# LIFE

## Light Instrument Frequency Encoding Format Specification

Version 1.0 - May 5, 2026

life-codec.org

A universal, open, hardware-agnostic format for encoding frequency relationships between sound and light, such that any sufficiently curious intelligence can reconstruct both from the file alone.

# 1. Introduction

LIFE is an open specification for encoding frequency relationships between acoustic and electromagnetic phenomena. It is to light composition what MIDI is to music, and what G-code is to fabrication: a universal instruction language that describes intent rather than implementation.

A .life file can represent a musical composition translated into light frequencies, a light sequence translated into sound, raw spectral data from a natural source, or an original composition authored directly in the LIFE format.

The format is designed to remain valid and decodable without external reference. Every .life file contains within itself the rules required to interpret it. This property makes LIFE suitable not only for contemporary audio-visual applications but for archival, scientific, and long-range communication purposes.

## 1.1 Design philosophy

- Relationships over absolutes. Frequencies are expressed as ratios, not raw Hz values. The ratio 3:2 is true everywhere in the universe. The convention of 440 Hz is not.
- Self-describing. A .life file contains its own codec definition. No external lookup table or software version is required to decode it.
- Reversible by default. Any .life file produced by translation must be reversible back to its source medium unless explicitly marked otherwise.
- Extensible without breakage. Unknown fields are ignored. The format grows without invalidating existing files.
- Human readable. JSON is the canonical encoding. A person with no software can open a .life file and understand its structure.

## 1.2 Relationship to existing standards

LIFE is not a replacement for MIDI, DMX, or OSC. It is a translation layer and archival format that sits above physical protocols. A LIFE-compatible system may output to any of these protocols at runtime. The .life file describes what should happen. The instrument decides how.

## 2. Terminology

| Field | Type | Description |
|---|---|---|
| codec | noun | The mathematical rules that translate between source and target frequency domains. |
| event | noun | A discrete moment in time containing one or more simultaneous frequency states. |
| voice | noun | A single frequency source within an event. Analogous to a MIDI channel or an instrument part. |
| ratio | noun | A frequency expressed as a relationship to the reference frequency.<br><br>The primary representation in LIFE. |
| wavelength_nm | noun | The electromagnetic wavelength in nanometres corresponding to a given frequency after codec translation. |
| intensity | noun | Relative amplitude or brightness, expressed as a float from 0.0 to 1.0. |
| reversible | flag | Indicates the file can be decoded back to its source medium without information loss. |
| lossless | flag | Indicates no frequency information was discarded during encoding. |
| preamble | section | The self-describing header of a .life file containing format version and decoding instructions. |
| extensions | object | Optional fields reserved for future capability without breaking existing parsers. |

# 3. File format

## 3.1 File extension and encoding

The canonical file extension is *.life*. The canonical encoding is UTF-8 JSON. No other encoding is defined in version 1.0.

A binary serialisation (*.life.bin*) may be implemented for performance-critical applications. Any binary serialisation must be fully derivable from and convertible to the canonical JSON form. The JSON form is always authoritative.

## 3.2 Top-level structure

Every *.life* file is a JSON object containing exactly three required top-level keys and one optional key.

```
{
  "preamble": { ... },
  "codec":    { ... },
  "events":   [ ... ],
  "extensions": { ... }     // optional
}
```

# 4. The preamble

The preamble is the Rosetta Stone of the file. It must contain sufficient information for a reader with no prior knowledge of LIFE to understand the structure of what follows. It is the first layer of the file and must appear before the codec and events.

