NEXCHROM · USER MANUAL

Workflows

Day-to-day operation of NexChrom: master data, CDS import, methods, calibration, execution, review and certificates.

Qualimetric · Nijmegen, The Netherlands · info@qualimetric.nl · nexchrom.nl · Version 2.89.0 · 3 November 2026

CONTENTS

1. Scope
2. Introduction
3. Page Orientation
4. Getting Started
5. Master Data Setup
6. Workflow A — Analysis via sample
7. Workflow B — Analysis via CDS file
8. Inspecting CDS Files Without Linking (CDS File Browser)
9. Workflow C — Calibration
10. Workflow D — External Certificate Import (PDF)
11. Dashboards
12. Maintenance
13. User Management
14. E-mail Alert Notifications
15. Upload Instrument Documentation
16. Troubleshooting and Common Issues

Scope

This user manual describes how to perform the day-to-day workflows supported by NexChrom as it currently exists. It guides a new user through master-data setup, CDS file import, method configuration, calibration, execution, result review, and certificate generation. It targets the application aligned with database schema DDL v2.12.0 (migrate_schema_v2.12.0.py).

Introduction

NexChrom is a Streamlit-based application for managing gas chromatography workflows. It runs against a SQLite database, stand-alone on one PC or as a central server that the users of the laboratory open in their browser (Settings › System, network access). All screens are reached from the left-hand navigation once the application is started. Two primary analysis workflows are supported:

- Analysis via sample — you start from an existing Sample, then link one or more CDS files and run the calculation.
- Analysis via CDS file — you start from CDS files on disk, and a Sample is created for you during import.

Page Orientation

Archiving instead of deleting

- Every master-data table (locations, CDS file formats, users, samples, sample types, instruments, methods, calibration models, certificates, components, aliases) shows **active records only** by default.

- Set **Archive status** to *Archived* to see the **archived records only** (never both at once); set it back to *Active* to return to the active ones. The **Archived At** column (the archive moment, on the tables that record it) appears in the *Archived* view only; in the *Active* view it would always be empty, so it is hidden there. The filter looks the same as the type filters and is always the rightmost control above the table. A table with extra columns hidden by default (Components: atom counts) has a **Columns** filter of the same kind directly left of it (*Standard* / *Atom counts*). On Samples the supplier, parent sample and number of linked CDS files, and on Instruments the data folder, CDS format, location, file name filter, documentation folders, notes and created-at are not grid columns at all: tick one instrument and edit them in the **Details of <instrument>** table under the grid — double-click a value (Location is a drop-down of the active locations, CDS format one of the active file formats), the edit turns yellow and is saved with *Save* of the grid or dropped with *Discard*.
- Select one or more rows and press **Archive**: the records get is_current = 0 (and an archive moment where the table records it). Nothing is deleted — analyses, calibration models, runs and files linked to the record stay where they are, and the record still resolves wherever it is referenced. In the archived view, **Unarchive** reverses it.
- Archived records are not offered in the selection lists of the other pages (instrument on Calibration and Analysis, methods on Execute). On *Analysis via sample* the sample list has an **Archive status** filter at the far right of the filter row (sample-type filter left, link status and archive status right): *Active* (default) or *Archived*; switching it reloads the sample list and clears the sample and file selection, but leaves the instrument and detector template chosen in section 1 as they are. Archived (historical) calibration models are not offered on Execute either: they stay readable on Calibration with their fits, and the runs calculated with them keep their results. CDS files and Analysis records are measurement records and cannot be archived.
- Both actions are written to the Change history. Use **Delete** only for records that were entered by mistake; records referenced by other data cannot be deleted anyway.

Every main CRUD page (Administration, Sample, Analysis, Method, Calibration, Execute, Review, Certificates) exposes an **'About this page'** expander directly below the page title. Expanding it shows a short info banner explaining what the page does and a workflow diagram that highlights the current page within the full NexChrom flow (Component → Sample → CDS files → Certificate → Method → Calibration → Analysis → Execute → Review → Analysis certificate). Some pages (Certificate, Calibration, Administration) only show the part of the flow they participate in. Use this expander whenever you want to confirm where a page sits in the overall workflow before you start entering data.

Filters and buttons

- No filter offers an *All* option: with nothing selected the table shows every record, and clicking the selected option again clears the filter. Category filters (sample type, location, certificate type) let you select several values together; two-way filters (Linked / Not linked, In use / Not in use, Matched / Unmatched, Named / Unnamed, Released / Unreleased, the Review status filters) take one value at a time.
- The controls above a table are **segmented filters** on one row: a type or location filter (the values; nothing selected = every value) at the far left, and at the far right *Columns* (*Standard* / the extra columns that are hidden by default, e.g. *Atom counts* on Components) followed by *Archive status* (*Active* / *Archived*), which is always the rightmost control. Clicking the selected option again clears it, which counts as the default.
- A filter that has nothing to switch is shown **disabled** with an explaining help text instead of being hidden — for example the *Peaks* filter of a detector without unidentified peaks in the CDS file browser.
- Wide tables — Users, Certificates, Components (with *Atom counts*), Locations, File formats, the Method tables, the Calibration tables, the peak tables of the CDS file browser, the component table of the certificate import and the Calculation results on Execute — keep a readable width per column and **scroll horizontally**; one text column (Notes, Description, Component, …) takes the free width, and the Select column and the identifying column (username, certificate number, component name) stay pinned on the left while you scroll. The

approval fields of a certificate (approval status, approved at/by, run, PDF, PDF hash) are read-only: the app fills them when a certificate is issued or approved.

- **Sorting**: click a column header to sort the table (a second click reverses, a third clears). The order stays while you tick rows or save; the date columns of the Certificates table (Issued At, Valid Until, Approved At, Created At, Archived At) sort by date, empty values last.
- **Action buttons** have a material icon, a short label and one fixed width, and sit on one row below the table: **Save** (blue), **Create** (green), **Delete** (red), **Discard** (grey) and **Archive** / **Unarchive** (amber). Other pages use the same colours: blue for the main action (*Calculate*, *Store for review*, *Execute*, *Approve*, *Issue certificate*, *Release*, *Link*), green for actions that create something (*Add*, *Duplicate*), red for *Reject*, grey for utilities (*Clear*, *Rename*, *Restore*, *Test*). Hover over a button for the detail its short label leaves out.

Feedback and messages

Save, Create, Delete and Discard actions display their success or error message as a **full-width banner** directly below the page title (or directly above the grid). The message is preserved across the automatic page refresh that follows an action, so confirmations no longer flash away. If a save is partially successful, the banner lists every row that was skipped — both new rows that never reached the database and edited rows whose required fields were blanked out — with the sample / component / certificate name for easy identification. Newly created rows now appear at the **top** of their grid, so they are always visible without scrolling.

Messages of one kind stand together in **one coloured box**: **red** — it blocks, resolve it first; **yellow** — look at it before you go on; **blue** — for information; **green** — done. One message stands as it is, several are a list; a message with details keeps them on lines of their own under a bold headline. Where one box holds what blocks two different decisions (Review), each decision has its own titled list. An explanation you do not need every time is in the help of the field or button: hover the ⓘ or the button.

Getting Started

Launch the Application

- Open a terminal in the project root.
- Activate the virtual environment: .venv\Scripts\activate on Windows or source .venv/bin/activate on macOS / Linux.
- From the project root, run: streamlit run app/main.py.
- Streamlit opens the application in the default browser.
- The login dialog opens: choose your user, tick *I have read and agree to the Software Licence Terms.* and click **Login** — logging in accepts the Qualimetric Software Licence Terms, published at nexchrom.nl/licence (the dialog names this link; the copyright line *© Qualimetric. All rights reserved.* closes the dialog). Behind the dialog the sidebar already shows the full menu with its icons, the same as after login; clicking a menu item before logging in brings the login dialog back. After login *Overview* opens.
- The NexChrom block at the top left of the sidebar (logo and name) is a link: a click opens the NexChrom page of the website (nexchrom.nl/nexchrom) in a new browser tab, also from the login screen; NexChrom itself stays open.

Select the Active Database

- Open the Settings page from the sidebar.
- In the Database tab, enter or browse to the SQLite database file (for example data_local/database/nexchrom_db_v2.12.0.db).

- Save the setting. Overview names the active database at the bottom of the page (Settings option *Show divider and active DB footer on pages*, on by default).

Verify Master Data Availability

- Check on the Administration page, tabs **Instruments** and **Components**, that the instruments and components are present (the sample types Sample, Quality Control, Calibration and Reference come with the database). Any missing master data blocks downstream workflows.
- If anything is missing, follow the relevant section in Master Data Setup below.

Master Data Setup

Register a Location and an Instrument

- Open the Administration page and create the Location on the Locations tab (name, address, building and other context).
- On the same page open the **Instruments** tab (the first tab of Administration; before NexChrom 0.10.0 Instrument was a page of its own) and create the instrument in the Instruments table. The *Filter on location* control at the top left of the table (the Location names from Administration; several can be selected together, nothing selected = every location, click a selected location again to clear it) narrows the table to those locations; new rows always stay visible. Select the Location and the CDS File Format that matches how the instrument exports its data.
- Give every physical instrument its **own data folder** (the working convention is `data_local/gc_data/<name>-<manufacturer>-<model>-<serial>/`) and point **data_source_path** at it; the folder name is a human-readable snapshot, the data_source_path link is authoritative. As a fallback for analyzers that must share one folder, **file_name_filter** restricts the listing to file names containing a substring (e.g. the serial number); leave it empty otherwise.
- On the Administration page, File formats tab, verify that the format you need (Agilent ChemStation, Scion CompassCDS, or Thermo Chromeleon*) is present. Add one if needed.

** Thermo Chromeleon archives (`.cmbx`) hold raw signals and no peak results. They can be viewed in the CDS File Browser (format *Thermo Chromeleon (.cmbx)*) with NexChrom's own, experimental integration, and on Analysis via sample their injections can be linked to a sample one by one, for testing (Workflow A, step 2): the areas are calculated but never stored as a run or a calibration until the integration is validated; a whole archive is refused. A native, validated import stays planned; for accredited work a Chromeleon instrument imports through CSV/Excel exports of its peak table (format *CSV/Excel/TXT*).*

Record the Scope and the QC Limits (first version)

The **Scope** tab on the Administration page holds the official accreditation scope — the registration header and the CMC lines per component, as on the RvA annex to the declaration of accreditation — next to the internal requirements. With the accreditation-scope check on, the results are compared with the CMC lines of the active scope (see *Derived limits* and *Accreditation scope at Execute and on the certificate* below); the internal requirements are recorded, not yet checked against analyses. Work top-down through its four sections:

- **Step 1 — Register the accreditation scope header.** Open the expander *Accreditation scope* and click *Create*: one row per annex / registration version. Fill in the **Laboratory** (a Location from the Locations tab; required), the **Registration** number (e.g. `K 999`; required) and the **Status** (*draft* / *active* / *suspended* / *withdrawn*; required, *active* by default), and, as printed on the annex, *Organisation*, *Valid from* / *Valid to* (dates as ISO `YYYY-MM-DD`), *Normative document* and *Body* (pre-filled with EN ISO/IEC 17025:2017 and RvA (Dutch Accreditation Council)), the location *Code* used on the annex (e.g. `Z`), *Replaces annex* (date of the previous

annex), *Status since* (e.g. the date of a voluntary suspension), *Head office*, *Register URL*, *Source file* (root-relative path of the annex PDF) and *Notes*. *Save* stores the header. Then tick **Select** on the scope you want to work on: the CMC lines and the Annex view below follow that one scope (ticking another scope moves the tick). The intention is one active scope per laboratory (not enforced yet).

- **Step 2 — Enter the CMC lines per section.** Open the expander *Calibration and measurement capabilities (CMC)*; it shows the lines of the scope ticked in step 1 (its label — registration · organisation · validity, a status other than active in brackets — stands above the table). Optionally choose the **Section** (*All sections* or one HCS code with its section, e.g. `RM 2 1 Natural gas`); the table reloads for that choice and lists the lines in annex order. *Create* adds a line pre-filled with the selected scope, the chosen section (else the section of the last line), the remark of that section, the location code of the header, the units `cmol/mol` and `%` and the next *Order* number. Per line enter the **Name on annex** (required, exactly as printed), the **Component** it maps to (leave it empty for a group line such as *Heptanes*, *Octanes* or a co-eluting pair, and name the constituents in *Notes*), *Range low* / *Range high* with *Range unit*, *CMC low* (the relative CMC at the low end, or the single value of the line) and *CMC high* (the relative CMC at the high end; leave it empty when the CMC is constant over the range) with *CMC unit*, *Remark* (e.g. `Analysis to ISO 6974`), *Loc.* (only when a line deviates from the header code) and *Order*. *Save* stores the lines. Without a ticked scope the section only shows *Select an accreditation scope in the table above (Select checkbox)*.
- **Step 3 — Check the Annex view.** Open the expander *Annex view*: it renders the ticked scope as on the RvA annex — the header block (annex to declaration of accreditation, normative document, registration number, organisation, laboratory with code, validity, replaced annex, status with date, register link) and a read-only table *HCS code / Measured quantity / Range / CMC / Remarks / Location* with one section row per section and one row per active line in order; ranges read as `(0.03 – 0.1) cmol/mol`, CMCs as `12.5 %` or `0.45 % – 0.1 %`, the header notes appear below the table. Compare it with the annex; nothing is edited here — correct the lines in step 2.
- **Step 4 — Record the internal requirements.** Open the expander *Requirements and scope limits*. Every row is one requirement or scope limit for a **Laboratory** (required), optionally narrowed to one **Instrument** and one **Component**: a *Requirement* text, a *Lower limit* / *Upper limit* with *Unit*, a *Reference* (standard or accreditation document) and *Notes*. Use it for what the laboratory is set up to measure beyond — or more strictly than — the official scope.
- Not yet checked: the environmental requirements of the scope (they need the ambient conditions per analysis); one active scope per laboratory is not enforced. Editing the scope follows the *Administration* row of the rights matrix.
- **QC limits from the history of a QC sample.** A QC sample is a cylinder of type *Quality Control* (or *Reference*) that you analyse repeatedly on an instrument — like any sample: link its CDS files, Execute, have the run approved on Review. Once it has been analysed for a while under conditions you judge normal and good, open the **Quality control** page, tab **Control settings**, block *Limits from QC history or certificate*: choose the **Instrument** (only instruments with QC analyses are listed, each with its number of QC samples) and then the **QC sample** analysed on it (with its number of analyses there; a QC sample without analyses is named below the drop-downs but cannot be chosen), basis **History** and the period (default: from the first approved run until today). The table shows per component n, *Centre* (the mean), *STD* (the standard deviation s), *RSD*, *Min* / *Max* and the bands: warning = mean ± 2·s, action = mean ± 3·s (Settings), next to the limits currently in force. The values of a row are rounded on its STD: STD to the significant digits of Settings › System › *Significant digits for uncertainties*, the other concentrations to the same decimal place, RSD with the same number of significant digits. A green banner means the set is **established** (at least 3 months and 13 runs by default); an orange one means it is below the minimum — you can still save it as **provisional** after ticking the confirmation, and replace it later. **Save as limits** (manager / admin) writes the limits and archives the previous set; its confirmation appears directly above the button.

