*ic*Listen Log File Formats

March 19, 2012

Ocean Sonics Ltd.
Hill House, 11 Lornevale Road,
Great Village, NS, B0M 1L0 Canada
Phone: +1 902 655 3000
www.OceanSonics.com

Table of Contents

Table of Figures

Table of Tables

1 Introduction

Some models of ***ic*****Listen** are capable of internally storing acoustic data in the time domain (waveform data), and/or in the frequency domain (power spectrum data).Also, Ocean Sonics' **Lucy** software is capable of logging both time series and frequency data that is scanned from ***ic*****Listen** devices. This document details the formats in which this data is stored.

2 Overview of Log Files

This section gives a brief overview of the file types used by ***ic*Listen** and **Lucy** for data storage.

For more detailed information on each file type, please refer to the *Detailed File Format Description* sections in the document.

2.1 File Formats

Currently ***ic*Listen** devices can produce 3 different file formats. Time series acoustic data is stored in standard WAV files, while acoustic frequency data is stored in a proprietary FFT file format, or a tab separated values format in a TXT file. **Lucy** is capable of storing data as standard WAV files, or as tab separated values (TSV) format in a TXT file.

Additional sensor data may also be retrieved from these file formats in some cases (See WAV, FFT and TXT file sections for more details).

2.2 File Storage Options

The following table outlines the file storage options for each ***ic*Listen** model and **Lucy**. For a more detailed look at what is stored by each model, refer to the *Detailed File Storage Capabilities* section.

Table 2-1: File Storage Capabilities

File Format	***icListen HF***	***ic*Listen LF**	***ic*Listen MF**	**Lucy**
WAV	All	All	None	v2.1 and higher
FFT	None	All	None	None
TXT(time)	None	None	None	All
TXT(frequency)	All	None	None	All

3 Detailed File Format Descriptions

This section describes the formats of all files used by ***ic*****Listen** and **Lucy** in detail.

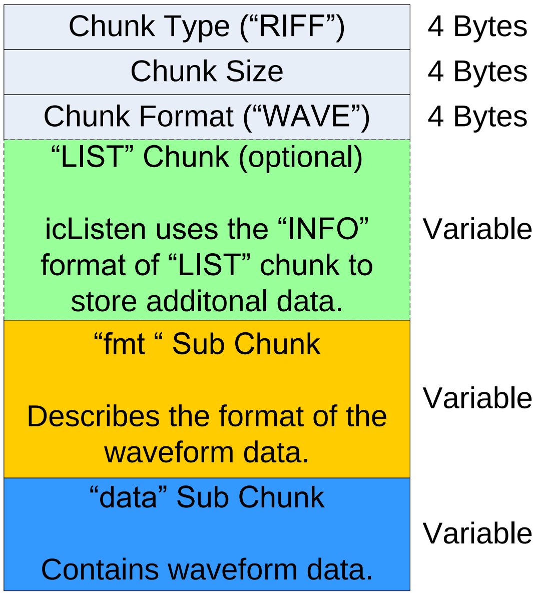
ic**Listen** devices will produce time series acoustic data in standard WAV files. ***ic*****ListenLF** devices will produce frequency data in a proprietary FFT file format, and ***ic*****ListenHF** devices will store frequency data in TXT format.

Lucy will store time series acoustic data either in standard WAV files or TXT files. FFT data is stored by **Lucy** in TXT format.

All files produced by ***ic*****Listen** or **Lucy** may be opened for playback using Ocean Sonics' **Lucy** Software (which is also used for command/control and data collection for ***ic*****Listen** devices). WAV files may also be opened in any software capable of dealing with standard WAV files. TXT files may be opened by virtually any spreadsheet or text editing program.

3.1 WAV File Structure

WAV files use the standard RIFF file structure, which groups the contents of the file into separate chunks. Each chunk contains its own header, which contains a 4-byte string indicating the type of chunk, and 32bit unsigned number indicating the size of that chunk in bytes (excluding the 8 header bytes). All chunks must be word aligned (size must be a multiple of 16bits). All data fields in WAV files are in little endian format.



