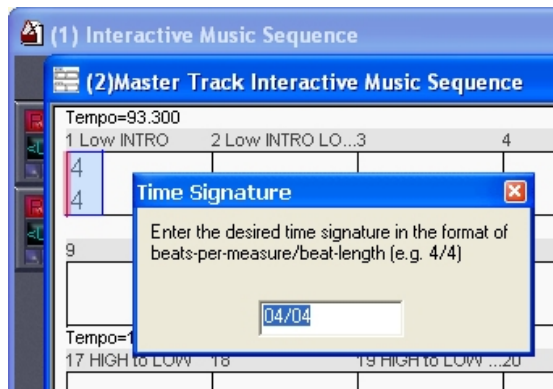
The Tempo area is the white bar above the Section bar. Again, right click to set, alter or delete Tempo settings. Of course there must be a tempo setting present at the start of the sequence, so you can't delete the first Tempo setting.



Our High Intensity loops were rendered at 118 bpm, the Low Intensity sections at 93 bpm. (There is a small discrepancy between the timing in IPS and the timing in the sequencing program that we used to generate these loops. So in IPS these values are actually 93.3 and 118.4 bpm respectively.)

## Time Signatures

You can also set different time signatures during the Sequence. The demo we're using here doesn't change time signature, so we won't be looking closely at this feature. But again, it's as simple as right-clicking in the Time Signature area (inside the grid squares of the Master Track)



The time resolution for changes in Time Signature, Tempo and Section is one measure (bar).

## Testing Section playback

---

To test the configuration of our Sections, we can play back the Sequence using the Master Track to emulate Section change calls. Try it for yourself with this Sequence. Just place the playback cursor back at the start of the Sequence, and then click the Play button on the Transport (or hit space).

You can see from the Master Track that the sequence begins with a short one measure Section called "Low INTRO", followed by a four bar section called "Low INTRO LOOP". If you look at the properties for these two sections, you'll notice that the intro part has no looping specified, whereas the intro loop region is marked to loop infinitely. And you can see the behaviour that this generates by watching and hearing the sequence play. The drum-roll in the intro plays once, and then a four bar loop plays on over and over.

That gets old very quickly! So let's try moving on to another section. In the Master Track, double click in the Section bar where it says "LOW to HIGH". An asterisk "\*" will appear by that section. Next time the current sequence reaches a valid exit point (in this case the next time the four measure loop completes), the play cursor jumps to the start of the "LOW to HIGH" section.

Just like the "Low INTRO" section, the "LOW to HIGH" section acts as a changeover passage, and accelerates the music from low tempo, low intensity mode to high tempo high intensity mode! This is accomplished using a two bar drum fill, during which the tempo is ramped up. Although this sample doesn't have a constant tempo, we cheated a little bit and set the Sequence tempo here to compensate for the length of this drum fill, so that the length of the wave is equal to two measures. ISACT does not currently have support for tempo ramping.

Again as with the intro region, this "LOW to HIGH" region is not looped, but is followed immediately by a looped four bar section.

You will also find “High INTRO” and “HIGH to LOW” sections, plus an “Interactive END” section. There is a small amount of redundancy, in that the loop section of each intensity’s intro is identical to the loop section of the section that changes from the other intensity to that intensity. But since sequences are composed of references to samples, rather than sample data, this is not of any practical concern.

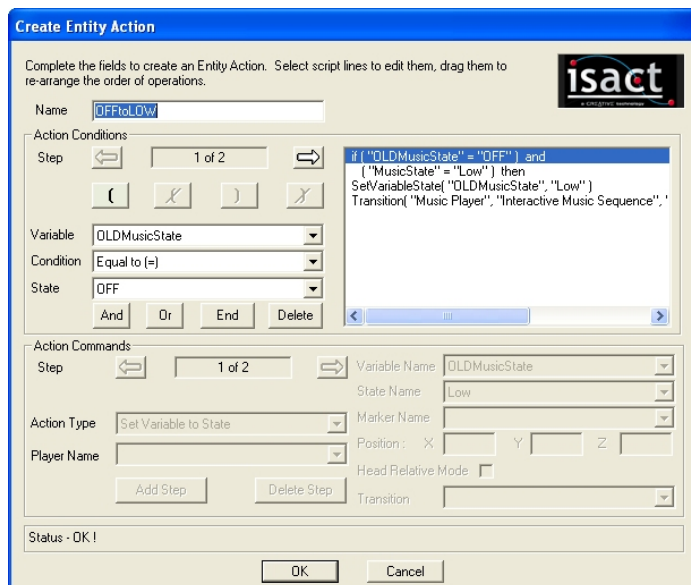
Now you should understand how a Metric Time Audio Sequence could be split into Sections using the Master Track. Its time to see how you can use a Sound Entity to create interactive music, taking input from the application and driving the sequence.

