VOSGAMES

Boom Recorder 7.21 MANUAL

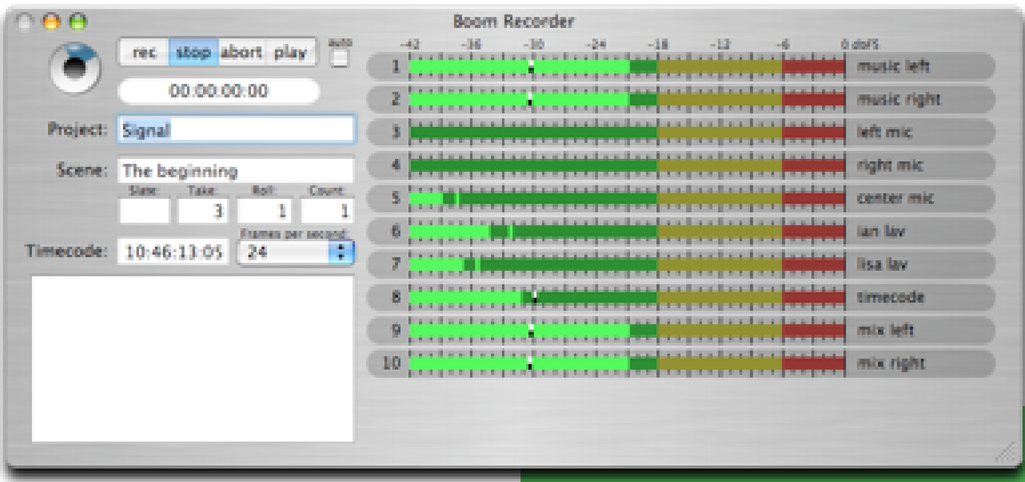
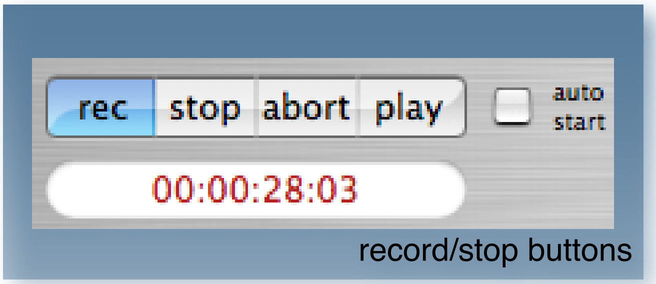


Table Of Contents

1. Recording and Playback

1.1. Tape transport



1.1.1. Rec

You start the recording by pressing the **rec** button, pressing the **⌘⌥R**, ⌘/ keys or pressing the play/pause button on the Apple Remote. When the recording is started the audio files are created and the preliminary metadata is added. Then the audio in the ring buffer, including the pre-record buffer, is added continuously to the file until the recording is stopped.

You can start a recording at any time, but there are times a recording will fail to start. Failures will be handled as if you pressed the abort button. Examples of ways for a start to fail are:

- The audio file(s) already exist, because the metadata produced the same filenames as before. Boom Recorder will never overwrite existing audio files.
- The selected destination folder no longer exists.
- Disk is full.

If you enable the "rec button starts new take during recording" preference, this button will stop and start a new recording automatically. This is useful to manually split recordings during a live concert for example.

1.1.2. Stop

You stop the recording or playback by pressing the **stop** button, pressing the **⌘⌥S** or ⌘/ keys or pressing the play/pause button on the Apple Remote. The audio still in the ring buffer up to the moment when the button was pressed is added to the audio file(s) and finally closed.

At the end of recording, the metadata fields on the main window are read again and the preliminary metadata is overwritten with the new information. The metadata is also written in a CSV (Comma Separated Values) sound log file.

After the recording the **slate**, **take** and **roll** fields are incremented and the **note** cleared, depending on the preferences. The **count** field is reset to one, See 1.1.3. abort for more information.

1.1.3. Abort

You may also stop a recording or playback by pressing the **abort** (or false start) button or by pressing the **⌘⌥A** keys. An abort is also triggered when an error occurred during the recording.

The **abort** function ends the recording in the same way as a normal **stop**, except that the **slate**, **take, roll** and **note** fields are left alone and the **count** field is incremented. When the **count** field is larger than one, the count is affixed to the end of the filename on the next recording. This way it is possible to record a take more than once, without overwriting file.

When an audio files has reached the 2 GB file limit the **abort** function is also called, but the recording will automatically restart. If the time between abort and restart is smaller than the pre-record buffer time the two audio files will be seamless.

1.1.4. Play

You may start a previous recording by pressing the play button or by pressing the ⌘⌥P keys. The audio will be played back on the same channels as when the recording was made, through the currently assigned device outputs. It is only possible to playback recordings that where made in the current session and may be selected from the sound log window.

You may **stop** or **abort** a playback without changing the current metadata. During playback you may also press **rec** to immediately start a new recording. The pre-record buffer is also available during playback. The timecode is like it had been jam-synced before the playback started.

Pressing **play** during playback will stop and restart the playback. It is not possible to press **play** during recording for safety reasons.

1.1.5. Auto

When this box is checked, the recording will start and stop automatically when a timecode signal is found and running. This will have a delay of a few seconds, but this can be compensated for with the pre-record buffer.

1.1.6. elapsed time

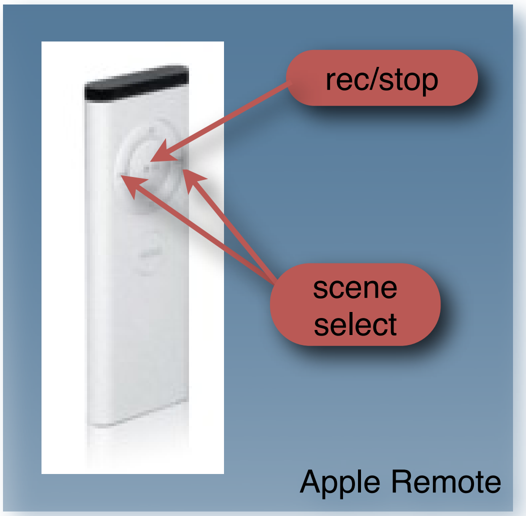
The elapsed time shows how many seconds of audio is recorded or played back. During the recording this counter turns red and shows the actual number of seconds that have been stored on this disk, including the pre-record buffer and excluding the number of seconds still in the ring buffer.

When the counter turns black, the recording is ended and will show the number of seconds that where recorded to disk.

During playback the counter is shown in green and shows how many seconds of the audio you have heard.