## 4.1 Required fields

| Field | Type | Description |
|---|---|---|
| format | string | Always the literal string “LIFE”. Identifies the file type unambiguously. |
| version | string | Semantic version of the LIFE specification used. Currently “1.0”. |
| created | string | ISO 8601 date or year of file creation. Provides temporal context for archival use. |
| scaling | object | Describes the mathematical method used to bridge source and target frequency domains. |
| reference | object | Defines the anchor frequency against which all ratios in this file are expressed. |
| spectrum | object | Defines the target electromagnetic range used for wavelength mapping. |

## 4.2 The scaling object

The scaling object must be self-explanatory. It must describe not just the value used but the reasoning behind it, so that the file remains interpretable without external documentation.

| Field | Type | Description |
|---|---|---|
| method | string | The scaling method. Version 1.0 defines “octave_halving” as the standard method. |
| constant | string | Human-readable expression of the scaling constant. Example: “2^40”. |
| value | integer | The numeric value of the scaling constant. Example: 1099511627776. |
| rational | string | Plain language explanation of why this constant was chosen. Required. |

## 4.3 The reference object

| Field | Type | Description |
|---|---|---|
| frequency_hz | float | The reference frequency in hertz. Conventionally 440.0 (concert A4). |
| ratio | string | The ratio of the reference to itself. Always “1:1”. |
| description | string | Human-readable description of what this reference represents. |

## 4.4 The spectrum object

| Field | Type | Description |
|---|---|---|
| min_nm | integer | Minimum wavelength of the target electromagnetic range in nanometres. |
| max_nm | integer | Maximum wavelength of the target electromagnetic range in nanometres. |
| description | string | Human-readable description of this range. |

## 4.5 Complete preamble example

```
"preamble": {
  "format":  "LIFE",
  "version": "1.0",
  "created": "2026-05-05",
  "scaling": {
    "method": "octave_halving",
    "constant": "2^40",
    "value":  1099511627776,
    "rational": "40 octaves separate the audible and visible spectra.
                 Dividing any acoustic frequency by 2^40 maps it into
                 the electromagnetic range while preserving all harmonic
                 relationships. This constant is physically derived,
                 not conventionally assigned."

  },
  "reference": {
    "frequency_hz": 440.0,
    "ratio":      "1:1",
    "description":  "Concert pitch A4."
  },
  "spectrum": {
    "min_nm": 380, "max_nm": 700,
    "description": "Visible light, violet to red."
  }
}
```

# 5. The codec

The codec defines the translation rules applied to produce this file. It must be fully specified within the file itself. A LIFE parser must be able to reverse the translation using only the codec object and the preamble.

## 5.1 Required fields

| Field | Type | Description |
|---|---|---|
| type | string | The codec type. Version 1.0 defines “harmonic_ratio” as the standard type. |
| source_medium | string | The origin medium. One of: “acoustic”, “electromagnetic”, “spectral”, “synthetic”. |
| target_medium | string | The destination medium. One of: “acoustic”, “electromagnetic”. |
| scaling_constant | integer | The numeric scaling constant. Must match preamble.scaling. value. |
| scaling_expression | string | Human-readable expression. Must match preamble.scaling.constant. |
| frequency_representation | string | How frequencies are stored in events. Must be “ratio” in version 1.0. |
| reference_frequency_hz | float | Must match preamble.reference. frequency_hz. |
| mappings | object | Explicit declaration of how each source property maps to each target property. |
| reversible | boolean | Whether this file can be decoded back to its source medium without loss. |
| lossless | boolean | Whether all source frequency information is preserved in this encoding. |

## 5.2 The mappings object

Mappings must be explicit and human-readable. They serve as documentation for future parsers and as a verification layer for current ones.

```
"mappings": {
  "pitch":  "frequency_ratio -> wavelength_nm",
  "velocity": "intensity_0_to_1 -> photon_flux_relative",
  "duration": "seconds -> seconds",
  "channel":  "voice_index -> light_source_index"
}
```

## 5.3 Complete codec example

```
"codec": {
  "type":                 "harmonic_ratio",
  "source_medium":        "acoustic",
  "target_medium":        "electromagnetic",
  "scaling_constant":     1099511627776,
  "scaling_expression":   "2^40",
  "frequency_representation": "ratio",
  "reference_frequency_hz": 440.0,
  "mappings": {
    "pitch":  "frequency_ratio -> wavelength_nm",
    "velocity": "intensity_0_to_1 -> photon_flux_relative",
    "duration": "seconds -> seconds",
    "channel":  "voice_index -> light_source_index"
  },
  "reversible": true,
  "lossless": true
}
```

# 6. Events

The events array is a time-ordered sequence of state declarations. Each event describes what exists at a given moment. LIFE events are declarative, not imperative. They do not say do this. They say this is.

