**Copyright Violation**

**Technical Specification**

**Version:** 0.1  **Date:** 2025‑04‑10  **Author:** Lawrence

**1  Overview**

“Copyright Violation” is a hybrid **VST3 plug‑in / Stand‑alone desktop application** that:

* **Tags** audio stems exported from *RipX* with rich metadata (Title, Artist, Album, Year, Genre, Key, BPM, Stem Type).
* **Indexes** and **browses** those stems inside any DAW (RipX DAW, FL Studio, Ableton, etc.).
* Allows **search, sort, preview, and drag‑drop** of stems into the host.

The tagging engine is embedded in the plug‑in so users never leave the DAW.

**2  Goals**

1. **One‑click tagging** of new stems using information auto‑extracted from the accompanying *.rip* project.
2. **Realtime library**: instantly see newly‑tagged stems, searchable by Key/BPM/Stem Type/etc.
3. **Cross‑platform** (Windows 10+, macOS 11+) with identical behaviour.
4. **Zero vendor lock‑in**: standard ID3v2.3 tags written into WAV/MP3/FLAC so any tool can read them.

**3  Glossary**

| **Term** | **Meaning** |
| --- | --- |
| **Stem** | Individual track (Vocals, Drums, Bass…) exported from RipX. |
| **Stem Type** | Category of a stem (Vocals, Drums, Instruments, FX, Other). |
| **Untagged queue** | List of stems missing one or more required tags. |
| **Library** | Tagged stems, fully searchable and previewable. |

**4  Functional Requirements**

**4.1  Tagging**

* **FR‑T1** Detect new audio files in a configurable *Rips Root* folder.
* **FR‑T2** Locate the matching *.rip* file (same base name, same folder).
* **FR‑T3** Extract <key> and <tempo> from *song.xml* inside the *.rip* ZIP.
* **FR‑T4** Derive **Stem Type** from filename or <layer name> inside *song.xml*.
* **FR‑T5** Present editable fields: Title, Artist, Album, Year, Genre.
* **FR‑T6** On **Tag All**, write ID3v2.3 frames:
  + TIT2 Title
  + TPE1 Artist
  + TALB Album
  + TDRC Year (YYYY)
  + TCON Genre
  + TKEY Key (e.g. “Bb”)
  + TBPM BPM (integer)
  + TXXX[STEM\_TYPE] Stem Type
* **FR‑T7** Remove stem from Untagged list once all mandatory tags exist.

**4.2  Library / Browser**

* **FR‑B1** Display table columns: Title, Artist, Album, Year, Genre, Key, BPM, Stem Type, File Name, Duration.
* **FR‑B2** Search bar filters rows (fuzzy match across all text fields).
* **FR‑B3** Column headers sort ascending/descending.
* **FR‑B4** Double‑click or spacebar previews stem via internal player.
* **FR‑B5** Drag‑and‑drop row → host inserts audio file.

**4.3  Configuration & Persistence**

* **FR‑C1** Remember last *Rips Root* folder.
* **FR‑C2** Cache metadata in SQLite to avoid full rescan on launch.
* **FR‑C3** Preferences dialog: theme (light/dark), audio preview buffer size, alias table for Stem Type detection.

**5  Non‑Functional Requirements**

* **NFR‑1 Performance** Index 10 000 stems < 5 s on SSD (using cache).
* **NFR‑2 Responsiveness** UI operations < 100 ms.
* **NFR‑3 Portability** Single code‑base builds on Windows & macOS.
* **NFR‑4 Extensibility** Modular architecture; new metadata fields pluggable.
* **NFR‑5 Stability** No crash may propagate to host; use JUCE *MessageManagerLock* safely.

**6  System Architecture**

┌──────────────────────────────────────────────┐

│ CopyrightViolation.dll │

│──────────────────────────────────────────────│

│ GUI Layer (JUCE) │

│ • UntaggedTab • LibraryTab │

│ • PreviewPlayer • Preferences │

│ │

│ Controller Layer │

│ • FolderWatcher • LibraryController │

│ │

│ Core Services │

│ • TagEngine (TagLib) │

│ • RipxParser │

│ • StemTypeResolver │

│ • MetadataCache (SQLite) │

│ │

│ Utils │

│ • SettingsStore (JSON) │

│ • AudioHelpers │

└──────────────────────────────────────────────┘

*All layers compiled once as JUCE* ***Stand‑alone + VST3*** *target.*

**7  Technology Stack**

| **Concern** | **Choice** | **Notes** |
| --- | --- | --- |
| GUI & Plug‑in SDK | **JUCE 7** | Generates VST3 & Stand‑alone. |
| Tagging library | **TagLib 1.13** | MIT‑licensed, WAV/MP3/FLAC/AIFF. |
| XML parsing | juce::XmlDocument | Already in JUCE. |
| Database | **SQLite 3** (via JUCE wrapper) | Cache only; not mandatory. |
| Build | **CMake** (preferred) or Projucer | CI‑ready. |
| Unit tests | **Catch2** | TagEngine, RipxParser. |

**8  Component Specifications**

**8.1  TagEngine**

* **read(const File&): StemMeta** – returns struct with all fields.
* **write(const File&, const StemMeta&)** – overwrites/creates ID3 frames.
* Uses TagLib; hides library details from upper layers.