1.2. Apple Remote

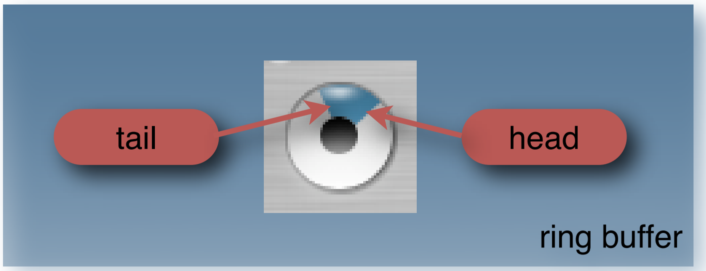
You can also use the Apple Remote to start and stop the recording using the **play/pause** button. You can also stop through the scenes with the **next** and **prev** buttons, the scenes are selected from the pre defined scene list. The take is automatically reset to one '1' when you change the scene to record.



You may want to make sure iTunes is not running, otherwise iTunes will react to the same button presses and start and stop playback at the same time.

1.3. The ring buffer

Because writing audio to a disk can not be done in real time, there needs to be a buffer between the audio interface hardware and the disk. The status of this ring buffer is shown in Boom Recorder as a wedge rotating in a clockwise direction. The wedge is colored blue when idling, red when recording and yellow during playback.



The leading edge is called the head of the ring buffer, here the audio interface is adding audio samples in the ring buffer. The trailing edge or tail is where Boom Recorder has written the audio data to the disk. In playback the head is where the audio file is inserted in the ring buffer, and the tail is where the audio device is playing from.

Normally when recording the wedge would be very thin, and occasionally the tail may slip and then later catch up with the head again. When the recording is stopped the audio that is not going to be written anymore appears as a partial blue wedge. When Boom Recorder is idle the wedge that is shown is used for pre-recording.

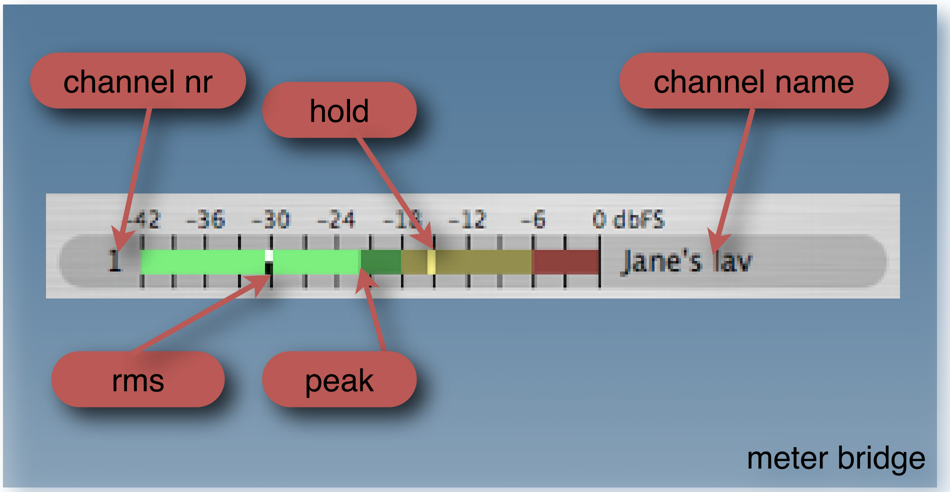
When the head is going faster than the tail; the disk or the computer is not fast enough to keep up with the audio, you may then need to reduce the amount of channels to record or lower the sample rate. When the ring buffer gets filled to 90 % the recording is automatically aborted.

The process that receives audio from the audio interface and adds it to the head of the ring buffer is timing critical. If the process is not finished before the next audio samples are available from the audio interface then audio data will be dropped. The recording will be aborted when this happens because next to the dropped audio, the timecode from that point on will not be aligned anymore.

If audio drops on your system you can either: close other programs that are using system resources or reduce the sample rate. If you are using a notebook you may want to use the external power as the notebook runs slower when on battery power. You may also want to change the buffer size in the hardware preferences.

2. Meter Bridge

The meter bridge shows the arming status, levels and names of the audio channels. The number of channels shows can be either manual selected or automatically, depending on the number of audio interface inputs.



2.1. Channel Number

The channel number of the most left part of a meter. This meter corresponds to the channel numbers in the patch bay.

2.2. Channel name

This shows the name of the channel as it is called by the device. You can change this name to something else if you like. This name is also stored in the meta-data of the audio file, and the names are remembered across invocations.

2.3. RMS

This shows the average volume of the channel over an interval of 0.3 seconds. It is acquired by taking the square value of each sample during the interval, which makes this a true RMS meter. The volume level is calibrated to be equal to the peak when the audio signal is a pure sinus.

2.4. Peak and Hold

This shows the peak volume of the channel over an interval of 0.1 seconds. It is acquired by taking the maximum absolute value of each sample in the interval. As the interval is guided by the user interface update, it is guarantied that you will be shown every maximum value.

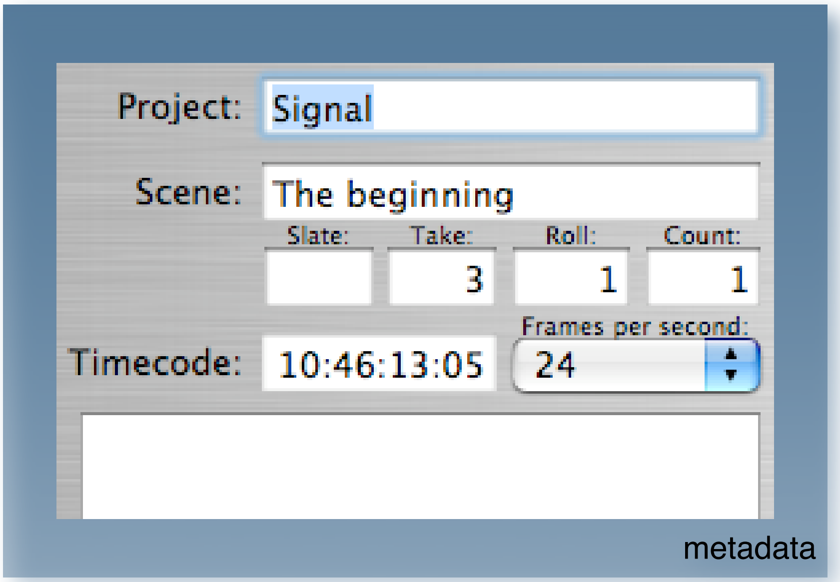
The peak is dampened and will return slowly to the left of the scale, comparable to a PPM meter. As it is difficult to keep your eye on the peak meter, the hold meter will show the highest peak value for 5 seconds. When the 5 second period is over the hold will show the then maximum peak level.