Chunk Type ("RIFF")	4 Bytes
Chunk Size	4 Bytes
Chunk Format ("WAVE")	4 Bytes
"LIST" Chunk (optional) icListen uses the "INFO" format of "LIST" chunk to store additonal data.	Variable
"fmt " Sub Chunk Describes the format of the waveform data.	Variable
"data" Sub Chunk Contains waveform data.	Variable

Figure 3-1: Basic WAV File Structure

Every WAV file will contain a RIFF chunk of type "WAVE". This chunk will at minimum will contain 2 sub-chunks (the "fmt " chunk, and "data" chunk). WAV files created by ***icListen*** devices also contain an "INFO" type "LIST" chunk, which contains an "ICMT" sub-chunk. This chunk is used for storing additional information about the WAV file.

3.1.1 “fmt “ Sub Chunk

Chunk ID (“fmt “)	4 Bytes
Chunk Size	4 Bytes
Compression Code	2 Bytes
Number of Channels	2 Bytes
Sample Rate	4 Bytes
Bytes Per Sec	4 Bytes
Block Alignment	2 Bytes
Bits Per Sample	2 Bytes
# of Extra Format Bytes	0/2 Bytes
Extra Format Bytes	Variable

Figure 3-2: “fmt “ Sub Chunk

The “fmt “ sub chunk describes the format of the waveform data in the following “data” sub chunk. The following table describes of each field in this chunk.

Table 3-1: “fmt “ Sub Chunk Field Descriptions

Field	Description
Compression Code	The compression type used by the waveform data. All ***ic*Listen** devices use type 1 (PCM/Uncompressed)
Number of Channels	The number of channels represented in the “data” chunk.
Sample Rate	The sample rate of the waveform data in Hz
Bytes Per Second(BPS)	This is the number of bytes of data per second (# Channels x Sample Rate x Bytes Per Sample)
Block Alignment	This value is the number of bytes per sample multiplied by the number of channels
Bits Per Sample	The number of bits per data point.
# of Extra Format Bytes	This field specifies the number of Extra Format Bytes will follow. This field does not exist for PCM/Uncompressed WAV files, and therefore does not apply to ***ic*Listen** devices.
Extra Format Bytes	The number and meaning of these bytes varies depending on the compression used. These bytes do not exist for PCM/uncompressed WAV files, and therefore are not present in ***ic*Listen** generated files.

The following table shows the settings for these bytes for each ***ic*Listen** Model that can store WAV files:

Table 3-2: “fmt “ Sub Chunk Fields by Model

Model	Compression	Channels	Sample Rate	BPS	Alignment	Bits Per Sample
HF	1	1	Varies	Varies	3	24
LF	1	1	Varies	Varies	3	24

3.1.2 “data” Sub Chunk

Chunk ID (“data”)	4 Bytes
Chunk Size	4 Bytes
Data	Variable

Figure 3-3: “data“ Sub Chunk

The data sub chunk contains the actual waveform data in the file, in the format described by the preceding “fmt “ sub chunk.

If the data is 8-bits per data point, it is considered unsigned data. Otherwise the data is considered signed.

If the data contains more than one channel, the data is interlaced (each sample contains data from each channel). See the figure below for a stereo data example.

Cunk ID (“data”)	4 Bytes
Chunk Size (12)	4 Bytes
Left Channel	2 Bytes
Right Channel	2 Bytes
Left Channel	2 Bytes
Right Channel	2 Bytes
Left Channel	2 Bytes
Right Channel	2 Bytes

Figure 3-4: 16 bit Stereo “data“ Sub Chunk example

3.1.3 “INFO” Chunk

Chunk Type (“LIST”)	4 Bytes
Chunk Size	4 Bytes
Chunk Format (“INFO”)	4 Bytes
Sub Chunk ID (“ICMT”)	4 Bytes
Sub Chunk Size	4 Bytes
ASCII String	Variable

Figure 3-5: “INFO” type “LIST” Chunk with “ICMT” Sub Chunk

The “INFO” chunk is a standard RIFF chunk, which may be used to add additional information to a WAV file. ***ic*Listen** devices only make use of the “ICMT” sub chunk for this purpose.