**8.2  RipxParser**

* **getKeyBpm(const File& ripx) → (String key, int bpm)**
* **getLayerNames(const File& ripx) → StringArray** (optional future use).

**8.3  StemTypeResolver**

* Regex / alias table maps filename → canonical Stem Type.
  + Example rules: (?i).\*voc.\* → *Vocals*; (?i).\*drum.\* → *Drums*.
* User‑editable via Preferences.

**8.4  FolderWatcher**

* Monitors root recursively using JUCE FileTreeMonitor.
* Debounces events (coalesce within 500 ms).

**8.5  MetadataCache**

* **stems(id INTEGER PK, path TEXT UNIQUE, mtime INTEGER, json TEXT)**
* On launch: compare mtime; refresh stale entries.

**9  File‑naming Convention**

* Default RipX export assumed: <Title>\_<StemType>.wav.
* Supported separators: space, underscore, dash.
* If filename ambiguous, fallback to <layer name> from song.xml.

**10  ID3 Frame Mapping**

| **Field** | **ID3 Frame** | **Format** |
| --- | --- | --- |
| Title | TIT2 | UTF‑8 text |
| Artist | TPE1 | UTF‑8 text |
| Album | TALB | UTF‑8 text |
| Year | TDRC | YYYY |
| Genre | TCON | UTF‑8 text |
| Key | TKEY | e.g. “C#m” |
| BPM | TBPM | Integer |
| Stem Type | TXXX (desc="STEM\_TYPE") | UTF‑8 text |

**11  User Interface Flow**

1. **Launch** plug‑in → select *Rips Root* (first run).
2. **Untagged Tab** lists new stems.
3. User fills blanks → **Tag All**.
4. Stems vanish from Untagged → appear in **Library Tab**.
5. Search/filter, preview, drag‑drop.

**12  Testing Strategy**

* **Unit tests** – TagEngine read/write round‑trip; RipxParser key/BPM extraction.
* **Integration** – End‑to‑end tagging of sample RipX project; verify tags with Kid3.
* **Performance** – Scan 10 k stems, measure <5 s on CI machine.
* **DAW host tests** – Load in RipX, FL Studio, Ableton, Reaper.

**13  Build & Deployment**

* **Prerequisites**: CMake ≥ 3.22, JUCE 7, TagLib headers, VST3 SDK (Steinberg).
* **Windows**: MSVC v143, set VST3\_SDK\_DIR env var.
* **macOS**: Xcode 14+, notarisation script TBD.
* **Artifacts**: CopyrightViolation.vst3, CopyrightViolation.exe (stand‑alone).

**14  Roadmap & Milestones**

| **Sprint** | **Deliverables** |
| --- | --- |
| **0** | Repo scaffold, CI, CMake, TagLib submodule. |
| **1** | TagEngine + unit tests. |
| **2** | RipxParser + StemTypeResolver. |
| **3** | CLI proof‑of‑concept (batch tagging). |
| **4** | JUCE GUI skeleton, FolderWatcher. |
| **5** | Untagged Tab functional. |
| **6** | Library Tab + Preview + Drag‑drop. |
| **7** | Preferences, SQLite cache. |
| **8** | Cross‑platform build, installers. |
| **9** | Beta testing, bug‑fix. |

**15  Licensing & Legal**

* Code under **MIT License**.
* TagLib (LGPL/MPL dual) – dynamic link on Windows, permissible.
* Respect personal‑use clause; users responsible for copyright of stems.

**16  Risks & Mitigations**

| **Risk** | **Likelihood** | **Impact** | **Mitigation** |
| --- | --- | --- | --- |
| RipX format change | Low | Medium | Parser unit tests on sample projects each release. |
| Host DAW crashes on plug‑in error | Low | High | Wrap critical calls in try/catch, never throw across VST ABI. |
| Large libraries slow | Med | Med | SQLite cache + threaded scan. |

**17  Open Questions**

1. Support for **CLAP** plug‑in format?
2. Auto‑renaming filenames with \_BPM\_Key suffix for FL Browser compatibility?
3. Optional cloud sync of metadata?

**End of Document**