The scale of the meter is measured in dBFS (dB Full Scale), where the right of the scale represents the maximum sample value. Sample values in Core Audio are represented as a number between 0.0 and 1.0. It is possible for sample value to have a higher value when the gain in software is set to high, such as is possible for the microphone on an iBook.

The minimum, warning, error and end values can be configured from the preferences panel.

3. Metadata

Metadata such as: timecode, project, scene, slate, take and roll will be stored in iXML, bext-chunk and a CSV spreadsheet file.



3.1. Project/Scene/Slate/Take/Roll/Note

You set the **project**, **scene**, **slate**, **take**, **roll** and **note** fields in the main window, the values in these fields may contain both digits and letters. However when the **slate**, **take** or **roll** fields are configured to automatically increment, only digits should be used.

These fields may be changed during the recording, when the recording is ended the current values are stored in the audio file.

Although the european **slate** field is added to the iXML- and bext-chunks of the audio files, this is not a standardized field, and there are no edit application that can process this. Check your work-flow to see if you want to use this field.

The titles in the popup part of the **scene** field can be entered in the Scene List Window.

You may use the Roll 2273 Window to calculate a **roll** number.

3.2. Retry field

The retry field increments when a recording is aborted, either by the user or by an error. The retry field is reset to one '1' when the recording is ended normally. You may also edit this field manu-ally.

When this field is larger then one '1' it will be appended to the name of the audio file(s). This so that the next recording will succeed without overwriting the old recording.

3.5. Timecode / Frame rate

This shows the current timecode that will be recorded into the audio files. The timecode is shown in black when a timecode signal is found. If a timecode signal is not found the timecode will free-run in lock with the sample rate and be shown in gray. This means if the sample rate is accurate enough you can jam-sync Boom Recorder at the start of the session.

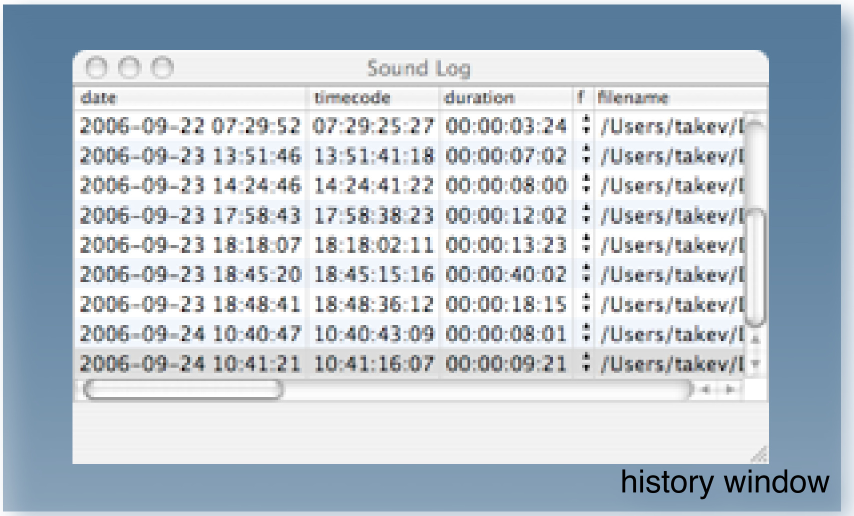
Clicking in the timecode field will change the field to shows the size of a timecode frame in samples. Clicking a second time will show the calculated sample rate in reference to the incoming timecode, this is useful for debugging problems between timecode and clock synchronization.

You should choose the same frame rate as the frame rate of the timecode signal that is fed to Boom Recorder, otherwise the timecode may be off by a full second. The following list shows the frame rate options that are possible in Boom Recorder:

format	description
23.97 fps	New american HD format for pull-down to 29.97 NTSC speed.
24 fps	Film speed.
25 fps	PAL speed.
29.97 fps	NTSC speed (non drop frame).
29.97 fps DF	NTSC speed (drop frame).
30 fps	uncorrected NTSC speed.
30 fps DF	uncorrected NTSC speed (drop frame), weird format.

In some cases the clock send to the audio interface is 47.952 Hz and the timecode is 29.97 fps when the sample rate of the audio hardware is set to 48000 Hz. In this case you will have to set the frame rate in Boom Recorder to 30 fps, this is because Boom Recorder does its calculations in reference to the 48000 Hz.

4. Sound log



You can open the sound log from the **Window** menu or using the ⌘**L** short cut key. In this window you can see the metadata that has been recorded to the audio file during this session.

When you start Boom Recorder all audio files in the selected audio folders are loaded and the metadata is added to the sound log. If you switch audio folders the Sound Log window will reflect the change.

Audio files made with other recorders are also read and their metadata is added to the Sound Log window as well. Not all audio files from other recorders can be read successfully.

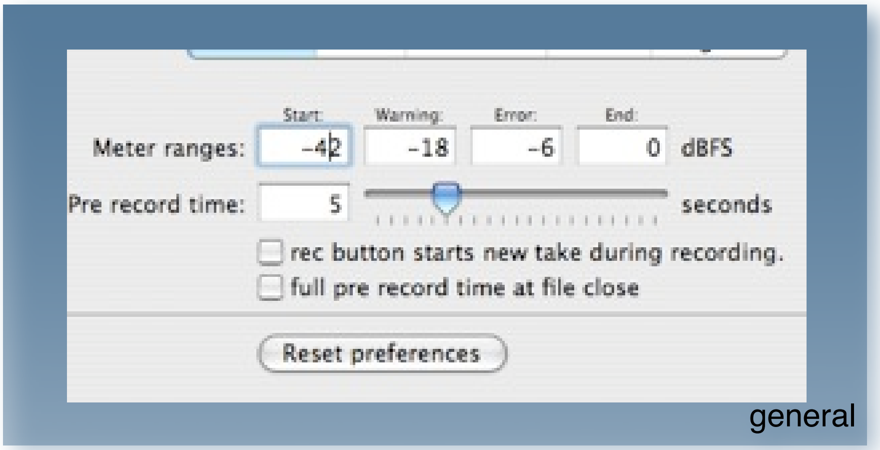
At the end of each recording the soundreport.xml files will show the same recordings as the sound log.

4.1. Reload

When pressing the reload button, Boom Recorder reads all audio files from Boom Recorder's audio folders and populate the Sound Log window. Then it will create new soundreport.xml files in each audio folder with the collected information.

5. Preferences

5.1. General



5.1.1. Meter ranges

Here you can set the dB values for the green, yellow and red parts of the meter scale. The default values are: –42, –18, –6, –0.

