

IsomerAPI development update

Changes since version 1.4.3 through version 2.0.1

Release date 23 October 2026 | Documentation date 24 October 2026

Purpose and scope

IsomerAPI 2.0.1 adds a shared database and decay core for the standalone application and LISE++ integration. Since the 1.4.3 baseline, development has also added database record editing, improved level schemes, and repaired Windows build and deployment behavior. This update explains the user-visible changes, the new interface, and the checks needed when maintaining or integrating the code.

Version 1.4.3, dated 23 August 2026, is the baseline and includes the move from w_Isomer to g_Isomer. Earlier startup, filtering and proxy-cache improvements belong to the previous development period.

Main developments

- Database editing now supports adding, updating and deleting records, with validation and protection of linked levels.
- Level schemes display half-lives in suitable units and preserve comparison signs. Selected-level decay plots load the complete isotope, including daughters excluded by table filters.
- IsomerCore provides a structured transmission result for an isotope. The GUI writes that result to text; LISE++ can use it directly in memory.
- MinGW and MSVC project and deployment changes address header resolution, legacy source selection, runtime dependencies and Windows startup behavior.

Hudson presentation at DNP2026

For Hudson's DNP2026 presentation, open the companion PDF below. Keep the PDF and this Word document in the same folder when copying or distributing them.

[Hudson DNP2026 presentation — IsomerAPI DNP2026 v2.pdf](#)

Release history after the baseline

Version and date	Development
1.4.4 24 Aug	Repaired compilation involving stylesheets, Constant.h and the level proxy.
1.4.5 24 Aug	Fixed NNDC access from decay-from-level plots; marked Save As as work in progress.
1.4.6 and 1.4.7 25 Aug	Moved styles and icons for LISE++ integration; refined uncertainty headers and isomer counts; repaired stale resource and project references.
1.4.8 23 Sep	Introduced the early database record editing utility.
1.4.9 and 1.4.10 26 Sep	Revised record editing, added range clear buttons and corrected disabled group box colors.
1.4.11 29 Sep	Improved half-life units and uncertainty signs; removed always-on-top scheme behavior; refined table labels.
1.4.12 10 Oct	Updated the About link and record editor formatting.
1.4.13 and 1.4.14 14 Oct	Repaired MSVC include paths and an include guard; made deployment portable; added runtime files; consolidated the release after build verification.
1.4.15 16 Oct	Kept one Edit Database Records menu command; prevented MSVC from compiling legacy files with matching names; updated the main form and About dialog.
1.4.16 23 Oct	Disabled the Windows console window at startup and replaced an invalid cursor resource with the built-in pointing-hand cursor.
2.0.0 23 Oct	Extracted IsomerCore and added structured transmission requests, branch preservation, shared-daughter handling and malformed-link warnings.
2.0.1 23 Oct	Fixed selected-level decay drawing by loading complete isotope data and merging all distinct gamma branches.

The 1.4.16 startup changes are recorded in release commit d72e7c2 and remain present in the current project. The current runtime version header proceeds from 1.4.15 to the 2.0 entries.

Database editing and level schemes

Edit database records

Open Options > Edit Database Records. The editor lists records under the current main-window filter. Select a row to edit it, or choose Add Record to create a new entry. Save updates an existing record; Add to Database inserts the new entry; Delete requests confirmation before removal. Closing or changing records prompts when edits are unsaved.

A record requires A of at least 1, Z between 0 and A, a positive LEVEL_ID, level energy, half-life and the contributor NAME. The editor checks numeric fields and verifies that FINAL_LEVEL_ID, when supplied, refers to an existing level of the same isotope. Existing records cannot change isotope identity or LEVEL_ID; create a separate record for that purpose.

Rows can share a level because the database stores separate gamma transitions. Editing shared level attributes updates those attributes across the rows for that LEVEL_ID. A new transition attached to an existing level must match its shared attributes. New entries receive a generated GAMMA_ID, while a new level is offered a suggested LEVEL_ID.

Writes use SQL transactions. The editor prevents deletion of the last row for a level still referenced by another transition. After saving, the changed row appears at the top even if it falls outside the main filters. On closing the editor, source choices and filtered results refresh when data changed.

Drawing and selection

Draw Scheme opens the drawing-choice dialog. For Decays from selected level, select a level in an Isomers table on the active page. Missing selection and unsupported Development View selection produce a warning. Filtered isotope plots remain a separate drawing choice.

In 2.0.1, the selected row determines the starting level. The application then reads the full isotope through IsomerCore, groups rows by LEVEL_ID and retains distinct GAMMA_ID branches. A short-lived daughter excluded by the main table filters can therefore still appear in the decay plot. This addresses the reported ⁷⁸Zn case involving level ID 1562 and gamma ID 4163.

Scheme windows are independent top-level windows with delete-on-close behavior. This supersedes the older owned-window arrangement that could keep a scheme above the main window. NNDC links, half-life unit selection and half-life comparison signs were also improved during the 1.4 series.