The “ICMT” sub chunk is a field containing a single NULL terminated ASCII comment string.

This string is used by ***ic*Listen** devices to give information about the scaling of the WAV data, as well as other sensor data. Different values are separated by commas in the string. The table below shows example string values used for different pieces of information. For more detail on which strings are stored by ***ic*Listen** and **Lucy**, refer to the *Detailed File Storage Capabilities* section.

Table 3-3: Example “ICMT” String Values

Field	**Example String**
Full Scale Voltage	“3.000000 V pk”
Hydrophone Sensitivity	“-169 dBV re 1uPa”
Humidity*	“22.7% RH”
Temperature*	“22.8 deg C”
Count At Full Scale Voltage	“8388608 = Max Count”

*This field may be padded with spaces depending on the magnitude of the value in the field

A full comment string with all of the above fields included would read:

> “3.000000V pk,-169 dBV re 1 uPa, 22.7% RH,   22.8 deg C, 8388608 = Max Count”

3.2 FFT File Structure

The FFT file format used by current ***ic*****Listen** devices is similar to that of WAV files

The first 8 bytes of the FFT file contain a 4 bytes ASCII string ("FFT_"), used to indicate the file type, and an unsigned 32bit number indicating the length of the file in bytes(excluding the first 8 bytes).

Following this, individual chunk headers are placed. Each chunk header contains a 32bit chunk ID indicating the chunk type, and a 32bit unsigned value indicating the number of bytes in that chunk (excluding the 8 header bytes).

The following table describes the different chunk types found in FFT files produced by ***ic*****Listen** units:

Table 3-4: FFT File Chunk Types

ID	Chunk Type	Description
0	FFT data	Frequency data, with formatting details
1	Temperature/Humidity Data	Temperature and humidity data

All data in the FFT file is little endian. All files produced by ***ic*****Listen** units contain a Temperature/Humidity chunk followed by an FFT data chunk.

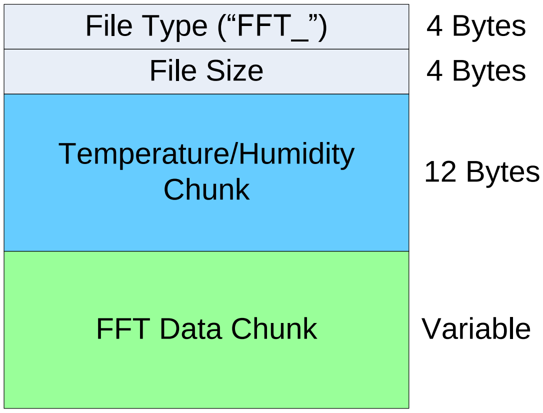
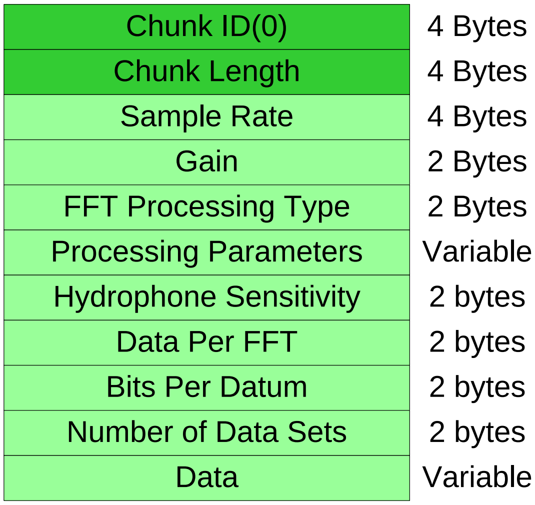