5.1.2. Pre record time

This defines the maximum length of the pre record buffer in seconds. Although you can set this value quite large, it will be limited depending on the sample rate and ring buffer length. A large pre record time also limits the playback buffer.

5.1.3. Rec button starts new take

When this function is enabled, pressing the **rec** button during a recording will stop and restart the recording as if a new take has been started. This is useful for manual splitting a recording during a live show.

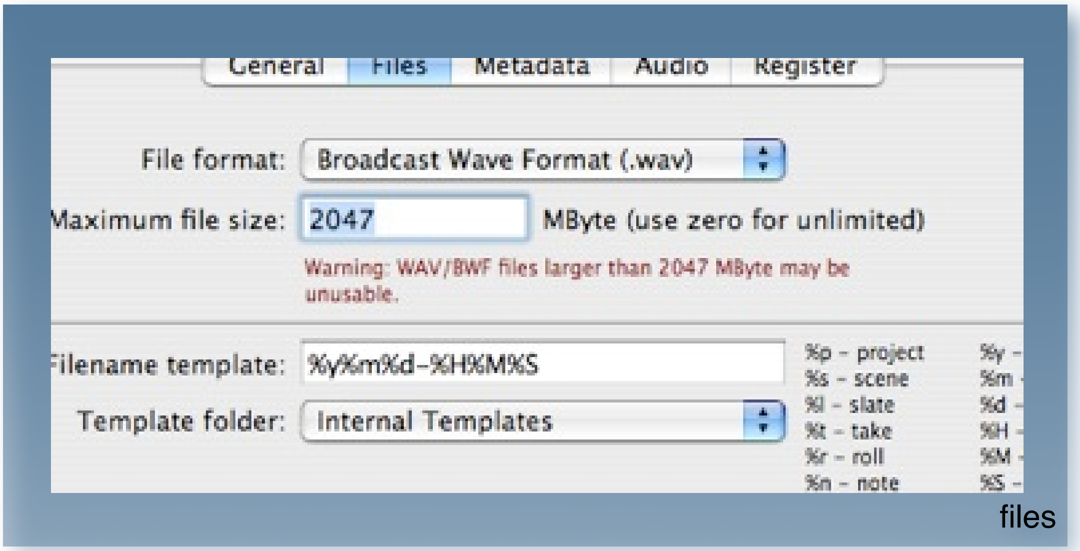
5.1.4. Full pre record time at file close

When this function is enabled the pre record buffer will overlap with the end of the last recording. This will make overlapping audio when a new recording is started because of a file size limit or in combination with the function described in 5.1.3.

5.1.5. Reset preferences

Pressing this button will cause all preferences and settings to be reset to their default values.

5.2. Files



files

5.2.1. File format

You can select the file format that Boom Recorder uses to write the audio into. There are currently two different formats:

- Core Audio Format (.caf)
- Broadcast Wave Format (.wav)

5.2.1.1. Core Audio Format (.caf)

Core Audio Format is Apple's new high resolution multichannel audio files. The most interesting feature of CAF files are that they are very robust; this makes them useful for long continues recordings such as for live event registration.

The frame accurate timecode is recorded in a "sync" marker at the start of the audio data. Other meta-data is stored in a iXML-chunk. Neither the timecode nor the iXML data can currently be read by Final Cut Pro.

5.2.1.2. Broadcast Wave Format (.wav)

The broadcasting industry has settled on a standardized audio format that is a descended of Microsoft's Wave files with extra meta data information embedded into the file.

Both bext-chunk, with sample accurate timecode, and iXML-chunk is added to the BWF file. Final Cut Pro will not be able to read data from either of these chunks. Read the section about the qttc-chunk for a solution for reading timecode in Final Cut Pro.

Normal BWF files have a file limit of 2 GB, which is easily reached when recording at high resolution, Boom Recorder will automatically switch to the next file when this limit is reached. Below is a table showing the maximum file duration in minutes compared to sample rate, sample depth and number of channels:

# ch	48 kHz 16 bit	48 kHz 24 bit	96 kHz 16 bit	96 kHz 24 bit
1	1490 min	992 min	745 min	496 min
2	745 min	496 min	372 min	248 min
4	372 min	248 min	186 min	124 min
8	186 min	124 min	93 min	62 min
16	93 min	62 min	46 min	31 min
32	46 min	31 min	23 min	15 min

5.2.3. Maximum file size

In this field you can tell at which file size Boom Recorder should continue to the next file. This useful if you like to make long audio recording more manageable for example to fit on a CD-ROM or DVD-ROM.

Most audio edit application are not able to work with files larger than 2 GB (some application van handle up to 4 GB). Boom Recorder will automatically change a BWF audio file into a RIFF64 WAV/ BWF file when the file becomes larger than 4 GB. Please check your workflow when you increase the maximum file size to above 2047 MB.

You can find information about the BWF structure and RIFF64 format on the following website:

http://www.ebu.ch/en/technical/publications/userguides/bwf_user_guide.php

5.2.4. Filename template

You can define how a filename is constructed from the supplied metadata in the preferences menu. The filename format consists of normal characters and place holders. The place holders start with a per cent '%' character followed by an optional digit for leading zeroes and a single character from the next list:

code	description
%p	This is replaced by the name of the **project**.
%s	This is replaced by the **scene** code.
%t	This is replaced by the **take** code.
%r	This is replaced by the **roll** code.
%n	This is replaced by the **note**.
%T	This is replaced by the **timecode** at the start of the recording.
%F	This is replaced by the **timecode format**.
%y	This is replaced by the current **year**.
%m	This is replaced by the current **month**.
%d	This is replaced by the current **day**.
%H	This is replaced by the current **hour**.
%M	This is replaced by the current **minute**.
%S	This is replaced by the current **second**.
%c	This is replaced by the channel name that is recorded.
%h	This is replaced by the channel number.
%f	This is replaced by the file number, this overrides the standard “-f01” suffix, so that you may put this in an other form and position.

By default an audio file uses the date and time as filename, this ensures that each recording will have a unique file name. The default sound log filename contains the date, thus will create a new sound log for each day.

5.2.5. Template Folder

After a recording Boom Recorder will write a soundreport.xml file in each audio folder. The sound report contains information for each audio file in the audio folder. You can force Boom Recorder to write the soundreport.xml from the Sound Log window.

The XML structure of the soundreport.xml has the following structure:
- <SESSION>, this is the root element.
- <RECORDING>, this element is added for each recording.
- <BWFXML>, this element contains the complete iXML chunk. it is added for each file in the recording.