Shared core and transmission data

Interface and database lifetime

`g_Isomer/core/IsomerCore.pri` exposes a C++17 module requiring Qt Core, Qt SQL and the QSQLITE driver, without a Widgets or LISE dependency. Construct `IsomerCore` after `QCoreApplication` and use it on its creating thread. Its managed connection opens an existing database file and reuses the connection for repeated opens of the same path. Release borrowed SQL handles, models and queries before destroying the core.

`TransmissionRequest` specifies `a`, `z`, `cascadeCutoffUs`, `minimumParentHalfLifeUs` and optional `parentLevelIds`. Defaults are a minimum parent half-life of 0.1 microseconds and a cascade cutoff of 20 microseconds. Energies are in keV. Internal isotope map keys use `Z` then `A`, while the request names `A` and `Z` explicitly.

`TransmissionResult` contains parent levels with collected gamma transitions, an error string and warnings. An empty levels vector with no error means no matching data. A nonempty error identifies a failed request. Nuclear data types are in the `IsomerData` namespace to avoid a collision with LISE's `Level` type.

Traversal and filtered parents

The request reads the complete isotope and groups repeated `LEVEL_ID` rows while retaining distinct gamma branches. Parent IDs and the minimum parent half-life restrict starting levels. Traversal stops at ground state or at a daughter whose half-life reaches or exceeds the cascade cutoff. Each reachable level is expanded once per parent, avoiding repeated shared daughters. Missing links retain measured gamma lines; cycles terminate with warnings.

`Calculate > Isomer Transmission` uses the filtered isotope to select parent IDs, then obtains daughter data from the complete isotope through the core. The GUI passes a minimum parent half-life of zero because its parent selection already comes from the filters. The dialog supplies the cutoff. Database errors stop the export; link warnings are shown with details while known gamma lines remain exportable.

The output keeps the existing semicolon-separated `LVL` and `GAM` records. `LVL` carries level energy, uncertainty, `IT` ratio, half-life, half-life uncertainty and spin. `GAM` carries gamma energy, uncertainty, level energy, level uncertainty, spin, gamma intensity and multipolarity. LISE++ can consume the same structured result without parsing this text file.

Physics and integration limits

The interface preserves stored gamma intensities and the established parent-isomer half-life and `IT`-ratio convention. Cascade collection does not solve coupled feeding kinetics or introduce new branch normalization. The LISE integration described in `core/README.txt` owns isotope caching and uses the same database under `LISErootPATH/lisecfg`. LISE adapter and production integration tests reside in the separate LISEcute project.

Build maintenance and verification

Current source locations

Location	Responsibility
IsomerAPI.pro	Qt modules, core inclusion, source lists, output directories and version metadata.
g_Isomer/L_isomerAPI_version.h	About version 2.0.1 and release date 23 October 2026.
g_Isomer/w_IsomerAPI_records.cpp	Record editor, validation and transactional writes.
g_Isomer/w_IsomerAPI_drawing.cpp	Selection, drawing choices and complete-isotope loading for selected-level decays.
g_Isomer/w_IsomerAPI_transmission.cpp	Transmission request construction and LVL/GAM export.
g_Isomer/core/IsomerCore.h and .cpp	Database lifetime, isotope loading and decay/transmission operations.
g_Isomer/core/tests/IsomerCore_test.cpp	Standalone Qt tests for database lifetime, filtering and legacy decay traversal.

Build and deployment changes

The project includes g_Isomer for header resolution and enables no_batch for MSVC to avoid selecting legacy sources with matching basenames. Windows console mode is disabled. MinGW output goes to _install and MSVC output to _install_MSVC. Both packages require lisecfg/nndc_DB_fullScan.sqlite beside the executable. Default builds keep debug traces disabled; rerun qmake with ISOMERAPI_DEBUG_TRACE=1 to enable them.

Checks for the next release

- Check About and executable metadata against the intended release. Current qmake versions are 2.0.1.1 on Windows and 2.0.1 elsewhere.
- Exercise record add, save and delete; reject invalid daughter links and deletion of a referenced level. Confirm shared level fields remain consistent.
- Plot a selected parent with daughters outside table filters and compare all branches with NNDC. Test 78Zn and verify ordinary window stacking.
- Compare a transmission export with the structured result. Check cutoff boundaries, multiple branches, shared daughters, missing links and cycles.
- Run the standalone core Qt tests and both compiler builds. The checked-in tests cover connection reuse and cleanup, filters, observed edits and legacy traversal; LISE-specific validation requires the LISEcute project.

Remaining development work

Save As remains labeled work in progress. Emissions in coincidence with selection is also marked work in progress and has no implemented drawing path. Keep these separate from the implemented transmission text export. Cached LISE data must be refreshed after database edits, for example by restarting LISE or changing its half-life limits as described in the integration notes.

Source references: current version history, IsomerAPI.pro, core/README.txt, IsomerCore interface and tests, record editor and drawing/transmission implementations; release commit d72e7c2 for version 1.4.16.