Figure 3-6: FFT File Structure Overview

3.2.1 FFT Data Chunk



Chunk ID(0)	4 Bytes
Chunk Length	4 Bytes
Sample Rate	4 Bytes
Gain	2 Bytes
FFT Processing Type	2 Bytes
Processing Parameters	Variable
Hydrophone Sensitivity	2 bytes
Data Per FFT	2 bytes
Bits Per Datum	2 bytes
Number of Data Sets	2 bytes
Data	Variable

Figure 3-7: FFT Data Chunk

This chunk contains the acoustic frequency data, as well as the information required to analyze this data. This table gives a description of each field:

Table 3-5: FFT Data Chunk Field Descriptions

Sample Rate	Sample rate in Hz at which raw data was collected
Gain	Applied gain in dB
FFT Processing Type	Type of FFT processing done. See actions below
Processing Parameters	Parameters vary based on FFT Type
Hydrophone Sensitivity	Sensitivity of the hydrophone in dB relative to 1µPa
Data Per FFT	Data points per FFT data set (ie: 1024pt FFT would yield 512 data points)
Bits Per Datum	Bits per data point
Number of Data Sets	Number of FFT data sets represented in the “Data” field
Data	Data in ½ dB units(2dB per count) relative to 1µV

The following table shows the available FFT processing types. See the following sections for more detail. For details on what processing types are available to each ***icListen*** model, refer to the *Detailed File Storage Capabilities* section.

Table 3-6: FFT Processing Types

Type Code	Type of Processing
0	Mean Average
1	Sliding Window
2	Peak Value Detect
3	Exponential Moving Average (IIR filter)
4	Sliding Window with Mean Average
5	Sliding Window with Peak Value Detect
6	Sliding Window with Exponential Moving Average

3.2.1.1 Sliding Window FFT Data

When sliding window processing is done, there is overlap in the data used to calculate FFTs. The following figures illustrate how data is shared between FFT calculations based on the percent overlap.

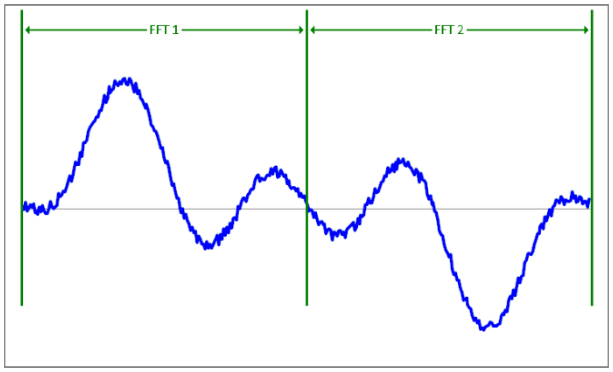


Figure 3-8: Sliding Window (0% overlap)

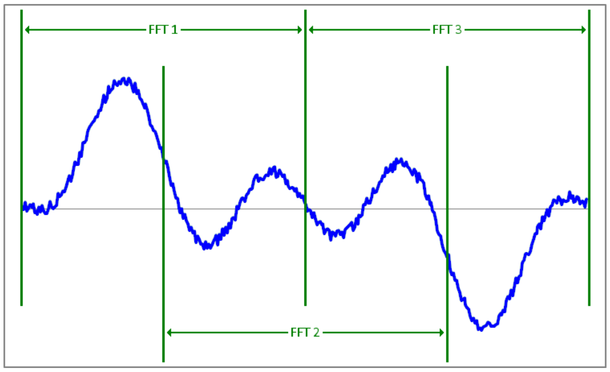


Figure 3-9: Sliding Window (50% overlap)

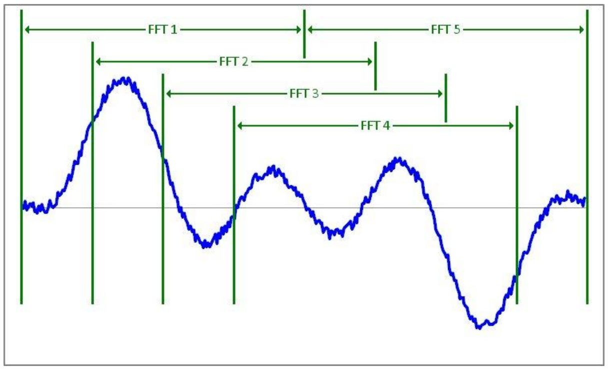


Figure 3-10: Sliding Window (75% overlap)

This processing type contains two parameters, which are used to calculate the overlap as follows:

$$\%\ Overlap = \left(1 - \frac{Parameter\ 1}{Parameter\ 2}\right) \times 100\%$$

Each data set in sliding window FFT data represents a single FFT calculation.