## Interactive music driven by Sound Entities

The scripting capabilities of the Sound Entity make this an ideal component to control playback of interactive music. Sound Entities are able to process input from the calling application in the form of state variables. Sound Entities can also drive Audio sequences at a section level. So it’s not hard to imagine a Sound Entity which plays different sections of a sequence according to a state being passed in.

Open the Sound Entity “Interactive Music Entity” within the interactive music demo project. This is one such Entity! It is controlled by a variable called “MusicState” which can be “High”, “Low” or “OFF” (default). A variable named “OLDMusicState”, with identical states allows us to store the previous state setting so that when the “MusicState” is changed, we can tell what state we are moving from, and what state we are moving to. This enables us to create actions for each state combination.

Open up one of the Actions; say “OFFtoLOW” and you’ll see how the mechanism works.

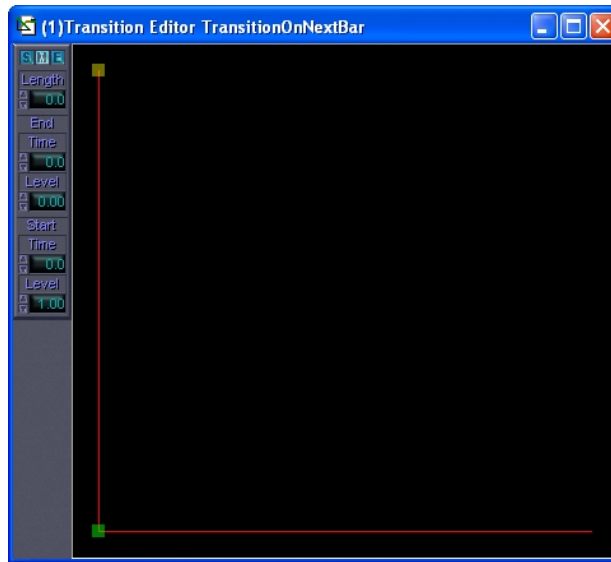


The action checks both the status of the current and previous state. If they match the requirements, then it sets the previous status to the new status, and then calls a Transition to change sections within the sequence.

## Transition, not Play

---

It's important to use the Transition call when you want to move from one section to another, even if you don't want a cross-fade between the two loops. This is because the Transition component also allows you to specify when the new content starts playing in relation to the old content. For this demo, we set up a transition that has no cross fade (so Time = 0.0), and transitions on the next marker. Check out the small icons in the top left of the transition dialog – S means “Start transition immediately”; M means “Start transition at next content Marker”; and E means “Start transition when content finishes”.



You can test the Entity for yourself. Simply go into audition mode, or Activate the Entity, and play round with the “MusicState” variable.

## Stings and motifs

The interactive music demo presented above is a simple affair featuring loops at two different intensities, plus special passages for intros, outros, and changeovers between intensities. However, since we've got this whole interactive tune going on, wouldn't it be nice to be able to add some occasional flourishes, perhaps when a particular game event occurs? And wouldn't it be even better if those flourishes were musical in nature, i.e. synchronised with the main music's rhythm.

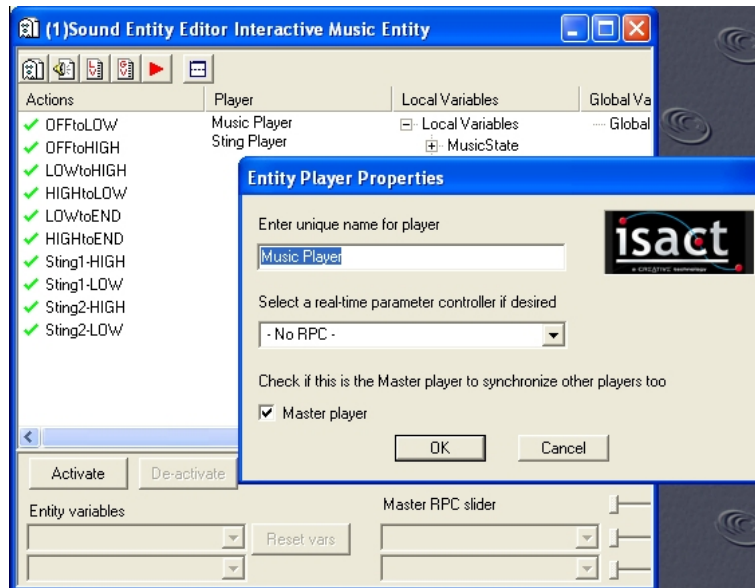
Whenever an Audio Sequence is playing, ISACT is constantly clicking its digital fingers to the musical beat on an internal timer. It is entirely possible to get other content to sync to this beat. To do this, there are a couple of things we have to set up. Firstly we need to define which content sets the beat (let's call that the 'primary' content). And secondly, we have to tell 'secondary' content how to sync to that beat.

