# XWAV File Format

High-frequency Acoustic Recording Packages (HARPs) record raw acoustic data in a compressed proprietary format on instrument internal storage disks during deployment. Once retrieved, images of raw disks are made during the first step of data processing. Recordings are then decompressed and reformatted to XWAVs, an enhanced loss-less audio format. For archiving purposes, these XWAV files have been compressed using a free loss-less audio codec (FLAC), with approximately a factor of 2X compression.

XWAVs are simply standard WAV files with extra information, such as the recordings’ high-precision timing, deployment location, site name, and instrument metadata, entered into the file header. **XWAV files can be read with** **any standard WAV file reader**. In this case, file readers will simply ignore the extra header information, and the files will function as WAV files.

The extra header information high-precision time is primarily used for sound localization and tracking. The tools contained within the freely available [*Triton* software package](https://github.com/MarineBioAcousticsRC/Triton) (available for MATLAB-based or [standalone](https://github.com/MarineBioAcousticsRC/Triton-Compiled) configurations). Use this information for accurate recording timing. Functions such as **rdxwavhd.m** within *Triton* can be used to read XWAV files into the MATLAB workspace and provide precisely timed extracted data chunks. All Triton scripts and functions are MATLAB m-files and can be read or edited using any text editor.

This audio data has been converted to [flac](https://xiph.org/flac/format.html) format from extended WAV (XWAV) format. This information has been preserved in the conversion to flac format by using the  [--keep-foreign-metadata](https://urldefense.proofpoint.com/v2/url?u=https-3A__xiph.org_flac_faq.html-23general-5F-5Fno-5Fwave-5Fmetadata&d=DwMFaQ&c=-35OiAkTchMrZOngvJPOeA&r=S4MI9VlVvkcXqMxzNk-ULZqsb--GLcmY7ayTBnBNZks&m=jMDMSDFeMS5HiakcyeJtfAavO-bQNHFZsKXXJeBEdjJOZpQNeGJxyke9Q-3Auxeo&s=0PZU3TPa1J_9R2eZIUVgRIjDGw_Nz91iLp2Dp0C8B3I&e=) flag when compressing the data. To retain this information when decompressing, the  [--keep-foreign-metadata](https://urldefense.proofpoint.com/v2/url?u=https-3A__xiph.org_flac_faq.html-23general-5F-5Fno-5Fwave-5Fmetadata&d=DwMFaQ&c=-35OiAkTchMrZOngvJPOeA&r=S4MI9VlVvkcXqMxzNk-ULZqsb--GLcmY7ayTBnBNZks&m=jMDMSDFeMS5HiakcyeJtfAavO-bQNHFZsKXXJeBEdjJOZpQNeGJxyke9Q-3Auxeo&s=0PZU3TPa1J_9R2eZIUVgRIjDGw_Nz91iLp2Dp0C8B3I&e=) flag should also be used. Some software may be able to use the flac files directly, without requiring decompression. However, as of writing (circa 2022), most existing tools capable of making use of the XWAV header information were not yet equipped to handle flac format directly.

**Additional Notes:**

* All HARP XWAV files use timestamp in the file names as a start time of the first sample of that file, but only to 1 s precision. Inferring subsequent event times based on samples elapsed since file start is common practice, but is **not recommended,** especially for recorders with imprecise or high-drift clocks used for analog-to-digital conversions. HARP clocks have a low drift on the order of 1 s / yr (10-8 s/s), but it is still recommended to use the timing details in the XWAV headers to get the most precise recording times.
* HARPs sampling at 200 kHz record acoustic data in 75 second “chunks”, termed “raw files”. Lower sampling rates produce longer raw files. Each XWAV file typically consists of 30 raw files. Timing at the level of each raw file is readily available in the XWAV header.
* The year values stored in xwav headers include only the last 2 digits. When converting to date numbers, you will need to add 2000.
* Recorded acoustic signal amplitudes read from XWAV files must be considered relative until corrected for sensor sensitivity (see Transfer\_Function\_readme). HARP recordings are currently 16-bit samples, so maximum XWAV amplitudes are +/- 215 (i.e., 32,768) integer counts. Ensure that ‘native’ values are preserved when reading the files, as some tools (e.g., MATLAB audioread function) will normalize the amplitudes by default, and this can result in incorrect amplitude estimates.