This soundreport.xml is converted into other file formats using XSLT templates. Currently three templates exists by default.
- soundreport.html.xsl, creates soundreport.html
- soundreport-long.html.xsl, creates soundreport-long.html
- soundreport.csv.xsl, creates soundreport.csv

When you press **⌘P** Safari is opened with the soundreport.html from the first audio folder. You can open the soundreport.html manually with Safari or Firefox. The soundreport-long.html contains an exhaustive description for each audio file created in this folder.

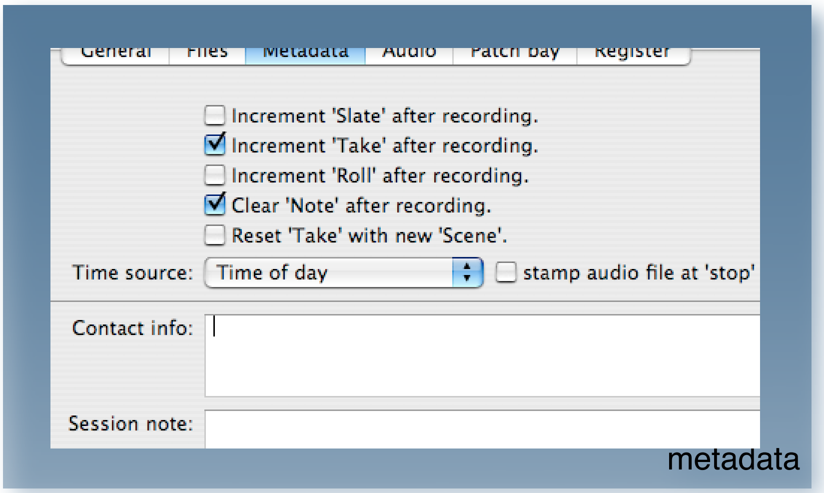
If you like to email the sound report or include it on the media with the audio, you will need to include soundreport.css together with the soundreport.html or soundreport-long.html.

The soundreport.csv is a Comma Separated Values file that can be read into spreadsheets and databases.

You can override the internal XML templates by selecting your own template folder, the files in this folder are copied at the and of each recording then the soundreport.xml file is processed by all XSL files in this folder. The filename is simply the filename of the xsl template with the ".xsl" extension chopped off.

5.3. Metadata

This preferences panel is used to configure how the metadata is used.



metadata

5.3.1. Increment 'Slate/Take/Roll' after recording

When enabled, the corresponding counter will increment after the recording is stopped.

5.3.2. Clear 'Note' after recording

When enabled, the note field is cleared after the recording is stopped.

5.3.3. Reset 'Take' with new 'Scene'

When changing the scene name, the take is reset to '1'.

5.3.4. Time source

Boom Recorder will be able to receive the timecode in different ways, currently there are six options:

setting	description
Zero	The timecode is always 00:00:00:00.
Session	The timecode starts counting from the beginning of the session.
Time of day	The timecode is the current time.
LTC SMPTE/EBU	The timecode is read from an audio channel.
Core Audio SMPTE	The timecode is read from the SMPTE field.
Core Audio word clock	The timecode is read from the word clock field.

5.3.4.1. Zero

The timecode in the display and in the audio file will be 00:00:00:00.

5.3.4.2. Session

The timecode will start freewheeling at the moment Boom Recorder is started.

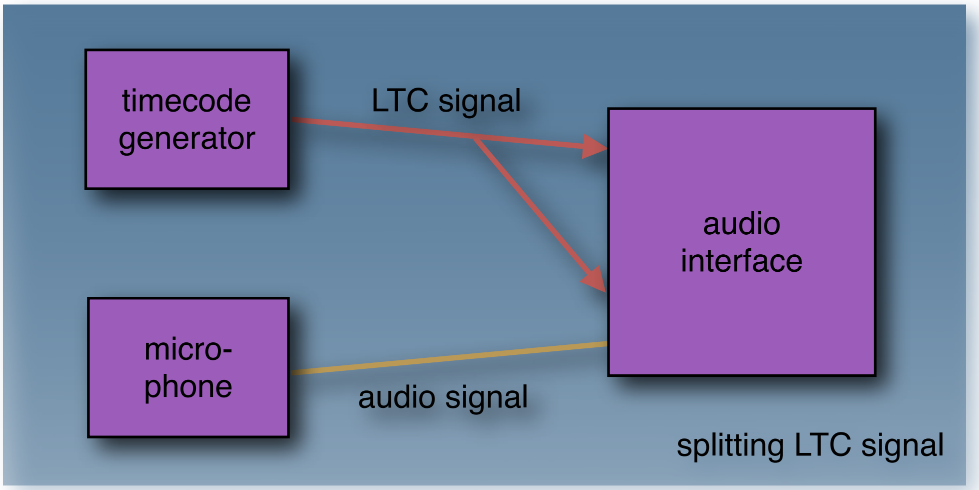
5.3.4.3. Time of day

The timecode will be set to the time of day each time a device property, such as the sample rate, changes. After this the timecode will be freewheeling in lock step with the sample rate of the audio interface.

5.3.4.4. LTC SMPTE/EBU

In this mode Boom Recorder will look for an audio channel on the selected audio interface, until it finds a LTC SMPTE/EBU timecode. When it finds the audio channel with the timecode it will show the current time in the Metadata tab in black.

If Boom Recorder loses its lock on the LTC (because of a bad signal, for example) the current time will freewheel and show in grey until it finds the LTC signal again. Don't worry when the LTC signal is lost during the recording, the timecode is only recorded in the audio file at the start of the recording.



In the LTC from channel mode the sample rate is not synced against the LTC signal, this is the job for the audio interface and you can select this from the **Clock Source** menu.

If you want both the timecode and sample rate sync, you need to split the LTC signal and feed one to an audio channel for the timecode and the second you feed to the clock input of the audio interface. Together with the BWF file format, this is the best way to get sample accurate timecode (Boom Recorder uses the front edge of the first bit of a timecode frame as the point where that timecode is located).

5.3.4.5. Core Audio SMPTE

The timecode is read from the SMPTE field during the I/O cycle, not every audio interface implements this feature.

5.3.4.6. Core Audio word clock

The timecode is read from the word clock field during the I/O cycle, not every audio interface implements this feature.

5.3.5. Stamp audio file at 'stop'

Normally the timecode that is read when you press the **rec** button is used to timestamp the audio file. With this option enabled the timecode that is read when you press the **stop** button is used instead.

5.3.6. Contact Info

This field can be filled in with the name and address of the recording engineer. This field is recorded in the iXML file, and is showed at the top of the page in the soundlog.xml file.

5.3.7. Session Note

This is a note that should be constant during the complete session. This field is recorded in the iXML file, and is showed at the top of the page in the soundlog.xml file.

