FiberQ 1.3.0 — Project versioning, safe migrations, and stable feature IDs

A user guide to the project-versioning features introduced in **FiberQ 1.3.0**. For the field-by-field data model, see the schema reference.

FiberQ 1.3.0 gives every project a **version marker**, upgrades **older projects automatically and safely** when you open them, and gives every feature a **stable identity** (`fiberq_uuid`) that survives edits, exports, and round-trips between tools and systems. Together these make FiberQ projects forward-compatible: today's data can be recognised and migrated by future releases without losing anything.

---

1. The project schema version

Each FiberQ project now records which **data schema** it was built against — the current schema version is **`1.0`**. This is separate from the plugin version you see in the QGIS Plugin Manager; the two advance independently.

The marker is:

- **saved into the QGIS project** when you save, and
- **mirrored into the GeoPackage** (a small `_fiberq_metadata` table) on a full project export, so the version travels with the data file itself.

You don't have to do anything to opt in — new projects are stamped automatically, and older ones are upgraded on load (next section).

---

2. Opening an older project (automatic migration)

When you open a project created by an **earlier** FiberQ version, FiberQ detects the older (or missing) schema marker and upgrades it in place. You'll see a message such as:

```
FiberQ: Upgraded project schema 0 -> 1.0 (assigned 50 UUIDs across 15 layers)
```

and a per-layer summary in the QGIS **Log Messages** panel (the FiberQ UUID tab). What the upgrade guarantees:

- **Lossless** — it only adds the identity field and fills it in. Your existing attributes and geometry are never modified or removed.
- **Idempotent** — running it again does nothing. Once a project is at `1.0`, re-opening it shows no upgrade message.
- **Safe on non-FiberQ / blank projects** — an empty QGIS project, or one with no FiberQ layers, is left completely untouched.
- **Never downgrades** — a project stamped newer than your installed plugin is left alone, so a newer project can't be silently rewritten by an older plugin.
- **Only stamps on success** — if the upgrade can't be written (for example a read-only or locked GeoPackage), the marker is left unchanged and the upgrade simply retries next time. The version never claims a migration that didn't actually reach disk.

> **Tip:** on a file-backed layer (a GeoPackage or a shapefile), the `fiberq_uuid` column and its values are written to the data file while the project opens. Only the version marker waits until you **save** the project (or export it to GeoPackage). If you close without saving, the next open finds every identity already in place and simply stamps the marker again.

---

3. Stable feature identity: `fiberq_uuid`

Every FiberQ feature now carries a **`fiberq_uuid`** — a stable, globally-unique identifier that stays with the feature no matter what happens to it:

- It is **added automatically** to every FiberQ layer and **filled on creation** for every element, across every drawing tool (routes, cables, optical slack, pipes, service areas, objects, poles, manholes, and all the enclosure/termination layers).
- It **survives edits and exports**, so a feature keeps the same identity across operations and between the plugin and any external system that consumes FiberQ data. This is the anchor for interoperability and for the open interchange-format work.
- It is a **standing invariant**, not a one-time conversion: an idempotent backfill runs on **every** project load, so features that appear later (imports, external edits, interchange round-trips) are healed and given an identity too.
- Splitting a feature (for example **Cut infrastructure**) gives each resulting half a **fresh** `fiberq_uuid` — the two parts are genuinely distinct features, not two copies sharing one identity.

`fiberq_uuid` is shown in the attribute table under the display alias **"FiberQ UUID"**.

4. What to check

- Open an existing FiberQ project → confirm the upgrade message appears once, and that every layer now has a **FiberQ UUID** column populated for all features.
- Save, close, and re-open → confirm **no** second upgrade message (the marker is now `1.0`).
- Create new elements with any tool → confirm each gets a `fiberq_uuid` immediately.

5. Quality assurance & compatibility

- **Full manual source review.** Ahead of the 1.3.0 release, the maintainer manually reviewed the **entire plugin source — every Python module** — in addition to the automated checks below.
- **Clean repository scan.** The plugin reports **zero findings** on the plugins.qgis.org Security & Quality scan: flake8 (with `--max-line-length=120 --ignore=E501`) and Bandit **at all severities** both report 0, and the repository's **Qt6 compatibility check** also passes with 0 enum errors — every QGIS enum access is scoped for PyQt6 / QGIS 4.
- **Automated tests on both Qt generations.** The pytest suite (including the new schema-version, migration, and `fiberq_uuid` invariant tests) runs green against **QGIS 3 / Qt5** and **QGIS 4 / Qt6**.
- **Manual acceptance testing across QGIS versions.** Beyond the automated suite, 1.3.0 was manually verified end-to-end on **QGIS 3.40 (Qt5)** and **QGIS 4.2 (Qt6)**, with both **newly created** projects and **older projects** migrated from a previous FiberQ version. Each combination was checked for:
- the on-load schema migration (`0 -> 1.0`) and its per-layer UUID summary;
- **idempotent re-opening** — after saving a migrated project, re-opening it shows no second upgrade and re-assigns no UUIDs;
- `fiberq_uuid` assignment on newly drawn features and backfill on existing ones;
- **cross-version opening** — a project migrated and saved under QGIS 3 / Qt5 opens cleanly, with no re-migration, under QGIS 4 / Qt6;
- layer rendering and styling (including the SVG map icons), the **Optical Schematic View**, **GeoPackage export**, **Cut infrastructure**, and new-element creation with the drawing tools — all with a clean QGIS message log (no errors).
- **Backward compatibility.** FiberQ 1.3.0 targets QGIS 3.22 LTR through QGIS 4 / Qt6, and opens projects from earlier FiberQ versions via the migration described above.

FiberQ is free software under the GPL-3.0-or-later licence. Source, issues, and releases: https://github.com/vukovicvl/fiberq. This release was developed with support from the NLnet NGI0 Commons Fund.