3.2.1.2 Mean Average FFT Data

For mean averaged FFT data, the stored data is FFT data that has been averaged over 'N' FFTs. There are two unsigned 16bit type parameters for this FFT type. The first is the 'N' value used in the mean calculation, and the second indicates the overlap (in FFT data sets) of data used between this mean calculation and the previous mean calculation.

The mean value for each frequency bin is calculated as follows:

$$Y_i{}^2 = \frac{1}{N}\sum_{j=0}^{N-1}\left|C_{i,j}\right|^2$$

Where:

N = Averaging Period
Y_i^2 = Signal Power of frequency bin
$C_{i,j}$ = FFT Coefficient
i = Frequency Bin Number
j = FFT Data Set Number

3.2.1.3 Peak Value Detect FFT Data

When the FFT type is peak value, the stored data contains the maximum value found for each frequency bin over 'N' FFT data sets. This type also has two 16bit unsigned parameters. The first is 'N', which represents the number of FFT data sets the peak was found over. The second parameter represents the overlap in FFT data sets used to find the peak values.

3.2.1.4 Exponential Moving Average (IIR Filter) FFT Data

When this processing type is used, the stored data represents the exponential moving average of all preceding FFTs performed. Two unsigned 16bit type parameters are included with this type. The first is the weighting factor (N) used in the calculation. The second value represents number of FFT's calculated between reported FFT data sets.

The exponential moving average is calculated for each frequency bin using the formula:

$$Y_{i,j}{}^2 = \frac{(N-1)\times Y_{i,j-1}{}^2 + \left|C_{i,j}\right|^2}{N}$$

Where:

N = Weighting Factor
$Y_{i,j}^2$ = Signal Power of frequency bin
$C_{i,j}$ = FFT Coefficient
i = Frequency Bin Number
j = FFT Data Set Number

3.2.1.5 Sliding Window with Mean Average FFT Data

This processing type performs a mean average on sliding window FFT data. There are 4 parameters with this processing type. The first 2 parameters are the same as those used by the *Mean Average* processing type. The second 2 parameters are the same as those used by the *Sliding Window* processing type.

3.2.1.6 Sliding Window with Peak Value Detect

This processing type takes sliding window FFT data, and then retains the maximum value detected for each frequency bin over 'N' FFT data sets. There are 4 parameters with this processing type. The first 2 parameters are the same as those used by the *Peak Value Detect* processing type. The second 2 parameters are the same as those used by the *Sliding Window* processing type.

3.2.1.7 Sliding Window with Exponential Moving Average

This processing type performs an exponential moving average of sliding window FFT data. There are 4 parameters with this processing type. The first 2 parameters are the same as those used by the *Exponential Moving Average* processing type. The second 2 parameters are the same as those used by the *Sliding Window* processing type.

3.2.2 Temperature/Humidity Chunk

Chunk ID(1)	4 Bytes
Chunk Length(4)	4 Bytes
Temperature	2 Bytes
Humidity	2 Bytes

Figure 3-11: Temperature/Humidity Chunk

This chunk contains two values, the first represents the temperature, and the second represents humidity at the time of logging.

Temperature is a signed 16bit value. It is measured in tenths of degrees Celsius.

Humidity is an unsigned 16bit value. It is measured in tenths of percent of relative humidity.

3.3 TXT File Structure

TXT files contain ASCII variables separated by tabs. These files may be read by virtually any text editor or spreadsheet program. When interpreted as tabular/spreadsheet data, tabs are equivalent to column divisions, and newline characters are row divisions.