## The Master Player

---

ISACT has the concept of the 'master player'. This is the player which other content can sync to. For obvious reasons, there can only be one master player at any one time. The tempo and time signature of the content played on that player will determine the points at which other content can be started.

Within a Sound Entity, you can set the Master Player by bringing up its properties (right-click on the player in the player list) and ticking the “Master Player” box. Now, any content triggered by the entity can sync to this player. Ideally this will be a player that is set up to play a Metric Time Audio Sequence; otherwise the whole thing doesn’t make so much sense!



Note that at run-time there is a potential problem. If there are multiple entities playing, each with a player marked as the master player, then what happens? Well, the answer is that the most recently instantiated master player takes precedence.

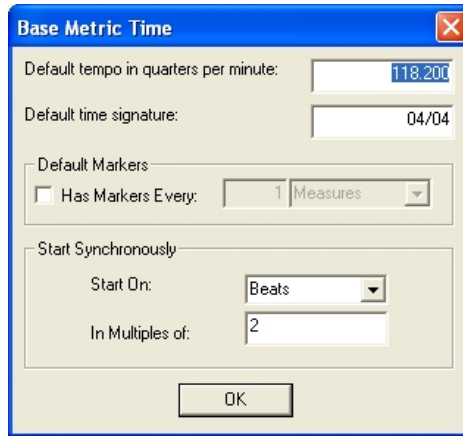
## Starting content on the beat

---

With a master player set up, we now need to tell any relevant secondary content how to synchronise. Of course, it’s up to you to make sure that the secondary content musically matches the primary content in terms of tempo, key, time signature etc... But ISACT can help to synchronise start time, which is the key ingredient that an interactive music playback system must provide.

Each and every bit of ISACT content (apart from the Sound Entity) has a set of properties for start synchronisation. Bring up the content’s property page and select “Base Metric Time”. The dialog allows you to set the time signature and tempo for the content. But more importantly it exposes settings for synchronous start. You can select the boundary type – Clocks, Beats, Markers or Measures – plus the resolution in boundaries.

The demo music project includes some musical flourishes that are named Stings. There are two variations of stings for both low and high intensity music. Check out their property pages, you’ll see how they are set up to trigger on rhythmic boundaries – some at the start of a measure (bar), others every two beats.



## Auditioning stings

---

Now let's play the stings over the music using the Sting Control. First of all open the Interactive Music Entity, and hit the Activate button. Set the "MusicState" variable to High and the music starts playing back. Open the Sting tool (*View->Tool Windows->Sting Controls*, or *Ctrl-Shift-N*). Drag the Stings High sound events into the Sting Control. When you want to hear a sting, select its row in the Sting Control and hit the play icon. The start-time synchronisation will be applied, ensuring that the musical flourish is played on the correct beat and the rhythm is maintained.

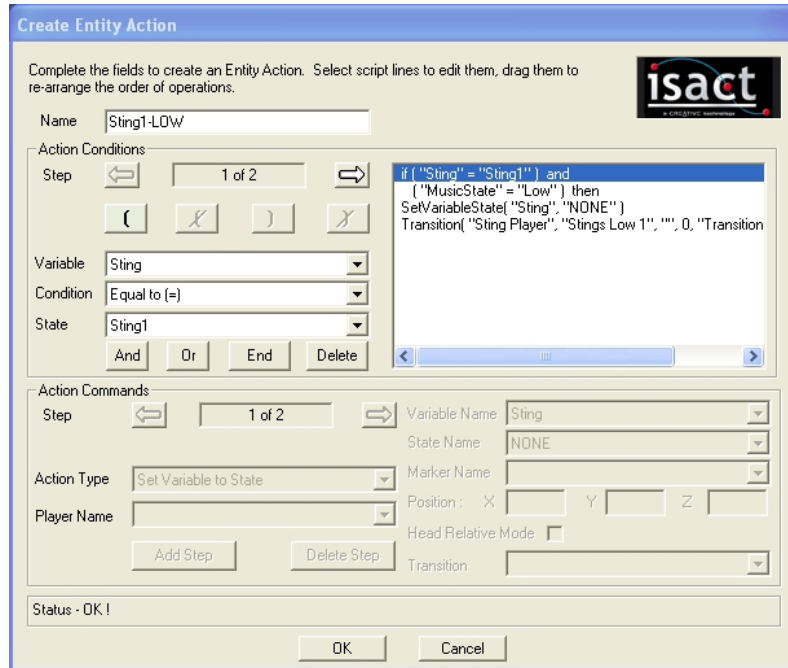
You'll notice that there are variants on the Sting Sound Events for both high and low intensity music.