- **The runs of the period.** Under the limits a table lists every run of the period with its status against these preview bands (the bands only, no Westgard rules): **Include** (ticked), *Run*, *Analysed*, **Status** — *fail* in red (outside the action band), *warning* in yellow (outside the warning band), *pass* in green —, *Outside the bands* (the components outside a band, e.g. *methane (fail), nitrogen (fail)*; hover a cell for a long list) and *Model*. Fails come first, then warnings, then passes, each by date; click a column header to sort otherwise. The line above the table counts them, e.g. *Runs in the period: 104 of 104 runs included · 1 fail · 5 warning · 98 pass*. Untick **Include** to leave a run out of the derivation: the limits above are derived again at once from the runs still included, and the line names the excluded run. The excluded run stays in the table, grey and in italics, with its status against the new bands. Nothing is stored before *Save as limits*, which notes the excluded runs with the saved limits (add the reason under *Notes*). Another instrument, QC sample or period starts again with every run included.
- **Exclude only a known incident.** Excluding a run derives the limits again and makes the bands narrower, so another run can then fall outside them — for example, in a year of two runs a week with one deviating run, excluding that run made another run a fail against the new bands. With action limits at 3·s about 0.3 % of the values fall outside the bands by chance (for a QC gas of 13 components analysed twice a week, 3 to 4 of the some 1300 values of a year). So exclude only a known incident, not every run that falls outside the bands.
- **QC limits from the certificate.** For a QC sample with a certificate (external QC gas, Workflow D) choose basis **Certificate**: every certified component gets warning = value ± 1·U and action = value ± 2·U (Settings). The table shows *Certified*, *U* and the bands next to the limits in force, rounded on U (U to the significant digits of Settings › System, the other values to the same decimal place). This is also the fallback when there is no history yet.
- **QC limits by hand.** Tab **QC limits**: *Create* a row with basis *manual* — *Component*, optional *Instrument* and *QC sample*, *Action low* / *Action high* (and optionally the warning bounds), *Unit*, *Notes* — and *Save*. Rows written by a derivation carry read-only provenance (centre, s, n, period, set by / at, provisional); re-derive instead of editing them. *Archive* takes a limit out of force.
- **Check a QC run.** Execute a QC sample like any other sample. In *Result*, under the results, the line **QC comparison** gives the overall verdict — pass, warning, fail, no limit — and the expander *QC comparison per component* shows every component against the limits in force: *Measured*, *Centre*, *STD* (the standard deviation of the limit), *Action low* / *Action high*, *Deviation %* (two decimals), *Status* and *Rules*. Measured, Centre and the limits are rounded on STD (STD to the significant digits of Settings › System, the others to the same decimal place); *Rules* names the violated Westgard rules and reads *all met* when none is violated (*no limit* without a limit). After *Calculate* it is a preview that stores nothing, a stored run keeps its verdict (the QC check runs at *Store for review*) and shows it on the Review page: the *QC* column in the runs table and the *QC comparison* block in the run detail — the same columns plus *Warning low* / *Warning high*, *Unit*, *z* (two decimals) and *Basis*, the *Status* cell in green (pass), yellow (warning) or red (fail); hover the headers *STD*, *Status*, *Rules* and *z* for their meaning. A run stored before limits existed has no verdict yet — a reviewer clicks **Run QC check now**; **Re-run QC check** repeats it against the limits in force now. **A failed QC run cannot be approved**: reject it and investigate the analyzer; a warning may be approved with a note. Both dashboards list failed QC runs of the last 30 days and pending runs with a warning (Overview under *Alerts*, Worklist as cards under *Act now* and *Needs attention*).
- **Westgard rules.** Tab **Control settings**, *Westgard rules*: tick the multirules you apply — 1_2s (warning), 1_3s, 2_2s, R_4s, 4_1s and the N-in-a-row rule (set N, e.g. 8, 10 or 12) — and *Save*. From then on every QC check and the control chart judge a result together with the preceding runs of the same QC sample; the violated rule codes appear in the *Rules* column on Execute and Review (*all met* when none is violated) and in the chart's hover text. Stored verdicts keep their rules; *Re-run QC check* on Review applies a new selection.
- **Two kinds of QC sample.** An **external QC gas** is a sample of type *Quality Control* with the supplier's certificate imported or registered on the Certificates page (origin external). An **internal QC gas** is a cylinder you analysed repeatedly as an ordinary sample: tab **Control settings**, *Promote a sample to QC* — choose the

sample, instrument and period (approved runs only; at least the minimum number of runs of Settings is advised), name the new QC sample and click *Promote to QC*; the confirmation appears directly above the buttons. To break a promotion off, click the grey **Discard** next to it: the choices go back to their defaults and nothing is stored (*Choices discarded — nothing was promoted.*). NexChrom creates the QC sample (parent = the original, same cylinder), an internal QC certificate with per component the mean and U = k·s of those runs (approval *draft* — approve it on Review › Certificates) and records the source runs. From now on link the cylinder's new CDS files to the new QC sample; its history already contains the parent's runs, so you can derive limits at once.

- **Keep an eye on validity.** Tab **Monitoring**, *Expiry monitoring*: what is over the date or approaching it — certificates (*valid until*), released calibration models (newest fit plus the validity months) and QC samples (their certificate) — with the level 🔴 / ⚠️ / 🔔 from the reminder windows you set under **Control settings › Reminders & warning** (months before the date per kind, and the model validity). Both dashboards repeat the urgent ones as alerts.
- **Derived limits.** Tab **Control settings**, *Derived limits*: switch the calibrated-range check, the calibration fit acceptance *GOF ≤ 2 (calibration fit acceptance, ISO 6143:2001 5.2.2)* — one row, *ISO 6143:2001, 5.2.2 (fixed, not a setting) — GOF (largest weighted deviation of a calibration point) ≤ 2 — every fit*; the Birge-ratio threshold is gone —, the expansion factor and the accreditation-scope check on or off and record when they are checked; their values are listed under *Monitoring › Control overview*, with *Max data age* from Settings › System › *Max calibration data age* (not listed when that is 0). With the scope check on, the Review page compares every result with the CMC range of the active accreditation scope of the instrument's laboratory (informative).
- **Accreditation scope at Execute and on the certificate.** The scope check runs on Review by default; tick the moments *execute* and *certificate* under *Derived limits* to see the scope line under the result table on Execute as well and to print an *Accreditation scope* section (with a *reported without accreditation* note for results outside the CMC range) on the analysis certificate.
- **Watch the QC sample.** Tab **Monitoring**, *Control chart*: choose the instrument, then one of the QC samples analysed on it, then the component (the same order as on the other pages; a QC sample without analyses is named below the drop-downs but not selectable) — the results over time against the limits in force (centre, dashed warning band, solid action band; green / orange / red points; hollow = run not approved), the counts below the chart and the *Points* table. *Control overview* lists every limit in force.
- Both pages work like every other master-data table: *Archive status* filter, *Save* / *Create* / *Delete* / *Discard* / *Archive*, change history (the Annex view is read-only; archived CMC lines drop out of it).

Configure Detector Templates per Instrument

Some GC systems report the same component on more than one detector (for example methane on both a TCD and a FID channel). The detector template records, per instrument, which detector's value should be used, so the Include choice is pre-filled at every import and only exceptions need manual correction.

- Open **Administration › Instruments** and select exactly one instrument in the Instruments table (Select checkbox) — the sections below it (Detector templates, Instrument component aliases and the two documentation expanders) follow that selection.
- An instrument can have multiple named templates (e.g. a different configuration for certain kinds of samples). Pick one in the **Template** selector; the buttons **Create template**, **Rename template**, **Make default** and **Delete template** manage them (the name field serves Create and Rename; the default template cannot be deleted and is the one preselected at Analysis import). An instrument without a template gets a default one named *Standaard* automatically at the first visit of this section or the first import; that creation is recorded in the Change history (DetectorTemplate, field *created*). Both import tabs offer a **Detector template** selector next to the instrument, so any template can be chosen per import.

- **Detectors table** — the registry of the instrument's detectors, with the literal CDS label, a short name and the standardized detector type (TCD / FID / other). Detectors register themselves automatically at the first import; the **Suggest from CDS data** button fills the table from analyses already imported and from the newest CDS files on disk. Assign a detector type where the warning flags one as untyped.
- **Preferred detector per component table** — one row per component naming its preferred detector. The **Suggest from CDS data** button adds one pending row per component seen in the instrument's CDS data; components seen on multiple detectors get one row per detector, and components that already have a row are checked against the data — when the data also shows another detector, the extra row is added and the conflict named in the message. Enable *Highlight duplicate component names*, disable **Include** on the wrong rows, then save — a component can keep only one preferred detector.
- Alternatively, resolve the duplicates once in an import grid and click **Save detector choices as instrument template** there; the template then fills itself from your actual choices.
- From then on, imports pre-fill Include on duplicate component rows automatically; rows from non-preferred detectors are stored with include = 0, so no measurement data is lost.

Maintain Components and Aliases

- Open **Administration › Components** (before NexChrom 0.10.0 Component was a page of its own). The tab holds three expanders: **Components** (open), **Global component aliases** and **Component role catalog**, followed by the CSV import / export, its Change history and its Help.
- In the **Components** expander, create the chemical components you need to report on (for example methane, nitrogen, CO2).
- In the **Global component aliases** expander, add every alternative name that might appear in your CDS files for the same component. Aliases are how NexChrom maps variant peak names back to the canonical component; aliases that hold for one instrument only are entered on Administration › Instruments (*Instrument component aliases*).
- One alias per component exists only once — either global or for one instrument. When you save an alias that already exists for the same component elsewhere, *Save* is refused with a message naming where it exists (*Alias X → component already exists as a global component alias / an alias of instrument Y. Remove it there first, then save it here …*); nothing is written until the conflict is resolved.
- In the **Component role catalog** expander, review the catalogue of role types (direct, indirect, fixed) a method can assign to a component; the assignment itself happens per method on the Method page.

Certificate composition

- The *Certificates* table has, next to *Filter on certificate type* (where several types can be selected together, e.g. calibration and qc), the filters **Approval** (Approved / Not approved), **PDF** (With PDF / Without PDF — is a source or frozen PDF on file?) and **Sample** (Linked / Not linked); none of them has an *All* option: nothing selected is no filter, and clicking the selected option again clears it. They combine and new rows always stay visible. *Without PDF* is the quick way to find the external certificates that still need their source document before they can be approved.
- The column **Supplier** says who supplied the certificate: VSL, NPL, NMi … for a certificate bought in — the PDF import fills it with that short name — and your own laboratory for an internal QC certificate and an analysis certificate NexChrom issued (Settings › System › *Laboratory*). Click its header to sort the table on it; edit it in the cell like the other header fields (pending until *Save*). A change withdraws the approval, as for the other header fields; the supplier of a generated certificate cannot be changed. The supplier of the cylinder stays on the sample (*Supplier* on the Sample page).
- For an external certificate the block **Source document** above the composition grid shows the PDF on file (switch **View the PDF on file** on for the inline viewer and the download) or warns that none is on file; choose

a PDF there and it is a pending change (yellow line, counted in the *Pending* caption, named in the logout dialog) that the **Save** of this section stores with its SHA-256; *Discard* drops it. Replacing the document of an approved certificate withdraws the approval.

- Select one certificate to see its components. **value** and **uncertainty** are shown with the GUM display rounding of the Execute certificate preview: the uncertainty to the significant digits of Settings › System › *Significant digits for uncertainties* (default 2), the value to the same decimal place, *Rel. Unc.* with the same significance (for example `0.0001010` and `0.0000040`); the stored values keep their full precision, and editing a cell starts from the stored value. The *Component values* on Review › Certificates use the same rounding. The **Rel. Unc.** column shows the relative expanded uncertainty as a percentage with two significant digits (for example `0.30 %`), the same presentation as *Rel. Unc.* on the Execute preview; it is derived from *value* and *uncertainty* and cannot be edited. A caption *Pending: +n / ~n / -n* under the grid counts the new, edited and deleted component rows that are not yet saved. Below it, **Sum of the component values** adds the values (unrounded, twelve decimals) with their unit — a composition far from 1 mol/mol points at a missing component, a wrong value or a wrong unit; rows with different units are not added, the caption says so instead. Review › Certificates shows the same sum under *Component values*.
- A **generated** certificate (the analysis certificate of a run, issued on Review › Runs; before NexChrom 0.9.0 it was registered on Execute) is shown as what it is: the block **Certificate PDF as issued by NexChrom** with the PDF made when it was issued, stored with its SHA-256 and never changed afterwards — *frozen* (expander *View the certificate PDF*, viewer and *Download the certificate (PDF)*), a check that its SHA-256 still matches the hash recorded at issue, and a line stating whether the ISO 6976:2016 physical properties are on the certificate (last page, with the reference conditions). Its row in the table and its composition are read-only — you cannot edit, save, create, delete or archive them here (*Archive* refuses it with a message); to take such a certificate out of use, withdraw it on Review. The origin *generated* is set only when Review issues the certificate and cannot be chosen by hand.

Maintain Sample Types

- Open the Sample page and select the Sample Types tab.
- Ensure the types you need exist (Sample, Calibration, QC, Reference).

Workflow A — Analysis via sample

Use this workflow when you already know the sample you want to analyze and want to attach CDS files to it afterwards.

Step 1 — Create the Sample

- Open the Sample page and select the Samples tab. Tick a sample: the read-only *No. of linked CDS files* in the **Details of <sample>** table under the grid shows how many CDS files are linked to it (empty = none yet; the details table also holds the supplier and the parent sample, editable by double-click and saved with *Save*) and the *CDS files* filter (*Linked* / *Not linked*; nothing selected = every sample) next to the sample-type filter separates measured from not-yet-measured samples; linking itself happens on the Analysis page.
- Click **Create** and fill in the sample details: name, sample type, description, and any identifiers you track.
- Save the record.

Step 2 — Link CDS Files on the Analysis Page

- Open the Analysis page. The tabs are ordered **Analysis via sample** → **Analysis via CDS file** → **CDS File Browser** (workflow tabs first, audit tab last).
- Select the **Analysis via sample** tab.