This distinction is fundamental. An imperative format encodes instructions for a specific machine. A declarative format encodes truth that any sufficiently capable system can act upon.

## 6.1 Event structure

| Field | Type | Description |
|---|---|---|
| t | float | Time offset in seconds from the beginning of the file. Non-negative. Non-decreasing. |
| voices | array | One or more voice objects active at this moment. May be empty to represent silence. |

## 6.2 Voice structure

| Field | Type | Description |
|---|---|---|
| id | integer | Voice identifier. Stable across events. Analogous to a MIDI channel. |
| ratio | string | Frequency expressed as a ratio to the reference. Example: "3:2". Primary field. |
| hz | float | Derived frequency in hertz. Provided for convenience. Must be consistent with ratio. |
| wavelength_nm | float | Derived electromagnetic wavelength in nanometres after codec translation. |

| Field | Type | Description |
|---|---|---|
| intensity | float | Relative amplitude or brightness. Range 0.0 to 1.0. Default 1.0. |
| colour | string | Hex colour approximation of wavelength_nm for human reference. Not canonical. |
| duration_s | float | Optional. Duration in seconds. If absent, duration extends to next event on this voice. |

### 6.3 The primacy of ratio

The ratio field is the authoritative representation of frequency in a LIFE file. All other frequency fields (hz, wavelength_nm, colour) are derived values provided for convenience and human readability.

A parser encountering a conflict between ratio and hz must use ratio. A parser generating a .life file must compute hz and wavelength_nm from ratio, not the reverse.

The ratio must be expressed as two colon-separated integers in their simplest form. The ratio 4:2 is invalid. The ratio 2:1 is correct.

## 6.4 Complete events example

```
"events": [
  {
    "t": 0.0,
    "voices": [
      {
        "id":          1,
        "ratio":       "3:2",
        "hz":          391.9954,
        "wavelength_nm": 562.16,
        "intensity":   1.0,
        "colour":      "#c9ff00"
      }
    ]
  },
  {
    "t": 1.2,
    "voices": [
      { "id": 1, "ratio": "2:1", "hz": 523.2511, "wavelength_nm":548.78,
        "intensity": 1.0, "colour": "#9eff00" },
      { "id": 2, "ratio": "1:1", "hz": 261.6256, "wavelength_nm":580.89,
        "intensity": 1.0, "colour": "#fffc00" }
    ]
  }
]
```

# 7. Extensions

The extensions object is a reserved namespace for capabilities not yet defined in the core specification. Its presence is optional. Its contents are implementation-defined.

A LIFE parser must silently ignore any extension field it does not recognise. It must not fail, warn, or alter its output as a result of encountering an unknown extension. This ensures forward compatibility.

## 7.1 Reserved extension keys

| Field | Type | Description |
|---|---|---|
| polarization | object | Electromagnetic polarisation state of each voice. Reserved for version 1.1. |
| phase | float | Phase offset in radians. Reserved for version 1.1. |
| coherence | float | Degree of coherence for laser or wave-specific outputs. Reserved for version 1.1. |
| spatial | object | 3D spatial position of each voice source. Reserved for version 1.2. |
| chemical | object | Chemical frequency encoding for molecular communication research. Reserved. |
| origin | string | Source attribution. May contain instrument name, location, or spectrometer ID. |

## 7.2 Extensions example

```
"extensions": {
  "polarisation": null,
  "phase":        null,
  "coherence":    null,
  "spatial":      null,
  "origin": "Spectrometer capture, dawn chorus, Burnaby BC, 2026-05-05"
}
```