5.3.8. bext-chunk

A bext-chunk is what sets a BWF (Broadcast Wave Format) file apart from a normal WAV file. More information about BWF files and in particular the bext-chunk can be found here:

http://www.ebu.ch/CMSimages/en/tec_doc_t3285_tcm6-10544.pdf

Here is a list of information that is recorded by Boom Recorder in this bext-chunk:
- description (more on that later)
- originator ("Boom Recorder")
- origination date (from the system date)
- origination time (from the system time)
- timecode (number of samples since 12:00 AM)
- coding history (how it was delivered by Core Audio, and put in the audio file)

These days the description field is used for extra metadata that was not included in the bext-chunk when they defined the standard. Because there is little space in the description field and because there is no clear standard, a better solution to store this extra information is in the iXML-chunk, which you can read about in the next section. The extra information that is added to the description by Boom Recorder is:
- SCENE
- SLATE
- TAKE
- TAPE (the roll field)
- SPEED (the timecode format)
- FILENAME (the name of the filename)

5.3.9. iXML-chunk

This new chunk replaces the overloaded description field in the bext-chunk and has been designed for the specific need of film recording. The metadata is stored in a readable self describing format called XML. More information about the iXML-chunk can be found here:

http://www.ixml.info/

The following is a list of tags that are supported by Boom Recorder, fields preceded with the star symbol '*' are not defined in the official iXML standard:
- IXML_VERSION (1.5)
- PROJECT
- SCENE
- *SLATE
- TAKE
- TAPE
- *COUNT (the retry field)
- FILE_UID
- NOTE
- SPEED.TIMECODE_RATE
- SPEED.TIMECODE_FLAG
- *SPEED.TIMESTAMP_TEXT (textual representation of the timestamp)
- SPEED.TIMESTAMP_SAMPLES_SINCE_MIDNIGHT_LO
- SPEED.TIMESTAMP_SAMPLES_SINCE_MIDNIGHT_HI
- SPEED.TIMESTAMP_SAMPLE_RATE
- SPEED.DURATION_SAMPLES
- *SPEED.DURATION_TEXT (textual representation of the duration)
- *SPEED.HARDWARE_SAMPLE_RATE (the actual rate of the audio interface)
- SPEED.DIGITIZER_SAMPLE_RATE (the sample rate after sample rate conversion, sample rate conversion is not yet implemented in Boom Recorder)
- SPEED.FILE_SAMPLE_RATE
- *SPEED.AUDIO_BIT_DEPTH
- *SPEED.AUDIO_ENCODING
- SYNC_POINT_LIST.SYNC_POINT.SYNC_POINT_TYPE ("ABSOLUTE")
- SYNC_POINT_LIST.SYNC_POINT.SYNC_POINT_FUNCTION ("SLATE_GENERIC")
- SYNC_POINT_LIST.SYNC_POINT.SYNC_POINT_COMMENT ("TC1")
- SYNC_POINT_LIST.SYNC_POINT.SYNC_POINT_LOW

- SYNC_POINT_LIST.SYNC_POINT.SYNC_POINT_HIGH
- SYNC_POINT_LIST.SYNC_POINT.SYNC_POINT_DURATION (0)
- *FILE_SET.FILENAME (filename, without folder)
- FILE_SET.TOTAL_FILES
- FILE_SET.FAMILY_UID
- FILE_SET.FAMILY_NAME
- FILE_SET.FILE_SET_INDEX
- TRACK_LIST.TRACK_COUNT
- TRACK_LIST.TRACK.CHANNEL_INDEX
- TRACK_LIST.TRACK.INTERLEAVE_INDEX
- TRACK_LIST.TRACK.NAME
- BEXT.* (for supported fields see the bext-chunk description)
- VOSGAMES.CONTACT_INFO (contact information of the recordist)
- VOSGAMES.SESSION_NOTE (note about the complete session)

5.3.10. qttc-chunk

Final Cut Pro 5.1.2 now can read timecode from BWF natively without the need for the BWFImporter.component plugin. If you have Final Cut Pro, please remove the plugin from the /Library/QuickTime folder.

Boom Recorder will also write a special chunk for work flows with Final Cut Pro involved. Final Cut Pro uses QuickTime to import BWF files, however it currently can not import the timecode, because the format of the timestamp is incompatble with QuickTime's. This chunk will store the timestamp in a QuickTime compatible way.

Using the BWFImporter.component bundled with Boom Recorder, Final Cut Pro will be able to read BWF files created with Boom Recorder and work with the timecode.

After you place the BWFImporter.component in /Library/QuickTime, QuickTime's default WAVE file reader is overridden and Final Cut Pro will be able to read the timecode. If you remove the BWFImporter.component from /Library/QuickTime, the original WAVE reader will be in use again.

With the introduction of QuickTime 7.1 a new QuickTime media handler chunk 'tc64' is been added when reading a BWF file. The current version of Final Cut Pro can not yet use this chunk for timecode.

The following table shows the format of the qttc-chunk, that may be implemented by other application developers to make compatible products.

offset	format	description	value
0	uint32_t	chunk ID	'qttc'
4	uint32_t	chunk size	20
8	uint32_t (big endian)	number of frames since midnight	
12	uint32_t	duration of a second	
16	uint32_t	divider	
20	uint32_t	frames per second	
24	uint32_t	timecode flags	

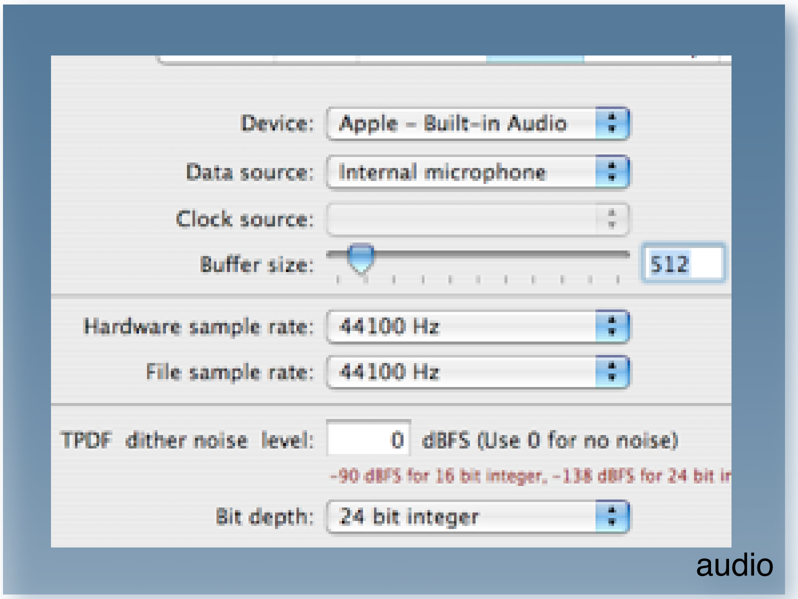
The number of frames-since-midnight-field is the conversion of number of samples since midnight that is stored in the bext-chunk, to frames per second. The number of frames need to be compensated for drop-frame if this is in use.