## Stings within the Entity

---

If you choose to encapsulate your Interactive Music within one Sound Entity, you can also play back individual components such as Sound Events from within the Entity component. This method comes with a big bonus – since the Sting variable is part of the Interactive Music Entity, the Actions that activate the stings can also 'see' the current setting for MusicState.

The Interactive Music Entity in the demo file features a variable called Sting. Check out the actions for playing stings – Sting1-HIGH and so on. Note how actions look for the combination of sting type and intensity and select the sting to play accordingly. Notice also how the Set Variable state command is utilised to reset the Sting variable state once the component has been triggered.



So the application just needs to set the music intensity and request the sting type, the Entity does the job of playing the sting that is consistent with the current intensity and resetting the variable. Try it for yourself!



## Run-time Editing with the Network Connection

ISACT features a very powerful tool for aiding the Sound Design process: Instead of creating lots of ISACT content for a game, then running the game to test how this content sounds, then exiting the game and making more changes, then running the game again and laboriously repeating this process until the sounds are how you want them, you can actually perform run-time editing with ISACT connected to the game over a network and edit your game sounds on the fly.

In order to use the network link from the application to the ISACT Production Studio, you need to use an alternative version of the ISACT runtime component (ISACTWin.dll) that includes network support. In case you are not sure which version of the ISACTWin.dll you are looking at, right-click on the DLL in Windows Explorer and bring up the Properties, go to the Version Tab and look at the 'Comments' entry – if it says "Network Build!" then you know this is the network-enabled version.

Let's try it out:

Exit ISACT

Firstly, you'll need to copy the following files:

C:\Program Files\ISACT\ISACT SDK\Win\Bin\NetworkBuild\isactwin.dll

C:\Program Files\ISACT\ISACT SDK\Win\Bin\AudioDrv.dll

Into this directory

C:\Program Files\ISACT\ISACT SDK\Win\Samples\CTester

Now run the following program

C:\Program Files\ISACT\ISACT SDK\Win\Samples\CTester\CTester.exe

This program plays some content created in ISACT and for the purposes of this demonstration will represent our 'game'.

The following screen will appear.

```
C:\Program Files\ISACT\ISACT SDK\Win\Samples\CTester\Debug\CTester.exe
ISACT Tester
Select an Open AL Device
1. SB Audigy 2 ZS Audio [C800]
2. Generic Hardware
3. Generic Software
1
Select Demo File:
1) ClockTick.sac (Sound Event Test)
2) NetTest (Sound Event, 3D Buffer, Path, and Global Effect Tests)
3) SimpleMusic.sac (Entity, 3D Buffer, Path Tests)
4) CarSimpleFinished (RPC Test)
5) Interactive3 (Transition & MT Audio Sequence Tests)
6) RandomClock (Random Test)
7) EventsAndEffectsInQueues (Queue and Timeline Tests)
2
'x' to exit...
```

Select the appropriate Open AL Device by keying '1' '2' or '3' then 'Enter'

When prompted, select the Demo File "NetTest" by keying '2' then 'Enter'

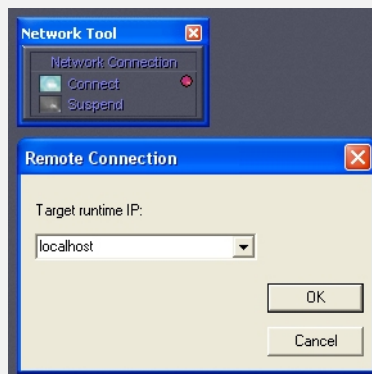
(NOTE : If you have the Windows Firewall installed you may see a Windows Security Alert asking you if you wish to keep blocking the application, select "Unblock" in order to enable the network connection.)