# 8. Complete file example

The following is a complete, valid .life file. It represents the opening moment of BWV 1074 (Canon a 4 completely invertible) by J.S. Bach, translated into the LIFE format using the standard 2^40 harmonic codec.

```
{
  "preamble": {
    "format":  "LIFE",
    "version": "1.0",
    "created": "2026-05-05",
    "source":  "BWV 1074 - Canon a 4 - J.S. Bach",
    "scaling": {
      "method": "octave_halving", "constant": "2^40",
      "value":  1099511627776,
      "rational": "40 octaves separate audible and visible spectra."
  },
  "reference": { "frequency_hz": 440.0, "ratio": "1:1",
                 "description": "Concert pitch A4." },
  "spectrum":  { "min_nm": 380, "max_nm": 700,
                 "description": "Visible light, violet to red." }
},
"codec": {
  "type": "harmonic_ratio",
  "source_medium": "acoustic", "target_medium": "electromagnetic",
  "scaling_constant": 1099511627776, "scaling_expression": "2^40",
  "frequency_representation": "ratio",
  "reference_frequency_hz": 440.0,
  "mappings": {
    "pitch":  "frequency_ratio -> wavelength_nm",
    "velocity": "intensity_0_to_1 -> photon_flux_relative",
    "duration": "seconds -> seconds",
    "channel":  "voice_index -> light_source_index"
  },
  "reversible": true, "lossless": true
},
"events": [
  { "t": 0.0, "voices": [
      { "id": 1, "ratio": "3:2", "hz": 391.9954,
        "wavelength_nm": 562.16, "intensity": 1.0, "colour": "#c9ff00" }
  ]},
  { "t": 1.2, "voices": [
      { "id": 1, "ratio": "2:1", "hz": 523.2511,
        "wavelength_nm": 548.78, "intensity": 1.0, "colour": "#9eff00"
        },
      { "id": 2, "ratio": "1:1", "hz": 261.6256,
```

```
        "wavelength_nm": 580.89, "intensity": 1.0, "colour": "#fffc00" }
    ]}
  ],
  "extensions": { "polarisation": null, "phase": null, "coherence": null
  }
```

# 9. Conformance

A conformant LIFE file must satisfy the following requirements.

## 9.1 Design philosophy

- The file must be valid UTF-8 encoded JSON.
- The top-level object must contain `preamble`, `codec`, and `events` keys.
- The preamble must contain all required fields defined in section 4
- The codec must contain all required fields defined in section 5.
- The `scaling_constant` in the codec must match the value in the preamble scaling object.
- The `reference_frequency_hz` in the codec must match the `frequency_hz` in the preamble reference object.
- All events must be ordered by non-decreasing `t` value.
- All voice `ratio` fields must be expressed as simplified integer ratios (e.g. 3:2, not 6:4).
- All voice `hz` and `wavelength_nm` values must be mathematically consistent with their ratio and the codec definition.

## 9.2 Recommended

- The `source` field in the preamble should identify the origin of the content.
- The `rational` field in the scaling object should be written in plain language sufficient for a non-specialist to understand.
- The extensions object should be included with null values for all reserved keys, as a forward-compatibility signal.
- Colour values should be included in voice objects for human readability, even though they are not canonical.

## 9.2 Forbidden

- A file must not omit the `ratio` field from any voice object.
- A file must not express ratio as a decimal. Only `integer`:`integer` form is valid.
- A file must not define extension keys that conflict with the reserved keys in section 7.1 for purposes other than those defined.
- A binary serialisation must not be distributed without a corresponding JSON canonical form.

# 10. Versioning

The LIFE specification uses semantic versioning. The version field in the preamble must reflect the specification version used to produce the file.

Minor version increments (1.0 to 1.1) may add new optional fields and new reserved extension keys.

They must not alter the meaning of existing fields. All version 1.x files must be parseable by any version 1.x parser.

Major version increments (1.x to 2.0) may alter the fundamental structure of the format. A version 2.0 parser encountering a version 1.0 file must attempt to parse it using the version 1.0 rules.

A parser encountering a version it does not recognise should attempt to parse the file using the closest known version and report the version mismatch.

# 10. Licence and openness

The LIFE specification is released into the public domain. No permission is required to implement, extend, or build upon it. No royalties or attributions are required.

The reference implementation (life-codec.com) is open source. Contributions to the specification and reference implementation are welcomed.

The goal of LIFE is to become a genuinely universal standard. Universal standards cannot be owned. They can only be tended.

LIFE - Light Instrument Frequency Encoding

Version 1.0 - May 5, 2026

life-codec.org