This integer needs to be stored in big endian format as QuickTime expects it in this format. The next four fields describe the number of frames and are normal little endian integers.

Below is a list on what needs to be in each of those four fields for each timecode format.

format	duration	divider	fps	flag
23.97	24000	1001	24	0
24	24000	1000	24	0
25	25000	1000	25	0
29.97	30000	1001	30	0
29.97 DF	30000	1001	30	1
30	30000	1000	30	0
30 DF	30000	1000	30	1
59.94	60000	1001	60	0
60	60000	1000	60	0

5.4. Audio



5.4.1. Device

You can select the audio interface from the device menu. After the selection this audio interface is initialized and all the other menus will show the current status of this device.

Please check the http://www.vosgames.nl/products/BoomRecorder/ website for the current compatibility with audio interfaces. Please let me know if you have changes for this list.

5.4.2. Data source

Some audio interfaces have the possibility to choose the data source, such as line or microphone. You can select these from this menu.

5.4.3. Clock source

Here you can select how the audio interface locks its sample clock. This is useable to get multiple digital devices in sync with each other. The clock source is different from the time source in that the time source selects where the timecode is coming from, while the clock source is used for synchronizing devices.

5.4.4. Buffer size

Here you can select the buffer size used by the device, it is possible this value will reset to some default value each time you start Boom Recorder. In most cases this default value is correct, if during recording you get a "Processor overload" error, you may want to change this value.

A larger value here does not always solve the problem, sometimes a lower value is better. Your milage may vary.

5.4.5. Hardware Sample rate

Most devices have multiple sample rates you can choose from, which you can select from this menu.

Often used sample rates are:

rate	description
44.1 kHz	CD-audio
48 kHz	DAT-audio (used for film)
88.2 kHz	double CD-audio
96 kHz	DTS and Dolby Digital film audio
192 kHz	high definition audio

Some audio interfaces have continues ranges of sample rates, these are not supported at this time. However, it may be possible to set the sample rate using an other application (the correct sample rate will than not be shown in Boom Recorder).

5.4.6. File Sample Rate

Here you can select the sample rate the audio file is stamped at, this would allow an audio editor to play the audio at a different speed than it was recorded at. This is useful for pull-up or pull-down.

5.4.7. TPDF dither noise level

This is the amount of noise to add to the audio before it is truncated to the required bit depth. This lessens the amount of distortion caused by quantization in exchange for a slightly higher noise floor.

In Boom Recorder this noise is added to the audio before the audio enters the ring buffer, in this way you can see the added noise on the meter bridge and hear it through the monitor.

The white noise is generated by hashing a counter using MD5 adding two of these random values together, like throwing two dice. The resulting noise is called TPDF (Triangular Probability Density Function). This type of noise is the best one to use if the audio is processed in any way after truncating. Other colors of noise or noise shaping should only be used after all processing have been done. For this reason Boom Recorder does not support other colors of noise or noise shaping.

The amount of noise to add is depended on the number of bits you would like to truncate to. The goal is to randomly throw the least significant bit.

bits	noise level
16	-90 dB
24	-138 dB

If you do not like to dithering at all, use the value 0.

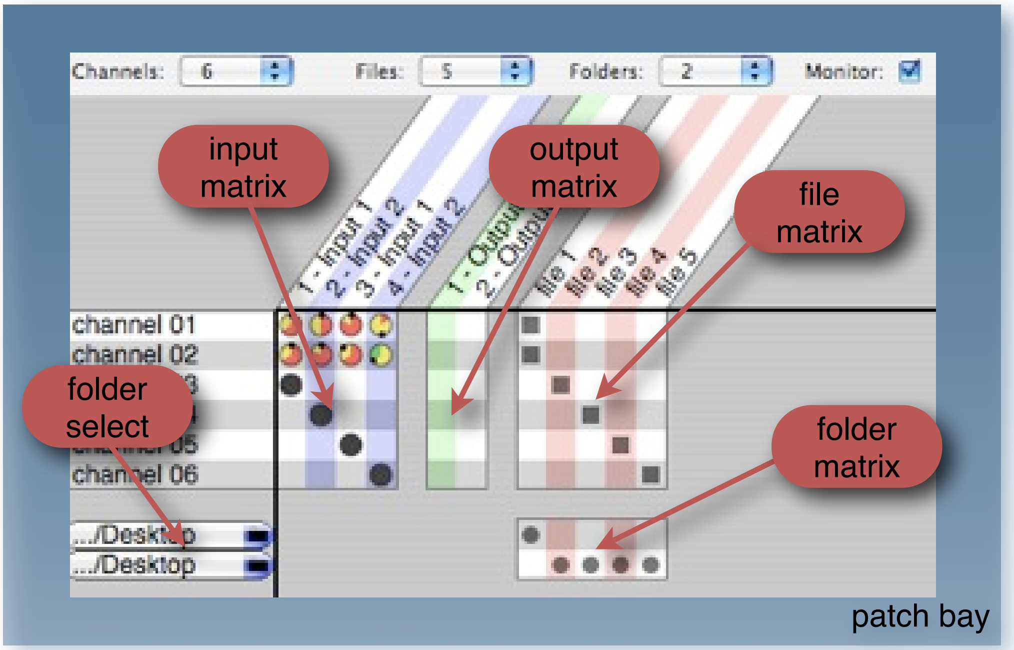
5.4.8. Bit depth

You can choose 16 bit integer, 24 bit integer or 32 bit float sample depths. The 16 bit and 24 bit sample depths are the sample formats that have been used for a long time as the default sample depths for most file format.

The new 32 bit floating point sample depth has a much higher dynamic range because the sample data is stored in an exponential format; more natural to our hearing. This format is the native format of Core Audio, and the 16 bit and 24 bit samples are truncated from these 23 bit floating point sample values. It is thus more natural to use the 32 bit floating point sample depth. However most production flows will only handle 16 bit or 24 bit samples.

5.5. Patch bay

The patch bay is where you tell Boom Recorder what to record and where to store the recording. In other words it shows how the audio is routed from the audio interface into the audio files.