- Section 1 **Select instrument** — pick the instrument that produced the CDS files. This selector sits at the top of the tab, separate from the sample and file sections.
- Section 2 **Select sample** — pick the sample. All samples are shown by default (samples are reusable: a calibration gas calibrates several instruments, and any sample can be re-analysed); the *Link status* column shows how many CDS files a sample has on each instrument, the selected instrument first (*5 CDS files on GC-2 · also 10 on GC-1*). **Instrument: Only …** lists only the samples with CDS files on the instrument chosen in section 1 — handy for looking around what an instrument has; it follows another instrument choice, and *Link status* is greyed out while it is on (click it again to list every sample). Use the **sample-type filter** (segmented control with the SampleType labels; several together, nothing selected = every type) to narrow by type, and the right-aligned **link-status control** (*Linked* | *Not linked*; nothing selected = every sample) to narrow by usage — *Not linked* quickly finds a freshly created sample. The file section has the same control with *Linked* | *Available*, where *Available* (default) shows importable files: unlinked ones plus files already linked to the selected sample. Click a selected option again to clear it.
- Section 3 **Select CDS file(s)** — browse to or select one or more CDS files that correspond to injections of this sample. The file table has a **Folder** column next to the file name: the sub-folder under the instrument's data path (*(data folder)*, in grey, for files directly in it — the *Data folder* of the instrument on Administration › Instruments; a line under the table says so when such a file is listed). *File integrity* compares each file with the hash stored at its import — *OK*, *Modified*, *Missing*, *Not imported* for a file no analysis refers to yet, *No hash stored* for an analysis imported without a hash (explained under *Inspecting CDS Files Without Linking*). Its columns keep a fixed width and the table scrolls horizontally.
- **Chromeleon archives (.cmbx)** (for testing): for an instrument on the format *Thermo Chromeleon (.cmbx)* an archive is listed as **one row per injection** — the column **Injection** reads *number · name · injection time*, and each injection has its own link status. **Name match** marks with ✓ the injections whose name equals the selected sample's; they are listed first. They are a proposal: tick the injections that belong to the sample yourself (a blank named *BLK* is never marked — it may still hold gas, so look at it in the CDS File Browser first). Each ticked injection becomes one analysis with the injection time as analysis time; its areas come from NexChrom's own integration of that injection. A component measured on two detectors (e.g. methane on the FID and on the Hayesep TCD) appears twice in the measurement table: untick *Include* on the detector that is not used, or set it in the instrument's detector template, before *Save*. The *Detector* column names the type next to Chromeleon's channel names — *FrontDetector (FID)*, *AuxLeftDetector (TCD)*, *AuxRightDetector (TCD)* — from the instrument's detectors (Administration › Instruments › *Detector templates*), else from the signal unit (pA: FID, mV: TCD); the first import fills an empty type there.
- **These areas are experimental** (not yet validated against Chromeleon): Execute calculates with them, but *Store for review* is refused; Calibration fits with them, but *Save results* is refused. For accredited work import the Chromeleon peak table as a CSV/Excel export.
- Two files with the same name in different sub-folders are two rows (the rows are keyed by the path under the data folder; the Folder column tells them apart).
- Confirm the import. NexChrom parses each file, creates an Analysis record, and stores the extracted peak responses as AnalysisResponse rows linked to the matched components.
- **The messages after Save**: green lists what was saved — *Added (n)* and *Removed (n)*, one file or injection per line; yellow only what needs your attention — files without response rows, peak names *Not recognised, not imported* (once when every file has the same, with where to add the alias) and a *Detector mix*; blue, for information, how names were recognised — by alias, alternative name or detector suffix, each name once.
- **Note on the route**: when the selected files are read otherwise than the analyses behind the instrument's released calibration model (another CDS or integration, e.g. a Compass file for a model built from CSV exports), a note in the yellow box above the buttons says so. Peak areas from another route are not comparable: Execute refuses such a sample with this model, and Calibration refuses to mix routes in one fit, unless an admin overrides it with a reason. Importing the files stays possible; NexChrom records with every

analysis which route it came (the parser that read the file). A whole Chromeleon archive (`.cmbx`) is never imported as one file — it holds no peak results; its injections are linked one by one with NexChrom's experimental integration, for testing (above).

- **Unticking a linked file** removes its analysis at *Save*. A file whose analysis a stored run or calibration rests on **stays linked** — the database itself refuses to delete such an analysis, so a stored run cannot disappear with it: as soon as you untick it, *Will stay linked — a stored result rests on it* in the yellow box above the buttons (together with the route note, when there is one) names the run(s) or model, and *Save* keeps it (*Not unlinked* …). Use *Re-import* to bring in new aliases or CDS names; never untick and re-tick for that.
- **Re-import** (button next to *Save* and *Discard* — hover it for this explanation): reads the ticked CDS files that are already linked to the sample again and replaces their measurements with what the file gives now — with today's aliases and detector template, and with the CDS names. The analysis itself stays (same id, link and time), and your *Include* choices stay where the same component comes back on the same detector. Use it after adding an alias (e.g. *n-Propane → propane*) or changing the detector template, instead of unticking, *Save*, ticking and *Save*. An analysis that a stored run or calibration rests on is only updated when its values stay the same (then only the CDS names are added); when the file now gives other values it is refused, because approved results must not change under their input. Files that are not on disk are skipped; a file changed since its import (🟠 *Modified*) is only re-read after ticking *Also re-read the files that changed since they were imported*. An analysis without a stored file hash (🟣 *No hash stored*) gets the hash of the file as it is read now — the message lists it under *File hash stored* — so *File integrity* checks that file from then on. The result is listed above the buttons; every re-import is in the *Change history*. With pending changes the button is disabled: *Save* or *Discard* first.
- **CDS files that are not on disk**: when none of the selected files is on disk, the table cannot re-read them. Without stored measurements it says there is nothing to show. With stored measurements the table shows them, with a note that peaks the files did not import cannot be listed.
- **Saved analyses show their CDS names**: the import stores with each measurement the name the CDS gave the peak and how NexChrom recognised it, so after *Save* the table still shows *CDS name* (e.g. *n-Propane*) and *Match* (e.g. *instrument alias*). Analyses imported before NexChrom 0.11.0 show an empty *CDS name* and *Match matched*; *Re-import* fills them. The *Unit* column shows the unit of each row.
- **Include choices are pending until Save.** Ticking or unticking *Include* in *Measurements from CDS file contents* changes nothing in the database yet: the caption *Pending: ~n include changes* under the table counts them, **Discard** drops them together with the unsaved file selection (*All pending file selection and Include changes discarded.*), and **Save** writes them to the saved measurements and applies them to the files imported in that same Save (on top of the detector template). The result message of Save or Discard appears directly above the Save button.
- **A component included twice blocks Save** — for example methane on two detectors, both with *Include* ticked: a red box directly above the buttons names the components once (*Save is blocked: the following components have multiple entries with Include enabled:* …). Untick *Include* on the entry you don't use in the measurement table; *Save* is enabled again.
- **Execute this sample**: once the sample has saved CDS files on the chosen instrument, the line *The sample has saved CDS files on this instrument — next step:* under the buttons ends with the link *Execute this sample (n CDS file(s) on this instrument)*, which opens Execute with this sample and instrument already chosen (Step 5).
- A legend under the table explains the **Match** column: *name*, *trivial_name*, *formula*, *iso_code* (the CDS name equals that field of the component), *alias* (Administration › Components), *instrument alias* (this instrument), *…, suffix X stripped* (recognised after removing the detector suffix), *matched* (imported earlier without a recorded source), *variant of <component>* (a detector variant that is deliberately not imported) and *unmatched*.
- If a peak name does not match a known component or alias, the table *Measurements from CDS file contents* names it above the table. Add the name as an alias — on Administration › Components (*Global component aliases*) when it holds for every instrument, or on Administration › Instruments (*Instrument component aliases*)

when it is specific to this instrument's method; an instrument alias wins over a global one — then re-select the file. A detector suffix such as *Methane (FID)* needs no alias. If the alias already exists for the same component — globally or on another instrument — Save is refused with a message naming where it exists (*Remove it there first*): one alias per component exists only once, global or for one instrument.

Step 3 — Choose or Define a Method

- Open the Method page. A method is instrument-independent — it meets an instrument only when a calibration model is built with it. The *In use on* column shows on which instruments a method is in use (number of active calibration models); the *Usage* filter (*In use* / *Not in use*; nothing selected = every method) at the top left separates methods in use from methods still waiting for a calibration.
- Select an existing method that fits your analysis, or create a new one.
- For each component the method reports, assign a role: direct (calibrated), indirect (calculated via RRF), or fixed. For a fixed component enter the concentration in the **Fixed value** column (mol/mol) and, optionally, its expanded uncertainty in **Fixed U**; a fixed role cannot be saved without a value. Execute reports the value as entered (even when the sample shows no peak for it) and includes it in the normalised composition.
- For indirect components, define the RRF rule: reference component and the relative response factor.
- Editing a method that calibration models or calculation runs already reference asks for an explicit confirmation first: roles and RRF rules are read live at (re)calculation time, so in-place changes alter what recalculations of existing runs produce. For semantic changes, use **Duplicate method as new version** (copies the method with all roles and RRF rules under a date-and-time-stamped name) and edit the copy. All role and RRF saves appear in the Change history.

Step 4 — Ensure a Calibration Model Is Released

- Open the Calibration page.
- Confirm that the instrument you used has a current calibration model covering the components in your method **and that its *Release* column reads *released*** — Execute only offers current, released models for new runs of an ordinary or calibration sample; a QC or reference sample is also offered a current model that is not released — draft, rejected or withdrawn — as the QC check of that model; an archived model is never offered.
- If there is none, build one using Workflow C (Calibration) below and have it released on the Review page (Workflow C, step 4).

Step 5 — Execute

- Open the Execute page — or click **Execute this sample** on *Analysis via sample* (Step 2): Execute then opens with the sample and the instrument already chosen. The page is one straight line of four parts, top to bottom, each an open expander: **Select Instrument & Sample**, **Checks**, **Result** and **Store for review** (the earlier page with run types and *Certificate Follow-up* is gone). When the sample already has stored runs, a notice above *Select Instrument & Sample* names the newest one (moment, review status, certificate) and the expander *Runs of <sample>* lists them all. A new run of the same CDS files does not replace an earlier one yet: an earlier pending run stays in the Review queue.
- **Select Instrument & Sample**
  - **Instrument** first, as on Analysis, Calibration and Quality control: a run uses the CDS files of one instrument only. The drop-down lists the active instruments with CDS files as *name (n to calculate)* and shows manufacturer, model, serial number and location under it.
  - The **sample table** lists the active samples with CDS files on that instrument, one row per sample: *Sample*, *Type*, *Cylinder no.*, *CDS files* (on this instrument), *Newest analysis*, *Last run* and its *Review status* (*no run yet* when there is none) — *Last run* is the newest stored run of the sample **on this instrument**, as *To calculate* is per instrument. Above it the **Sample type** filter (nothing selected when you arrive = every type, so the

calibration and QC gases still to calculate are in view too; click a type to narrow the table to it) and **Show**: *To calculate* (default — samples whose newest CDS file on this instrument is not yet in a pending or approved run: no run, a rejected run, or newer CDS files) or *All* (also the samples that are done). The number on the type buttons follows *Show*; a chosen instrument or sample type stays chosen when the count in its label changes (after *Store for review*, or switching *Show*) and when you come back from another page. Tick one sample; the caption under the table names its CDS files (number, analysis span, cylinder). **All CDS files of the sample on this instrument are averaged** (expander *CDS files used*); files of the sample on other instruments are named but not used.

  - **Method** — only methods with a usable calibration model on this instrument are offered, and the **calibration model follows from the method** (*Calibration model: <name> · released · fitted <date> · follows from the method*). A model is usable when it is current (not archived) and **released**; a QC or reference sample is also offered every current model that is not released — **draft**, **rejected** or **withdrawn** — labelled *<status> — only as the QC check of this model* (Workflow C, Step 4), so a withdrawn or rejected model can be checked and released again on Review. With two usable models for the method a *Calibration model* drop-down appears, the released one preselected (the release on Review keeps one released model per instrument and method — Workflow C, Step 4 —, so two released models of one method occur only for models released before that rule). An archived model is never offered, and an ordinary sample is never offered a model that is not released; when no model is usable the page says what is missing and where to solve it (no model on the instrument, archived models only, or the model is not released yet). The expander *Method details* shows the component roles and RRF rules.

- **Checks**
  - The prerequisites of the calculation: CDS files with responses, a released method (an **admin** may override an unreleased method: tick *Override the unreleased method (admin only)* and give the reason; it is stored with the run and shown as an alert on both dashboards), the model's instrument matching the CDS files, the RRF rules of indirect components and the values of fixed components (a fixed component without a value in the method is a **warning** in Checks; *Calculate* is not disabled by it, but then fails as a whole with *Calculation failed: Fixed component '…' has no fixed value in the method …* — enter the value first). **Method and model must match**: every direct component of the method needs a fit in the model; a mismatch is an error, corrected at the base (recalibrate with a gas that contains the component, or have a method version released in which it is indirect). The errors stand together in **one red box** and disable *Calculate*; the warnings stand in **one yellow box** and do not; the passed checks are collapsed under *n check(s) passed* (opened, a list in two columns). The block ends with *Ready to calculate* or *Not ready*.
  - The **validating QC run** (not for a QC or reference sample) is **one line**: NexChrom proposes the **nearest approved** QC run of the same instrument that **passed (or warned)**, analysed within the **QC window of the sample type** around the sample's analyses (Settings › System › Validation (QC): *Sample* default ± 7 days, *Calibration* default ± 1 day). A green line confirms it; an orange line says it lies outside the window, or that there is none. **Change** opens the choice: another QC run within the window, *(none)*, or — after ticking *Also offer QC runs outside the window* — one of the 5 nearest outside it. A QC run that is not approved, has no QC verdict, was rejected or failed cannot be chosen; while no QC run within the window is chosen, those within the window are listed one line each with the reason — e.g. *QC run 248 · QC-NG-01 · 2026-10-15 14:15 — no QC verdict yet: run its QC check on Review first* — and **Open on Review** at the end of the line where approving it or running its QC check is the next step. A missing or distant QC run is a notice, never a block. The choice is stored with the run and shown on the Review page and on the certificate. When the choice is lost (for example after a visit to another page) the proposal is chosen again. On a database without the QC migrations the line is replaced by the caption *QC coverage is not installed in this database (run the QC migrations): no validating QC run.*
  - An **experimental integration** (in the yellow box): the sample's CDS files include injections of a Chromeleon archive whose areas come from NexChrom's own, not yet validated integration. *Calculate* works; *Store for*

review is refused, with the reason in the red box above its button.