You should be hearing a voice repeating "one, two, three". Keep this program running and now open ISACT and open the project NetTest.sac from the directory:

C:\Program Files\ISACT\ISACT SDK\Win\Samples\CTester

Examine the different components that make up this project under the *Project File* tab. This is the content that the CTester program is currently using.

Now select "View" > "Tool Windows" > "Network" to bring up the Network Tool window, and then click the *Connect* button to bring up the Remote Connection dialog



The dropdown menu is where you specify the IP address of the computer running the game you want to connect to. Since CTester is running on the same machine as ISACT, just type localhost into this box and hit *OK*.

The connect button should now be lit indicating that a connection has been established.

Open the 'Count' Sound Event and change the minimum and maximum pitch settings. You will hear the pitch of the content being played by CTester change.

Try experimenting by changing the settings in the 3D Buffer Controller '3DB1' and in the Spatial Path 'Spatial Path 01' as well. As you make these changes you will hear that what CTester is playing will respond accordingly. Saving the project will make the changes permanent, and the next time you run CTester, these changes will be in place.

As you can see, when designing audio content for a game this is a very powerful feature: it allows you to play the game and identify where you'd like to make alterations to your sounds, then make these alterations using ISACT whilst still in the game until they are just how you want them.

In a game there would be potentially many sounds happening at once. What happens if there are just too many sounds playing for you to be able to accurately alter the sound you want? Simple, you can use ISACTS built in Content Mixer which enables you to isolate the sound you wish to work on and either mute it or solo it.

For more information on the Content Mixer, see the 'Miscellaneous other stuff' section under the 'Content Mixer' subheading.

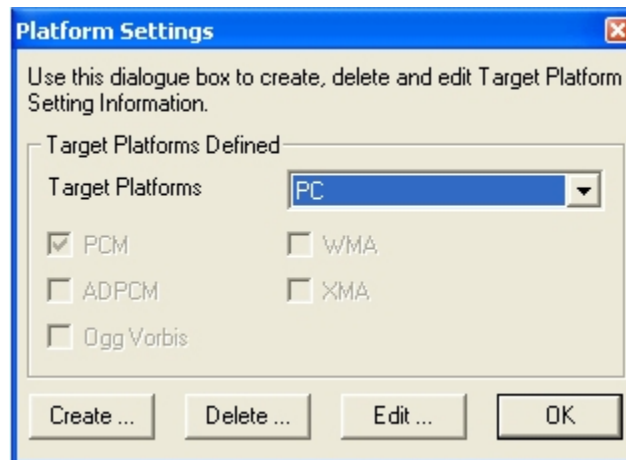
## Target Platform Settings

ISACT allows you to create your own target platforms and select the target compression formats that you want to use on each platform, using the 'Platform Settings' dialog. You can then assign a target compression format to samples on a per platform basis.

During a build, ISACT's Bank Builder application will use these settings and perform the sample compression you specified.

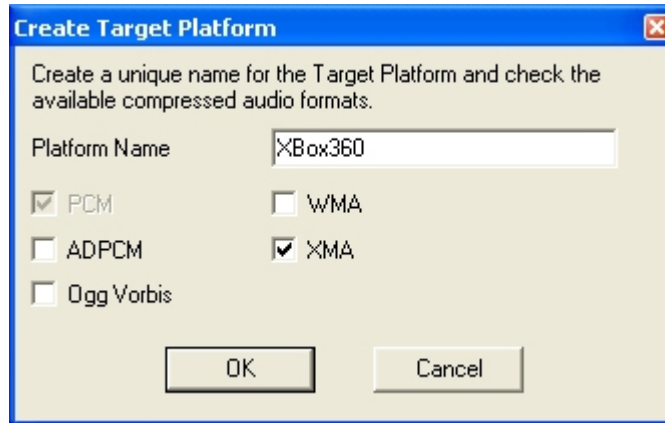
Let's try creating a new platform with appropriate compression formats called Xbox 360:

Access the 'Platform Settings' dialog by selecting "File" > "Target Platforms" from the main menu.



This dialog allows you to view the Target Platforms that have already been defined along with their supported compression types. The "Create...", "Delete..." and "Edit..." buttons give you the options of defining a new target platform, or deleting / editing an existing one.