5.5.1. Channels

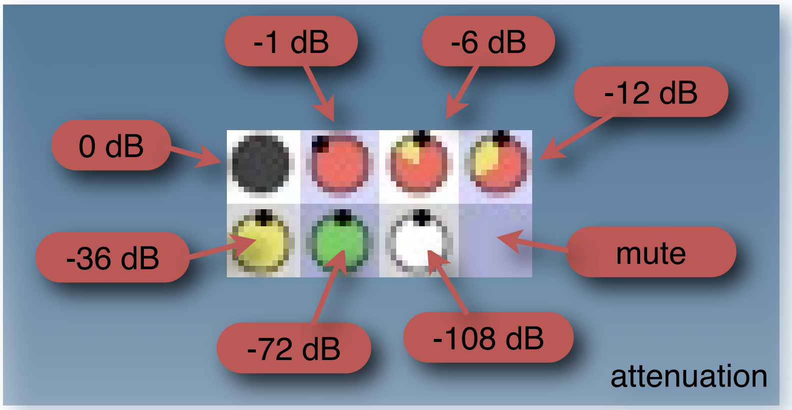
With the **Channel** pull down field you can select the number of channels Boom Recorder will use. You can see the channels as the bus of Boom Recorder. The channels make up the rows of the in-put-, output- and file-matrix and alternate between white and grey. Each channel is also shown on the main window with a meter.

With Boom Recorder Pro you can select up to 64 channels, with Boom Recorder Lite you can only select up to 2 channels. No channels are shown if you have not licensed Boom Recorder and the trial period has passed.

5.5.1. Input Matrix

All the inputs of an audio interface are placed in columns in the input matrix and alternate from white to blue. Each cell in the input matrix connects an input from the audio interface to a channel. To connect an input to a channel you click in the corresponding cell. Each connection you make will cost some processing time.

You can change the attenuation by 1 dB of each connection by pressing the '–' and '+' keys on the keyboard while hovering the mouse pointer over the corresponding cell. A one decibel change is shown as a dot rotating 1/6th around the circle, a six decibel change is shown as a 1/6th change of a pie slice, a 36 decibel change is shown in different colors.



5.5.2. Output Matrix

The output matrix is pretty much the same as input matrix, except it routes the channels back to the audio interface. The output channels are used when monitoring or playing back a recording.

5.5.3. File Matrix

With the **Files** pull down field you can select how many audio files need to be created with each recording. Files are shown as alternating in red and white columns in the patch-bay. In the file matrix you can select which channels are recorded into each file. In the example above a total of five files will be created in each recording, 1 stereo file and 4 mono files.

5.5.4. Folder Matrix

From the **Folder** pull down field you can select into how many folders the audio files can be stored. Folders are shown in rows below the channel rows. An open dialogue appears when you press on one of the **folder select** buttons.

The cells in the **folder matrix** connect each file to a destination folder.

5.5.5. Monitoring

When monitoring is enabled the outputs and inputs are coupled through the channels while in idle or recording. There is a small latency involved as the input and output are buffered by the audio interface and its drivers.

5.6. Register

5.6.1. Rregistered To

Put here the name you have received when you ordered Boom Recorder.

5.6.2. Registration Code

register menu

Put here the registration code you have received when you ordered Boom Recorder.

5.6.3. Buy Online

This button will open the VOSGAMES store web page: http://www.vosgames.nl/store/

5.6.4. Verify

After you have entered the registration code, you can press the verify button to start the verification process.

6. Scene List

You can add a pre defined scene list in this window. This allows you to select the scene name quickly from the main window.

You can add a new scene title, remove a scene title and insert a scene title.

7. Roll 2273

Roll 2273 may be a solution for getting a unique roll number. The algorithm is simply incrementing the roll number by ten '10' for each day since January 1, 2000. Then add the media number.

The only limitation is that you will not be able to create more than 10 media per day.

In the Roll 2273 you can calculate the roll number by selecting a date and entering the media number. You can press **Today** for the roll for today's date. You can then set the roll in the main window using the **Set roll** button.

You can also type in the **Roll** field in this window to calculate the date. You will have to press the **Return Key** after you type in the text fields for the change to commit.

roll 2273

8. Spectrum Analyzer

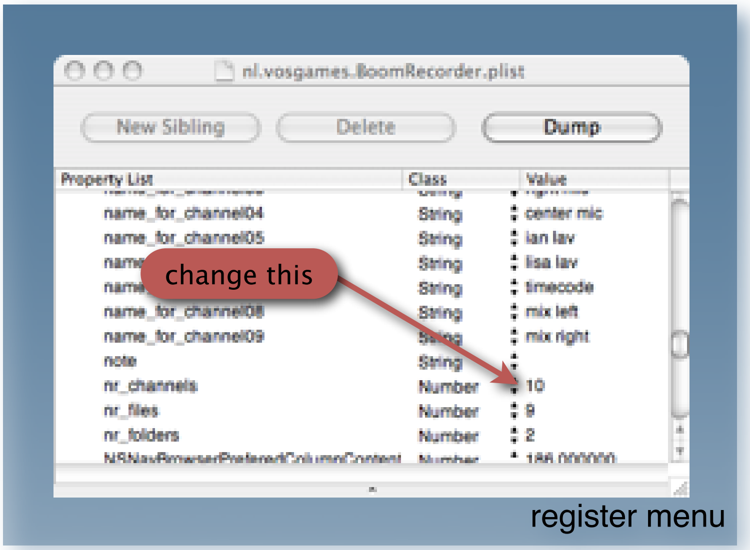
The spectrum analyzer shows the audio in the frequency domain, this is also called a RTA (Real Time Analyzer). Currently the RTA can not be used to analyze a play back.

At the top of this window you can select the channel to read and the sensitivity of the display.

spectrum analyzer

9. Performance

9.1. Number of channels



The size of the bus defines how much memory Boom Recorder will use, if you select too many channels here you computer may stop responding. If you have done so:

- close Boom Recorder or reset the computer.
- Edit the nl.vosgames.BoomRecorder.plist file in your home Library/Preferences folder with the Property List Editor application.
- Change the value nr_channels property

10. AppleScript

Boom Recorder supports a few commands and variables for scripting. The application is called "BoomRecorder" (without the whitespace) in AppleScript.

10.1. Commands

- record
- stop
- abort
- play

10.2. Variables

- project
- scene
- slate
- roll
- take
- timecode
- counter
- channel(s)

10.3. channel class

- name
- input
- output
- file(s)

10.4. File class

- enabled

10.5. Example: "start at specific timecode"

```
set timestamp_found to 0

repeat until timestamp_found = 1
	tell application "BoomRecorder"
		set current_timecode to get timecode
	end tell
	if current_timecode > "22:56:00:00" then set timestamp_found to 1
	delay 1
end repeat

tell application "BoomRecorder" to record
```