- **Peak areas from another route** (in the red box) when the sample's analyses were read otherwise than the analyses the calibration model was fitted from (another CDS or integration, e.g. Compass files for a model built from CSV exports), or carry another area unit for a component: a model is valid only for peak areas made the same way, so *Calculate* is disabled. The message has the headline *Peak areas from another route than calibration model <model>*, then the two routes on lines of their own — *Sample: …* and *Model: …* (for a unit difference *Area unit of <component>: … (sample) · … (model)*) — then what to do: link the sample's files of the model's route, or calibrate with this route; the last line says that only an **admin** can override it: tick *Override the route difference (admin only)* below and give the reason. The error then becomes a warning in the yellow box (… — *admin override*, the same lines and *Reason: …*), and the reason is stored with the run and shown on Review › Runs (*Route override*).
- A model that was saved from more than one route under an admin override is named in the yellow box — *Calibration model <model> was saved from more than one route or area unit under an admin override* — with the reason on the next line.
- A **QC or reference sample** needs no validating QC run: its result is compared with its own QC limits in *Result*; with a draft model the run is the QC check of that model.

- **Result**
  - **Calculate** computes the composition and its expanded uncertainties; **nothing is stored** (badge *Preview — not stored*). The coverage factor k of the calculation (Settings › System › Uncertainty) is kept with the result. The warnings of the calculation (for example the EXTRAPOLATION warning below) stand together in one yellow box under the badge.
  - The per-component table (numeric ISO-code sorting) with the sum of unnormalized mol fractions and the k below it. *Uncertainty* and *Mol fraction* are rounded as on the certificate: the uncertainty to the significant digits of Settings › System › *Significant digits for uncertainties* (default 2), the mol fraction to the same decimal place (display only; the stored values keep their precision). Uncertainties follow GUM propagation — calibration covariance for direct components, RRF relative standard uncertainties per detector type (ISO 6974-2 Annex B) for indirect components — and are expanded with that k. The **Cal. range (mol/mol)** column shows the calibrated range of the fit — the lowest and highest certified concentration of the calibration gases used — and **Range status** tells whether the calculated result lies *within*, *below* or *above* it (indirect components show *ref-below* / *ref-above* when their reference component was extrapolated); *within* is green, the flagged values are red and an EXTRAPOLATION warning names them. The result is still calculated and stored with the flag — it is a review signal, not a block, and the Review page shows the range next to each result so it can be read against your accreditation scope. An empty range means the model was saved before the range was recorded: re-fit and re-save it on the Calibration page.
  - With the scope check on at the moment *execute* (see *Accreditation scope at Execute and on the certificate*), the accreditation-scope line sits under the table.
  - Expander **Physical properties (ISO 6976)** — for every sample type, also calibration and QC gases: choose the *Combustion temperature*, *Metering temperature* and *Reference pressure* (defaults from Settings › System › ISO 6976 reference conditions, normally 25 °C / 0 °C / 101.325 kPa) and read the molar mass, compression factor, calorific values (molar, mass and volume basis), density, relative density and Wobbe indices — ideal and real gas — with U (the k of the calculation) and relative U, rounded as ISO 6976 prescribes. The notes under the table (one list) state the conditions and the basis; a component without a propagated uncertainty is named in a yellow box under them (*No propagated uncertainty for: … (contribution counted as 0).*). For an ordinary sample the toggle **Include on the certificate** adds a last page *Physical properties - ISO 6976:2016* to the analysis certificate (page 1 points to it); the choice and the reference conditions are stored with the run at *Store for review* and cannot be changed afterwards. Leave it off for a certificate without that page.

When a component has no ISO Code or its name is not the ISO 6976:2016 name of that number, the section names it instead of calculating — correct it on Administration › Components.

- For a **QC sample** the **QC comparison** against the limits in force (see *Check a QC run*), marked *preview, not stored* until the run is stored.
- For an **ordinary sample** the expander **Certificate preview** (open): the analysis certificate of this result as a formatted PDF, as it will be issued — the application and version in the header (*Generated with NexChrom …*), sample details, instrument, method, calibration provenance (model, the regression methods of its fits as their labels on the *Regression* line — EIV, OLS, WOLS or GLM, and *through-origin fit* or *2-point line* for a fall-back fit on fewer than three points, e.g. *EIV, through-origin fit* —, calibration gases, data age), the GUM-rounded results table with normalized values (significant digits from Settings › System › *Significant digits for uncertainties*, default 2), an uncertainty statement (expansion factor, ISO 6974-2 RRF basis), the validating QC run and the run provenance (*Executed by*, *Approved by*), with the status banner *PREVIEW*. Inline, or as a download. A QC, reference or calibration gas gets no analysis certificate.

- **Store for review**
  - Optional **Run notes** (stored with the run, before the automatic note naming the sample, model, method and CDS files), then **Store for review** — for a QC or reference sample calculated with a model that is not released (draft, rejected or withdrawn) the button reads **Store as QC check of <model>**. It needs the edit right on Execute, clear checks and a database with the run columns of `migrate_run_certificate_v2.10.0.py`; the edit right and the checks are verified again when the click arrives.
  - **The result is calculated again at Store**, with the inputs as they are at that moment (calibration fits, included peaks, method). When it differs from the result shown, nothing is stored and the page says *Storing failed: the inputs changed since Calculate (calibration fits, included peaks or the method): press Calculate again and check the result before storing it* — so a stored run is always the result you saw.
  - It stores a CalculationRun (run type *manual*) with its AnalysisResult rows and the validating QC run, and **fixes what its certificate is built from**: the **executor** (your user name), the **coverage factor k** of the calculation and, for an ordinary sample, the **certificate options** — the significant digits, whether the ISO 6976 page is included and at which reference conditions, and the CDS files used. A later change in Settings does not change the certificate of a stored run. For a QC sample the QC check runs at once and is stored with the run; when it cannot run, the run is stored anyway and the page warns *The QC check of the run could not run: … Run it on Review › Runs.*
  - The page then shows *Stored run n · pending review*, the green confirmation *Run n stored. It now waits on Review › Runs for a second person …* with the link **Open run n on Review** at the end of its row, the values stored with the run (read-only; the ISO 6976 choice also under *Physical properties*) and the certificate preview of the stored run. Nothing else is done on Execute: the run waits on Review for a second person.
  - What keeps *Store for review* disabled stands in one red box above the button: on a database without the run columns of `migrate_run_certificate_v2.10.0.py` *This database cannot store the executor, k and certificate options with the run yet: run data_local/database/migrate_run_certificate_v2.10.0.py --apply first.* (with or without the ISO 6976 option: have the migration applied), and for areas of the experimental integration *Not stored: …*.

Step 6 — Review the Run and Issue the Certificate (second person)

- A **manager or admin** opens the Review page, tab **Runs** (filter *Pending*) — or clicks the run in the Overview's *Review queue*, opens its card on the Worklist (*Review now*), or clicks *Open run n on Review* on Execute; each opens Review on the Runs tab with that run selected. The **Reviewer** card under the introduction shows who is reviewing, that every decision is recorded under that name with the moment and the notes, and the open

work (runs pending review, approved runs without certificate, certificates, models, methods); an operator or auditor sees *view only* there with the reason, and the page read-only.

- Above the runs table sit the **Instrument** selection (left; *All instruments* by default, lists the instruments with stored runs, each with its number of pending runs, e.g. *GC-1 (3 pending)*), the **Sample type** filter (next to it; several types together, nothing selected = every type) and the **Review status** filter (right): *Pending*, *Approved*, *Rejected* and *No certificate* (approved runs of an ordinary sample without an active certificate). The columns *Run* (the run number) and *Analysed* tell several runs of one sample apart. On a database without `migrate_run_certificate_v2.10.0.py` the tab warns *This database has no record of the executor, k and certificate options of a run yet: four eyes cannot be checked and certificates use the current settings. Run … migrate_run_certificate_v2.10.0.py --apply.*
- Tick the run: the summary is a **detail card** — grouped label / value rows (Sample, Calculation, Review, …) with the statuses as coloured chips (green = approved / released, amber = pending / draft, red = rejected / withdrawn / fail); the *Review* group names *Executed by* (the user who stored the run) and the certificate (*not issued yet* until it is) — followed by the *Calculated results*: *Mol fraction* and *Uncertainty* rounded as on the certificate (the uncertainty to the significant digits of Settings › System › *Significant digits for uncertainties*, the mol fraction to the same decimal place), *Rel. U* (the uncertainty relative to the mol fraction, in %), *Unit*, *Cal. min* / *Cal. max* (the calibrated range, 5 decimals), *Position in range (%)* (where the result lies in the calibrated range: 0 % = Cal. min, 100 % = Cal. max; below 0 or above 100 is outside it) and *Range* (*within* in green, *below* / *above* in red); hover a column header for its meaning; the first column stays pinned while scrolling and the last one fills the width. When a result lies outside the calibrated range the EXTRAPOLATION warning follows, then the line on the validating QC run.
- The card carries an **Earlier decisions** group whenever the record was decided before (moment, user, verdict and note, rebuilt from the Change history) and a chip *previously rejected* / *previously withdrawn* when one of those verdicts was negative — a rejection stays traceable after a later approval.
- **The certificate itself.** For a run of an ordinary sample — every sample type except *qc*, *ref* and *calibration*, so also a sample without a type or of a custom type — the block **Certificate** shows the document you sign, as on paper: before issue the **preview** built from what was stored with the run (its k, significant digits and ISO 6976 choice, named in the caption) — exactly what will be issued, then with the status APPROVED and your name and the moment under *Approved by*; after issue the **frozen PDF** of the certificate (never rebuilt: a missing file is shown as an error). A run stored before NexChrom 0.9.0 has no stored certificate options: its certificate is issued with its k (back-filled by the migration with the Settings k at that moment), the significant digits at the moment of issue and without an ISO 6976 page (the caption says so). The header of the certificate names the application and version that made it: *Issued with NexChrom v0.14.0 (Qualimetric)* on the issued certificate (*Generated with …* on the preview). When the run was calculated with another version of NexChrom than the one that issues the certificate, the header says so as well (*Run calculated with NexChrom …*), and the Run provenance always lists *Calculated with* (*not recorded* for a run stored before NexChrom 0.9.0). The version is fixed when the run is stored and when the certificate is issued, and never changes afterwards.
- Enter *Review notes* (stored with the decision; there is no separate Save — the help of the field says so) and decide. For a run of an **ordinary sample**:
  - **Approve and issue certificate** — approves the run and issues its analysis certificate in one action: the PDF is frozen to data_local/certificates/generated/ with its SHA-256 and recorded as a Certificate (origin *generated*, approval *approved*, basis *approved_run*, approved and issued by you at the moment of the approval, the validating QC run copied) with the normalized composition as CertificateComponent rows. The message names the certificate, with the link **Open <no> on the Certificates page** at the end of its row (e.g. *Open NC-RUN-00142 on the Certificates page*).
  - **Approve without certificate** — approves the run only. It is then listed under the filter *No certificate*, where **Issue certificate** issues it later, also in another session; Overview's Review queue links *n approved run(s) without certificate* to that filter, and Worklist shows them as cards (*Certificate to issue*). The certificate then

names under *Approved by* the reviewer who approved the run, with the moment of that approval; the one who issues it is recorded on the Certificate (created by / approved by, issued at = the moment of the issue). The notes box still holds the approval notes of the run: only a text that differs from them is added to the certificate notes as *Issue note:* ….
    - **Reject** — not fit for a certificate; the sample is offered again on Execute (*To calculate*). **Reopen** takes a verdict back.

- For a run of a **QC, reference or calibration gas** the buttons are **Approve**, **Reject** and **Reopen**: such a gas gets no analysis certificate.
- **What blocks a decision** stands in one red box between the notes and the buttons, as a list under *Cannot approve / release:* and a list under *Cannot issue the certificate:* (the same box on every tab of Review; an admin's way out, such as the release without a QC check, in italics under it). **Four eyes**: the user who stored the run (*Executed by*) cannot approve it or issue its certificate — unless an admin, who may approve their own work (an info line says so; the decision is recorded under the admin's name). The rule is skipped while the database holds a single active user, and for runs stored before NexChrom 0.9.0 (no executor recorded). A failed QC check, a superseded run or a run without results cannot be approved. Issuing is blocked — *Approve without certificate* stays possible — when the run already has an active certificate, the ISO 6976 choice stored with it cannot be calculated, the run's calibration model is no longer released, the model was **fitted again after the run** (the run's results come from the earlier fit: calculate the sample again on Execute), or a certificate file of the same name already exists (a certificate file is never overwritten: move the file first). **Reject** and **Reopen** are refused while an approved certificate hangs on the run: withdraw the certificate first (Review › Certificates).
- The decision is stored with your username and the moment; *Review history of this record* shows every earlier verdict.

Step 7 — The Issued Certificate

- The Certificates page lists the issued certificate with its approval status; the approval columns are read-only there, and a generated certificate is frozen: *Save*, *Delete* and *Archive* refuse it.
- **Withdraw** on Review, tab **Certificates** (tick the certificate), takes a certificate out of use: status *withdrawn*, the PDF stays. The run then has no active certificate and is listed under *No certificate* again; **Issue certificate** issues a new certificate under the run number with a reissue suffix — for example *NC-RUN-00142-R1*, then *-R2*, … — and its own PDF file. The withdrawn certificate and its file are kept: a certificate file is never overwritten.
- **Legacy drafts.** A generated certificate registered on Execute before NexChrom 0.9.0 may still have the approval *draft*. It is approved on Review, tab **Certificates** (filter *Pending*), or from the Overview's *Review queue* or its card on the Worklist: the detail card (header with status chips, the run and its review status, *Earlier decisions* with the *previously …* chip where applicable), the component values — with a **Rel. Unc.** column, the relative expanded uncertainty as a percentage with two significant digits, as on the Certificates page — and the switch **Certificate PDF as issued by NexChrom** (the frozen PDF: viewer and *Download the certificate (PDF)*; its help explains *frozen*). The path of that PDF is the row *Certificate PDF* under the card. **Approve** releases it (approval status *approved*, reviewer and moment recorded, approval basis *approved_run*); it needs the approved run behind it. Imported, hand-entered and internal QC certificates are approved on the same tab with the four-eyes check against the source document (Workflow D, Step 4).

Workflow B — Analysis via CDS file

Use this workflow when you want to start from CDS files on disk and let NexChrom create the Sample for you during import.

Step 1 — Import CDS Files

- Open the Analysis page and select the **Analysis via CDS file** tab (the middle tab in the new ordering).
- Browse to the CDS files you want to import.
- Choose the matching instrument and confirm the import.
- NexChrom creates a new Sample from the file metadata, registers each file as an Analysis, and extracts AnalysisResponse rows.