XWAV files have the following general layout:

| 1 XWAV file | Standard WAV header | RIFF header |
| --- | --- | --- |
| Format Chunk |
| Additional XWAV header | HARP Chunk |
| XWAV directory | HARP dir subchunk 1 |
| HARP dir subchunk 2 |
| . . . |
| HARP dir subchunk 30 |
| Data | Data Chunk 1 |
| Data Chunk 2 |
| . . . |
| Data Chunk 30 |

Detailed layout within the WAV and XWAV headers is as follows:

|  | Field Name  *PARAMS.xhd. precedes each* | Length in Bytes | Start | End | Format | # of Elements | Example |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Standard WAV header** | | | | | | | |
| Riff Header | ChunkID | 4 | 0 | 3 | uchar | 4 | “RIFF” |
| ChunkSize | 4 | 4 | 7 | uint32 | 1 | filesize-8 |
| Format | 4 | 8 | 11 | uchar | 4 | “WAVE” |
| Format Chunk | fSubchunkID | 4 | 12 | 15 | uchar | 4 | “fmt “ |
| fSubchunkSize | 4 | 16 | 19 | uint32 | 1 | 16 |
| AudioFormat | 2 | 20 | 21 | uint16 | 1 | 1 |
| NumChannels | 2 | 22 | 23 | uint16 | 1 | 1 |
| SampleRate | 4 | 24 | 27 | uint32 | 1 | 200000 |
| ByteRate | 4 | 28 | 31 | uint32 | 1 | 400000 |
| BlockAlign | 2 | 32 | 33 | uint16 | 1 | 2 |
| BitsPerSample | 2 | 34 | 35 | uint16 | 1 | 16 |
| **SUBTOTAL** |  | **36** | **0** | **35** |  |  |  |
| **Additional XWAV Header** | | | | | | | |
| HARP Chunk | hSubchunkID | 4 | 36 | 39 | uchar | 4 | “harp” |
| hSubchunkSize | 4 | 40 | 43 | uint32 | 1 | 56+30\*32 |
| WavVersionNumber | 1 | 44 | 44 | uchar | 1 | 0 |
| FirmwareVersionNumber | 10 | 45 | 54 | uchar | 10 | 1.xxxyyyzz |
| InstrumentID | 4 | 55 | 58 | uchar | 4 | “01 “ |
| SiteName | 4 | 59 | 62 | uchar | 4 | “ABCD” |
| ExperimentName | 8 | 63 | 70 | uchar | 8 | “EXP12345” |
| DiskSequenceNumber | 1 | 71 | 71 | uchar | 1 | 1 |
| DiskSerialNumber | 8 | 72 | 79 | uchar | 8 | 12345678 |
| NumOfRawFiles | 2 | 80 | 81 | uint16 | 1 | 1 |
| Longitude | 4 | 82 | 85 | uint32 | 1 | -17912345 |
| Latitude | 4 | 86 | 89 | uint32 | 1 | 8912345 |
| Depth | 2 | 90 | 91 | uint16 | 1 | 5555 |
| Reserved | 8 | 92 | 99 | uchar | 8 | 00000000 |
| **SUBTOTAL** |  | **64** | **36** | **99** |  |  |  |

|  | Field Name  *PARAMS.xhd. precedes each* | Length in Bytes | Start | End | Format | # of Elements | Example |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Additional XWAV Header cont.**  *Repeated for each RawFile(k) n=32(k-1), usually 30* | | | | | | | |
| HARP dir Subchunks | year(k) | 1 | 100+n | 100+n | uchar | 1 | 7 |
| month(k) | 1 | 101+n | 101+n | uchar | 1 | 12 |
| day(k) | 1 | 102+n | 102+n | uchar | 1 | 31 |
| hour(k) | 1 | 103+n | 103+n | uchar | 1 | 23 |
| minute(k) | 1 | 104+n | 104+n | uchar | 1 | 59 |
| secs(k) | 1 | 105+n | 105+n | uchar | 1 | 59 |
| ticks(k) | 2 | 106+n | 107+n | uint16 | 1 | 999 |
| byte\_loc(k) | 4 | 108+n | 111+n | uint32 | 1 | 1066 |
| byte\_length(k) | 4 | 112+n | 115+n | uint32 | 1 | 30000000 |
| write\_length(k) | 4 | 116+n | 119+n | uint32 | 1 | 60000 |
| sample\_rate(k) | 4 | 120+n | 123+n | uint32 | 1 | 200000 |
| gain(k) | 1 | 124+n | 124+n | uint8 | 1 | 1 |
| padding | 7 | 125+n | 131+n | uchar | 7 | 0000000 |
| **SUBTOTAL** |  | **32+n** | **100** | **131+n** |  |  |  |
| **Data Chunk** | | | | | | | |
| Data Chunk | dSubchunkID | 4 | 132+n | 135+n | uchar | 4 | “data” |
| dSubchunkSize | 4 | 136+n | 137+n | uint32 | 1 | datasize |
| DATA |  | 138+n |  |  |  |  |

For additional information see comments in rdxwavhd.m, wrxwavhd.m, and initdata.m in Triton.

For typical full HARP XWAV files, file size will be 900,001,068 bytes or 30 raw files (n) 30,000,000 bytes/rawfiles & 1068 byte header.