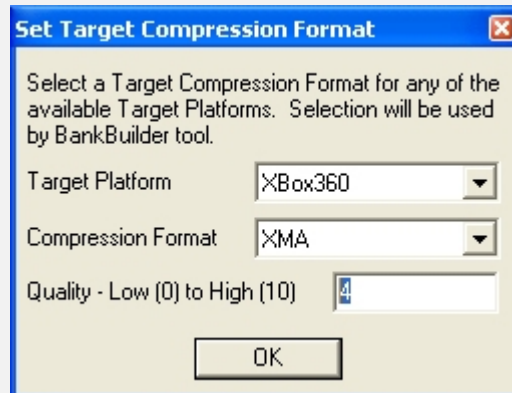
Click the "Create..." button to bring up the 'Create Target Platform' dialog



This dialog allows you to specify the Platform Name and which compression types you want to be supported. In the 'Platform Name' textbox type Xbox360 and select the XMA checkbox. Click OK. NOTE: Not all compression formats are supported on all platforms, e.g. XMA is only available on the 360, and WMA is only available on Windows and Xbox. If in doubt talk to the programming team to find out what compression formats are available for each platform.

Your newly created Xbox360 platform will now be available for selection from the 'Set Target Compression Format' dialog.

Click OK to exit the Platform settings dialog and click on the Sample Bank tab. Right-click on a sample bank and select "Target Format..." from the menu.



Select Xbox360 from the Target Platform drop down menu and select your required Compression Format and quality level.

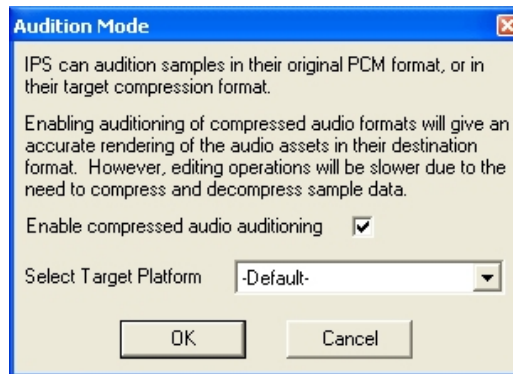
When you perform a build, an Xbox360 bank will be built with the compression format you have just specified.

You can also set the Target Compression Format on a per sample basis by Right-clicking on an individual sample and selecting "Target Format..." from the menu.

## Auditioning of Compressed Audio Formats

ISACT gives you the option of auditioning your samples either in their original PCM format, or in the target compression format you have specified for the sample and platform in question.

If you wish to audition samples in their target compression formats, select "File" > "Audition Mode..." to bring up the 'Audition Mode' dialog.



Check the 'Enable compressed audio auditioning' checkbox. Then select your Target Platform from the drop down menu. Now whenever you play a sample, it will be auditioned with the compression option you have specified for that particular Target Platform and sample. NOTE: Some compression formats, e.g. XMA cannot be auditioned due to the lack of decoders for the PC platform.

Let's say you enabled compressed audio auditioning and selected a target platform such as '-Default-'. Double clicking your Sample Bank would bring up the Sample Bank Editor:

Look at the picture above, if you play BeeChunk2.wav, it will be actually be auditioned as an OGG VORBIS file as that is the Target Format selected for that sample on the '-Default-' platform.



## Conclusions

This tutorial has covered how to create the basic ISACT components using IPS, with practical examples.

You can use Audio Samples to store sound data, with 3D Buffer Controllers holding simple 3D parameters. Sound Events allow you to respond to triggers with dynamic audio, and Spatial Paths provide the capability to move a sound in 3D space. Global Effects allow you to set up EAX hardware effects processing, and Real-time Parameter Controllers can be used to create mappings between input variables and 3D sound properties.

The 3D Sequence enables you to create tempo driven loop based 3D compositions. And the Sound Entity provides complex control over other ISACT components, driven by variables passed by the application.

Sound Events can be sequenced in Timelines, and general Content can be easily strung together using the Sound Queue. The Randomizer enables you to play back any content chosen randomly from a list.

ISACT is a very flexible system, and there's still more that you can do. The most exciting advanced features enable you to synchronise all your content in tempo, helping you to create a truly musical interactive soundtrack in 3D.

### Contact Details

---

To give feedback or ask questions on any aspect of ISACT, get in touch with your local Creative Labs Developer Relations contact.

USA

Liam Byrne, [lbyrne@creativelabs.com](mailto:lbyrne@creativelabs.com), +1 408 546 6452

Europe

Edward Cooper, [edward\\_cooper@cle.creative.com](mailto:edward_cooper@cle.creative.com), +44 (0) 1784 476 775