Step 2 onward

Continue with Workflow A from Step 3 (Choose or Define a Method). On Execute (Step 5) the imported sample is listed on its instrument under *Show: To calculate*; its certificate is issued on Review together with the approval of the run (Step 6).

Inspecting CDS Files Without Linking (CDS File Browser)

Use the **CDS File Browser** tab on the Analysis page (the last of the three tabs) when you want to look at a CDS file on disk without linking it to a sample. The tab is structured as:

- **1. Instrument & folder selection** — pick the instrument whose data folder you want to browse.
- **2. Select CDS file** — single-select grid of files discovered in the instrument's data folder, with the columns *Filename*, **Folder**, *File integrity*, *Modified* and *KB*. *Folder* is the sub-folder under the instrument's data path, as on the import tabs — *(data folder)* in grey for a file directly in it, explained in a line under the table — so the same file name in two sub-folders is told apart.
- **3. File metadata** — shows the raw metadata fields parsed from the selected file. For Agilent files this now includes *Dilution*, *Multiplier*, *Inj Volume* and *Actual Inj Volume* taken from `Report00.CSV`, plus a warning banner when ChemStation reported warnings on the run.
- **Chromatograms** — for a Thermo Chromeleon archive, an Agilent ChemStation `.D` directory and a Scion CompassCDS `.DATA` file: the signals of the file with the peaks, see below.
- **4. Peak results per detector** — shows raw per-detector peak data as it appears in the file (no totals, averages or derived values). Each detector or signal has a **Peaks** filter: *Named* (default) lists the identified peaks, *All (n unknown)* adds the unidentified ones; for a detector without unidentified peaks the filter is disabled and its help text says so.

The values shown in sections 3 and 4 are straight from the file via the shared parser — nothing is recalculated or aggregated, so the browser is safe to use as a quick audit view.

File integrity compares the file on disk with the hash stored when it was imported: *OK* (unchanged since the import), *Modified* (changed since the import), *Missing* (no longer on disk), *Not imported* (no analysis refers to the file yet) and *No hash stored* (imported without a hash, for example an analysis recorded before the hash was kept; it cannot be checked until a *Re-import* on *Analysis via sample* stores the hash). The import tabs use the same labels. For an Agilent `.D` directory the hash covers the names and sizes of the files in it, not their contents.

Chromatograms draws each signal of the file in its own frame, on a shared time axis; zoom with the mouse, and the hover gives the time and the real signal value. **Log range [decades]** (0 to 10) sets the y-axis: 0 is linear, above 0 a logarithmic axis that shows small peaks next to large ones. A log axis draws no values of zero or below (a TCD baseline lies around 0 mV), so each detector signal is lifted until its lowest point lies that many decades below its highest: the panel title gives the shift (e.g. *log axis: signal + 0.1422 mV*), the hover the real value. More decades show smaller peaks and more baseline noise. The field starts at the default of *Settings › System › Chromatograms*

(5 decades unless changed there); a change here holds for this view only and is not stored. *Log range* changes the plot only, never the peak results. A signal that is no detector, such as the ambient pressure, stays linear.

Agilent ChemStation (.D):

- *Chromatograms* draws the signal file of every signal (`*.ch`): the detectors first, in the order of the report, then other signals such as the ambient pressure. The peaks of the CDS report are marked at their retention time — triangles, green with the compound name, grey for an unnamed peak; the hover gives the retention time, area, area %, peak type and width. ChemStation integrated these files: there are no integration settings, and the peak results below are the CDS's own.
- In red, the integration of ChemStation itself, read from the sequence result file (`.acaml`) in the sequence folder next to the `.D` directories: per peak its baseline from begin to end with a marker at both ends, and the expected retention time of an identified compound as a dotted line. The hover names *Integration: Agilent ChemStation* with the baseline code (e.g. *BB*, *BV*, *VV*). The method's retention-time windows are not in the file, so no window is shaded. A baseline is drawn only for a peak of the report that the result file holds with the same retention time and area, so every baseline drawn belongs to the area NexChrom imports.
- A blue box *Integration of the CDS* says what is not drawn: the number of peaks of the report without a baseline, with their peak types (integrated by hand in the report after the sequence was processed: *MF*, *FM* or *MM*), and the number of peaks the result file holds that the report does not list (mostly small unnamed peaks). Without a result file the box says so, and only the triangles are drawn.
- A `.D` without signal files, or a signal file of another format version than NexChrom reads (the `.ch` files of version 179), is named instead of drawn.

Scion CompassCDS (.DATA):

- *Chromatograms* draws the signal of every channel of the file, with the sampling interval and number of points the file states: the channels with peak results first, in the order of the tables below, then a channel without results. The peaks of the CDS's peak results are marked at their retention time (green with the compound name, grey for an unnamed peak; the hover gives the retention time, height, area and area %); a compound of the method that the CDS did not find (height and area 0) is not marked. CompassCDS integrated these files: there are no integration settings, and the peak results are the CDS's own.
- In red, short vertical ticks: the begin and end of each peak as CompassCDS integrated it, from the peak table of its channel in the file — a valley between two peaks once for both — with the expected retention time of an identified compound as a dotted line. The file stores no baseline levels at begin and end and does not say what its identification window means, so no baseline is drawn and no window is shaded. The hover of a tick names *Integration: Scion CompassCDS* with the boundary code (*S* on the baseline, *V* a valley), the fused group and the CDS's baseline at the retention time.
- A blue box *Integration of the CDS* counts the marked peaks without begin and end (the peak table of the channel they are drawn on does not hold them) and counts the peaks of the tables that no marked peak pairs with (not drawn). A channel whose signal does not add up is named instead of drawn.
- **Known limitation:** in some Scion files NexChrom gives the peaks of one detector the name of another — two detector names swapped, or a few peaks filed under another detector than the channel that measured them. Imported, their areas are stored under the wrong detector name, and the detector template may then include the wrong measurement. The CDS File Browser shows it: on such a channel the triangles stand beside the peak tops instead of on them, and the blue box counts peaks without begin and end. Look at a new series of Scion files in the browser before you import it, and leave a file whose triangles do not stand on the tops out of the import. When such a file is imported together with consistent files, the *Detector mix* warning after *Save* names the component.

File format: under the instrument the browser preselects the instrument's own CDS file format (*(instrument default)*). Choose another format to look at files of another kind in the same folder, e.g. *Thermo Chromeleon*

(.cmbx) for an InterScience / Thermo instrument that imports through CSV exports. The choice is for viewing only; the import tabs keep the instrument's format.

Thermo Chromeleon archives (.cmbx): an archive holds a whole sequence — every injection with its raw signals and the processing method — but no peak results.

- *File metadata* shows the sequence (name, instrument, Chromeleon version, status, processing and instrument method, export date) and one row per injection (name, position, type, status, injection time, volume, signals and the lab's own fields such as *Customer* or *SampleLocation*). **Injection to inspect** chooses the injection for the sections below; the first injection that is not a blank is preselected. The ambient pressure of the injection (mean over 2.8–3.0 min, as the lab's Chromeleon layout uses it) is shown, not applied. An archive without a sequence (e.g. an exported report definition) is reported as such.
- *Chromatograms* draws every signal of the injection, each in its own frame: per component of the processing method the expected retention time (dotted) and window (shaded), and NexChrom's integration — baselines in red, apexes as triangles (green with the component, grey unidentified; × for a peak in a period in which the method inhibits integration, the periods are listed under the peak table). Three settings tune the integration (*Height threshold*, *Smoothing*, *Slope threshold*); zoom with the mouse. The fourth, in the same row, is **Log range [decades]** (above): it changes the plot only — the axis values are shifted, the integration and the peak results stay the same — and the ambient pressure stays linear.
- *Peak results per detector* lists per detector every component of the processing method with the NexChrom component it maps to (*Match* as on the Analysis tabs, e.g. *Methane (Hayesep)* → methane by stripping the suffix), the expected retention time, window and peak match, and the peak found: apex, start, end, height and area (signal unit · min), or a note why none was found. Under the table, as short lists: a line over the full width with how many components were found and peaks integrated and the signal noise, and below it in two columns the *Other peaks* (time, area, *inhibited*) on the left and the events of the method on the right (a single list stands on the left), the events as periods — *Events applied* (e.g. *InhibitIntegration: 0.000–2.000 min, from 7.921 min*) and *Not applied yet* (e.g. *LockBaseline: 21.031–21.546 min (2)*); an *off* without an *on* is named, so a doubtful event in the method shows.
- **The integration is experimental and not validated** against Chromeleon's peak table (yellow banner); nothing is stored here. Use it to look at an archive, not for results. Linking injections to a sample is done on *Analysis via sample* (above).

Workflow C — Calibration

Use this workflow to build a new calibration model for an instrument.

Step 1 — Register the Calibration Analyses

- Ensure the calibration samples exist on the Sample page, classified with a Calibration sample type.
- Import the CDS files for your calibration injections on the Analysis page and link them to the calibration samples.

Step 2 — Build the Calibration Model

- Open the Calibration page. The **Instrument** drop-down at the top left of *Calibration models* scopes the whole page: it lists instruments only, as *name (n models)*, and shows manufacturer, type, serial number and location in small type under it; *Archive status* on the right of the same row switches the table to archived models. The first column of the table after the selection tick is the model **Id** (read-only,).
- Open the Calibration page.
- Select the instrument.

- Select the calibration analyses to use as input. The table shows one row per sample **and analysis session** (CDS files grouped by analysis date); by default only the most recent session of each sample is selected. Select an older session to deliberately reuse it, or several sessions of one sample to pool them (a warning names the pooled sample). For a model that already holds a fit, a message under the table compares your selection with the input set of the saved fit: *Selection matches the input of the saved fit*, or *Selection differs from the saved fit of this model* with *Added: / Removed:* per sample @ session — a new regression would then run on other input than the stored one.
- *Calibration input data* lists every calibration point: every component with a certified value in the selected gases is fitted alike, whatever its role in the method (broad fitting). The **Detector** column shows which detector(s) the included replicates come from; a point whose replicates mix detectors is red and a warning names it — fix the Include choices of that file on the Analysis page before trusting the fit.
- **Select a point** in that table to see the replicate peak areas behind its mean: one row per CDS file with its deviation from the mean, z-score and a Grubbs outlier flag, the statistics (n, mean, SD, RSD, u(mean)) and a control chart with the mean and the ±1 s, ±2 s and ±3 s limits of the included replicates. Untick **Include** to leave a replicate out of this model's regression input, or press **Exclude n flagged outlier(s)**; **Restore all** undoes it for this point. These changes are **pending in the session** — the caption *Pending: +n excluded / -n restored* counts them — and are written only when you click **Save** next to them (stored with the calibration model, logged in the Change history, recorded in the fit's input set and copied when you save as a new model); **Discard** reverts them. The input table above and the regression keep using the *saved* exclusions, so **Execute regression is disabled while exclusions are pending** — save or discard them first. The analysis itself and other models are untouched. The Grubbs test is advisory — with few replicates it only catches gross outliers.
- Select the regression method: **OLS**, **WOLS**, **GLM** or **EIV** (pre-selected). Every option has a one-line caption saying what it does and where its GOF comes from, and the `?` next to the choice explains them. **EIV** (errors-in-variables, ISO 6143; uses uncertainties on both peak area and certified value) is the recommended method for traceable calibration. Whatever the method, **every fit is tested the same way**: on the **GOF** (goodness of fit; the standard writes it as Γ), the largest weighted deviation of a calibration point from its adjusted point, $|x - \hat{x}|/u(x)$ or $|y - \hat{y}|/u(y)$ — the fit passes when GOF ≤ 2 (ISO 6143:2001, 5.2.2), a limit fixed by the standard, not a setting. Under EIV both deviations count; OLS (ordinary least squares), WOLS (weighted by $1/u^2$(certified)) and GLM (robust Huber reweighting) take the peak area as exact, so their GOF is the largest $|y - \hat{y}|/u(y)$ with the certificate uncertainties only — stricter than the EIV GOF on the same data. OLS is kept as an unweighted reference: on data spanning several decades it ignores the low-concentration points and its GOF goes far above 2 — which is the point of the comparison, not a defect.
- Every component with three or more calibration points gets, next to the orders with intercept, the **1st-order through-origin variant *1 (origin)***: under EIV the slope-only line, under the other methods the line without the constant term. It is a tested fit like the others (GOF, ν = n − 1), can be chosen in *Used* (which offers e.g. 1, 1 (origin), 2, 3), is saved with fit type *origin* and the method of the run, and is never proposed by *Optimal* — the intercept test tells you whether it is worth a look (next step). Orders 2 and 3 through the origin are not offered. At exactly two points, behind its own checkbox, the exact line with intercept (*1*) and the through-origin fit (*1 (origin)*) are both computed and both selectable in *Used*; *Used* starts at the exact line, and the b_0-test only labels *Optimal*: the intercept of the exact line is significant when $|b_0|$ exceeds $k \cdot u(b_0)$, with k the coverage factor of Settings › System › Uncertainty (default 2) — *Optimal* reads *1*, or *1 ($b_0 \approx 0$)* when the intercept is not significant.
- Click Execute. NexChrom computes all feasible polynomial orders (1..min(3, n-2)) in one pass and shows the results grid with one row per component. The message above the grid sums the run up: *Computed all orders for n component(s)*, the components skipped for fewer than 3 points, and — when a component has no order that passes its criterion — *No order passes its criterion for …* (the box is then yellow instead of green); for 2-point components it reports the b_0-test with its criterion and k spelled out (*b_0-test on 2-point components ($|b_0|$*

> 2·u(b_0) counts as significant; k = 2, the coverage factor of Settings › Uncertainty): intercept not significant for … — Optimal reads 1 (b_0≈0); the through-origin variant 1 (origin) can be chosen in Used; significant intercept for … — line with intercept.). A component whose regression failed is not in the grid: a red box *Regression failed — these components are not in the results* names it with the reason.