All TXT files generated by ***ic*****Listen**/**Lucy** contain several rows of header information at the start of the file, followed by rows of either FFT data or time series data. The types of data stored in TXT format by **Lucy** are FFT, Time_Series, and Calibration. ***ic*****ListenHF** stores only FFT data in TXT format.

The following table shows the fields contained in each row of data after the header, with example values for each.

Table 3-7: Example TXT data row contents

	Time	Comment	Guest	Temperature	Humidity	Gain	Sequence	Data
Example	hh:mm:ss	"Test"	5.50	22.0	27.3	12dB	1002	98 89…

Here are example headers for the FFT, Time_Series, and Calibration log types.

FFT file header section:

```
icListenHF       v1.1.03
File Version: 3  File Type: FFT
This file name:  /home/icListen/Data/Fft_20120307T02-05-44.txt
Computer Id:     icListenHF       User Id:          "1205"
Data Log File Created: 2012-03-07 2:05:44
CLIENT   icListenHF        ID      icListenHF        PERSONNEL       icListenHF
icListenHF
Sample Rate: 512000       Data Reference: -120dB re 1V      Phone Sensitivity (dBV re 1uPa): -169      FFT
Rate: 125.0
Time     Comment  Guest    Temperature      Humidity Gain     Sequence FFT..
```

Time_Series file header section:

```
Instrument Concepts' Lucy     V2.4.1
File Version: 2     File Type: Time_Series
This file name:     c:\Example\TimeSeries_20110817_142351.txt
Computer Id:        "ICSS-TP5"          User Id:   "Alan"
Data Log File Created         17 Aug.2011         14:23:51
CLIENT    "None"     ID       "Test"     PERSONNEL          "Me"
"icListen LF # 2 Ready   Firmware v2.4.03"
Sample Rate: 4000   Phone Sensitivity (dBV re 1uPa): -171
Time      Comment   Guest     Temperature         Humidity  Gain      Sequence  Time Series Data..
```

Calibration file header section:

```
Instrument Concepts' Lucy         V2.4.1
File Version: 2  File Type: Calibration
This file name:  C:\Documents and Settings\Alan\Application Data\Lucy\LF_CAL.txt
Computer Id:     "ICSS-TP5"       User Id:          "Alan"
Data Log File Created     10 Aug.2011      10:09:32
CLIENT   "None"    ID      "Test"    PERSONNEL       "Me"
"icListen LF # 202 Ready  Firmware v2.4.03"
Sample Rate: 4000        Data Reference: dB re 1nV Phone Sensitivity (dBV re 1uPa): -171      FFT Rate: 0.5
Time     Comment  Guest    Temperature      Humidity Gain     Sequence FFT..
```

4 Detailed File Storage Capabilities

Some comments in the WAV header, and certain types of FFT data, may not be stored by all ***ic*Listen** units or **Lucy** Versions. The following table outlines in detail, what is stored in each field, by each ***ic*Listen** model and by **Lucy**.

Table 4-1: Detailed File Storage Capabilities

File Type	**ICMT fields/FFT Types**	***ic*Listen HF**	***ic*Listen LF**	***ic*Listen MF**	**Lucy**
WAV Files	Peak Voltage & Phone Sensitivity	All	All versions	N/A	v2.1 and up
	Humidity & Temperature	All	v2.3 and up		v3.0 and up
	Count At Full Scale Voltage	All	None		v3.0 and up
FFT Files	Mean Average	N/A	v2.3 and up	N/A	N/A
	Sliding Window		v2.3 and up		
	Peak Value Detect		v2.3 and up		
	Exponential Moving Average		v2.3 and up		
	Sliding Window with Mean Average		v2.3.01 and up		
	Sliding Window with Peak Value Detect		v2.3.01 and up		
	Sliding Window with Exponential Moving Average		v2.3.01 and up		

Glossary

dB	Decibels
FFT	Fast Fourier Transform
Hz	Hertz
µPa	Micro Pascals
µV	Micro Volts
TSV	Tab Separated Variable
V	Volts