- In *Calibration results* pick the active fit per component from the dropdown (*Used*); it starts at the proposed order **with intercept** in every case. The *Optimal* column proposes the lowest order with **GOF ≤ 2** — the same test for every method — and reads the plain order, *2*, with the note *2 (b_0≈0)* only when the intercept test of that order ($|b_0|$ against $k \cdot u(b_0)$, k = the coverage factor of Settings › Uncertainty, 2 by default) finds the data consistent with a zero intercept. At order 1 the note is a hint that the through-origin variant *1 (origin)* is worth a look, never an automatic switch: *Used* only changes when you change it. ISO 6143 takes the simplest admissible function *unless the fits differ considerably*, and so does NexChrom: when the next order also passes the test and improves the fit clearly — the extra coefficient lowers TSSD by more than 3.84, the 95 % point of χ^2 with one degree of freedom — that order is proposed instead, step by step up the orders, never onto an order that fails (hover the *Optimal* header for the rule; the caption under the grid explains *Fit test* and *Optimal* in short). When **no order passes**, *Optimal* reads "—" and *Used* starts at the model set on *Settings › System › Calibration/Regression* (by default the 1st order with intercept; the lowest converged order where that model is not available), and the fit stays *Fail*; the message above the grid and a yellow warning under it name the component. A higher order rarely cures one deviating level: look at the *Calibration points* of that component first (the orange level, the weighted deviations — below) before changing *Used*. You can accept the proposal or override it. The grid shows one **GOF** column — **GOF (≤ 2)** — and the **Fit test** (*Pass* / *Fail*; "—" where the fit cannot be tested, such as a through-origin fit on fewer than three points, or where no GOF is stored). Hover a header for its meaning in plain words: the GOF says how far the worst point lies from the curve, in standard uncertainties (up to 2 passes). Because every method is tested on the same statistic, the GOF values of different methods can be compared.
- A fit that fails — GOF above 2, under every method — has its GOF in **red / bold** in the results grid and in the component-detail metric, its *Fit test* on *Fail*, and one warning banner below the grid lists the components: *The GOF exceeds the ISO 6143 limit of 2 for: …*. The limit is fixed by the standard; there is no threshold to adjust. The check is advisory only. A failing fit typically points to inconsistent measurement data (outliers, drift, mis-integrated peaks) — review the input rather than raising the order; the *Calibration points* table shows which level to look at first (below).
- If a component has fewer than three usable calibration points it is skipped with a warning naming the component and the offending sample(s); the other components still fit. Two opt-in checkboxes handle these cases, always shown and disabled when no component in the input has that number of points: for a **single** point a 1st-order through-origin fit, for exactly **2 points** the exact line with intercept and the through-origin fit, both selectable (*Used* starts at the exact line; the b_0-test labels *Optimal*). Adequacy cannot be tested for these fits: their GOF and *Fit test* read "—" in the results and in the stored model (on Calibration and on Review), they are never flagged — no red GOF and no warning banner, also not for the 2-point through-origin fit, which has one degree of freedom — and the notes of the analysis certificate say so (the Execute page carries no note). The detail of the *1 (origin)* variant shows the b_0-test of the line with intercept (*Intercept b_0 test (line with intercept)*) and its verdict: *consistent with a zero intercept — this through-origin variant is justified* or *significant intercept — the line with intercept (1) is the better choice*.
- Select a component in the results grid to open its **regression detail** — the badge says *New regression — not saved*: metrics (method, order, the GOF — *GOF (limit 2):* for every method —, red when the fit fails, with a tooltip in plain words; calibration range; max calibration error and its sample), the coefficients with their uncertainties and |b|/u(b), the covariance and correlation matrices, the fit statistics in groups (Fit; *Goodness of fit* written out in full: TSSD, *Birge ratio (BR) = √(TSSD/ν), information* — never red —, for every method the row **BR test, χ^2 95 %, information** (e.g. *Pass (limit 1.540 at ν = 4; p = 0.32)* or *Fail (limit …)*; "—" where the fit cannot be tested or has no GOF: TSSD against the 95 % point of χ^2 at ν degrees of freedom, so the limit on

the Birge ratio depends on ν — 1.96, 1.73, 1.54, 1.45, 1.35 at ν = 1, 2, 4, 6, 10; a check mark on *Pass*, red on *Fail*, and it decides nothing (the question mark at *Goodness of fit* says so): a *Fail* next to a passing GOF means that every point lies within its limits but the scatter as a whole is larger than the stated uncertainties lead to expect, a reason to look, not a verdict), *GOF (ISO 6143:2001, 5.2.2; limit 2)* — red when it fails —, R^2 and the intercept test (on a through-origin variant *Intercept b_0 test (line with intercept)* with the verdict whether the variant is justified); Calibrated range; Residual diagnostics: runs test on the residual signs, standardised residuals, Shapiro–Wilk normality), the **Calibration points** with residuals, for EIV the two signed weighted deviations $(x - \hat{x})/u(x)$ and $(y - \hat{y})/u(y)$, for OLS, WOLS and GLM the column $(y - \hat{y})/u(y)$ alone (red above 2, under every method: the level and axis that fail GOF ≤ 2), the χ^2 contributions and the column **Without this level (GOF)** — the GOF of a re-fit with the same method and order without that level ("—" when the re-fit has fewer than one degree of freedom or stopped without a verified optimum). When the fit fails and passes without one level, that row is **orange**: the level whose removal helps most; the caption names it with the other levels without which the fit would also pass. Orange is a pointer to investigate that level (certificate, cylinder, integration, injections), not advice to drop it; when another order passes with all levels the caption says so, the cause being more likely the order than a single level; when no single level makes the fit pass, the caption says so (several levels deviate, or the order may be too low). Then the calibration curve (hover: peak area, certified value, χ^2 contribution and the GOF without this level) and the residual plots (y for all methods, x additionally for EIV; hover: per-point relative error): the points are dark blue, the orange level orange and larger, and a point whose weighted deviation exceeds 2 red (every method) — the same colours in all three plots. A bow in the residuals of a first-order fit shows up as few runs and a large max |z| — try the next order.

- If a fit *stopped without a verified optimum* (column *Conv* = No with the reason, and a warning above the overview), the EIV iteration was halted by its divergence guard: the coefficients are the last accepted iterate. This nearly always goes together with a very high Birge ratio and means the calibration points do not match their stated uncertainties — check the CDS files linked to the calibration samples and their certificates before saving. *Optimal* never proposes such an order when a converged one exists.

Step 3 — Save the Model

- When the fit looks acceptable, save it. Saving into a model that already holds a fit (or that calculation runs reference) is guarded: choose **Save as new model** (recommended; a dated name is suggested) or an explicit overwrite.
- The suggested name for a new model carries date **and time** (models are often re-fitted a few times before one is kept); existing date/time suffixes are stripped first so stamps never stack.
- **One route per calibration**: when the input of the fits comes from more than one route (e.g. Compass files and CSV exports), or a component has more than one area unit, a red line above *Save results* says so and *Save results* is disabled — calibrate each route separately. Only an **admin** can override it: tick *Save with more than one route anyway (admin only)* and give the reason; the reason is stored on the model (column *Route override* in the models table), in the change history, and shown on Review and Execute. Saving later fits from one route clears it.
- With *Save as new model*, NexChrom creates the new model as current (is_current = 1, activated_at set) and archives the previous model (is_current = 0, archived_at set) with its component fits intact; replicate exclusions travel with the new model.
- A saved model starts as **draft** (columns *Release*, *Released By*, *Released At* in the models table, read-only). Saving new fits into a model that was already released puts it back to *draft* — the flash message says so — because the release covered the previous fits.
- The **method and instrument of a model with a release decision** (released, withdrawn or rejected) cannot be changed in the *Calibration models* table: *Save* refuses it with the red message *Not saved — the method or instrument of a model with a release decision cannot be changed: … Withdraw the model on Review and fit a*

new model on the other method instead (one released model per instrument and method). A model is checked and released for one instrument and method; a model for another method is a new model.

Step 4 — QC Check of the Calibration, then Release (second person)

- **Analyse a QC sample in the calibration session.** Run at least one QC / reference sample on the instrument together with the calibration gases (the same day; the QC window of *Calibration* in Settings › System › Validation (QC), default ± 1 day around the first and last calibration analysis). Link its CDS file(s) like any sample.
- **Calculate it with the new model.** On Execute choose the instrument, tick the QC sample (sample type *Quality Control*) and choose the method: a QC or reference sample is offered the current draft model too (*<model> · draft — only as the QC check of this model*; with a released model of the same method next to it, choose the draft one under *Calibration model*). *Calculate* shows the QC comparison; **Store as QC check of <model>** stores the run and the QC check runs automatically (see *Check a QC run*). A rejected or withdrawn model that is still current is offered to a QC or reference sample in the same way (*withdrawn — only as the QC check of this model*), so it can be checked and released again. An ordinary sample is never offered a model that is not released. The expander *QC sample in this calibration* on the Calibration page shows, top to bottom:
  - a **status line** with the run that decided it — *QC check passed · run n of <QC sample> (analysed …) — ready for release on Review*, *QC check with a warning …*, *QC check failed … · n component(s) outside the action limits* or *QC check without verdict … · no QC limits to check against* — or what is missing (*QC check missing · …*; *QC check not required (Settings) · …* when Settings do not require it); hover it for the rule;
  - one caption with the calibration session, the number of calibration analyses and the QC window;
  - the QC / reference analyses in the window as a **table**: *Sample*, *Type*, *Analysed*, *Run with this model* (the newest, or *not yet calculated*), *Verdict*, and *Earlier runs* with their verdicts when an analysis was calculated more than once;
  - while the release is blocked, one yellow box *The release on Review is blocked:* with the reasons and, when the QC analysis is not calculated yet, the next step.
- A **manager or admin** opens the Review page, section **Calibration models** (filter *Pending*; the **Instrument** selection left of it narrows the table to the models of one instrument, each listed with its number of draft models, *All instruments* by default), ticks the model (or clicks it in the Overview's *Review queue*, or opens its card on the Worklist): the detail card (instrument, method, release status as a chip, *Earlier decisions* with a *previously rejected / withdrawn* chip where applicable) and the fits per component (fit type, the regression method as its label — a fall-back fit reads *through-origin fit* or *2-point line* —, the polynomial order (the number of coefficients minus one, so a through-origin fit reads 1), **GOF (≤ 2)** — as stored with the fit, for every method, 4 decimals —, the **Fit test** (Pass / Fail; the GOF red when the fit fails, "—" where it cannot be tested: the 1-point and 2-point fall-backs, the 2-point through-origin fit included, or where no GOF is stored), R^2 with 6 decimals, TSSD with 4 significant digits, df, Conc. min / max with 6 significant digits, fit moment; hover a header for its meaning in plain words; the first column stays pinned while scrolling) and the block **QC check of this calibration** — the same block as on the Calibration page (status line, session caption, table with *Earlier runs*), the run details on the Runs tab. The release is blocked while no QC sample was analysed in the session, the QC sample was not yet calculated with the model, the check has no limits to check against, or the check failed: the reasons stand in the red box *Cannot approve / release:* above the decision buttons. A warning may be released with a note; a failure means the analyzer did not perform within the QC limits — investigate and recalibrate.
- Under the model card the full-width block **Decisions and notes** holds the waiver reason, the route override (*Saved from more than one route*, with the status line *Route* on the card), the current release notes, the model notes and the numbered decision history (oldest first, the current decision marked), so the card itself stays short.

- **Release** puts the model into use — Execute now offers it for new runs of every sample type — and withdraws every other current, released model of the same instrument and method: **one released model per instrument and method**. The message names them (*Withdrawn as superseded (one released model per instrument and method):* …), the withdrawn model gets the release note *Withdrawn by <user> on <moment>: superseded by the release of model '<name>' (id n) for the same instrument and method.*, the released model the note *Supersedes the released model(s) '…' of the same instrument and method (withdrawn).*, and every change stands in the Change history. Models released before this rule existed are not touched: when two released models of one instrument and method remain, withdraw one by hand. **Reject** marks it as not fit; **Withdraw** (on a released model) takes it out of use again, runs already calculated keep their results; a withdrawn or rejected model can be released again after a new QC check (Execute offers it to a QC or reference sample as the QC check of that model). Notes and the reviewer are stored; the history is shown per model.
- **Changing a model that was decided on.** You do not need to withdraw a model to change it: fit it again on the Calibration page and save (overwrite, or *Save as new model*). New fits put a released, withdrawn or rejected model back to **draft** (the flash and the release notes say so, an earlier QC waiver is cleared) and it appears in the Review queue again for its QC check and release. *Withdraw* is for taking a model out of use, not a step before editing.
- **Release without a QC check**: when no QC sample of the calibration session has a verdict — no QC measurement was made, or it is not calculated with the model yet — an **admin** may still release the model (the red box then ends with *As admin you can release without a QC check below (tick the box and give a reason).*): tick *Release without a QC check (admin only)*, enter the reason (for example a certificate correction with no QC measurement in the session) and click *Release*. The reason is stored on the model (*QC waiver* column on the Calibration page, *Waiver reason* on Review) and in the release notes; Overview shows an orange warning for the model (Worklist a card under *Needs attention*) until a QC run with it passes. A **failed** QC check can never be waived, and a manager cannot give the waiver.
- Models that existed before this release step was introduced were released automatically by the migration (released by *migration*); withdraw them here if they should not be in use.
- Reopening a model later: with no fresh results, *Calibration results* lists the stored fits (order, method — a fall-back fit reads *through-origin fit* or *2-point line* —, *GOF (≤ 2)* and Fit test, R^2, calibrated range, save moment, data age); a fit without a stored GOF reads "—" there — select it and the detail shows the GOF of its re-fit. Select one to see its full regression detail — badge *Saved fit* — with the calibration points rebuilt from the recorded input set (a warning tells you when the underlying data changed since the fit was saved; for EIV a re-fit confirms the stored coefficients and supplies the weighted deviations and, for a fit saved without it, the GOF; *Without this level* is computed on the rebuilt points). Historical (archived) models stay readable here, and the runs calculated with them keep their results; Execute does not offer them for a new calculation. Every save and archiving step is recorded in the Change history (AuditLog) on the Calibration page.

Workflow D — External Certificate Import (PDF)

Use this workflow to bring a reference certificate issued externally (VSL, NPL, NMi) into NexChrom so you can link it to samples as a fixed-value source.

Step 1 — Upload the PDF

- Open the Certificates page and select the tab **Import from file**.
- Choose the certificate PDF under *Upload certificate PDF* and click **Parse PDF** (the button reads *Parse PDF (with OCR)* for a scanned document).

- NexChrom first attempts to extract text from the PDF layer. If the document is a scanned image, it automatically falls back to OCR.
- It identifies the certificate format (VSL / NPL / NMi) and extracts certified component values.

Step 2 — Review and Save

- Review the extracted values on screen. Switch **View the uploaded PDF** on (above *Certificate details*, off by default) to see the PDF next to *Certificate details* and *Components* and compare the values with it before you save; viewing stores nothing.
- Read the **red messages under the component grid**: a value that cannot be an amount fraction — above 1 mol/mol (the unit was probably not recognised, e.g. a missing × 10^{-2} exponent or a mmol/mol header), not positive, an uncertainty larger than its value, or a sum of the fractions above 1 mol/mol — is reported there instead of being saved silently. Check the unit and the exponent of those rows against the PDF and correct them before *Save certificate*.
- Correct any parsing errors by editing the grid.
- Check the **Supplier** under *Certificate details*: a known supplier is proposed with its short name (VSL, NPL, NMi; the help of the field shows how the PDF names it) and is stored with the certificate, in the column *Supplier* of the Certificates table.
- Verify the validity date and cylinder number are correct. A cylinder number that differs from the linked sample's is a **critical** finding: a red notice on the Certificates page and on Review, and a 🔴 alert on the Overview (an *Act now* card on the Worklist) — a certificate belongs to one cylinder, so correct the link or the number.
- Save. There is no verification step on this tab: the certificate is stored as a **draft** under your name (*Entered by*). A second user checks every value against the source PDF and approves it on the Review page (Step 4); until then it is not offered as certified values on Calibration or as a basis on Quality control.
- After the save the uploader is empty and the confirmation (certificate number, draft, entered by) stays at the top of *Upload certificate PDF* until you choose the next PDF.
- NexChrom writes a Certificate row and a CertificateComponent row per reported component. The certificate status defaults to active; the approval status is draft.
- On save the PDF itself is stored under `data_local/certificates/imported/<certificate no>.pdf` and its SHA-256 hash is recorded on the certificate (columns *PDF* and *PDF Hash* in the Certificates table), the same way the generated analysis certificates keep their PDF. Uploading a PDF that was already imported under another certificate number shows a warning naming that certificate before you save.

Step 3 — Link to a Sample

- Open the Sample page or the Certificates page and link the imported certificate to the sample that represents the material it certifies.
- Or create the sample on the import tab itself before you save (*Link to sample* › **Create new sample**): the name and the cylinder number come from the certificate, and the **Sample type** is proposed from the certificate type chosen under *Certificate details* (calibration → *Calibration*, qc → *Quality Control*). The proposal follows a change of the certificate type until you choose a type yourself — that choice is kept.
- Fixed components in future Methods can now reference this certificate as their value source.
- A certificate entered **by hand** in the Certificates table has no PDF yet: tick it, choose the supplier's PDF in *Certificate composition* › **Source document** (a yellow line and the *Pending* caption show it as an unsaved change) and press **Save** in that section — the PDF is then stored with its SHA-256 like an imported PDF; *Discard* drops it. A PDF on file is shown by the switch **View the PDF on file** (inline viewer and download). Without a source document the certificate cannot be approved.

Step 4 — Check and Approve on Review (four eyes)

- A **different user** than the one who entered the certificate (a manager or admin) opens the Review page, section **Certificates** (filter *Pending*; the **Type** filter left of it — Analysis / Calibration / QC, several together, nothing selected = every type — narrows the table; the column **Created by**, just before *Approved by*, names the user who entered the certificate, or who issued it for a generated one: that user cannot approve it), and ticks the certificate: the detail card names *Entered by* and, under *Certificate*, the **Supplier** (check it against the PDF with the other header fields), and the **Source PDF** switch — on by default for a certificate still to be approved; switch it on yourself for an approved one — shows the viewer next to the component values.
- Compare every component value and uncertainty with the PDF (the **Sum of the component values** under the table is a quick plausibility check: far from 1 mol/mol points at a missing component, a wrong value or a wrong unit), correct the certificate on the Certificates page if something is wrong (the check then starts again), tick *I have checked every component value and uncertainty against the source document*, enter notes and click **Approve**. The approval is recorded under your name with the moment.
- Two rules block the approval: the source PDF is missing (upload it first) and the four-eyes rule — the reviewer is the user who entered the certificate. Internal QC certificates (created by the QC promotion) follow the same four-eyes approval; their source are the approved runs.
- A later change of the values, the header or the source document of an approved certificate withdraws the approval automatically (the certificate is a draft again, the reason is noted) so the changed certificate is checked again.

Dashboards

The menu holds two equivalent dashboards: **Overview**, the panel switcher, and **Worklist**, the same alerts and open work as piles of cards. They are the first two items of the menu, Overview on top. Both read the same sources, so an alert or a waiting review appears on both. Both are always open to every user (they are not in the rights matrix), and neither writes anything to the database: they only show, and take you to the page where you act.

Overview

Overview is the landing page (it opens after login): the alerts, the work waiting for the second person, the status per instrument and the routes through the application.

- **Panel switcher**: the four sections **Alerts**, **Review queue**, **System status** and **Workflows** are shown one at a time. The bar above them holds four equally wide blocks, the active one filled in its panel colour (red, blue, green, purple), with the number of items small in parentheses (alerts, queue entries, instruments; none behind Workflows); click a block or the ◀ / ▶ arrows and the chosen panel slides in. Alerts opens first; your choice is kept while you are logged in. Every panel opens with a centred coloured title band and a subline.
- **Alerts**: six tiles on top (critical, warning, caution, certificates, calibration models, quality control), then every alert as a line sorted by severity — 🔴 critical first, then 🟠 warning, 🟡 caution, 🔵 informational, 🟢 ok — and the legend of the coloured dots at the bottom. Every alert is a link: a certificate alert (over date, expiring, not linked, cylinder mismatch — the mismatch is critical) opens the Certificates page with that certificate ticked and its composition open; a calibration-model alert (without coefficients, expiring, released without a QC check) opens the Calibration page on the model's instrument with the model ticked; a QC-run alert opens Review with that run; a QC-sample alert opens Quality control › Monitoring with instrument and sample preselected. The page switches without a reload.
- **Review queue**: what waits for the second person in four columns — runs pending review (run, sample, instrument, stored date, a QC fail mark), certificates awaiting approval (imported, hand-entered and internal certificates, and legacy draft generated certificates), calibration models awaiting release and methods awaiting

release — six per column with a count (… *and n more on the Review page*; *nothing waiting*) and an *Open Review* link; every entry opens Review with that record ticked, its filter set and its tab open. Under the runs the link *n approved run(s) without certificate* opens Review › Runs under the filter *No certificate*, where *Issue certificate* issues them; these runs are not counted in the queue total.

- **System status**: six tiles (runs pending review, draft certificates, draft models, failed QC runs in 30 days, provisional QC limits, over date / warning), then one line per instrument — traffic light, released model with fit date and age, last QC run with verdict and age, QC limits, pending runs — each a link to the model on Calibration; the legend and the *Expiring / expired* line below.
- **Workflows**: one tab per route — *Analysis via sample*, *Analysis via CDS file*, *Calibration* and *Execute → analysis certificate* (Select Instrument & Sample, Checks, Result, Store for review, Review: approve and issue the certificate, Certificate issued) — with the steps in the order you walk them, including the second person's review, approval and release steps (blocks in the Review blue). Every step has a filled arrow button that opens its page and, next to it, a collapsed expander: open it to read what to do on that page (section, button, result, what can block) and to find an *Open <page>* link.
- The Review page itself shows its four parts — **Runs**, **Certificates**, **Calibration models**, **Methods** — as tabs: reviewing runs, certificates, models and methods are separate jobs. Via the menu you arrive on *Runs*; a link from the Overview's Review queue, or the button of a card on the Worklist, opens the tab of that record.
- **Methods need a release too**. A method is complex (component roles, RRF rules, fixed values), so a second person checks it: a new or changed method is a *draft*; on Review › *Methods* a manager or admin who did not make the last change ticks it, checks the roles and RRF rules and presses **Release**. Until then the Calibration page refuses to save a fit on the method and Execute refuses to store a run with it. An **admin** may override that with a reason (tick the override box, enter the reason): the reason is stored on the model or run, logged in the change history and shown as a 🔴 alert on the Overview (a card under *Needs attention* on the Worklist) until the method is released. Saving component roles or RRF rules of a released method puts it back to draft. The methods that existed before this step were grandfathered as released.
- At the bottom of the **Workflows** panel (below the three routes) the expander **Complete workflow overview** renders the full NexChrom flow as a fixed grid: the left column holds Component alias (joined to Component by a dotted line without arrowhead), Component, Sample and CDS files; the middle column Certificate (known values), Method, Analysis and Detector template; the right column Analysis certificate, Review, Calibration and Execute; the labelled lines show what feeds what (certified values, components & roles, cal. samples, model, …).

Worklist

Worklist shows the same alerts and open work as **piles of cards**: one card per thing to do, so you see at a glance how much is waiting where, and one click takes you to the record. The page has two tabs, each with a count badge: **Needs attention** (the badge is red while *Act now* holds a card, grey otherwise) and **Pick up where you left off**.

- **Needs attention** — four stacks side by side, each with a big coloured count, its name and a line below:
  - **Act now** (red, *Blocks results or certificates*): a certificate, calibration model or QC sample over its expiry date; a cylinder mismatch between a certificate and its linked sample; a QC run that failed its QC check (last 30 days).
  - **Needs attention** (orange, *Fix before it becomes a problem*): an expiry within the warning period; a calibration model with direct components without coefficients; a model released without a QC check (admin waiver); a model or run made with an unreleased method (admin override); a certificate not linked to a sample; a QC run with a QC warning.
  - **Coming up** (yellow, *Reminders and minor issues*): the 1st, 2nd and 3rd expiry reminders.

    - **Waiting for review** (blue): runs pending review, approved runs without certificate (one card for all of them), certificates awaiting approval, and calibration models and methods awaiting release. The line below the name says whose turn it is, e.g. *5 for you · 4 waiting for others*, *All 9 are yours* or *All 9 wait for others*.
    - A stack without cards shows a thin line *Nothing here*.
- **Pick up where you left off** — the unfinished work along the three routes, one pile per step that holds work, in process order (steps without work are not shown). The head of a pile shows the count in the colour of its route (Samples indigo, Calibration teal, Quality control violet), the step name and a line with the route and where you continue or whose turn it is (e.g. *Samples · on Execute*, or on a review step who may act). The steps:
    - Samples: **To analyse** (a sample without an imported analysis → Analysis), **To calculate** (analyses of a sample on an instrument that are newer than its last run → Execute with instrument and sample preselected; *Rejected — calculate again* when its last run was rejected), **To review** (runs pending review), **Certificate to issue** (an approved run without certificate → Review › Runs, filter *No certificate*).
    - Calibration: **Method to release**, **Certificate to enter** (a calibration gas without certificate → Certificates), **Certificate to approve**, **To fit** (an instrument with calibration analyses imported after the newest fit of its current model, or with calibration data but no model → Calibration), **Model to release**.
    - Quality control: **Certificate to enter** (a QC gas without certificate; reference gases are left out, their limits can come from the QC history), **Certificate to approve**, **To calculate** (QC or reference gas analyses not yet calculated → Execute), **To review** (QC runs pending review), **Limits to set** (a QC or reference gas analysed on an instrument without QC limits there, or with provisional limits only → Quality control).
    - When nothing is left the tab says *Nothing for you right now.* or *Nothing to pick up.*
- **Only mine | All users** (left above the piles of *Pick up where you left off*, with counts): *Only mine* hides the work that waits for someone else; *All users* shows everything.
- **Instrument menu** (top right on both tabs): *All instruments* ▾ opens a list of the instruments, each with its status dot and its number of cards; choose one to see only the cards of that instrument, choose *All instruments* again to see everything. Certificates belong to no instrument and are hidden while one instrument is chosen; an empty stack then reads *Nothing for* followed by the instrument name. The menu and the *Only mine* | *All users* choice are kept while the browser tab stays open.
- **Leafing through a pile.** A pile shows its top card with the cards below it as edges. Under a pile of more than one card the round arrows **‹ 1 of 5 ›** leaf through it while the cards stay small; with the mouse on the pile the ← / → keys do the same. **Show all** deals the pile out into a column; **Stack them** puts it back. Your own turn lies on top: cards that wait for someone else are sorted to the back.
- **A small card** shows its kind (e.g. CERTIFICATE), an icon top right, the title, the record, a line of details and, bottom right, the card **reference** (e.g. RV-147). The full red edge is reserved for *Act now*.
- **Open a card** by clicking it: it moves to the centre and opens large over the dimmed page, with the stack or route step, *n of m*, the reference, the facts, *Why it matters* and *What to do* (on *Pick up where you left off*: *Where it stands* and *Next step*). The arrows (or ← / →) page through the pile. One button takes you to the record — for example *Open certificate*, *Review now* or *Continue on Execute* — and opens that page with the record selected (Analysis, and Certificates for a certificate still to enter, open without a selection). **Esc** or a click beside the card puts it back.
- **Who acts on a review card.** A review card says whether it is your turn. Reviewing needs a manager or admin with the edit right on Review, and the four-eyes rules of the Review page apply while more than one user is active: you do not approve a run you stored (an admin may), a certificate you entered, or a method you changed last. The label at the top right of the card tells you: *You* (yours to do), *You (own run)* (an admin's own run), *You stored it*, *You entered it*, *You changed it*, *Needs a manager* (your access role cannot review) or *Needs a reviewer*. A card that waits for someone else is greyer with a dashed edge; opened, it shows the strip *Waiting for someone else* with the reason and a quiet button *View on Review*. Issuing the certificate of an approved run can be done by any reviewer.

- **The card reference** — two letters for the kind of action and the number of the record, e.g. RV-147 for run 147 awaiting review, CA-12 for certificate 12 awaiting approval, TC-31.4 for sample 31 to calculate on instrument 4. Use it to tell a colleague which card you mean: the same situation has the same reference for everyone and on both tabs, as long as it lasts. On an opened card the reference is a button: click it to copy it (it shows *Copied*). References are not stored; once the work is done the card and its reference disappear.

Maintenance

Database Integrity Checks

- Open the Settings page and select the Maintenance tab.
- Run the integrity check. The system reports anomalies such as orphan foreign keys or missing required values.
- Follow the reported guidance to resolve any findings.

Database Export (Backup)

- From the Maintenance tab, run the database export operation.
- Store the exported copy outside the project directory if you need an off-site backup.

Settings

The Settings page is organised in six tabs (User settings, System, Sensors, Maintenance, Database, SQL). Sections on the System and Maintenance tabs are rendered as default-closed expanders so the view stays calm.

- **User settings** controls display preferences: table view height, show/hide DB footer, show/hide Import/Export controls, and whether page selections are cleared on navigation.
- **System** exposes Server configuration, E-mail (SMTP) configuration, the Alert check schedule (scheduler not yet wired up), **RRF defaults** (relative standard uncertainties of calculated RRFs per detector type, ISO 6974-2 Annex B: FID 2 %, TCD 10 %), **Calibration/Regression** (Model when no order passes, default the 1st order with intercept: the model *Used* starts at when no order passes the fit test; u(peak area) of a level with a single injection, default the spread of the other levels, or a fixed value, relative or absolute; u(certified) of a value without a certificate uncertainty, a fixed standard uncertainty, default 1 % relative; Max calibration data age, default 365 days, 0 = off, used to warn when the analyses behind a stored fit grow older than the limit; every calibration fit is tested on the fixed GOF ≤ 2 of ISO 6143:2001 5.2.2) and **Uncertainty** (u(peak area) of a sample with a single injection, default the spread of the calibration levels, or a fixed value: one injection has no spread of its own, and Execute names the components; the app-wide coverage factor k, default 2, used to expand the result uncertainties of a calculation and to convert to standard uncertainties at calibration input; a run stores the k of its calculation at *Store for review* and its certificate is issued with that k). *Significant digits for uncertainties* and the *ISO 6976 reference conditions* are stored with a run in the same way, so a later change does not alter the certificate of a stored run. **Validation (QC)**: the **QC window per sample type**, one row per type — *Sample* (default ± 7 days: on Execute the validating QC run of a sample run is proposed within it, nearest first) and *Calibration* (default ± 1 day: the QC sample of a calibration session counts for the release of the model when it lies within the window around the first and last calibration analysis; on Execute the same window applies to a calibration gas); a *Quality Control* or *Reference* run has no window, it is judged against its own QC limits — and the checkbox **QC sample required for the release of a calibration model** (default on); the section has its own *Save*.
- **System › Laboratory** holds the name of your own laboratory. NexChrom records it as the supplier of the certificates it makes — the analysis certificates issued on Review and the internal QC certificates of *Promote a sample to QC* — in the column *Supplier* of the Certificates table; an issued certificate keeps the name it got,

and the name is not printed on the certificate. The field proposes the organisation of the accreditation scope (Administration › Scope) until you save your own name; save an empty field to follow the scope again.

- **System › Chromatograms** holds **Default Log range [decades]** (0 to 10 in steps of 0.5, default 5): the value the *Log range* field of the chromatograms on *Analysis › CDS File Browser* starts at, for Chromeleon archives, Agilent `.D` and Scion `.DATA` alike. 0 gives a linear y-axis; above 0 a logarithmic y-axis on which each detector signal is lifted until its lowest point lies that many decades below its highest (the shift in the panel title). The field on the Analysis page can be changed per view; that change is not stored. Only the plot changes, never the peak results. The section has its own *Save* (*Saved: chromatograms open with Log range 5 decade(s).*).
- **System › Log** shows where NexChrom writes its messages for troubleshooting (for example when the Aura2 cannot be reached or comes back): by default `%LOCALAPPDATA%\Qualimetric\NexChrom\logs\nexchrom.log` on the PC that runs NexChrom, not in the data and not in the terminal. Below the file its size; at 1 MB the file moves on to `.1` and at most three older files are kept. *View the log* shows the messages, newest first, at most 200; a message with a traceback under it counts as one. Four filters narrow it down, over the whole file: **Level** (click a level to switch it off or on; all are on at first), **Source** (the part of NexChrom that wrote the message, for example *nexchrom.ambient* for the Aura2; none chosen = all), **Period** (*All*, *Last hour*, *Today*, *Last 7 days*) and **Search** (any text in the source or the message, upper or lower case alike). The caption says how many messages match (*Filtered: 3 of 120 message(s) match.*); when none does, a notice says so. **Refresh** reads the file again, **Download the log file** saves the whole file (for example to send with a question) and, while a filter is active, **Download the filtered lines** saves only the matching messages, oldest first, as *nexchrom_filtered.log*. Everyone may view it; only an administrator can choose another file: enter the full path of a file, **Save** (the log continues there at once, the previous file stays where it is), **Discard** puts back the saved one and **Use the default location** puts the default in the field. When the file set cannot be used (a network drive that is gone), NexChrom writes to the default location and the section says so.
- **Sensors** connects the optional **Aura2** ambient sensor (temperature, air pressure, relative humidity); without the sensor NexChrom works as before and this tab can be left as it is (off) — for example when the GC itself already records temperature, pressure or humidity with every injection. In **1 · Use** tick *Sensor connected / in use* and *Always show measured values on screen*, and choose the **Connection**: **USB-C** (the Aura2 on a USB port of the PC that runs NexChrom; *Auto-detect* finds it, or choose the port) or **WiFi** (the network name, e.g. `aura2-tester1.local`, or the IP address; **Search** lists the Aura2s on the local network). In **2 · Readout & display** set the readout interval and the units (°C / °F / K; hPa / kPa / bar / mmHg). **Save** stores the settings, **Discard** restores the saved ones. Under **Live readout**, **Read out** reads the Aura2 now with the values on screen — also before saving, so a connection can be tried first — and names the device (label, chip ID, serial number of the temperature / humidity sensor, firmware, port or address); **Reset** forgets the found port or WiFi connection. With *Always show* on, a pill at the top of every page shows the measured values, e.g. ⚡ — 🌡 *20.0 °C* · 🕒 *1008.4 hPa* · 💧 *53.1 %RH*. The plug pulses briefly at every new measurement. When a readout fails the last values stay, dimmed with ⏳, and NexChrom tries again within 10 s; after three failures in a row the pill turns red — *Aura2 sensor not found · check the connection* — and hovering over it shows the reason (e.g. the port is in use by another program, or the WiFi host is not reachable). The sensor is read in the background, so logging in and working never wait for it; right after the login or a change of the connection the pill may show *Aura2: first measurement follows…* for a few seconds. The values are shown only: they are not stored and do not enter any calculation. With *Auto-detect* NexChrom searches for the Aura2 by sending a short identification request to every serial port of the PC, and after a failed readout it searches again at the next attempt (about every 10 s while no Aura2 answers). When other serial devices are connected to the PC, choose the port of the Aura2 instead of *Auto-detect*. NexChrom opens the port at every readout, so another program that needs the Aura2's port (for example a firmware update) works only while the sensor is switched off here. The log (System › Log) records when the Aura2 drops out, a different reason, a reminder every half hour while it stays away, and when it is back — not every attempt.
- **Maintenance** offers the database export, data-integrity checks and record counts — each in its own expander.

- **Database** holds just the active database path.
- **SQL** is the direct SQLite console, moved here from a separate page; it offers SELECT and Script execution modes against the active database.

The path of the settings file is displayed above the Save / Discard buttons at the bottom of the page.

User Management

Manage user accounts on the Administration page, USERS tab.

Create a User

- Open the Administration page and select the USERS tab (the tabs are Instruments, Components, Locations, File formats, Users and Scope). The tab holds two sections, each an expander open by default: **Users** and **Access roles & rights** (neither depends on the ticked user: the second is per access role; the minimum alert level for e-mails lives on Settings › System › E-mail (SMTP) configuration) (the Locations and File formats tabs hold one table each and therefore no expander — the rule: one section under a tab is shown bare, two or more sections are expanders).
- Click Create user to add a new row.
- Fill in username (required), full name, email, phone, and address fields.
- Select the user's laboratory Role from the dropdown (Laboratory Manager, Lab Supervisor/Team Leader, Laboratory Technician / Technician II, Lab Assistant/Assistant Technician, Research Scientist/Analyst, or Principal Investigator). This role is informational.
- Select the user's **Access role** (operator, manager, auditor or admin; default operator) — the role the permission system will use; admin always keeps full rights.
- Tick the Email alerts checkbox if this user should receive alert e-mails.
- Click Save to write to the database.
- The minimum alert level (🔴 Critical, 🟠 Warning, 🟡 Caution) at which subscribed users receive e-mails is set on the Settings page, System › E-mail (SMTP) configuration, next to the SMTP settings; here you only decide *who* receives them (column *Email alerts*).

Roles & rights

- Below the Alerts section, the **Roles & rights** matrices define per access role which pages may be viewed and edited (expanders *View* and *Edit*, one checkbox per page). The rows *Instrument* and *Component* govern the Instruments and Components tabs of Administration, the row *Administration* its other four tabs (Locations, File formats, Users, Scope).
- Only an admin can change the matrices; other users see them read-only. The admin row always keeps full rights — changes to it are ignored.
- Change the ticks and click **Save** (the standard blue button; it stays disabled while nothing is pending) to store the matrix — audited, visible in the Change history; **Discard** next to it drops the unsaved ticks. The caption *Pending: ~n rights changes* counts the ticks that differ from the saved matrix.
- The matrix is enforced on every page: without the view right the page is not shown (a lock notice instead), without the edit right the page shows a *View only* notice under its title and every write button (Save, Create, Delete, Archive, Execute, Approve, …) is disabled; Discard, Clear and read-only actions stay available. Review decisions and the saving of Quality-control limits keep their own, stricter gates on top.
- On Administration each tab follows its own row: a tab whose row denies the view right is left out, and each tab shows its own *View only* notice. By default an operator or manager can therefore still edit instruments and

components, while Locations, File formats, Users and Scope are view only for everyone but an admin. The page opens when the role may view Administration, Instrument or Component.

E-mail Alert Notifications

Configure SMTP so NexChrom can send the alerts of the dashboards as e-mails to subscribed users.

Configure the SMTP Server

- Open the Settings page and select the System tab.
- Expand E-mail (SMTP) configuration.
- Enter SMTP host, port, security mode (STARTTLS, SSL or none), username, password and From address.
- The password is stored in the operating system credential store (Windows Credential Manager), not in the settings file.
- Click Send test e-mail to verify delivery to your own user account's e-mail address.

Schedule Alert Checks

- In the same System tab, below SMTP configuration, tick "Send e-mail immediately when an alert occurs" if you want each new alert to trigger a message.
- Set the Daily check time for the scheduled digest.
- A yellow notice states that the dispatch scheduler itself is still being wired up; the settings you enter are persisted so they take effect as soon as the scheduler goes live.

Subscribe Users to Alerts

- On the Administration page (USERS tab), tick the Email alerts checkbox for each user who should receive alert messages, and make sure their e-mail address is filled in.
- On the Settings page (System › E-mail (SMTP) configuration) set the minimum alert level so users only receive e-mails at or above that severity.

Upload Instrument Documentation

Keep the manuals, specifications and troubleshooting guides of your instruments with NexChrom: each instrument has a documentation folder of its own, and the instruments of one make/model share a second folder, so that a common manual is kept only once.

Upload Files

- Open **Administration › Instruments** and tick the instrument in the Instruments table (Select checkbox).
- Below *Instrument component aliases* the expanders **Instrument-specific documentation** and **Shared make/model documentation** open for that instrument (without a ticked instrument they ask you to select one; before NexChrom 0.10.0 this was a Documentation tab with its own drop-down).
- If no documentation folder exists yet, click Create documentation folder.
- Use the drag-and-drop area or Browse files to select one or more files (PDF, TXT, MD, DOCX, XLSX).
- Click Upload to save the files to the instrument's documentation folder.
- Documents that apply to every instrument of one make/model belong in the shared section — uploading or deleting there affects all instruments that point to that shared folder.
- To delete a file, select it in the grid and click Delete.

View or Download a Document

- Select exactly one file in either table: a download button appears, plus an inline preview for PDF, TXT and MD files (DOCX and XLSX are download-only).

Troubleshooting and Common Issues

A peak in a CDS file was not matched to a component

Add the exact peak name as an alias of the canonical component: on Administration › Components, expander *Global component aliases*, when the name holds for every instrument, or on Administration › Instruments, expander *Instrument component aliases* of the selected instrument, when it is specific to that instrument (an instrument alias wins over a global one). Re-select the CDS file (or re-import it) and the response will now be recognized.

The triangles of a Scion file stand beside the peaks

On *Analysis › CDS File Browser › Chromatograms* the peaks of a Scion `.DATA` file are drawn on the channel of the detector NexChrom reads for them. When the triangles of one channel stand beside the peak tops and the blue box counts peaks without begin and end, NexChrom has given those peaks the name of another detector — a known limitation for some Scion files (see *Inspecting CDS Files Without Linking*). Leave that file out of the import; when it was imported already, check the *Detector* of its measurements on *Analysis via sample*.

Execute is blocked by a validation error

The *Checks* part of the Execute page shows which component, response, or calibration mapping is missing; *Calculate* stays disabled while an error holds (the red box); a warning (the yellow box) does not disable it. Typical causes:

- The selected Method requires a component that is not present in the linked analyses (import more CDS files or change the method).
- Method and model do not match: a direct component of the method has no fit in the calibration model (recalibrate with a gas that contains it, or have a method version released in which it is indirect).
- No method is offered at all: the instrument has no current, released calibration model (release the draft model on Review › Calibration models, or build a new model on Calibration).
- A fixed component has no value in the method: this is a warning in *Checks* (it names the component), not a red check — *Calculate* stays enabled but fails as a whole with *Calculation failed: Fixed component '…' has no fixed value in the method (Method page, column Fixed value).* Enter the value in the *Fixed value* column of the component role on the Method page.

I deleted a sample and its certificate disappeared from filters

Certificates use ON DELETE SET NULL for sample_id, so the certificate itself is preserved when its sample is deleted — only the link is cleared. Re-link the certificate to a different sample if needed.

I want to rerun an old analysis with the original calibration model

This is not offered: Execute lists only current (not archived) calibration models — released ones, and for a QC or reference sample also a draft, rejected or withdrawn one (as the QC check of that model). Calibration models are preserved with is_current flags rather than overwritten, so the earlier runs keep their results, their model and their certificate, and remain traceable on Review; an archived model and its fits stay readable on the Calibration page. A new calculation of the same CDS files uses the current model of the instrument.

The certificate of an approved run is missing

The run was approved with *Approve without certificate*, or its certificate was withdrawn. Open Review › Runs with the filter *No certificate* (or Overview's *n approved run(s) without certificate*, or a *Certificate to issue* card on the Worklist) and click **Issue certificate**. When that button is disabled, the message above it says why: an active certificate exists already, the ISO 6976 choice stored with the run cannot be calculated, the calibration model is no longer released or was fitted again after the run (calculate the sample again on Execute and have the new run approved), or a certificate file of the same name exists in data_local/certificates/generated/ (a file is never overwritten: move it first).

Where are the messages of NexChrom?

In the log file, not in the terminal: Settings › System › *Log* shows where it is (by default `%LOCALAPPDATA%\Qualimetric\NexChrom\logs\nexchrom.log`), its newest messages under *View the log* — filter on *Level WARNING* and *ERROR*, or *Search* a word such as *Aura2*, to find the one you need — and offers it as a download. What you need to know while working the app shows itself (for example the Aura2 pill at the top); the log is for finding out afterwards what happened.

The app reports a missing table or column

This typically means the active database was initialized from a different DDL version or misses an additive migration. Check the Metadata table on Settings › **SQL** with `SELECT * FROM Metadata`: `database_version` names the DDL version (2.12.0 for the current schema; `migrate_schema_v2.12.0.py` sets it) and every applied migration leaves one row `migration:<script>` (each `migrate_*.py` writes it with `--apply`). Run the migration that is missing (`CHANGELOG.md` lists them per release), or switch databases, or re-initialize using data_local/database/init_db_v2.2.0.py against the current DDL.